Document Number: 81395-8498820
Revision: B
CAGE Code: 81395

MICROIR™

SOFTWARE
INTERFACE CONTROL DOCUMENT

TWV640
UNCOOLED 12 µm (MICRON) THERMAL IMAGER

June 24, 2016

Prepared By:

Information & Electronic Systems Integration Inc.
2 Forbes Road
Lexington, MA 02421

DISTRIBUTION A – PUBLIC RELEASE UNLIMITED

The export of this product is governed by the United States Department of Commerce under ECCN 6A003.b.4.b

DISCLAIMER AND COPYRIGHT

This document gives only a general description of the product(s) and service(s) and, except where expressly provided otherwise, shall not form any part of any contract. From time to time, changes may be made in the products or the conditions of supply.

Revision History

Rev	Date	Comment
-	10/27/2015	Initial release per ECO EOB57783
A	05/12/2016	Revised IAW ECO EOB67860
B	06/24/2016	Revised IAW ECO EOB68352

Table of Contents

1. Introduction

The TWV640 and TWV640i camera cores can be controlled by the provided MicroIR™ GUI application running on a personal computer or through customer-designed software or embedded systems.

This document describes the software interface for the TMV640/TMV640i infrared camera core family of products. Its purpose is to provide the available commands and a concise description of each command. All commands are in an easy to read ASCII format and can be communicated to the unit via Universal Serial Bus (USB) or a Universal Asynchronous Receiver Transmitter (UART) interface.

1.1 Interfacing with TWV640 Camera Cores

There are two standard hardware configurations of the TWV640 cores; a base configuration with everything needed to image and a configuration with an added interface board.

- Base configuration is ideal for embedded systems and has two methods for control – 3 pins for the industry standard USB 2.0 HS and an RS-232 port for a UART.
- Added interface board has individual standard connectors for micro USB, Camera Link and analog video

While some of the timing and electrical details of the UART and USB protocols are different, the commands and command structure described in this document applies to both interfaces.

NOTE: For detailed information on all electrical interfaces to and from the camera units, see the Electrical Interface Control Document.

1.2 Camera Models and Their Interfaces

The TWV640 and TWV640i are very similar in terms of commands available and what function each command represents, and unless noted the user should assume the command is the same across all cameras.

The TWV640 supports 30 and 60 Hz frame rates and TMV-640i only supports a 7.5 Hz frame rate. Although the TWV640i does not support the 30 Hz and 60 Hz frame rates, the command interface is the same.

Abbreviations and Acronyms

Abbreviation	Description
ASCII	American Standard Code for Information Exchange
EEPROM	Electrically Erasable Programmable Read-Only Memory
FOC	Fine Offset Calibration
FPA	Focal Plane Array
FPGA	Field Programmable Gate Array
GUI	Graphical User Interface
ICD	Interface Control Document
IR	Infrared
NTSC	National Television System Committee
NUC	Non-Uniformity Correction
OLED	Organic Light Emitting Diode
PAL	Phase Alternative Line
UART	Universal Asynchronous Receiver Transmitter
ROI	Region Of Interest
USB	Universal Serial Bus
USB HS	Universal Serial Bus High Speed
VGA	Video Graphics Array (640x480)

2. Communicating with the Camera

The base configuration of the TWV640 can be configured and monitored via an ASCII interface through either USB or UART, while the configuration with an interface board has only a micro USB port. Refer to the Electrical ICD document for more detailed information on each connector. The serial ASCII protocols are designed to be intuitive and this document provides the details necessary so that the commands may be used and called properly.

2.1 ASCII Commands

For formatting purposes, special characters are omitted from command descriptions in this document and the TWV640 cameras can accept commands with or without the special formatting. This section describes the optional full formatting for commands that may make system communication more robust. There are two places special characters may appear in the command and response of a unit – before the command and after the command. Each command can be prepended with the total ASCII character count. The end of the command must be terminated with carriage return and newline characters.

For example, if the system intends to send the command 'polarity set black' to a camera core, the message can either be simply the text of the command or the following:

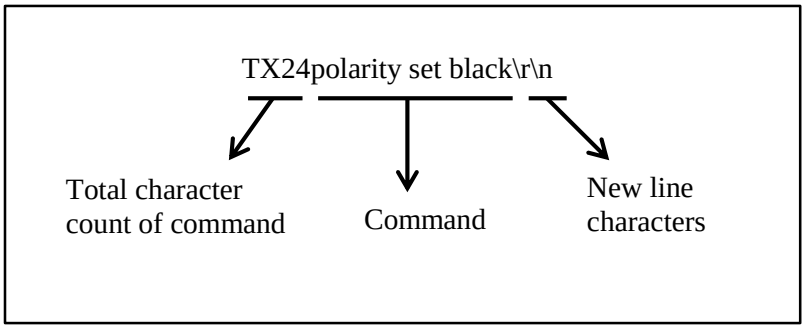

Figure 2-1: Optional command formatting

Before the command the following text is added - TX24. The section before the command includes the letters TX following by a two digit number that represents the total character count of the command, including any special characters.

2.2 Understanding the ASCII Response

Each ASCII command results in a response that demonstrates the unit has received and is processing the command. The response is in the following format:

TX*##MESSAGE/r/n*

where:

- ## = Two digit decimal ASCII number with the number of bytes in the message, including the carriage return /r and new line /n characters
- *MESSAGE* = is the text that describes the status of the command

NOTE: For documentation purposes only, the responses noted in section 3 do not show the "TX##".

The following is an example of how the *polarity set black* command is documented with its response:

ASCII Command: polarity *set black*

Response: Processing: polarity set black
>

The response can be further separated into two parts – processing and the end of the command. 'Processing: polarity set black' means the unit has received the command successfully and is currently executing the command. The '>' character indicates the unit is ready for another command. There is a large variation in how much time each command takes to finish. Instead of setting a constant time, the system should wait for the receipt of the '>' character before sending another command.

2.3 USB Settings

See Table 2-1 for the USB serial port settings. Currently the serial port settings cannot be changed.

Table 2-1: USB Serial Port Settings

Serial Port Configuration Parameter	Value
Baud Rate	115,200 bps (bits per second)
Parity	NONE
Data Bits	8 bits per byte
Stop Bits	1 stop bit per byte
Encoding	ASCII

2.4 UART Settings

See Table 2-2 for the UART serial port settings.

Table 2-2: UART Serial Port Settings

Serial Port Configuration Parameter	Value
Baud Rate	19,200 bps (bits per second)
Parity	NONE
Data Bits	8 bits per byte
Stop Bits	1 stop bit per byte
Encoding	ASCII

3. List of Software Commands

This section lists the supported commands. Depending on the product, there may be additional commands, and some commands listed here may not be available for your product. To determine the version of the embedded software build, send the **version info** command.

The commands are broken into three parts – root command, subcommand and arguments. The root command is the first word of the command. The subcommand is the second word and any remaining parts are the argument(s). Not all commands have all three parts, see Figure 3-1 below for an example.

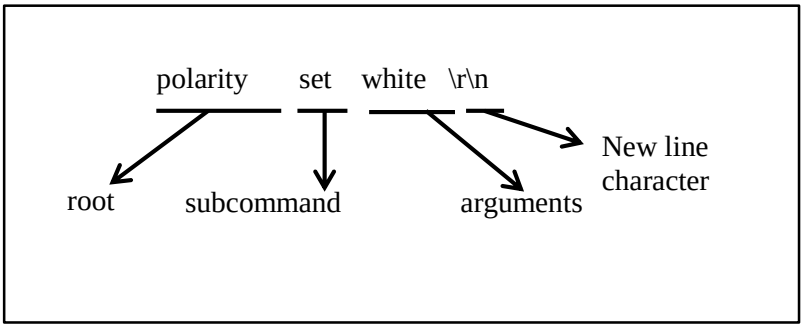

Figure 3-1: Command Structure

The most helpful command in the list is **help**, which displays all the root commands that are available. To further explain each command, subcommand and arguments, type *HELP <ROOT CMD>* (for example, **help version**). See the remainder of Section 3, as well as Section 4, for more details about individual commands.

NOTE - Commands are not case sensitive.

3.1 Adaptive Color Command (adaptive_color)

The camera units allow custom graphics to overlay over active imagery. The adaptive color command allows the color of the additional custom graphics to be dependent upon the color of the displayed imagery. For example, if the custom graphics to be overlaid over an image are white and imagery under the custom graphics end up white, the graphic overlay would become invisible. The adaptive color allows the color of the custom graphic to be changed if the average of the region of interest (ROI) underneath the custom graphic crosses either low or high thresholds.

Note on the region of interest – the shape is actually a square with a side of (radius * 2 + 1) with the center of the square falling on (x-center, y-center). If the radius is zero, the ROI is a single pixel (x-center, y-center). For example, if the radius is 5 the ROI is an 11 pixel by 11 pixel square.

Root Command: adaptive_color

Sample ASCII Command: adaptive_color *read*

Sample Response:
```
Processing: adaptive_color read
AVG = 126.315750
>
```

Table 3-1 adaptive_color command

Subcommand	Argument	Description
roi	x-center	Establishes a region of interest (ROI) that defines average value of what is considered underneath the custom graphic. All pixels in region of interest are averaged and that value is used to determine where background stands in relation to adaptive color threshold.
	y-center	
	radius	
high	r_component	If the average value of the ROI pixels is above the set threshold, this setting sets the new color of custom graphic pixels in Red, Blue, and Green format.
	g_component	
	b_component	
low	r_component	If the average value of the ROI pixels is below the set threshold, this setting sets the new color of custom graphic pixels in Red, Blue, and Green format.
	g_component	
	b_component	
thresh	high_threshold	Sets the threshold for the adaptive_color command.
	low_threshold	
enable	true | false	Enables (true) or disables (false) adaptive color setting.
read	-----	Returns ROI average pixel value

3.2 Auto Contrast Enhancement Level Command (auto_ce_lvl)

The contrast enhancement level represents a tradeoff between image dynamic range and contrast. Higher enhancement level makes smaller emittance differences more discernable but reduces the visible dynamic range. By default, the enhancement algorithm uses the majority of the image for enhancement of the entire image – a rectangle with (x-position, y-position) coordinates of (23, 16) and width of 591 and height of 445. The window can be changed using the auto_ce_lvl command, but must be at least 10 pixels wide and 10 pixels high.

Root Command: auto_ce_lvl

Sample ASCII Command: auto_ce_lvl *get*

Sample Response:
```
Processing: auto_ce_lvl get
Auto CE Gain Level = 2
>
```

Table 3-2 auto_ce_lvl command

Subcommand	Argument	Description
default	new_gain_level	Sets the default contrast enhancement level. Valid range of 0 to12.
get	-----	Returns the auto-contrast enhancement level.
set	new_gain_level	Sets the contrast enhancement level to a number between 0 and 12.
window	x-position	Sets the window in the image the algorithm uses to set the dynamic range of the entire image. The x-position and y-position are relative to the upper left corner of the image, and width increases from left to right and height increases up to down.
	y-position	
	height	
	width	

3.3 Color Enhanced Imagery Command (color)

Instead of an image ranging between black and white to represent different infrared emittance levels, there is a "false color" option to instead have an imagery that ranges from blue to red. In all settings, blue represents black and red represents white. Therefore, if the polarity of the image is set to white-hot then the color red would represent higher infrared emittance. If the polarity of the image is set to black-hot then the color blue would represent higher infrared emittance. Color is only available in 24-bit display mode.

Root Command: color

Sample ASCII Command: color *enable true*

Sample Response: Processing: color enable true
>

Table 3-3 color command

Subcommand	Argument	Description
enable	true | false	Enables or disables color look up table in the infrared imagery.

The following bars illustrate the mapping of black and white to "false colors" for white-hot video.

Black White

Blue Red

3.4 User Configurable Graphics Overlays Command (usr_overlays)

The TWV640 camera core allows graphics to be overlaid on live imagery. The usr_overlays command can be used to manipulate already existing overlay graphics on the camera core in terms of where it shows up in the imagery and its duration. Generating overlay graphics is described in the MicroIR GUI User's Guide.

Root Command: usr_overlays

Sample ASCII Command: usr_overlays *show splash*

Sample Response:
```
Processing: usr_overlays show splash
>
```

The usr_overlays list command returns a list of existing graphic overlays through the ASCII interface:

Sample ASCII Command: usr_overlays *list*

Sample Response:
```
Processing: usr_overlays list
Start Listing Gui Object Item Names:
gui_overlay_1
gui_overlay_2
gui_overlay_3
End Listing Gui Object Item Names
>
```

Table 3-4 usr_overlays command

Subcommand	Argument	Description
enable	true | false	Enable or disable user overlays.
hide	overlay_name	Disables custom graphic overlay named overlay_name.
list	-----	Returns a list of graphic overlays stored on the camera core.
move	overlay_name	Moves center of graphic overlay named overlay_name to (x-position, y-position) relative to upper left corner of live imagery.
	x-position	
	y-position	
movec	overlay_name	Moves center of graphic overlay named overlay_name to (x-position, y-position) relative to center.
	x-position	
	y-position	
layer	overlay_name	Set the layer of a user defined overlay to overlay_name to layer (1=foreground, 2=midground, 3=background)
	layer	
refresh	-----	Updates list of available user-defined overlays.
show	overlay_name	Enables custom graphic overlay named overlay_name.
timer	overlay_name	Shows user-defined overlay overlay_name for time_in_seconds amount of time on the screen, after which overlay is no longer shown.
	time_in_seconds	

3.5 Display Command (display)

The TWV640 allows direct access to the settings of the eMagin VGA family of OLED displays. Using this command, the settings of an eMagin VGA OLED connected to the unit can be queried or set.

Root Command: display

Sample ASCII Command: display *dimctl 00*

Sample Response: Processing: display dimctl 00
Finished DIMCTL update
>

Table 3-5 display command

Subcommand	Argument	Description
dimctl	brightness_level	Sets display brightness level, hexadecimal value that can range from 0 to 7F without 0x.
idrf	final_idrf_value	Populated by the factory.
read	register_address	Read eMagin register value at register_address.
readp	register_address	Read eMagin EEPROM register values.
shift	x-offset	Sets image offset relative to center of display. x-offset and y-offset are signed integers.
	y-offset	
startup	-----	Starts display if not already started
write	register_address	Overwrite eMagin register value at address register_address and value data.
	data	
writep	register_address	Write eMagin EEPROM register values.
	data	

3.6 Fine Offset Calibration Command (foc)

There are two ways of calibrating each unit with non-uniformity correction (NUC) – using the scene or a blade if available. Using a blade shutter does not require any particular kind of scene, but the resulting calibration does not include effects from the lens and any physical variables past the shutter. Calibrating to the scene requires a uniform scene, with something in front of the camera that covers the entire frame of view that is uniform, but results in calibration that includes the lens in front of the camera. The unit must have a shutter in order to calibrate with a blade and currently the base configuration employs the use of a Nanomotion shutter. If a shutter is not available the unit will display an error letting the user know that the I2C interface failed.

Every time a unit calibrates, even if automated, there is a message through ASCII on the USB and UART ports. The output states "Starting FOC" on shutter start and "FOC Complete" when calibration has been finished:

Root Command: foc

Sample ASCII Command: foc *scene*

Sample Response:
```
Processing: foc scene
Starting FOC
FOC Complete
>
```

A similar message is also output when the camera calibrates automatically:

Response:
```
Starting FOC
FOC Complete
```

Automated shuttering is depending on three system parameters, the units being seconds:

FINE_OFFST_CAL.FOCSHUTTERINCR - INCREMENT

FINE_OFFST_CAL.FOCSHUTTERMAX - MAXIMUM

FINE_OFFST_CAL.FOCSHUTTERMIN – MINIMUM

The first shutter happens after the first MINIMUM amount of seconds, from then on the period between shutters is incremented by the INCREMENT amount of seconds until it reaches the MAXIMUM. For example, if the MINIMUM is 95 seconds, INCREMENT is 90 seconds and MAXIMUM is 270 seconds: The first shutter is at 95 seconds, the MINIMUM. The second shutter is at 95 seconds + 185 seconds (95 seconds + 90 seconds). The third shutter is at 95 seconds + 185 seconds (second shutter) + 270 seconds (270 < 185 + 90) From then on a shutter will occur every 270 seconds.

Table 3-6 foc command

Subcommand	Argument	Description
auto	on | off	Enables or disables automatic fine offset calibration.
blade	-----	Calibrates with a blade.
scene	-----	Calibrates unit to scene.
reset	-----	Resets a hung FOC.
debug	-----	Prints debug information.

3.7 Flash Frame Capture Command (framecapture)

As many as 60 frames can be captured to memory and then selectively displayed according to an index number. The frames are indexed in the order which they were saved. The saved images can then be displayed instead of live imagery.

Root Command: framecapture

Sample ASCII Command: framecapture *print norm*

Sample Response:
```
Processing: framecapture print norm
Begin Stored Frames
7 64040000 n=8 p=-1
8 640e0000 n=-1 p=7
End Stored Frames
>
```

The sample response above shows that there are two captured frames with index numbers 7 and 8, and to display the frame captured with index 7 instead of live imagery:

ASCII Command: framecapture display 7

Response:
```
Processing: framecapture display 7
addr = 0x64040030
>
```

The address of the image is always the framecapture print norm address plus 0x30, and the numbers following the hexadecimal address printed by the print subcommand representing the previous and next image numbers.

Table 3-7 framecapture command

Subcommand	Argument	Description
capture	-----	Saves current frame to memory.
clear	------	Clears the entire area of Flash memory dedicated to imagery regardless of how many images are saved, and similarly takes the same amount of time regardless of how many images exist.
delall	-----	Deletes all saved frames.
delete	frame_index	Deletes frame with index frame_index from memory.
display	frame_index	Displays frame at index frame_index.
live	-----	Returns to live imagery from displaying a saved frame
print	norm	Returns USB ASCII text with location and index number of all saved frames.

3.8 Set Frame Rate Command (framerate)

Change the default frame rate of the camera; the default frame rate is saved to flash memory and will be reverted to at startup, but does not change the current operating frame rate. Analog output supports both 30 and 60Hz in the TWV640 core and 7.5Hz in the TWV640i. USB output is supported on all of the frame rates in both cores.

Root Command: framerate

Sample ASCII Command: framerate *get*

Sample Response:
```
Processing: framerate get
frameRate = 1
>
```

When using the framerate *get* command on a TWV640, a framerate of 0 represents 30 Hz, and a frameRate of 1 represents 60 Hz. If the framerate get command is used on TWV640i, 0 is always returned.

Table 3-8 framerate command

Subcommand	Argument	Description
default*	30 | 60*	Sets camera default frame rate to either 30 frames per second or 60 frames per second, does not change the current frame rate
get	-----	Gets number representing current frame rate.

* Not available in TWV640i.

3.9 Display Gamma Command (gamma)

This command allows manual gamma calculation mode if an eMagin VGA OLED display is used with the camera core. For more information on the purpose of gamma calculation or what the command should do, please refer to eMagin documentation.

This command allows manual gamma calculation mode.

Root Command: gamma

Sample ASCII Command: gamma *calc*

Sample Response:
```
Processing: gamma calc
Get Gamma Coeffs
gammaCoeff[0] = 428, grayLevel[0] = 255
gammaCoeff[1] = 479, grayLevel[1] = 128
gammaCoeff[2] = 517, grayLevel[2] = 64
gammaCoeff[3] = 571, grayLevel[3] = 32
gammaCoeff[4] = 645, grayLevel[4] = 16
gammaCoeff[5] = 740, grayLevel[5] = 8
gammaCoeff[6] = 857, grayLevel[6] = 4
gammaCoeff[7] = 1009, grayLevel[7] = 2
Update LUT
Write Gamma Table final to FPGA
Done!
```

Table 3-9 gamma command

Subcommand	Argument	Description
calc	-----	Calculates gamma table.
default	-----	Gamma calc uses default values.
print	-----	Print the gamma table.
shape	[0-2] 1 for normal	Set gamma shape.
vgn	-----	Sample the VGN value.
wait	(false | true)	Set the gamma wait flag.

3.10 Parameter Command (param)

Set, read and save system parameter values. Parameters are normally used for product or user specific data and are stored in Flash memory so they survive a power cycle.

A parameter value set with the set command is only used in the current session. Parameters can be saved using the save command so the system boots with the saved parameter values on the next restart or power cycle.

Root Command: param

Sample ASCII Command: param *get MICRO_IR.FOC_FRM_MODE*

Sample Response: Processing: param get MICRO_IR.FOC_FRM_MODE
PARAM MICRO_IR.FOC_FRM_MODE = 0
>

Table 3-10 param command

Subcommand	Argument	Description
get	item_name	Returns value of parameter item_name.
set	item_name	Changes parameter item_name. Changes are not saved if unit is shut down.
	value	
save	-----	Saves all current parameters to Flash memory. If unit is restarted, parameters will be set to the current values.

3.11 Change Infrared Polarity Command (polarity)

The command changes the polarity of the imagery in terms of whether white or black represents higher infrared emittance. There is also a mode in which the camera outlines regions of the image with similar infrared emittance named "edge". The edge mode does not allow switching between white-hot and black-hot modes, and displays the majority of the image as black and outlines similar emittance regions in white. In other words, edge only works in white-hot.

Root Command: polarity

Sample ASCII Command: polarity *get*

Sample Response: Processing: polarity get
Polarity is WHITE
>

Table 3-11 polarity command

Subcommand	Argument	Description
get	-----	Returns current polarity of camera.
set	white | black | edge	Sets system polarity to white, black or edge: • White: • Black: • Edge: Represents system mode that outlines regions with similar infrared emittance.
default	white | black	Sets default polarity of imagery to either white or black, the value is saved to Flash memory.
startup	-----	Run post startup polarity.

3.12 Version Command (version)

The version command returns the version of the different components of the camera.

Root Command: version

Sample ASCII Command: version *info*

Sample Response: Processing: version info
MicroIR Version Athena2 Lex 2.3.0, built on 09:29:32 May 2 2016 by Ace.Coder
>

Table 3-12 version command

Subcommand	Argument	Description
info	-----	Returns the version of the embedded software in the unit.
fpga	-----	Returns the version of the FPGA firmware in the unit.

3.13 Sharpness Command (sharpness)

Live imagery can be enhanced by either sharpening or introducing slight blurring. The default setting of the camera is no sharpness enhancement and currently there is no way to change the default.

Root Command: sharpness

Sample ASCII Command: sharpness *set 0*

Sample Response: Processing: sharpness set 0
>

Table 3-13 sharpness command

Subcommand	Argument	Description
set	0-4	Sets level of sharpness processing of the live imagery: • 0: Introduces slight imagery blurring. • 1: Leaves imagery as is. • 2 – 4: Introduces progressive level of sharpening.

3.14 Temperature Sensor Command (temp_sen)

The command returns the reading of the interface board temperature sensor in ASCII text. This command will only operate correctly if the provided interface board is used.

Root Command: temp_sen

Sample ASCII Command: temp_sen *read*

Sample Response: Processing: temp_sen read
Amb Temp = 0x305 (48.312500 deg C)
>

To convert the hexadecimal value returned to a decimal temperature, first convert the hexadecimal into a decimal number (two's complement decode) and then multiply the result by 0.0625.

Table 3-14 temp_sen command

Subcommand	Argument	Description
read	-----	Returns interface board temperature sensor output.

3.15 Output Video to Analog Command (video_analog)

Sets up the unit to output to Analog video and disables all other video modes.

Root Command: video_analog

Sample ASCII Command: video_analog *default*

Sample Response: Processing: video_analog default
>

Table 3-15 video_analog command

Subcommand	Argument	Description
default	-----	Sets default output video format to Analog.
NTSC	-----	Sets analog video output format to NTSC.
PAL	-----	Sets analog video output format to PAL.
disable	-----	Disable analog video.

3.16 Output Video to USB Command (video_usb)

Sets up the unit to output to USB video and disables all other video modes.

The raw imagery from the camera core is a 640x480 resolution (VGA), 16-bits per pixel black and white. Color optimizations and customizations, for example overlays, require signal processing that makes the imagery 24-bits per pixel. To be able to display color optimized imagery or custom overlay, the mode of the imagery is 24-bits per pixel.

Root Command: video_usb

Sample ASCII Command: video_usb *auto_overlays*

Sample Response: Processing: video_usb auto_overlays
>

Table 3-16 video_usb command

Subcommand	Argument	Description
auto	-----	Switches output video to 16-bit auto-contrast mode.
auto_overlays	-----	Switches output video to 24-bit pixel mode to allow color or custom GUI overlays.
default	-----	Sets default output video format to USB.
manual	-----	Switches output video to 16-bit, non-uniformity corrected but not contrast enhanced video
disable	-----	Disables video from being output over USB.

3.17 Output Video to 16-Bit Parallel Command (video_16bit)

- To be updated in future build

3.18 Electronic Zoom Functionality Command (zoom)

Electronic zoom makes a selected region of the image the entire image displayed; currently 0.5x, 1x, 2x, 4x and 8x settings are supported.

Root Command: zoom

Sample ASCII Command: zoom *get*

Sample Response:
```
Processing: zoom get
Centered on 320, 240, Zoom is 1X
>
```

Table 3-17 zoom command

Subcommand	Argument	Description
get	-----	Returns center point and current electronic zoom level unit is set to (0.5X, 1X or 2X).
set	zoom_level	Sets window that electronic zoom displays as entire 640 by 480 image, with new image center set at (x-center, y-center) relative to upper left corner of imagery. Valid zoom level values – ".5x", "1x"," 2x", "4x" and "8x" If x-center or y-center are not entered, their defaults of 320 and 240 are used.
	x-center (optional)	
	y-center (optional)	

3.19 Nanomotion Control Commands (nanomot)

Nanomotion is the manufacturer of the motor used to drive the shutter blade. This command allows the user to send Nanomotion commands directly to the shutter blade motor.

Root Command: nanomot

Sample ASCII Command: nanomot exec calibr

Sample Response:
```
Processing: nanomot exec calibr
Shutter Calibration Done
>
```

Table 3-18 nanomot command

Subcommand	Argument	Description
exec	calibr | sparam | rparam | getvar	Execute Nanomotion command.
	<data>*	

*See Nanomotion documentation.

3.20 Shutter Control Commands (shutter)

Controls the operation of the FPA shutter blade. The shutter setup function is part of system startup.

Root Command: shutter

Sample ASCII Command: shutter drive open

Sample Response: Processing: shutter drive open
>

Table 3-19 shutter command

Subcommand	Argument	Description
drive	open | close	Open or Close the shutter.
setup	-----	Run shutter setup function.

3.21 Overlay Commands (overlay)

Demonstrate overlay capability. A “Hello World” overlay can be displayed, moved and cleared.

Root Command: overlay

Sample ASCII Command: overlay hello

Sample Response: Processing: overlay hello
>

Table 3-20 overlay command

Subcommand	Argument	Description
enable	false | true	Enable or disable.
hello	-----	Display “Hello World” overlay.
time	-----	Display “Hello World” for 10 seconds.
move	x	Move “Hello World”.
	y	
clear	-----	Clear “Hello World”.
refresh	-----	Updates the overlay.

3.22 Crop Commands (crop)

This commands controls where you are shifting the crop of visible imagery. Imagery can only be shifted vertically by 10 rows in either direction.

Root Command: crop

Sample ASCII Command: crop *shift 10*

Sample Response: Processing: crop *shift 10*
>

Table 3-21 crop command

Subcommand	Argument	Description
shift	-10-10	Shifts cropped imagery by up to 10 rows up or down.

3.23 Vertical Video Direction Control Commands (v_dir)

Vertical video direction control allows you to flip the imagery coming out of the unit vertically.

Root Command: v_dir

Sample ASCII Command: v_dir *flip true*

Sample Response: Processing: v_dir flip true
>

Table 3-22 v_dir command

Subcommand	Argument	Description
flip	true | false	Enables or disables vertical flip of imagery

3.24 Horizontal Video Direction Control Commands (h_dir)

Horizontal video direction control allows you to flip the imagery coming out of the unit horizontally.

Root Command: h_dir

Sample ASCII Command: h_dir *flip true*

Sample Response: Processing: h_dir flip true
>

Table 3-23 h_dir command

Subcommand	Argument	Description
flip	true | false	Enables or disables horizontal flip of imagery

3.25 Delay Command Processing (delay)

Insert a delay of command processing in milliseconds.

Root Command: delay

Sample ASCII Command: delay 100

Sample Response: Processing: delay 100
>

Table 3-24 delay command

Argument	Description
ms	Delay time in milliseconds.

3.26 Help Command (help)

Root Command: help

Sample ASCII Command: help *framerate*

Sample Response: Processing: help framerate
FRAMERATE GET – Get the current frame rate
FRAMERATE DEFAULT – Set the device default frame rate
>

Table 3-25 help command

Subcommand	Argument	Description
-----	-----	Returns all available commands with short description of each command.
command	-----	Returns details about command along with available subcommands and arguments.

4. Command Quick Reference

See Table 4-1 for a list of commands with their available subcommands and arguments. Refer to the section 3 for additional information.

Table 4-1: Command List

Commands	Subcommands and Arguments	Reference
adaptive_color	roi x-center y-center radius high r_component g_component b_component low r_component g_component b_component thresh high_threshold low_threshold enable true | false read	3.1
auto_ce_lvl	default new_gain_level get set new_gain_level window x-position y-position height width	3.2
color	enable true | false	3.3
usr_overlays	enable true | false hide overlay_name list move overlay_name x-position y-position movec overlay_name x-position y-position layer overlay_name layer refresh show overlay_name timer overlay_name time_in_seconds	3.4
display	dimctl final_idrf_value idrf final_idrf_value read register_address readp register_address shift x-offset y-offset startup write register_address data writep data	3.5

Commands	Subcommands and Arguments	Reference
foc	auto off | on blade scene reset debug	3.6
framecapture	capture clear delall delete frame_index display frame_index live print norm	3.7
framerate	default 30 | 60* get	3.8
gamma	calc default print shape 0 | 1 | 2 vgn wait false | true	3.9
param	get item_name set item_name value save	3.10
polarity	get set white | black | edge default white | black startup	3.11
version	info fpga	3.12
sharpness	set 0 | 1 | 2 | 3 | 4	3.13
temp_sen	read	3.14

Commands	Subcommands and Arguments	Reference
version	info fpga	3.12
video_analog	default ntsc pal disable	3.15
video_usb	auto auto_overlays default manual	3.16
zoom	get set zoom_level x-center y-center	3.18
nanomot	exec calibr | sparam | rparam | getvar data	3.19
shutter	drive open | close setup	3.20
overlay	enable false | true hello time move x y clear refresh	3.21
crop	shift number_rows	3.22
v_dir	enable true | false	3.23
h_dir	enable true | false	3.24
delay	ms	3.25
help	command	3.26

* Not available in TWV640i.

5. Parameters

Name	Description	Information
MICRO_IR. FOC_FRM_MODE	Frame mode when FOC is activated	Data Type: int32_t Range: 0 for default legacy, 1 for freeze frm. See Enumeration type, FOC_Frm_Mode_t Default: 0 Engineering Units: None
MICRO_IR. DEFAULT_VIDEO	Default video output	Data Type: int32_t Range: 0 for USB, 1 for CameraLink, 2 for Analog Default: 0 Engineering Units: None
MICRO_IR. DEFAULT_CONTRAST	Default contrast mode	Data Type: int32_t Range: 0 for Manual, 1 for Auto, 2 for Auto Overlays Default: 1 Engineering Units: none
MICRO_IR. DEFAULT_CE_GAIN	Default contrast enhancement gain value.	Data Type: int32_t Range: 0 to 12 (0x0 to 0xC) Default: 5 Engineering Units: none
MICRO_IR. DEFAULT_CROP_SHIFT	Default crop shift value	Data Type: int32_t Range: -10 to 10 Default: 0 Engineering Units: Rows
MICRO_IR. COLOR_EN	Enable color LUT by default	Data Type: int32_t Range: 0 false, 1 true Default: 0 Engineering Units: none
MICRO_IR.V_DIR_FLIP	Enable vertical flip by default	Data Type: int32_t Range: 0 false, non-zero true

		Default: 0 Engineering Units: none
MICRO_IR.H_DIR_FLIP	Enable horizontal flip by default	Data Type: int32_t Range: 0 false, non-zero true Default: 0 Engineering Units: none
MICRO_IR. SHARPNESS	Value to initialize sharpness	Data Type: int32_t Range: 0 to 4 Default: 1 Engineering Units: 0 is blurred, 4 is very sharp
MICRO_IR. BAUD	Specifies the baud rate of the UART	Data Type: int32_t Range: 9600 * number Default: 0 Engineering Units: 9600 baud increments
MICRO_IR. MIN_SPLASH	Specifies the minimum time to display splash screen	Data Type: uint32_t Range: uint32 Default: 1000 Engineering Units: ms
MICRO_IR. ANALOG_MODE	Specifies the mode for analog video output, NTSC or PAL.	Data Type: int32_t Range: 0 (NTSC) or 1 (PAL) Default: 0 (NTSC) Engineering Units: boolean
ADV739X. SUBCAR_FREQ_NTSC	32-bit decimal representation of sub carrier frequency.	Data Type: int32_t Range: int32 Default: Engineering Units: hz
ADV739X. SUBCAR_FREQ_PAL	32-bit decimal representation of sub carrier frequency.	Data Type: int32_t Range: int32 Default: Engineering Units: hz
LWNR. SCENE_EN	Enable scene in LWNR.	Data Type: int32_t Range: 0 (disable) or 1 (enable)

		Default: 1 (enable) Engineering Units: boolean
LWNR. COLUMN_EN	Enable column in LWNR.	Data Type: int32_t Range: 0 (disable) or 1 (enable) Default: 1 (enable) Engineering Units: boolean
LWNR. RANK_ORDER_EN	Enable rank order in LWNR.	Data Type: int32_t Range: 0 (disable) or 1 (enable) Default: 1 (enable) Engineering Units: boolean