

AGEMA 550/570 Control 1.00

Reference Guide



Contents

AGEMA 550/570 Camera Control Reference Guide	3
About the Camera Control.....	3
Properties.....	5
AmbientTemperature Property.....	5
AtmosphericTemperature Property	5
AutoConnect Property	5
Battery Property.....	5
CalibTitle Property	6
Cooler Property	6
Date Property.....	6
Device Property	6
DeviceCount Property.....	7
Digital Property.....	7
Emissivity Property.....	7
EstimTrans Property.....	7
FieldMode Property	8
Filter Property	8
FilterArtNo Property.....	8
FilterName Property	8
FilterSerialNo Property.....	9
Focus Property.....	9
LensName Property	9
LensSerialNo Property.....	9
LensArtNo Property.....	10
Level Property	10
NoiseReduction Property	10
ObjectDistance Property	11
Palette Property	11
Port Property	11
Power Property	11
Range Property	12
ReferenceTemperature Property.....	12
RelativeHumidity Property.....	12
ScannerArtNo Property.....	12
ScannerName Property	13
ScannerSerialNo Property.....	13
SeqFile Property	13
SeqIndex Property	13
SeqName Property.....	13
SeqPath Property	14
Shutter Property.....	14
Span Property	14
StartCondition Property.....	15
StartTime Property	15
StartValue Property	15
Status Property	16
StopCondition Property	16

StopValue Property.....	16
StoreCondition Property	16
StoreValue Property	17
TemperatureUnit Property	17
TrigSource Property	17
Zoom Property.....	18
Version Property	18
Methods.....	19
AboutBox Method	19
AutoAdjust Method	19
Button Method	19
Connect Method	19
Disconnect Method.....	20
ExternalCorrection Method	20
GetAbsImage Method.....	21
GetDeviceName Method.....	21
GetDeviceStatus Method	22
GetFile Method	23
GetImage Method.....	23
GetNoise Method.....	23
GetRangeLimits Method.....	24
GetScale Method	24
Optimize Method	24
Recording Method.....	24
Save Method	25
SetNoise Method	25
SetScale Method.....	26
ShowCameraInfo Method	26
ShowDeviceInfo Method	26
Events.....	27
CardEvent Event	27
CriticalEvent Event	27
ScannerEvent Event	27

Index

28

AGEMA 550/570 Camera Control Reference Guide

About the Camera Control

Description

This Camera Control communicates with an AGEMA 550/570 camera. You can use it to switch measurement range, control the image filtering, view camera status etc. You can also retrieve images from it.

File names

CAM550A.TLB	Type library
CAM550A.OCX	Binary load file

Remarks

Its properties, methods and events

This control has methods, properties and events by which you can interact with it. The methods and properties allow you to give commands to the control and to retrieve information from it. The events can tell you that something important has happened, for instance that the user has pressed the disconnect button.

The distinction between methods and properties is quite subtle. Properties can be assigned one single value, which methods normally can't. Methods can take parameters, which properties normally don't. The intermediate case, a property with parameters, exists and is handled by this guide as a method because that is the way in which it is regarded by Visual Basic.

This reference guide lists the properties, methods and events in separate sections, in alphabetic order.

Copyright

© by FLIR Systems AB, 1996-1998. All rights reserved worldwide. No parts of the software including source code may be reproduced, transmitted, transcribed or translated into any language or computer language in any form or by any means, electronic, magnetic, optical, manual or otherwise, without the prior written permission of FLIR Systems AB, Box 3, S-182 11 Danderyd, Sweden.

This reference guide may not, in whole or part, be copied, photocopied, reproduced, translated or transmitted to any electronic medium or machine readable form without prior consent, in writing, from FLIR Systems AB.

Limitation of Liability

FLIR Systems computer software is supplied as is and FLIR accepts no responsibility or liability and makes no warranties whatsoever whether expressed or implied with regard to the same, including but not limited to its quality, performance, merchantability or fitness for a particular purpose.

FLIR specifically disclaims the implied warranties of merchantability and fitness for a particular purpose.

FLIR shall not be liable for any direct, indirect, special, incidental or consequential damages, whether based on contract, tort or any other legal theory.

Trademarks

Thermovision® is a registered trademark of FLIR Systems AB.

Microsoft® and MS-DOS® are registered trademarks, and Windows™ is a trademark of Microsoft Corporation.

Quality Assurance

The Quality Management System under which these products are developed and manufactured has been certified in accordance with the standard for ISO 9001.

Properties

AmbientTemperature Property

Gets or sets the object ambient temperature in Kelvin. Affects the temperature calculations.

Syntax

object.AmbientTemperature = *ambtemp*
ambtemp = *object.AmbientTemperature*

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>ambtemp</i>	Float	Value range between 200 and 5811 K

AtmosphericTemperature Property

Gets or sets the atmospheric temperature in Kelvin. Affects the temperature calculations.

Syntax

object.AtmosphericTemperature = *atmtemp*
atmtemp = *object.AtmosphericTemperature*

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>Atmtemp</i>	Float	Value range between 200 and 5811 K

AutoConnect Property

Property for connecting automatically when the control object is created.

Syntax

object.AutoConnect = *flag*
flag = *object.AutoConnect*

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>flag</i>	Boolean	TRUE=Connect to camera upon control creation FALSE=Connect when ordered

Battery Property

Returns the current battery status as a text string.

Syntax

status = *object.Battery*

<u>Item</u>	<u>Description</u>	<u>Value</u>
-------------	--------------------	--------------

<i>Object</i>	Control object name	For example Ctrl500.
<i>Status</i>	String	õökõ õlowõ or õemptyõ

CalibTitle Property

Gets the calibration title of the camera.

Syntax

title = *object*.**CalibTitle**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>title</i>	String	Calibration title

Cooler Property

This property controls the camera detector cooler (AGEMA 550) function.

Syntax

object.**Cooler** = *state*

state = *object*.**Cooler**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>State</i>	String	õonõ õstdbyõ or õoffõ

Date Property

Sets or gets the camera date and time.

Syntax

object.**Date** = *date*

date = *object*.**Date**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>date</i>	DATE	Camera date and time

Device Property

Sets or gets the device type for image retrieval. You must select a device type before attempting to connect to the device.

Syntax

object.**Device** = *device*

device = *object*.**Device**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Device</i>	Short	0=PC Card 1=ITEX Frame grabber

.....

DeviceCount=Device not selected

DeviceCount Property

Gets the maximum number of supported devices.

Syntax

devcount = *object*.**DeviceCount**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Devcount</i>	Short	Number of supported devices

Digital Property

Controls the camera digital mode. The digital mode must be switched on in order to enable the digital link. The digital mode is automatically switched on when calling the Connect Method.

Syntax

object.**Digital** = *mode*

mode = *object*.**Digital**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Mode</i>	String	õonõ or õoffõ

Emissivity Property

Gets or sets the camera emissivity constant. Affects the temperature calculations.

Syntax

object.**Emissivity** = *eps*

eps = *object*.**Emissivity**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>eps</i>	Float	Value range between 0.01 and 1.0

EstimTrans Property

Gets or sets the estimated transmission. Affects the temperature calculations.

Syntax

object.**EstimTrans** = *tao*

tao = *object*.**EstimTrans**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.

<i>tao</i>	Float	0.0 ó 1.0 (Zero means use calculated transmission)
------------	-------	--

FieldMode Property

Controls the camera field mode. Relevant only for AGEMA 550 cameras.

Syntax

object.**FieldMode** = *mode*
mode = *object*.**FieldMode**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Mode</i>	Long	0=Interpolated mode. Half of the lines are interpolated 1=Dual freeze. Same as 0 except that frozen images uses dual field mode. 2=Dual field mode. Half of the lines are from the previous frame.

Filter Property

Controls the camera filter. Use the Range property to change measurement range. Changing measurement range may cause the filter to be activated or deactivated.

Syntax

object.**Filter** = *mode*
mode = *object*.**Filter**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Mode</i>	String	õonõ or õoffõ

FilterArtNo Property

Gets the part number of the current camera filter.

Syntax

filterpn = *object*.**FilterArtNo**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>filterpn</i>	String	For example õ194176õ

FilterName Property

Gets the name of the current camera filter.

Syntax

filtername = *object*.**FilterName**

<u>Item</u>	<u>Description</u>	<u>Value</u>
-------------	--------------------	--------------

<i>object</i>	Control object name	For example Ctrl500.
<i>filtername</i>	String	For example öNOFö

FilterSerialNo Property

Gets the serial number of the current camera filter.

Syntax

filtersn = *object*.**FilterSerialNo**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>filtersn</i>	String	For example ö34ö

Focus Property

This property controls the camera focus function.

Syntax

object.**Focus** = *state*

state = *object*.**Focus**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>State</i>	String	öupö = Focus towards infinity ödownö = Focus towards near focus öoff ö= Stop focusing

LensName Property

Gets the name of the camera lens.

Syntax

lensname = *object*.**LensName**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>lensname</i>	String	For example öFOV24ö

LensSerialNo Property

Gets the serial number of the camera lens.

Syntax

sn = *object*.**LensSerialNo**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>sn</i>	String	For example ö1004ö

LensArtNo Property

Gets the part number of the camera lens.

Syntax

pn = *object*.**LensArtNo**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Pn</i>	String	For example ö194579ö

Level Property

Sets or gets the image level in raw pixels. Use SetScale Method to set the scale limits in temperature.

Syntax

object.**Level** = *level*

level = *object*.**Level**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>level</i>	Long	Value range is 0 to 65535.

Line Property

Set the control in line scanning mode. Please note that the methods GetAbsImage, GetAbsROI and GetAbsLine does not work in line scanning mode.

Syntax

object.**Line** = *line*

line = *object*.**Line**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>line</i>	Short	0=Normal image scanning mode 1-240=Line scanning mode

NoiseReduction Property

Sets or gets the camera noise reduction factor.

Syntax

object.**NoiseReduction** = *factor*

factor = *object*.**NoiseReduction**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Factor</i>	Short	Value range 0 to 7

ObjectDistance Property

Gets or sets the camera object distance in meters. Affects the temperature calculations.

Syntax

object.**ObjectDistance** = *distance*
distance = *object*.**ObjectDistance**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Distance</i>	Float	Value range between 0 and 9999 m

Palette Property

Sets or gets the current camera palette.

Syntax

object.**Palette** = *string*
string = *object*.**Palette**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>String</i>	String	õironö or õgreyö

Port Property

Sets or gets the serial port for communicating with the camera.

Syntax

object.**Port** = *port*
port = *object*.**Port**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>port</i>	Short	0-3 (COM1-COM4)

Power Property

Sets or gets the device power status.

Syntax

object.**Power** = *flag*
flag = *object*.**Power**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Flag</i>	Boolean	TRUE = Power On, FALSE=Power Off

Range Property

Sets or gets the camera logical measurement range.

Syntax

object.**Range** = *range*
range = *object*.**Range**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Range</i>	Short	Depends on camera calibration. Value range is 0 - 5.

ReferenceTemperature Property

Gets or sets the camera reference temperature in Kelvin.

Syntax

object.**ReferenceTemperature** = *reftemp*
reftemp = *object*.**ReferenceTemperature**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Reftemp</i>	Float	Value range >= 0

RelativeHumidity Property

Gets or sets the relative humidity. Affects the temperature calculations.

Syntax

object.**RelativeHumidity** = *relhum*
relhum = *object*.**RelativeHumidity**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Relhum</i>	Float	Value range between 0.0 and 0.99

ScannerArtNo Property

Gets the part number of the camera.

Syntax

pn = *object*.**ScannerArtNo**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Pn</i>	String	For example ö193567ö

ScannerName Property

Gets the model name the camera.

Syntax

name = *object*.**ScannerName**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Name</i>	String	For example öAGEMA 570ö

ScannerSerialNo Property

Gets the serial number of the camera.

Syntax

sn = *object*.**ScannerSerialNo**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Sn</i>	String	For example ö340001ö

SeqFile Property

The **SeqFile** property sets or gets image file mode.

Syntax

object.**SeqFile** = *flag*

flag = *object*.**SeqFile**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Flag</i>	Boolean	TRUE=all images in one file (.SEQ) FALSE=images in separate files (.IMG)

SeqIndex Property

The **SeqIndex** property sets or gets image file index counter, i.e. the number which is added to the sequence name to form the image or sequence file name.

Syntax

object.**SeqIndex** = *counter*

counter = *object*.**SeqIndex**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Counter</i>	Long	File name index counter.

SeqName Property

The **SeqName** property sets or gets the image sequence name. See also **SeqIndex**.

Syntax

object.**SeqName** = *name*
name = *object*.**SeqName**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Name</i>	String (BSTR)	Image sequence name.

SeqPath Property

The **SeqPath** property sets or gets the directory where images should be stored.

Syntax

object.**SeqPath** = *path*
path = *object*.**SeqPath**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Path</i>	String (BSTR)	Storage path (for image sequence files).

Shutter Property

The **Shutter** property controls the camera shutter.

Syntax

object.**Shutter** = *state*
state = *object*.**Shutter**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>State</i>	String	õonö or õoffö

ShowToolTip Property

The **ShowToolTip** property enables or disables the display of tool tips.

Syntax

object.**ShowToolTip** = *flag*
state = *object*.**ShowToolTip**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>Flag</i>	Boolean	TRUE = Tool tips visible FALSE=Tool tips disabled

Span Property

Sets or gets the display range in raw pixels. Use SetScale Method instead.

Syntax

object.**Span** = *span*
span = *object*.**Span**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>span</i>	Long	Value range is 0 to 65535.

StartCondition Property

Sets or gets the start recording condition.

Syntax

object.**StartCondition** = *condition*
condition = *object*.**StartCondition**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>condition</i>	Short	0=Start on software trig 1 (see Recording Method) 1=Start on software trig 2 (see Recording Method) 2=Start at specified time (see StartTime Property) 3=Start at external trig (see TrigSource Property)

StartTime Property

Sets or gets the start time for image recordings.

Syntax

object.**StartTime** = *time*
time = *object*.**StartTime**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>value</i>	DATE	Start recording time if StartCondition is 2.

StartValue Property

Sets or gets the delay time (in seconds) for external trig start recording condition.

Syntax

object.**StartValue** = *value*
value = *object*.**StartValue**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>value</i>	double	Delay in seconds if StartCondition is external trig

Status Property

Not used.

StopCondition Property

Sets or gets the stop recording condition.

Syntax

object.**StopCondition** = *condition*

condition = *object*.**StopCondition**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>condition</i>	Short	0=Stop at software trig 1 (see Recording Method) 1=Stop after N recorded images (see StopValue) 2=Stop after specified time interval (see StopValue) 3=Stop at software trig 2 (see Recording Method) 4=Stop at external trig (see TrigSource Property)

StopValue Property

Syntax

object.**StopValue** = *value*

value = *object*.**StopValue**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>value</i>	double	Delay in seconds if StopCondition is external trig (4) Recording duration in seconds if StopCondition is 2.

StoreCondition Property

Sets or gets the recording condition.

Syntax

object.**StoreCondition** = *condition*

condition = *object*.**StoreCondition**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>condition</i>	Short	0=Record at highest possible speed 1=Record at time interval (see StoreValue Property) 2=Record at software trig 2 (see Recording Method) 3=Record every N:th image (see StoreValue Property) 4=Record at external trig (see TrigSource Property)

StoreValue Property

Sets or gets recording interval in images. If set to one, record all images, if set to 2 then record only every 2:nd image and so on.

Syntax

object.StoreValue = *value*
value = *object.StoreValue*

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>value</i>	double	Image interval if StoreCondition is 3. Recording interval in seconds if StoreCondition is 1.

TemperatureUnit Property

Sets or gets the temperature unit.

Syntax

object.TemperatureUnit = *unit*
unit = *object.TemperatureUnit*

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>unit</i>	Temperature unit (short integer)	0 Celsius 1 Fahrenheit 2 Kelvin

TrigSource Property

Sets or gets the external trig source type.

Syntax

object.TrigSource = *sourcetype*
sourcetype = *object.TrigSource*

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>sourcetype</i>	Short	0=Use parallell interface as trig source 1=Use COM 1, CTS signal as trig source 2=Use COM 2, CTS signal as trig source 3=Use COM 3, CTS signal as trig source 4=Use COM 4, CTS signal as trig source 5=Use LPT 1 (378h), DB-17 as trig source 6=Use LPT 2 (278h), DB-17 as trig source

Remarks

Option 5 and 6 are valid only for Windows 95 platforms.

Zoom Property

Sets or gets the camera zooming factor.

Syntax

object.**Zoom** = *zoom*

zoom = *object*.**Zoom**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>zoom</i>	Float	>= 1.0, <= 4.0

Remarks

The zooming is performed around the centre of the IR image.

Version Property

Returns the camera control version string.

Syntax

version = *object*.**Version**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>version</i>	String	E.g ðAGEMA 550/570 Camera Control, ver 1.01ð

Methods

AboutBox Method

You can use the **AboutBox** method to bring up the About box of the camera Control

Syntax

object.**AboutBox**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.

AutoAdjust Method

Perform a camera internal image correction using the shutter.

Syntax

status = *object*.**AutoAdjust**(*flags*)

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>status</i>	Function return status.	0 = OK 10 = Not connected
<i>flags</i>	Short (bitmask)	Bit 1=Adjust camera level Bit 2=Adjust camera span Bit 3=Use image histogram for adjustment Bit 4=No shutter offset map

Button Method

Simulate the camera buttons.

Syntax

status = *object*.**Button**(*str*)

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>status</i>	Function return status.	0 = OK 10 = Not connected
<i>str</i>	String	First arg: ent / esc / f1 / f2 / u / d / r / l Optional second arg: p / r Example: ðentö (press and release enter button) Example: öf1,pö (press f1 button)

ClearLineBuffer Method

Clear the line buffer and set all pixels to zero.

Syntax

object.**ClearLineBuffer**

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.

Connect Method

Connect with the camera using the specified port or let the control decide which port to use.

Syntax

status = *object*.**Connect**(*port*)

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>status</i>	Function return status.	0 = OK See GetDeviceStatus Method for device error codes.
<i>port</i>	Serial port to use for connection.	0 = Use default port 1-4 = Use COM1-COM4

Disconnect Method

Disconnect the camera and exit the digital transfer mode.

Syntax

status = *object*.**Disconnect**()

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example IRIImage1.
<i>status</i>	Function return status.	0 = OK 10 = Not connected

ExternalCorrection Method

Perform a camera external image correction.

Syntax

status = *object*.**ExternalCorrection**(*images*)

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>status</i>	Function return status.	0 = OK 10 = Not connected
<i>images</i>	Long	Number of images used in correction. Default is 5.

GetAbsImage Method

Get an image with only pixel data from the camera. If successful, a memory block will be allocated for the image pixels, and it is the responsibility of the caller to deallocate the memory. The pixels are compensated for temperature drift and detector non-linearity. The image format is 320 x 240 pixels and each pixel is 16 bit wide. This method does not work in line scanning mode.

Syntax

status = *object*.**GetAbsImage**(*pHandle*)

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>status</i>	Function return status.	0 = Successful image creation See GetImage Method for image acquisition error codes and GetDeviceStatus Method for device error codes.
<i>pHandle</i>	Pointer to long	Handle to memory block

GetAbsROI Method

Get an image region of interest with pixel data from the camera. If successful, a memory block will be allocated for the image region pixels, and it is the responsibility of the caller to deallocate the memory. The pixels are compensated for temperature drift and detector non-linearity. The caller determines the image format and each pixel is 16 bit wide. This method does not work in line scanning mode.

Syntax

status = *object*.**GetAbsROI**(*pHandle*, *x*, *y*, *dx*, *dy*)

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>status</i>	Function return status.	0 = Successful image creation See GetImage Method for image acquisition error codes and GetDeviceStatus Method for device error codes.
<i>pHandle</i>	Pointer to long	Handle to memory block
<i>x</i>	Short	Upper left x coordinate of region
<i>y</i>	Short	Upper left y coordinate of region
<i>dx</i>	Short	Region width
<i>dy</i>	Short	Region height

GetAbsLine Method

Get an image line with pixel data from the camera. If successful, a memory block will be allocated for the image line pixels, and it is the responsibility of the caller to deallocate the memory. The pixels are compensated for temperature drift and detector non-linearity. The image line is 320 pixels wide and each pixel is 16 bit wide. This method does not work in line scanning mode.

Syntax

status = *object*.**GetAbsLine**(*pHandle*, *line*)

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>status</i>	Function return status.	0 = Successful image creation See GetImage Method for image acquisition error codes and GetDeviceStatus Method for device error codes.
<i>pHandle</i>	Pointer to long	Handle to memory block
<i>line</i>	Short	Image line (0-239)

GetDeviceName Method

Get the name of a device. Use the property DeviceCount to get total number of supported devices.

Syntax

name = *object*.**GetDeviceName**(*device*)

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>name</i>	String (BSTR)	Device name.
<i>device</i>	Short	Device index (zero based).

GetDeviceStatus Method

Get device status.

Syntax

status = *object*.**GetDeviceStatus**(*device*)

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>status</i>	Short	0 = Device is OK 1 = Device is not present 2 = Device is busy 3 = Device driver is missing 4 = Device driver version is incorrect 5 = Failed to load device firmware 6 = Device configuration error 7 = No device is selected 8 = Failed to load device driver 9 = Failed to establish serial connection 10 = Device is not connected
<i>device</i>	Short	Device index (zero based).

GetFile Method

Get the name of a recorded image file from the list of recordings. Use the method Recording to query the current number of recorded files.

Syntax

fname = *object*.**GetFile**(*index*)

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>fname</i>	String (BSTR)	Image file name.
<i>index</i>	Long	Index in recording file list

GetImage Method

Get an image from the camera in FLIR proprietary format. If successful, a memory block will be allocated for the image, and it is the responsibility of the caller to deallocate the memory. The image can be passed into the IR image control for image display and analysis.

Syntax

status = *object*.**GetImage**(*pHandle*)

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>Object</i>	Control object name	For example Ctrl500.
<i>status</i>	Function return status.	0 = Successful image creation 11 = Cannot allocate image buffer 12 = Invalid image 13 = Device hardware error 14 = Timeout waiting for image 15 = Camera configuration error 16 = Image acquisition aborted See GetDeviceStatus Method for device error codes.
<i>pHandle</i>	Pointer to long	Handle to image memory block

GetNoise Method

Get camera noise reduction factor and level.

Syntax

status = *object*.**GetNoise**(*pfactor*, *plevel*)

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>status</i>	Function return status.	0 = OK
<i>pfactor</i>	Pointer to short	0 (off) to 15 (max reduction)
<i>plevel</i>	Pointer to short	0 to 7 (max level)

GetRangeLimits Method

Get the current camera range limit information.

Syntax

phy_range = *object*.**GetRangeLimits**(*range*, *pLo*, *pHi*, *pFilter*)

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>phy_range</i>	Short	Camera physical measurement range
<i>range</i>	Short	Logical measurement range (0 ó 5).
<i>pLo</i>	Pointer to long integer	Returns minimum range temperature. Zero if range does not exist.
<i>pHi</i>	Pointer to long integer	Returns maximum range temperature. Zero if range does not exist.
<i>pFilter</i>	Pointer to boolean	Returns TRUE if filter is used.

GetScale Method

Get the camera display range in temperature (Kelvin).

Syntax

status = *object*.**GetScale**(*pLo*, *pHi*)

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>status</i>	Function return status.	0 = OK
<i>pLo</i>	Pointer to double	Returns low display range in Kelvin.
<i>pHi</i>	Pointer to double	Returns high display range in Kelvin.

Optimize Method

Optimize image retrieval for PC Card device.

Syntax

object.**Optimize**(*type*, *wait_states*)

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>type</i>	Short	0 = PCIC compatible adapter 1 = Cirrus Logic PCIC compatible adapter
<i>wait_states</i>	Short	Memory wait states (0-3). Default is 3.

Recording Method

Perform an image recording action.

Syntax

status = *object*.**Recording**(*action*)

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>status</i>	Function return status.	Return value depends on parameter <i>action</i> . The query functions 5 and 9 will return 0 (FALSE) or 1 (TRUE). The query function 8 will return a value >= 0.
<i>action</i>	Short	0=Signal software trig 1 1=Signal software trig 2 2=Pause recording 3=Stop recording 4=Update recording parameters 5=Query if recording is active 6=Abort recording 7=Clear list of created files 8=Query number of recorded files 9=Query if recording is paused 10=Show recording status 11=Hide recording status 12=Resume recording 13=Scan current directory for existing recordings 14=Enable image retrievals during recording 15=Disable image retrievals during recording 16=Signal external trig

Save Method

Save an image to file (FLIR proprietary format).

Syntax

status = *object*.**Save**(*fname*)

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>status</i>	Function return status.	0 = OK 21 = General file error 22 = File access denied 23 = File path error 24 = Media full See also GetImage Method for image acquisition error codes.
<i>fname</i>	String (BSTR)	File name (absolute path specified).

SetNoise Method

Set camera noise reduction factor and level.

Syntax

status = *object*.**SetNoise**(*factor*, *level*)

<u>Item</u>	<u>Description</u>	<u>Value</u>
-------------	--------------------	--------------

<i>object</i>	Control object name	For example Ctrl500.
<i>status</i>	Function return status.	0 = OK 10 = Not connected
<i>factor</i>	Short	0 (off) to 15 (max reduction)
<i>level</i>	Short	0 to 7 (max level)

SetScale Method

Set the camera display range in temperature (Kelvin).

Syntax

status = *object*.**SetScale**(*dwLo*, *dwHi*)

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.
<i>status</i>	Function return status.	0 = OK
<i>dwLo</i>	Double precision	Low display range in Kelvin.
<i>dwHi</i>	Double precision	High display range in Kelvin.

ShowCameraInfo Method

Show camera information panel.

Syntax

object.**ShowCameraInfo**()

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.

ShowDeviceInfo Method

Show device information panel. A device must be selected before calling this function. See the Device Property.

Syntax

object.**ShowDeviceInfo**()

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500.

Events

CardEvent Event

The **CardEvent** event occurs when a device changes state.

Syntax

Private Sub *object*_CardEvent ([*id* **As Short**])

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500
<i>id</i>	Event identifier	1 = A device is ready for connection 2 = Selected device is connected 3 = Device is disconnected

CriticalEvent Event

The **CriticalEvent** event occurs when the camera makes an important state change. For example to signal low battery voltage.

Syntax

Private Sub *object*_CriticalEvent ([*id* **As Short**])

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500
<i>id</i>	Event identifier	0 = Cooling state change (see Cooler Property) 1 = Battery state change (see Battery Property) 2 = Digital link state change (see Digital Property)

ScannerEvent Event

The **ScannerEvent** event occurs whenever certain camera data has been updated. The event is fired when the camera measurement range or any object parameters is changed.

Syntax

Private Sub *object*_ScannerEvent ([*id* **As Short**], *status* **As Short**)

<u>Item</u>	<u>Description</u>	<u>Value</u>
<i>object</i>	Control object name	For example Ctrl500
<i>id</i>	Event identifier	0x602 = Object distance 0x603 = Ambient temperature 0x604 = Emissivity 0x605 = Atmospheric temperature 0x606 = Relative humidity 0x60c = Reference temperature 0x702 = Temperature range 0x21f = Auto adjust event
<i>status</i>	Event status	Not used. Always 0.

Index

C

Copyright..... 3

D

Description of the camera control..... 3

F

File names..... 3

L

License..... 3

Limitation of Liability 4

Q

Quality Assurance 4

T

Trademarks..... 4

U

Usage..... 3