Date
2016-05-23

Basic ICD FLIR IR Camera - PC

Contents

1 Connectivity Overview

1.1 Physical interfaces

From the camera, two physical interfaces for data transfer can be supported.

- Ethernet
- USB device

Analog video or DVI may also exist, but is not considered a data transfer interface.

1.2 Low level protocols

On the mentioned physical interfaces, it is possible to run different low level protocols.

1.2.1 Ethernet

For Ethernet, only TCP/IP is supported. The camera should seamlessly work on any LAN, provided that a proper IP adress, netmask and possibly gateway is set in the camera.

No FLIR specific device drivers are required, so any type of computer and operating systems supporting TCP/IP should work.

1.2.2 USB

IP, USB Video Class and USB Mass Storage Device can be supported for USB.

For USB Video Class and USB Mass Storage Device, no FLIR specific device drivers are required. FLIR can provide drivers for IP via USB.

1.3 Functionality

The two main ways of controlling with the camera are through the command control interface and through the resource control interface. Image streaming, file transfer and other functionality is provided through the Services interface.

1.3.1 Command control

Commands can be given to the command shell in the camera. Some commands are standard commands like for instance `cd` that operate directly on the camera themselves. Others rely on camera resources (see below), for instance the `level` command. It is possible to run independent instances of the command shell with the telnet or ssh service on established TCP/IP connections.

1.3.2 Resource control

A lot of (but not all) software functionality is exposed through something called “software resources”. Those familiar with the Microsoft® Windows registry will recognize the idea.

However, in the camera a resource node can also represent a software function that, upon read or write, actively interacts with the camera software.

- Resources are organized in a hierarchy (like in a tree).

- Resource nodes can be data holders (of for instance calibration data).
- Resource nodes can be connected to hardware (like for instance to internal temperature sensor values).
- Resource nodes can be connected to software (like for instance to spotmeter values)
- Resource nodes have a type (for instance double, int32, ascii) and certain attributes (read-only, read/write).
- Resources can be reached through commands, like `rls` and `rset`, and through the IP FLIR Resource socket service.

1.3.3 Services

Independent of physical interface, it is possible to access the system using TCP/IP with the exposed services described here: telnet, ftp, http, CIFS, FLIR resource socket, FLIR RTP. More than one service and possibly more than one instance of the service may be run at the same time "simultaneously" for instance 2 telnet's and one ftp session together towards the same camera.

On a USB device interface, USB Video Class and USB Mass storage device are also available as services.

1.3.3.1 telnet

Command control used mainly for manual typing. Typical clients are the standard telnet command on a PC or "teraterm".

1.3.3.2 ssh

Command control over secure link used mainly for manual typing. This protocol is only available on cameras using Linux OS.

1.3.3.3 ftp

File transfer to/from the camera using a ftp client software on a PC. Typical clients are the standard ftp command on a PC. Use ftp login with user `flir` and password `3vlig`.

1.3.3.4 sftp

File transfer to/from the camera using sftp client software on a PC. Use sftp login with user `fliruser` and password `3vlig`. This protocol is only available on cameras using Linux OS.

1.3.3.5 http

This is the web service. Typical clients are Microsoft Internet Explorer and Firefox.

1.3.3.6 FLIR resource socket

It is possible to directly read and write nodes of the software resource tree from a PC. Standard sockets are used, but there are no standard client programs available.

1.3.3.7 Image streaming

1.3.3.7.1 Set-up

The available streams are described and presented using SDP (Session Description Language, RFC 2327). The SDP content is accessed using RTSP (Real Time Streaming Protocol, RFC 2326).

The RTSP/DESCRIBE command lists the available streams.

MPEG4	Compressed video in three sizes. (640x480, 320x240, 160x128)
FCAM	FLIR usage only.
raw	IR-signal or temperature linear IR-image (two types) in two or three sizes. (640x480, 320x240, 160x120) The pixels are transferred in network byte order (Big Endian).

Table 1

The RTSP/SETUP command establishes an RTP-based transport session using one of the formats.

The RTSP/GET_PARAMETER command gets the current "framerate" and "format".

The RTSP/SET_PARAMETER command sets the current "framerate" and "format".

The RTSP/PLAY and RTSP/PAUSE commands control the image stream.

The RTSP/TEARDOWN command closes the transport session.

1.3.3.6.2 MPEG4

The MPEG4 streams use RTP/UDP/IP for transport (RTP=Real time Transport Protocol, RFC 1889). The MPEG bit stream is packetized according to RFC 3016.

On the receiver side, FLIR supplies a DirectShow component (Win32, PC platform) which is able to receive the MPEG4 bit stream. The component is able to receive the MPEG4 bit stream according to RFC 3016. The bit stream is reassembled and forwarded as video samples of FOURCC type MP4V. Several MPEG4 decoders may be used, for example from 3ivx ($7 per license) or a free decoder (ffdshow). The DirectShow component can be used by any application that wishes to display the MPEG4 video stream.

1.3.3.6.3 IR streams

The raw IR streams use RTP/UDP/IP for transport (RTP = Real time Transport Protocol, RFC 1889). The transport format is according to RFC 4175 (RTP Payload Format for Uncompressed Video). The raw image frame rates are up to about 7.8 Hz.

These raw formats are provided:

0	16 bit un-compressed IR image linear in signal
1	16-bit un-compressed IR image linear in temperature, resolution 0.1 K (range 0 -6553 K)
2	16-bit un-compressed IR image linear in temperature, resolution 0.01 K (range 0 -655 K)

Table 2

On the client side, FLIR supplies a DirectShow component (Win32, PC platform) which is able to setup and receive the IR streams. The IR stream is reassembled and forwarded as samples of FOURCC type Y160 (this FOURCC type is only preliminary at this stage). The DirectShow component can be used by any application that wishes to display the IR stream or want to grab samples from the stream.

1.3.3.7.2 USB Video Class streams

Both IR and video streaming can exist. The streams comply with the USB Video Class specification version 1.0 and are controlled from the receiving end. The image frame rate on an USB 2 connection

may reach 25 Hz. The FOURCC type UYVY (Standard YUV 4:2:2 video) is provided.

1.3.3.7.3 DHCP

The camera supports the client part of the Dynamic Host Configuration Protocol (DHCP).

1.3.3.8 Remote detection

1.3.3.8.1 Multicast DNS (Bonjour)

To query Bonjour for local FLIR IR cameras, use:

- Service name: flir-ircam
- Protocol type: _tcp
- Domain: local

Name	Value (example)	Explanation
model	S	
ID	A320	Camera ID
GID	Gen A	Generic ID
SI	FFF_RTSP	Streaming Interface
SIV	1.0.0	Streaming interface version
CI	RTREE	Command interface
CIV	1.0.0	Command interface version

Table 3: TXT records

For more information, see http://www.dns-sd.org

2 Command Protocol Specification

2.1 Introduction

This section describes the command interface used by the telnet server in the camera.

Depending on the camera model, all commands may not be available or functional in a particular camera.

2.2 Syntax

The basic idea is that the command protocol shall be easy to interpret by both human users and machines. The commands and any response that are sent over the communication channel, only uses ASCII printable characters. The user interface is not so different from several well-known standards used for terminal sessions (e.g. UNIX-shell command interpreter). In fact, this protocol is implemented with the help of DOS shell commands.

2.2.1 Commands

The commands (or system input) follow the syntax:

`<command><WS><parameters><CR>`

where:

`<command>`	is the character string identifying one of the commands described in detail below,
`<parameters>`	specifies one or more parameters to the command, all separated by <WS>, and possibly identified with special flag parameters e.g. -y.
`<WS>`	white space - command and parameters are separated with one or many ASCII Space or Tab characters,
`<CR>`	specifies the byte ASCII control character CR that ends each command request.

The command can also be a query for the current setting:

`<command><CR>`

2.2.1.1 Parameters

Valid parameters can be of the following types:

Integer	ASCII string representation of an integer value. Space, dot or comma is not allowed (E.g. 115200).
String	ASCII string (E.g. off). If the string contains more than one word, the string has to be delimited by double quotation marks. E.g. “Likethis!”

Character	ASCII character (E.g. +).
Float	ASCII string representation of a decimal value. Dot is used to separate decimals (E.g. 0.5)

2.2.2 2.2.1.2 Example

Typical commands:

```
level 1000 <CR>
ps -P appcore<CR>
```

2.2.3 OK Responses

If the command has parameters to alter the current camera settings the response (system output) is:

```
<CR><LF>path>
```

where:

<CR><LF>	specifies the two-byte ASCII control characters CR and LF that ends every line of output data
\flashfs\system>	is the shell prompt indicating that the camera is ready to receive next command at path \flashfs\system. The format of this prompt can be changed by the "prompt" command into something more convenient.

If the command is a query, typically a command without any parameters, there will be a response with the following syntax:

<CR><LF><result><CR><LF>path>

where:

<result>	is one or many results given by that command, all separated by <WS>, and if the result takes more than one line there will be a <CR><LF> line break inserted between every line.

The camera always responds to a command with a <CR><LF> and the “path>”prompt to announce that it is ready to run next command.

This is also the case for an empty command (a single <CR>). This can be used to check if camera is ready and running. Note: Incoming commands are buffered and input buffers can overflow if commands are sent too frequently. (1 Hz is not a problem.)

2.2.3.1 Example

The command: level<CR>

could give the following result:

```
<CR><LF>1000<CR><LF>path>
```

while the command:

`level 1000<CR>`

will get the shorter response:

`<CR><LF>path>`

2.2.3.2 Initial response

The camera initially responds to power up with an initial message followed by the "\>" prompt:

`<Initial Message String><CR><LF>\>`

where the <Initial Message String> is an undefined character string (not containing <CR><LF>\>).

When the <CR><LF>\> prompt is received the camera is ready to receive the first command.

2.2.4 Error responses

If the command is somehow erroneously received there is a response indicating the error.

Depending on the kind of error the responses will be:

<CR><LF>Bad command or file name<CR><LF>	when a command not is recognized by the interpreter,
<CR><LF>Reason<CR>LF>	when a parameter to the command is of wrong type or out of range, or if the command failed by other reasons, e.g. hardware not ready.

When the <CR><LF>path> prompt is received again the camera is ready for the next command.

2.3 Command survey

Command	Class	Usage
attrib	DOS	Displays or changes file attributes.
bt	Camera	Simulate a button being pressed.
cd / chdir	DOS	Changes the current directory used for command shell or displays its name.
copy	DOS	Copies a file on the file system.
date	DOS	Set the current date.
del / delete / erase	DOS	Deletes files from file system.
delay	DOS	Wait.
dir	DOS	Displays a list of files and subdirectories in a directory.
echo	DOS	Displays a message or switches command echoing on or off. Useful in bat-files.

exit	DOS	Exits (terminates) the command line interpreter.
freeze	Image	Stop or start the flow of live images.
fsver	Utility	Reports free file system space
ipconfig	Utility	Shows information about ip interfaces.
ifconfig	Linux	Shows information about ip interfaces.
kitcrc	Camera	Checks that the software kits of the camera are OK.
level	Image	Displays and optionally sets level of image color transformation.
memory	Utility	Report current memory usage.
mkdir / md	DOS	Creates a directory.
move	DOS	Move existing file/directory to new name
netstat	Utility	Displays protocol statistics and current TCP/IP network connections.
nuc	Image	Non uniformity image correction.
palette	Image	Change the colors of the image.
path	DOS	Displays or sets a search path for executable files.
ping	Utility	Test IP communication.
prompt	DOS	Change the DOS prompt
rcd	Resource	Change working resource tree path.
rset	Resource	Set a new value on a given resource.
route	Utility	Shows route Information. Manipulates Network Routing Tables.
span	Image	Displays and optionally sets span of image colour transformation.
start	Utility	Runs a command in the background.
store	Image	Store current image in a file.
time	DOS	Set the current time
type	DOS	Display the contents of a text file.
usbfn	Utility	Changes USB device port behavior.
version	Camera	Display version of camera hardware and software components

Table 4

2.4 Command details

The following command definitions follow the syntax described above, but for reasons of readability, when specified in this section, it will not be at the same level of detail.

The “path>”prompt, the “CR” the “LF” the “space” and the “whitespace” will all be there even if omitted in the following sections.

Parameters enclosed in brackets are optional, e.g. [-o <option>] where -ois an optional flag and <option> is its parameter.

Response is only specified for the case of a query command.

2.4.1 DOS Commands

2.4.1.1 attrib

Displays or changes file attributes.

Syntax	`ATTRIB [+R | -R] [+A | -A] [+S | -S] [+H | -H] file [/S [/D]]`		
Param.	Parameter	Type	Value
	file	string	File name
	attribute	string	+ Sets an attribute - Clears an attribute R Read-only file attribute A Archive file attribute S System file attribute H Hidden file attribute
	option	string	/S Processes matching files in the current directory and all subdirectories /D Processes directories as well
Response.	Type ATTRIB without a parameter to display the attributes of all files		

2.4.1.2 cd/chdir

Change the current directory or display its name

Syntax	`CHDIR [path]`
	`CHDIR [..|-]`
	`CD [path]`
	`CD [..|-]`		

Param.	Parameter	Type	Value
	path	string	Directory name

	short path	string	.. parent directory - previous directory
Response.	Type CD without a parameter to display the current directory.		

2.4.1.3 copy

Copy one or more files to another location.

Syntax	`COPY [/V][/Y|/-Y][/A|/B] source [/A|/B] [+ source [/A|/B] [+ ...]] [destination [/A|/B]]`		
Param.	Parameter	Type	Value
	source	string	Specifies the file or files to be copied.
	source options	string	/A Indicates an ASCII text file. /B Indicates a binary file.
	destination	string	Specifies the directory and/or filename for the new file(s).
	destination options	string	/V Verifies that new files are written correctly. /Y Suppresses prompting to confirm you want to overwrite an existing destination file. /-Y Causes prompting to confirm you want to overwrite an existing destination file.

2.4.1.4 date and time

Set the real time clock of camera.

Syntax	`DATE [/T][date] TIME [/T][time]`		
Param.	Parameter	Type	Value
	date	string	Date
	time	string	Time
	options	string	/T display only
Response.	Type DATE or TIME without parameters to display the current setting and a prompt for a new one. Just press ENTER to keep the same setting.		

2.4.1.5 del / delete / erase

Delete one or more files.

Syntax	`DEL [/N /T /Q /W /Z] files`
	`DELETE [/N /T /Q /W /Z] files`
	`ERASE [/N /T /Q /W /Z] files`		

Param.	Parameter	Type	Value
	Files	string	Specifies the file(s) to delete.
	Options	string	/T Display total number of deleted files and freed disk space. /Q Quiet. /W Wipe. Overwrite the file with zeros before deleting it. /Z Zap (delete hidden, read-only and system files).

2.4.1.6 delay

Pause for n seconds or milliseconds.

Syntax	`DELAY [/m] n`		
Param.	Parameter	Type	Value
	N	integer	time period.
	options	string	/m specifies that n is in milliseconds, not seconds

2.4.1.7 dir

Display a list of files and subdirectories in a directory.

Syntax	`DIR [path][filename] [/A] [/B] [/L] [/N] [/S] [/P] [/W] [/4]`		
Param.	Parameter	Type	Value
	path, filename	string	Specifies directory, and/or files to list.

	options	string	/A Displays files with HIDDEN SYSTEM attributes default is ARCHIVE and READ ONLY /B Uses bare format (no heading information or summary). /L Uses lowercase. /N New long list format where filenames are on the far right. /S Displays files in specified directory and all subdirectories /P Pauses after each screen full /W Prints in wide format /4 Display four digit years.

2.4.1.8 echo

Displays a message or switches command echoing on or off.

Syntax	`ECHO [ON | OFF]`
	`ECHO [message]`
	`ECHO .`		

Param.	Parameter	Type	Value
	message	string	Text to display
	options	string	ON Switches echoing on OFF Switches echoing off . (dot) Prints an empty line
Response.	Type ECHO without a parameter to display the current ECHO setting.		

2.4.1.9 exit

Exit the command line interpreter.

Syntax	`EXIT`

2.4.1.10 mkdir/md

Create a directory.

Syntax	`MKDIR path` `MD path`		
Param.	Parameter	Type	Value
	path	string	Directory name

2.4.1.11 move

Move files and rename files and directories.

Syntax	`MOVE [path]filename1[,...] destination (moves files)` `MOVE [path]dirname1 dirname2 (renames directories)`		
Param.	Parameter	Type	Value
	path, destination, dirname	string	Directory names
	Filename	string	File name

2.4.1.12 path

Displays or sets a search path for executable files.

Syntax	`PATH [path[;...]]`		
Param.	Parameter	Type	Value
	path	string	Directory name
Response	Type PATH without a parameter to display the current PATH setting.		

2.4.1.13 prompt

Change the command prompt.

Syntax	`PROMPT [text]`		
Param.	Parameter	Type	Value
	text	string	Wanted prompt The prompt can be made up of normal characters and the following special codes: $A & (Ampersand) $B | (pipe) $C ((Left parenthesis) $D Current date $E Escape code (ASCII code 27) $F) (Right parenthesis) $G > (greater-than sign) $H Backspace (erases previous character)

			$L < (less-than sign) $N Current drive $P Current drive and path $Q = (equal sign) $T Current time $V OS version number $_ Carriage return and linefeed $$ $ (dollar sign) $+ Displays the current depth of the directory stack Type PROMPT without parameters to reset the prompt to the default setting.
Response.	The set command displays the current path setting.		

2.4.1.14 rmdir/rd

Remove a directory.

Syntax	`RMDIR path` `RD path`		
aram.	Parameter	Type	Value
	path	string	Directory to remove

2.4.1.15 rem

Start a comment line in a batch file.

Syntax	`REM [Comment]`		
Param.	Parameter	Type	Value
	comment	string	Comment text

2.4.1.16 rename/ren

Rename a file/directory or files/directories.

Syntax	`RENAME [path][directoryname | filename] [newdirectoryname | newfilename]` `REN [path][directoryname | filename] [newdirectoryname | newfilename]`		
Param.	Parameter	Type	Value

	path, directoryname	string	Directory name
	filename	string	File name

2.4.1.17 type

Display the contents of text files.

Syntax	`TYPE [path]filename`		
Param.	Parameter	Type	Value
	path	string	Directory name
	filename	string	File name

2.4.2 Camera commands

2.4.2.1 Button simulation

This command could be used for a simple remote control. It simulates press/release of buttons on the camera used for direct control.

Syntax	`bt <keyswitch> [<key>]`
	`bt <flank> <keyswitch>`
	`bt <key>[<key1>..[<keyN>]]`
	`bt (empty command will send a space)`		

Param.	Parameter	Type	Value

	keyswitch	string	-a <key> simulate <Alt>+<key> -u simulate joystick up button -d simulate joystick down button -l simulate joystick left button -r simulate joystick right button -e simulate Enter key -t simulate Tab key -c simulate Escape key -b simulate Backspace key - simulate Minus key f1 simulate Left function button f2 simulate Right function button -g simulate Laser button -s simulate Snapshot button -o simulate Open (recall) image button -p simulate Power on/off button -m simulate Mode button -f simulate Freeze button -A simulate Manual/Auto button -- simulate Adjust left key -+ simulate Adjust right key -* simulate Adjust press key
	Flank	string	-P simulate a press -S simulate a release Can be used to simulate a long press.
	Key	string	Normal characters are sent as they are.

2.4.2.2 Software kit check

Make a consistency check of the installed software

Syntax	`kitcrc -c <kitfile|dir>`		
Param.	Parameter	Type	Value
	kitfile	string	Kit revision file
	dir	string	Directory containing kit revision files
	options	string	-c check consistency

2.4.2.3 Restart

This command resets the camera and makes a complete restart.

Syntax	`Restart`

To make a restart that will restore the factory defaults, you will have to use the menu system of the camera.

2.4.2.4 Version info

This command returns the version and ID of all hardware and software of the system.

Syntax	`version`
Response.	One row for each item: <camera name> <part number and revision> <serial> <hw item name> <part number and revision> <serial> <sw item name> <version> <edit> <date> <build type>

2.4.3 Image commands

2.4.3.1 freeze

This command controls the frame grabber and what the video circuit shows. Live or the still image just grabbed. Frame grabber freeze does not stop the image streams.

Syntax	`freeze <state>` `freeze`		
Param.	Parameter	Type	Value
	State	String	on, off
Response.	Type freeze without a parameter to display the current <state>		

2.4.3.2 level

Set the level of the center position in the pixel value span.

Syntax	`level <value>` `level`		
Param.	Parameter	Type	Value
	Value	Integer	0 -65535
Response.	Type level without a parameter to display the current <value>		

The level will only be kept in Manual adjust mode.

2.4.3.3 Non uniformity correction

Calculates a new image offset map. The map is made with the camera looking at an internal shutter.

Syntax	`nuc`

The command can to take up to 10 seconds.

2.4.3.4 Palette

This command changes or displays the current palette.

Syntax	`palette [-r] [-e] palfile`
	`palette [-r] [-e] -u`
	`palette`		

Param.	Parameter	Type	Value
	palfile	string	Palette file name to update from.
	options	string	-r reverse palette -e turn on extremes (above/below colors) -u update current palette instead of applying a new file.
Response.	Type palette without a parameter to display the current palette settings.		

2.4.3.5 Recall

This command recalls a previously stored image from the file system.

Syntax	`recall <path and file>`		
Param.	Parameter	Type	Value
	path_and_file	string	Image file to read.

The command can be expected to take a few seconds.

2.4.3.6 span

Set the pixel value span around the level value.

Syntax	`span <value> span`		
Param.	Parameter	Type	Value
	value	Integer	0 -65535
Response.	Type span without a parameter to display the current <value>		

The span will only be kept in Manual adjust mode.

2.4.3.7 store

This command stores currently displayed image as a file. The file format FFF mentioned below is FLIR Systems propriety file format

Syntax	`store [mode] [options] <path and file> [2nd path and file]`		
Param.	Parameter	Type	Value
	path_and_file	string	File name
	mode	string	(no flag) Radiometric FFF -j Radiometric JPEG -v Visual JPEG -y Visual FFF -s Sequence FFF -m Combo IR and visual (must have 2 filenames)
	options	string	General: -p PNG compressed pixels JPEG options: -e JPEG only (non-radiometric) -o No overlay -c Cut image with overlay to IR image size

The command can be expected to take quite a few seconds.

2.4.3.8 Test patterns

The camera has a number of built-in test pattern images. They can be retrieved through the normal recall command if you set the file name to ###TESTPATTERN1###, ###TESTPATTERN2###, ###TESTPATTERN3###, ###TESTPATTERN4###, ###TESTPATTERN5### or ###TESTPATTERN6###. To remove the test pattern, recall another image or switch to live image distribution.

2.4.4 Resource tree commands

Available resources are described in the Camera Resource Reference Guide. Command errors are reported in human readable format.

2.4.4.1 rcd

Change your current default resource tree node path.

Syntax	`rcd <resource path>` `rcd`

Param.	Parameter	Type	Value
	resource_path	string	Current default resource tree path. The root node is assumed if no path is given.

2.4.4.2 rls

Syntax	`rls [switches] [resource name [resource name...]]`		
Param.	Parameter	Type	Value
	resource_name	string	Wanted resource name, with or without full path.
	switches	string	-r list recursive -l list using long format -t dump complete transparent data (req. -l) -d force integer decimal output
Response.	Short format		
	resource_name value		
	Long format		
	rwcdrwcdrwcd0p*1 user group <a> resource_name "value" Explanation: First rwcd = owner read,write,create,delete rights flags. Second rwcd = group read,write,create,delete rights flags. Third rwcd = other read,write,create,delete rights flags. 0p*1 = subscribers, persistance, not default, users counts and flags <a> = node type (ascii/integer/double/index/entry) For resources with fixed string values, all alternatives are listed as well.		

2.4.4.3 rpwd

Display your current default node path of the resource tree.

Syntax	`rpwd`
Response.	The current default node path

2.4.4.4 rset

Syntax	`rset [options] <resource name> <value> [ ; <resource name> <value> ... ]`

Param.	Parameter	Type	Value	
	resource_name	string	Resource name, with or without full path.	
	Value	-	New value of the resource	
	options	string	-s Wait for subscribers	

2.4.5 Utility commands

2.4.5.1 Report free file system space

Syntax	`fsver -s`

Response	Example:
	`Sector summary: 52122 in use`
	`30500 free`
	`2050 discarded`
	`0 cleaning`
	`0 allocated`
	`0 with unknown status`
	`File system on DSK1: is OK`

2.4.5.2 IP interface information

Syntax	`ipconfig [/? | /all | /renew [adapter index] | /release [adapter index]]`		
Param.	Parameter	Type	Value
	adapter index	Integer	Use ipconfig /all to find the adapter index value
	options	string	/? Display help message. /all Display full configuration information. /release Release the IP address for the specified adapter index /renew Renew the IP address for the specified adapter index

Response	Example:
	`Windows IP configuration`
	`Ethernet adapter [RNDISFN1]:`
	`IP Address ..... : 192.168.0.2`
	`Subnet Mask ..... : 255.255.255.0`
	`Default Gateway ... : 192.168.0.1`
	`DNS Servers ..... : 192.168.0.1`		

2.4.5.3 Network statistics

Displays protocol statistics and current TCP/IP network connections

Syntax	`netstat [-e] [-n] [-s] [-p proto] [-r] [interval]`		
Param.	Parameter	Type	Value
	interval	integer	number of seconds to pause between refreshes. Press Ctrl-C to stop
	options	string	-e Displays Ethernet statistics. -n Displays addresses and port numbers in numerical form. -s Displays per-protocol statistics. -p Displays statistics for one protocol (IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, or UDPv6) -r Displays the routing table.
Response	Example of `netstat -p UDP`		

	`UDP Statistics:`
	`Datagrams Received = 235`
	`No Ports = 4847`
	`Receive Errors = 0`
	`Datagrams Sent = 113`
	`Number UDP entries = 4`		

2.4.5.4 Test IP network communication

Syntax	`ping [-t] [-l size] [-n count] [-i TTL] [-v TOS] [-w timeout] address`		
Param.	Parameter	Type	Value
	address	string	The remote IP number to check
	options	string	-t Ping until interrupted by Ctrl-C. -l Specify send buffer size. -n Specify number of attempts (default 4). -f Don't fragment. -i Specify time to live (default 128, max 255) -v Specify type of service (default 0, max 255) -w Specify timeout (in milliseconds)

2.4.5.5 Update the routing table

Shows route information. Manipulates Network Routing Tables

Syntax	`route [-f] command [destination] [MASK netmask] [gateway] [METRIC metric] [IF interface]`	
Parameter	Type	Value
command	string	PRINT Displays a route ADD Adds a route DELETE Deletes a route CHANGE Modifies an existing route
destination	string	Specifies the destination host.
gateway	string	Specifies the associated gateway.
options	string	-f Clears the routing table prior to running the command. MASK Specifies a network mask (default 255.255.255.255). METRIC Specifies a metric value (inverse priority). IF Specifies an interface number (default the best one).

Example of "route print":

```
Interface List
     0x10003   02 40 7f 07 23 81      RndisFn!
     Active Routes
The no. of entries is 8
```

Destination	Netmask	GatewayAddress	Interface	Metric
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.2	30
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.0.0	255.255.255.0	192.168.0.2	192.168.0.2	30
192.168.0.2	255.255.255.255	127.0.0.1	127.0.0.1	30
192.168.0.255	255.255.255.255	192.168.0.2	192.168.0.2	30
224.0.0.0	240.0.0.0	192.168.0.2	192.168.0.2	30
255.255.255.255	255.255.255.255	192.168.0.2	192.68.0.2	1
255.255.255.255	255.255.255.255	192.168.0.2	192.168.0.2	1

2.4.5.6 Display the amount of system memory

Syntax	`memory`

Response	Example: 56% memory load. 87,613,440 bytes total physical RAM. 38,936,576 bytes available physical RAM. 0 bytes total page file. 0 bytes available page file. 33,554,432 bytes total virtual memory. 31,195,136 bytes available virtual memory.

2.4.5.7 Start a command in the background

Syntax	start <command>		
Param.	Parameter	Type	Value
	Command	string	A command that can be run in the background.

2.4.5.8 Change USB port behaviour

Composite ports (CompositeFN) can have more than one function at the same time. Such USB ports cannot be changed with this command.

Syntax	usbfn <funktionname>		
Param.	Parameter	Type	Value
	functionname	string	RNDIS Mass_Storage_Class Video_Class

3 Resource Socket Protocol Specification

3.1 Introduction

This section describes the TCP/IP resource interface used to read, write and iterate through the software resources of the camera. Subscriptions to value changes can also be made.

3.2 About the Resource Tree

A large part of the internal and external communication in the cameras uses the so called "resource tree". Resources are a way to expose functionality and data in a module to others, both internally and as an exported interface to external applications.

Resources are organized as a tree structure.

It is possible to add one resource as child to another, thus building a hierarchy of resources.

Individual resources have ASCII strings as identifiers. When they are inserted into the tree structure there are methods to address them using a name parsing algorithm that combines the resource identifier strings along "the way". From one specific resource node it is then possible to find child nodes several levels down.

There is one (1) global root node in the camera.

3.2.1 Node types

All individual resource nodes have a type. There is a fundamental difference between parent nodes and leaf nodes.

3.2.1.1 Parent node

Parent nodes may have child nodes.

Type	Comment
Entry	Supports children. May have a Create operator.
Index	Supports automatic naming. Normally, the index node names are increasing numbers.
Mount point	Supports mount of resource trees from other applications

Table 7: Parent node types

An existing entry node might support a Create operator. When it is used, it adds children as an “instanceof what it represents”, in form of a new index child, and its predefined children. That index node might then support the Delete operator to remove itself and its children.

3.2.1.2 Leaf nodes

Leaf nodes cannot have children

Type	Meaning

bool	boolean value (true/false)
int	32 bit integer value
double	IEEE double precision floating point value
ASCII string	An ascii string of arbitrary size
UNICODE (UCS-2)	An unicode string of arbitrary size
Transparent	Data bytes of size defined by node

Table 8: Leaf node types

Value limits are supported (except for the transparent type).

For strings, the limits are either defined as an array of valid strings or a maximum string length.

3.2.1.3 Access and ownership attributes

Attribute	Value	Comment
ownerID	16 bit integer	set by resource creator
groupID	16 bit integer	set by resource creator
access rights	field data	

Table 9: Access and ownership attributes

Access rights are ORed together from "read", "write", "create" and "delete"

When a resource is accessed, a check is made for the attempted access method. The check compares the caller's current ownerID/groupID against the node ownerID/groupID and permissions.

If the grant fails, an error code is returned

3.2.1.4 Other attributes

Attribute	Meaning
Persistent	node data will be included in dump
Sync	node will automatically try to retrieve its original value from the file at insertion
Subscr	node is subscribable
Notify parent	node will notify parent's subscribers
Rename	node may be renamed
Default	node has a "default" value and can be skipped when dumping to file

Table 10: Other attributes

3.3 Resource transport protocol

3.3.1 General

The resource transport protocol is byte-oriented.

The client sends requests, and the server replies.

The response might be a simple acknowledgement, or contain data, depending on the command.

If a subscription has been set up, automatic notification messages can be sent from the server.

Message	Format
request	[header] [command] [type & data]
response type 1	[header] [err-response] [error]
response type 2	[header] [ok-response]
response type 3	[header] [ok-response] [type & data]
notification	[type] [magic value] [data]

Table 11: Message formats

For request and response type 3 messages, the type information is sometimes omitted.

Resource servers use TCP/IP as carrier protocol. The well-known TCP server port is Hex 5678 = 22136.

To be able to access the server data, you have to login with the RESLIB_COMM_AUTH_HOST request.

The client opens a TCP/IP connection on the server and sends a request. The server will close the connection after the response, unless the client sends an "open session" command. Then the connection stays open until it is closed by either side.

All numerical values are sent in network byte order (big endian).

The resource paths are sent as 8-bit ASCII strings.

Some commands use NULL-terminated strings, other use pascal strings with length count first.

3.3.2 Header

A header is sent first in each request or response.

The header is 8 bytes long. It contains the length of the rest of the message.

Byte	Value
0	Hex FE
1-3	[reserved] = 0
4-7	binary length the rest of the message

Table 12: Header format

Example (open session command): Hex FE 00 00 00 00 00 00 01 C0

3.3.3 Data Types

Data type	Value	Length	Interpretation
BOOL	1	1	0 FALSE 1 TRUE
INT32	2	4	network byte order signed 32-bit integer
DOUBLE	3	8	network byte order IEEE double precision floating point
ASCII	4	string length + NULL	
UNICODE	7	(string length + NULL) * 2	
TRANSPARENT	8	Variable	
ENTRY	5		
INDEX	6		
MOUNTPOINT	9		

Table 13: Supported data types

3.3.4 Basic commands

This is a description of the fundamental resource tree protocol commands. There are more elaborate commands available in some versions of the protocol.

All command and response values are one byte long.

Some commands have an alternative command byte value. Use it when the server must wait for all subscribers to finish before sending the response.

Each command has its own unique ok-response byte. This byte is sent as the default ok response. Command specific output formats are described by the commands.

If the command fails, an error response byte Hex FF is sent instead followed by an error code.

Byte	Output
0	error code

Table 14: Output format for the err-response Hex FF

3.3.4.1 RESLIB_COMM_READ_DATA

Command byte: Hex 01
OK response byte: Hex 02

Byte	Input
0-(n-1)	resource name (NULL terminated), length n bytes
n	1 (copy flag)

Table 15: RESLIB_COMM_READ_DATA input

Byte	Output for all types except transparent
0	data type
1-	Data

Table 16: RESLIB_COMM_READ_DATA output

Byte	Output for transparent
0	data type
1-4	size of transparent data
5-8	revision of transparent data
9-12	Zero
13-	transparent data (NATIVE byte order)

Table 17: RESLIB_COMM_READ_DATA output

3.3.4.2 RESLIB_COMM_WRITE_DATA

Command byte: Hex 11 (if wait for subscribers: Hex 13)
OK response byte: Hex 12

Byte	Input for all types except transparent
0	data type
(1-n)	resource name (NULL terminated), length n bytes
(n+1)-	value

Table 18: RESLIB_COMM_WRITE_DATA input

Byte	Input for transparent
0	data type
(1-n)	resource name (NULL terminated), length n bytes
(n+1)	1 (copy flag)
(n+2)-(n+5)	size of transparent data
(n+6)-(n+9)	revision of transparent data
(n+10)-(n+13)	zero
(n+14)-	transparent data (NATIVE byte order)
Byte	Input for transparent
last 4 bytes	timeout in ms (only command Hex 13)

Table 19: RESLIB_COMM_WRITE_DATA input

3.3.4.3 RESLIB_COMM_CREATE_INSTANCE

Command byte: Hex 21 (if wait for subscribers: Hex 23)

OK response byte: Hex 22

Byte	Input
0-(n-1)	resource name (NULL terminated), length n byte
n-(n+3)	timeout in ms (only command Hex 23)

Table 20: RESLIB_COMM_CREATE_INSTANCE input

3.3.4.4 RESLIB_COMM_CREATE_NAMED_INSTANCE

Command byte: Hex 24 (if wait for subscribers: Hex 25)

OK response byte: Hex 22

Byte	Input
0-(n-1)	resource name (NULL terminated), length n byte
n-(m-1)	index name (NULL terminated), length m bytes
m-(m+3)	timeout in ms (only command Hex 25)

Table 21: RESLIB_COMM_CREATE_NAMED_INSTANCE input

3.3.4.5 RESLIB_COMM_DELETE_INSTANCE

Command byte: Hex 31 (if wait for subscribers: Hex 33)

OK response byte: Hex 32

Byte	Input
0-(n-1)	resource name (NULL terminated), length n byte
n-(n+3)	timeout in ms (only command Hex 33)

Table 22: RESLIB_COMM_DELETE_INSTANCE input

3.3.4.6 RESLIB_COMM_READ_ATTRIBUTES

Command byte: Hex 41
OK response byte: Hex 42

Byte	Input
0-(n-1)	resource name (NULL terminated), length n byte

Table 23: RESLIB_COMM_READ_ATTRIBUTES input

Byte	Input
0	data type
1	read (owner) attr (BOOL)
2	write (owner) attr (BOOL)
3	create (owner) attr (BOOL)
4	delete (owner) attr (BOOL)
5	subscr attr (BOOL)
6	persistent attr (BOOL)
7-10	open count
11-14	num subscribers
15	sync attr (BOOL)
16	rename attr (BOOL)
17	notify parent attr (BOOL)
18	hinted hex (BOOL)
19	hinted bitfield (BOOL)
20	hinted time_t (BOOL)
21	hinted default value (BOOL)
22	read (group) attr (BOOL)

23	write (group) attr (BOOL)
24	create (group) attr (BOOL)
25	delete (group) attr (BOOL)
26	read (other) attr (BOOL)
27	write (other) attr (BOOL)
28	create (other) attr (BOOL)
Byte	Input
29	delete (other) attr (BOOL)
30	[reserved]
31-32	user id
33-34	group id

Table 24: RESLIB_COMM_READ_ATTRIBUTES output

The hint flags indicate how an int32 resource should be interpreted.

3.3.4.7 RESLIB_COMM_READ_LIMITS

Command byte: Hex 45
OK response byte: Hex 46

Byte	Input
0	length of resource name (=n)
1-n	resource name

Table 25: RESLIB_COMM_READ_LIMITS input

Byte	Output for int32
0	data type
1-2	number of limit items
3-6	lower limit
7-10	upper limit

Table 26: RESLIB_COMM_READ_LIMITS output

Byte	Output for double
0	data type
1-2	number of limit items

3-10	lower limit
11-18	upper limit

Table 27: RESLIB_COMM_READ_LIMITS output

Byte	Output for plain ascii string
0	data type
1-2	number of limit items
3	0
4-7	max length

Table 28: RESLIB_COMM_READ_LIMITS output

Byte	Output for ascii string with fix values
0	data type
1-2	number of limit items
3-	NULL-separated allowed strings

Table 29: RESLIB_COMM_READ_LIMITS output

Byte	Output for plain unicode string
0	data type
1-2	number of limit items
3-4	0
5-8	max length

Table 30: RESLIB_COMM_READ_LIMITS output

Byte	Output for Unicode string with fix values
0	data type
1-2	number of limit items
3-	NULL-separated allowed strings

Table 31: RESLIB_COMM_READ_LIMITS output

3.3.4.8 RESLIB_COMM_ITERATE_CHILD

Command byte: Hex 53
OK response byte: Hex 54

Byte	Input
0	0 (iterate first) 1 (iterate last) 2 (iterate next) 3 (iterate prev)
1	length of resource name (=n)
2-(n+1)	resource name (parent for first/last, sibling for next/prev)

Table 32: RESLIB_COMM_ITERATE_CHILD input

Byte	Output type
0	length of resource name
1-	resource name

Table 33: RESLIB_COMM_ITERATE_CHILD output

Attempts to iterate past the first or last child give an error RESLIB_COMM_OPERATION_FAILED response.

3.3.4.9 RESLIB_COMM_ADD_SUBSCR

Command byte: Hex A0
OK response byte: Hex A1

Byte	Input for UDP notifications
0-3	magic number
4-5	port number
6	0 (one-shot subscription) 1 (eternal subscription)
7	0 (UDP)
8	length of resource name (=n)
9-	resource name

Table 34: RESLIB_COMM_ADD_SUBSCR input

The format of the notification is described after the last command.

3.3.4.10 RESLIB_COMM_REMOVE_SUBSCR

Command byte: Hex A2

OK response byte: Hex A3

Byte	Input for UDP notifications
0-3	magic number
4-5	port number
6	0 (one-shot subscription) 1 (eternal subscription)
7	0 (UDP)
8	length of resource name (=n)
9-	resource name

Table 35: RESLIB_COMM_REMOVE_SUBSCR input

3.3.4.11 RESLIB_COMM_SESSION_OPEN

Command byte: Hex C0
OK response byte: Hex C1

This command tells the resource protocol server to keep the TCP/IP socket open instead of closing it after each command.

3.3.4.12 RESLIB_COMM_SESSION_CLOSE

Command byte: Hex C2
OK response byte: Hex C3

This command tells the resource protocol server to close the TCP/IP socket.

3.3.4.13 RESLIB_COMM_AUTH_HOST

Command byte: Hex D0
OK response byte: Hex D1

Byte	Input
0	length of user name (=n) 4
1-n	user name (ASCII) root
n+1	length of password (=m) 5
(n+2)-(n+m+2)	password (ASCII) 3vlig

Table 36: RESLIB_COMM_AUTH_HOST input

The authentication lasts for about 5 minutes.

3.3.5 Notification

Subscription notification is delivered via UDP. To set it up, use the RESLIB_COMM_ADD_SUBSCR command. It is the only time a resource server sends messages without a client request. A notification is 512 bytes long with no header.

Byte	Value
0-3	subscription type (=1)
4-7	data type
8-11	magic value
12-15	boolean value

Table 37: BOOL resource notification

Byte	Value
0-3	subscription type (=1)
4-7	data type
8-11	magic value
12-15	integer value

Table 38: INT32 resource notification

Byte	Value
0-3	subscription type (=1)
4-7	data type
8-11	magic value
12-19	double value

Table 39: DOUBLE resource notification

Byte	Value
0-3	subscription type (=1)
4-7	data type
8-11	magic value
12-511	first 499 characters (maximum) + NULL

Table 40: ASCII resource notification

Byte	Value
0-3	subscription type (=1)
4-7	data type
8-11	magic value
12-511	first 248 characters (maximum) + NULL (2 bytes)

Table 41: UNICODE resource notification

3.3.6 Error Codes

Name	Hex Value
RESLIB_COMM_OK	00
RESLIB_COMM_PATH_NOT_FOUND	A0
RESLIB_COMM_GET_NOT_SUPPORTED	A1
RESLIB_COMM_SET_NOT_SUPPORTED	A2
RESLIB_COMM_CREATE_NOT_SUPPORTED	A3
RESLIB_COMM_DELETE_NOT_SUPPORTED	A4
RESLIB_COMM_OPERATION_FAILED	A5
RESLIB_COMM_OPCODE_NOT_FOUND	A6
RESLIB_COMM_UNKNOWN_TYPE	A7
RESLIB_COMM_TYPE_MISMATCH	A8
RESLIB_COMM_WRONG_INDATA_SIZE	A9
RESLIB_COMM_PERMISSION_DENIED	AA
RESLIB_COMM_AUTH_FAILED	AB
RESLIB_COMM_SESSION_FAILED	AC
RESLIB_COMM_NAME_NOT_UNIQUE	AD

Table 42: Error codes

4 Camera Resource Reference Guide

4.1 Introduction

This section describes the software resources supported by the FLIR Systems camera for end-customer use. Their names and behavior should be stable enough over the product lifetime, although no absolute guarantee can be given.

Depending on the camera model, all described resources may not be available or functional on a particular camera.

4.1.1 General information

A lot (but not all) of the software functionality is exposed through a feature called “software resources”. Those familiar with the Microsoft® Windows registry will recognize the idea.

However, in the camera a resource node can also represent a software function that, upon read or write, actively interacts with the camera software.

- Resources are organized in a hierarchy (like in a tree).
- Resource nodes can be data holders (of for instance calibration data).
- Resource nodes can be connected to hardware (like for instance to internal temperature sensor values).
- Resource nodes can be connected to software (like for instance to spotmeter values)
- Resource nodes have a type (for instance double, int32, ascii) and certain attributes (readonly, read/write).

4.1.2 Non-documented resources

There are a number of non-documented resources. These resources are for FLIR internal usage only. Their interpretation and/or existence may change from one camera software version to another without any written notice, and end-user are strongly discouraged to use any non-documented resources even if found.

4.1.3 Indirect resource access

The camera resources are read and changed indirectly by the different interfaces of the camera.

For instance a graphical user interface (buttons and video overlay) manipulates the camera by reading/writing resources in code. The web browser interface interacts with the resources by using asp files (within the camera file system) that translates resource names to presentable values upon access. URL's sent to the camera by clicking on links could also manipulate resources.

4.1.4 Direct resource access

It is also possible to read/manipulate the resources directly.

- There is a FLIR TCP socket server interface (TCP/IP port Hex 5678).
- Through the telnet commands `rcd`, `rls` and `rset`.

To be able to access the TCP socket server you have to log in with an authentication message. When you access the resources through the telnet server, login is not required.

The camera allows simultaneous resource changes from various sources, but the effects may be unpredictable.

4.1.5 Notation

The resource paths are truncated, and each previous level is reduced to a dot. Example: `.image.services.nuc.commit` is listed as `....commit`.

Attribute	Explanation	Comment
r	read allowed	(r) means the resource always reads the same, or the read value is meaningless
w	write allowed	(w) means that write only is to be performed by task owning resource
p	persistent	(p) means the persistence flag may or may not be set
s	subscription allowed	

4.2 Resource list by category

4.2.1 Network resources

Name	Type	Attr.	Explanation

.services.net.name	ascii	r	Camera host name (read only)
.services.net.interface.LAN90001	entry	-	Branch for the Ethernet interface data
... mac	ascii	r	Interface MAC address (read only)
... DHCP	bool	rw	true = acquire IP address from DHCP server false = use static IP address
... address	ascii	rw	Static IP address (when DHCP is false)
... netmask	ascii	rw	Static IP netmask (when DHCP is false)
... gateway	ascii	rw	IP address of gateway (when DHCP is false)
... commit	bool	ws	true = apply and announce changes

Table 43

Network resources for the LANSSP1 interface are only possible to change through the GigEVision control protocol.

4.2.2 Image stream resources

USB Video Class streaming is not set-up using any Camera resources.

IP image streaming is set up through the Real Time Streaming Protocol (RTSP) and transmitted through the Real Time Transport Protocol (RTP) on demand to client computers. These resources only determine conditions, and do not set up any streams.

Name	Type	Attr.	Explanation
.rtp	entry	-	Branch for RTP related resources
..maxBR	int	rw	Maximum bit rate at MPEG encoding (Kbits/second). Default is 2000 Kbits/second. The bit rate limits affect the encoding quality. Do not set the maximum value to low
..mcaddress	ascii	rw	Multicast address to use if multicast transmission is requested by the client.
..mcenable	bool	rw	Multicast support. Default true.
..motion	int	rw	Encoding motion detection level. Range 010. Default value = 0 (motion detection disabled). A low value means higher sensitivity for detecting motion.

..pframes	int	rw	MPEG group of picture mode indicates the number of predicted frames between intermediate frames. Range 0-65535. Default value = 0. An intermediate frame is a fully encoded frame.
..frequency	double	rw	Current frame rate in Hz
..quality	int	rw	Encoding quantization level. Range 1-31. Default value = 7. A low value means higher quality picture. Encoding quality is also dependent on the maximum bit rate allowed.
..status	ascii	r	Displays a list of all connected clients.
..ttl	int	r	Multicast TTL (Time To Live) value. Normally the TTL value is negotiated by the client who initiates the multicast image stream. This value is the default in case the client does not specify any TTL.
.tlut	entry	-	Branch for temperature linear image parameters (Normally controlled by the Real Time Streaming Protocol)
..active	bool	rw	true = Temperature linear mode false = Raw mode (pixel value linearity)
..format	int	rw	0 = 100 mK resolution (0 - 6535 K) 1 = 10 mK resolution (0 -653 K)

Table 44

4.2.3 Camera body resources

4.2.3.1 Camera on/off resources

Name	Type	Attr.	Explanation
.system.restart	bool	w	Forces an immediate restart of the camera when set to true.
.power.actions	entry	-	Branch for power down actions
...down	bool	w	Forces an immediate hibernation when set to true.
...factorydefault	bool	w	Forces an immediate restart with factory default settings when set to true.
...off	bool	w	Forces an immediate power off when set to true.
.appl.startup.status	entry	r	Branch for camera startup status

....currentItemNum	int	rs	If currentItemNum equals the value of numItems in the same branch, all applications in the camera are started.

Table 45

4.2.3.2 Version resources

Name	Type	Attr.	Explanation
.version.product	ascii	r	Branch for version information about the camera as a whole
...date	ascii	r	Camera manufacturing date
...name	ascii	r	Camera name
...article	ascii	r	Camera part number
...revision	ascii	r	Camera revision
...serial	ascii	r	Camera serial number
.calib.general	entry	-	Branch for information about the camera calibration
...date	ascii	r	Calibration date
...calibStage	ascii	r	Calibration version
.version.kits.XXX	ascii	r	Branch for version information for camera software kit XXX
....name	ascii	r	Software kit name
....date	ascii	r	Release date for XXX
....ver	ascii	r	Version of XXX
.version.sw.XXX	entry	-	Branch for version information for camera software XXX
....date	ascii	r	Release date for XXX
....ver	ascii	r	Version of XXX
.version.fW.XXX	entry	-	Branch for version information for camera firmware XXX
....date	ascii	r	Release date for XXX
....ver	ascii	r	Version of XXX
.version.hw.XXX	entry		Branch for version information for camera hardware XXX
....article	ascii	r	Part number of camera hardware XXX

....revision	ascii	r	Revision number of camera hardware XXX
....serial	ascii	r	Serial number of camera hardware XXX

Table 46

4.2.3.3 Video resources

Name	Type	Attr.	Explanation
.image.flow.video.mode	ascii	rw	PAL or NTSC Forces an immediate change of video mode, if changed.
.gui.system.hideGraphics	bool	rws	Hides the image scale and measurement graphics, if set to true.

Table 47

4.2.3.4 Focus resources

Name	Type	Attr.	Explanation
.system.focus	entry	-	Branch for lens focus control.
...areaHeight	int	rw	Height of the centered measurement box for autofocus
...areaWidth	int	rw	Width of the centered measurement box for autofocus
...position	int	rws	Refocus to absolute position
...maxPos	int	rw	Maximum absolute position
...minpos	int	rw	Minimum absolute position.
...speed	int	rws	Refocus with a certain speed value: -100..-1 move towards near focus 1..100 move towards far focus 0 stop
...autofull	bool	rw	true=Perform coarse autofocus (using complete focus range)
...autofast	bool	rw	true=Perform fine autofocus (using proximity focus range)
...state	string	r	BUSY while autofocus is operating.

Table 48

4.2.3.5 Temperature range change resources

All cameras keep a list of "calibration cases" in its resource branch .calib.ccase.X. There is normally one case for each temperature range of each lens of the camera. The list item positions can vary between cameras, even if they have the same lenses and temperature ranges. Some of the cases in the list might be disabled. In some cameras, these cases are sub-divided into sub-ranges. More about that below.

In order to switch to a particular calibration case, you have to know its number, the case index. And, in order to find that number, you have to know the name of the current lens. This is how it can be done:

1. Fetch the current lens identification from the last part of the value of ".image.ccase.lens".
2. Write the lens id into ".image.ccase.query.le".
3. Write "ds" into ".image.ccase.query.ds", "ap" into ".image.ccase.query.ap" and "fi" into ".image.ccase.query.fi".
4. Read the list of valid case index numbers from the end of the value of ".image.ccase.query.result"
5. Inspect these cases (for instance their temperature limits) and select one.
6. Write the desired case index value into ".image.services.range.caseIndex".
7. Set ".image.services.range.commit" to true.
8. Wait until ".image.services.range.status" becomes zero, or a message appears in ".image.services.range.failCause". The range switch can take up to 10 seconds.

Name	Type	Attr.	Explanation
.image.ccase.ccase	ascii	rs	The current calibration case record. For example ".calib.ccase.0"
.image.ccase.lens	ascii	rs	The current lens calibration record. The lensID is the string after the last dot. For example ".calib.lens.leFOL38$326700097$"
.image.ccase.query	entry	-	Branch for queries of valid calibration cases.
....ds,ap,fi,le	ascii	rw	ID resources for detector settings (ds), apertures (ap), filters (fi) and lenses (le). Set them to "ds", "ap", "fi", and "le" respectively to cover all combinations. Example: Set le to "leFOL38$326700097$" to find all cases for that lens.
....result	ascii	r	The result of a particular query. For example: "ds_we_ap_fi_leFOL38$326700097$: 0 1 2" means that you've found cases 0, 1 and 2.
.calib.ccase.X	entry	-	Data for calibration case X
....descr	ascii	r	Human readable name of the case
....TMax	double	r	Upper limit in Kelvin
....TMin	double	r	Lower limit in Kelvin
....enabled	bool	r	True, if the case is enabled.
.image.services.range	entry	-	Branch for range switching

....caseIndex	ascii	rw	Desired case by index value
....commit	bool	ws	Set to true to perform a range switch.
....status	int	rs	Zero when the switch has been made.
....failCause	ascii	r	Human readable result

Table 49

4.2.3.6 Sub-range changes

In some cameras, the calibration cases are sub-divided into sub-ranges. In such cameras, the procedure to change both range and sub-range is a lot less complicated. Just set the `.image.services.TMax.TMax` resource to the desired upper limit, and wait until `.image.services.status` becomes 0. The camera selects the best existing sub-range available and changes `TMax` and `TMin` to the obtained limits.

Name	Type	Attr.	Explanation
`.image.services.TMax`	entry	-	Branch for range and sub-range switching, when sub-ranges exist.
`....TMax`	double	rw	Obtained upper limit in Kelvin
`....TMin`	double	rw	Obtained lower limit in Kelvin
`....status`	int	rs	Zero when the switch has been made.

Table 50

4.2.3.7 Non-uniformity correction resources

Name	Type	Attr.	Explanation
`.tcomp.services.autoNuc.active`	bool	rws	True if the automatic image non-uniformity correction is enabled.
`.image.services.nuc.commit`	bool	ws	Forces an immediate non-uniformity correction.
`.image.services.nuc.shutter`	bool	rws	True if usage of the internal shutter flag is enabled.

Table 51

A non-uniformity correction can take up to 10 seconds.

4.2.4 Image resources

4.2.4.1 Image geometry resources

Name	Type	Attr.	Explanation

.image.sysimg.basicImgData.geometricInfo	entry	-	Branch for the geometrical properties of the image
.... firstValidX	int	r	First valid column of the image (0-based, leftmost)
.... firstValidY	int	r	First valid row of the image (0-based, top- down)
.... imageHeight	int	r	Overall height of the image, including invalid rows.
.... imageWidth	int	r	Overall width of the image, including invalid columns.
.... lastValidX	int	r	Last valid column of the image
.... lastValidY	int	r	Last valid row of the image
.... pixelSize	int	r	Number of bytes per pixel, normally 2. Can be used to determine if the image is MSBF or LSBF oriented.
.... minMeasRadius	int	r	Radius of the circle that has to be covered in order to make the center pixel accurate. 0 = not defined.

Table 52

4.2.4.2 Image mode resources

Some cameras have sensor for normal light and these are the resources for controlling MSX/Thermal/Visual modes. Only valid for cameras that have a visual sensor.

Name	Type	Attr.	Explanation
.image.services.channel	entry	-	Branch for the mode of the image source
....active	string	rw	Set image mode. IR/VISUAL/FUSION
....commit	bool	rw	Commit the mode set in active resource.

4.2.4.3 Image zoom resources

Name	Type	Attr.	Explanation
.image.zoom	entry	-	Branch for the zoom properties of the image
...zoomFactor	double	rw	Current zoom factor (1.0 to model dependent max)

Table 53

4.2.4.4 Image distribution resources

Name	Type	Attr.	Explanation
`.image.sysimg.basicImgData.distrData.distrLive`	bool	rs	True if the camera image is live.
`.image.state.live.set`	bool	ws	Forces a live image, if set to true.
`.image.state.freeze.set`	bool	ws	Forces a still image, if set to true.

Table 54

4.2.4.5 Image time resources

Name	Type	Attr.	Explanation
`.image.sysimg.basicImgData.imageInfo`	entry	-	Branch for time and trig status of the image
.... `imageMilliTime`	int	r	Number of milliseconds within the last second.
.... `imageTime`	int	r	Number of seconds since 1/1 1970 00:00, in UTC
.... `timeZoneBias`	int	r	Number of minutes separating the current time zone from UTC.
.... `trigCount`	int	r	Number of trig flanks detected since the last reset.

Table 55

4.2.4.6 Image temperature scale resources

Name	Type	Attr.	Explanation
`.image.contadj.adjMode`	ascii	rws	`auto` or `manual` scale limits.
`.image.sysimg.basicImgData.extraInfo`	entry	-	Branch for the temperature scale limits. Set `adjMode` to `manual` to be able to change the limits.
... `highT`, `lowT`	double	rws	Scale limits, in Kelvin.
... `high`, `low`	double	rws	Scale limit, in pixel value.
... `levelT`, `spanT`	double	rws	Center and span of the scale, in Kelvin.
... `level`, `span`	double	rws	Center and span of the scale, in pixel value.

Table 56

4.2.4.7 Image color resources

Name	Type	Attr.	Explanation
.image.sysimg.palette .readFile	ascii	rw	Set to the full path of a palette file, to use that palette. If no path is given, "\FlashFS\system" is assumed.
.image.sysimg.palette .paletteData	entry	-	Branch for data from the current palette
... reversed	int	rws	1 = reversed order of the colors. 0 = normal order.
... useExtremes	int	rws	1 = display particular colors outside the limits of the temperature scale.
... useUnderOverflow	int	rws	Bit 0: 1 = display a particular camera underflow color. Bit 1 : 1 = display a particular camera overflow color.

Table 57

4.2.4.8 Image storage resources

Image storage and retrieval can take quite a few seconds.

Name	Type	Attr.	Explanation
.image.services.store	entry	-	Branch for image storage
....fileName	ascii	rw	Ascii file path and name. (Do not set both names.)
....fileNameW	unicode	rws	Unicode file path and name. (Do not set both names.)
....format	ascii	rws	"FFF" = Pure radiometric image "FFF+PNG" = Compressed pure radiometric image "JPEG" = Non radiometric jpeg image "JPEG+PNG" = Jpeg image and compressed radiometric image
....overlay	bool	rws	True if scale and measurement graphics is included in jpeg format.
....overwrite	bool	rws	True if existing files may be overwritten.
....quality	int	rws	Quality setting for jpeg images (1% to 100%).
....commit	bool	ws	Orders the storage of an image
.image.state.recall	entry	-	Branch for image retrieval

.. set	ascii	rw	Read from an ascii file path and name. (Do not set both names.)
....setW	unicode	rws	Read from an unicode file path and name. (Do not set both names.)

Table 58

4.2.4.9 Test pattern image resources

The camera has a number of built-in test pattern images. They can be retrieved through the normal .image.state.recall resources if you set the file name to ###TESTPATTERN1###, ###TESTPATTERN2###, ###TESTPATTERN3###, ###TESTPATTERN4###, ###TESTPATTERN5### or ###TESTPATTERN6###. To remove the test pattern, recall another image or switch to live image distribution.

4.2.5 Measurement resources

4.2.5.1 Object parameters

These parameters describe the IR-physics of the object being imaged, and of the atmosphere and equipment through which the IR-radiation passes.

Name	Type	Attr.	Explanation
`.image.sysimg.basicImgData.objectParams`	entry	-	Branch for image object parameters.
... `ambTemp`	double	rws	Temperature of the surroundings reflected in the object, in Kelvin.
... `atmTemp`	double	rws	Atmospheric temperature in Kelvin.
... `emissivity`	double	rws	Object emissivity (0.001 to 1.0)
... `estAtmTransm`	double	rws	Estimated atmospheric transmission. If it is 0, the camera will calculate a transmission value from the atmospheric temperature, object distance and relative humidity values.
... `extOptTemp`	double	rws	External Optics temperature in Kelvin. Used for heat shields, close-up lenses etc.
... `extOptTransm`	double	rws	External Optics transmission. (0.001 to 1.0) Set to 1.0 if no external optics is present.
... `objectDistance`	double	rws	Distance to the object in m.
... `relHum`	double	rws	Relative humidity of the air. (0.0 to 1.0, 30% = 0.30)

Table 59

4.2.5.2 Measurement rate

The measurement loop of the camera does not have to be activated. It will always run as soon as there is a valid image available and some measurement function is active.

Name	Type	Attr.	Explanation
`.image.measure.count`	int	r	Increased whenever a measurement loop is finished. Will eventually wrap around.
`.image.measure.frequency`	double	rws	Desired measurement rate in Hz. The current number of active measurement functions will also affect the rate.

Table 60

4.2.5.3 Spotmeter functions

Name	Type	Attr.	Explanation
`.image.sysimg.measureFuncs.spot.X`	entry	-	Branch for spotmeter X
.... `active`	bool	rws	Set to true to activate this spotmeter.
.... `ambTemp`	double	rws	Temperature of the surroundings reflected in the object, in Kelvin. Will only be used with this spotmeter, and if bit 2 in `parMask` is set.
.... `calcMask`	int	rws	Bit field enabling and disabling result calculations Set bit to 1 to enable. Bit 0: Reserved Bit 1: Temperature Bit 2-: Reserved
.... `emissivity`	double	rws	Object emissivity (0.001 to 1.0) Will only be used with this spotmeter, and if bit 0 in parMask is set.
.... `objectDistance`	double	rws	Distance to the object in m. Will only be used with this spotmeter, and if bit 1 in `parMask` is set.
.... `parMask`	int	rws	Bitmask indicating what extra parameters to include in the measurement. Only change the bits described in this table! The other ones are reserved!
.... `valueS`	double	rs	The spotmeter pixel value.
.... `valueT`	double	rs	The spotmeter temperature, in Kelvin.
.... `valueValid`	ascii	rs	Status of the value: • "U" = Undefined • "=" = Valid • "<" = Less than • ">" = More than • "*" = Outside the image
.... `x,y`	int	rws	Image pixel coordinates of the spotmeter. Top left corner is 0,0.

Table 61

4.2.5.4 Box functions

Name	Type	Attr.	Explanation
`.image.sysimg.measureFuncs.mbox.X`	entry	-	Branch for measurement box X
.... `active`	bool	rws	Set to true to activate this box.
.... `ambTemp`	double	rws	Temperature of the surroundings reflected in the object, in Kelvin. Will only be used with this box, and if bit 2 in `parMask` is set.
.... `avgS`	double	rs	Average in pixel values within the box.
.... `avgT`	double	rs	Average in Kelvin within the box.
.... `avgValid`	ascii	rs	Status of the average, see spotmeter `valueValid`.
.... `calcMask`	int	rws	Bit field enabling and disabling result calculations Set bit to 1 to enable. Bit 0-1: Reserved Bit 2: Maximum value Bit 3: Maximum value position Bit 4: Minimum value Bit 5: Minimum value position Bit 6: Average value Bit 7: Standard deviation value Bit 8: Median value Bit 9: Reserved Bit 10: IsoCoverage Bit 11-: Reserved Max (or Min) results will only be visible on the image if both bit 2&3 (or bit 4&5) are set.
.... `emissivity`	double	rws	Object emissivity (0.001 to 1.0) Will only be used with this box, and if bit 0 in `parMask` is set.
.... `height`	int	rws	The height of the box.
.....`isoCoverage`	double	rs	Percentage (0-100) of the area pixels covered by isotherm 1.
.... `maxS`	double	rs	Maximum pixel value within the box.
.... `maxT`	double	rs	Maximum Kelvin value within the box.
.... `maxValid`	ascii	rs	Status of the maximum, see spotmeter `valueValid`.
.... `maxX,maxY`	int	rs	Image pixel coordinates of the maximum. Top left corner is 0,0.

.... medianS	double	rs	Median of the pixel values within the box.
.... medianT	double	rs	Median of the Kelvin temperatures within the box.
.... medianValid	ascii	rs	Status of the median, see spotmeter valueValid.
.... minS	double	rs	Minimum pixel value within the box.
.... minT	double	rs	Minimum Kelvin value within the box.
.... minValid	ascii	rs	Status of the minimum, see spotmeter valueValid.
.... minX, minY	int	rs	Image pixel coordinates of the minimum. Top left corner is 0,0.
.... objectDistance	double	rws	Distance to the object in m. Will only be used with this box, and if bit 1 in parMask is set.
.... parMask	int	rws	Bitmask indicating what extra parameters to include in the measurement. Only change the bits described in this table. The other ones are reserved.
.... sdevS	double	rs	Standard deviation of the pixel values of the box.
.... sdevT	double	rs	Standard deviation of the Kelvin temperatures of the box.
.... sdevValid	ascii	rs	Status of the standard deviation, see spotmeter valueValid.
.... relHum	double	rws	Relative humidity of the air. (0.0 to 1.0, 30% = 0.30)
.... x,y	int	rws	Image pixel coordinates of the top left corner of the box.
.... width	int	rws	The width of the box.

Table 62

4.2.5.5 Circle functions

Name	Type	Attr.	Explanation
.image.sysimg.measureFuncs.mcircle.X	entry	-	Branch for measurement circle X
.... active	bool	rws	Set to true to activate this circle.
.... ambTemp	double	rws	Temperature of the surroundings reflected in the object, in Kelvin. Will only be used with this circle, and if bit 2 in parMask is set.
.... avgS	double	rs	Average in pixel values within the circle.

.... avgT	double	rs	Average in Kelvin within the circle.
.... avgValid	ascii	rs	Status of the average, see spotmeter valueValid.
.... calcMask	int	rws	Bit field enabling and disabling result calculations Set bit to 1 to enable. Bit 0-1: Reserved Bit 2: Maximum value Bit 3: Maximum value position Bit 4: Minimum value Bit 5: Minimum value position Bit 6: Average value Bit 7: Standard deviation value Bit 8: Median value Bit 9: Reserved Bit 10: IsoCoverage Bit 11-: Reserved
.... emissivity	double	rws	Object emissivity (0.001 to 1.0) Will only be used with this circle, and if bit 0 in parMask is set.
.....isoCoverage	double	rs	Percentage (0-100) of the area pixels covered by isotherm 1.
.... maxS	double	rs	Maximum pixel value within the circle.
.... maxT	double	rs	Maximum Kelvin value within the circle.
.... maxValid	ascii	rs	Status of the maximum, see spotmeter valueValid.
.... maxX,maxY	int	rs	Image pixel coordinates of the maximum. Top left corner is 0,0.
.... medianS	double	rs	Median of the pixel values within the circle.
.... medianT	double	rs	Median of the Kelvin temperatures within the circle.
.... medianValid	ascii	rs	Status of the median, see spotmeter valueValid.
.... minS	double	rs	Minimum pixel value within the circle.
.... minT	double	rs	Minimum Kelvin value within the circle.
.... minValid	ascii	rs	Status of the minimum, see spotmeter valueValid.
.... minX, minY	int	rs	Image pixel coordinates of the minimum. Top left corner is 0,0.
.... objectDistance	double	rws	Distance to the object in m. Will only be used within the circle, and if bit 1 in parMask is set.
.... parMask	int	rws	Bitmask indicating what extra parameters to include in the measurement. Only change the bits described in this table. The other ones are reserved.
.... radius	int	rws	Radius of the circle.
.... sdevS	double	rs	Standard deviation of the pixel values of the circle.

.... sdevT	double	rs	Standard deviation of the Kelvin temperatures of the circle.
.... sdevValid	ascii	rs	Status of the standard deviation, see spotmeter valueValid.
.... x,y	int	rws	Image pixel coordinates of the center of the circle.

Table 63

4.2.5.6 Line functions

Name	Type	Attr.	Explanation
.image.sysimg.measureFuncs.mline.1	entry	-	Branch for the measurement line
.... active	bool	rws	Set to true to activate the line.
.... ambTemp	double	rws	Temperature of the surroundings reflected in the object, in Kelvin. Will only be used with this line, and if bit 2 in parMask is set.
.... avgS	double	rs	Average in pixel values on the line.
.... avgT	double	rs	Average in Kelvin on the line.
.... avgValid	ascii	rs	Status of the average, see spotmeter "valueValid".
.... calcMask	Int	rws	Bit field enabling and disabling result calculations Set bit to 1 to enable. Bit 0-1: Reserved Bit 2: Maximum value Bit 3: Maximum value position Bit 4: Minimum value Bit 5: Minimum value position Bit 6: Average value Bit 7: Standard deviation value Bit 8: Median value Bit 9: Reserved Bit 10: IsoCoverage Bit 11-: Reserved Max (or Min) results will only be visible on the image if both bit 2&3 (or bit 4&5) are set.
.... curpos	tnt	rws	Position of the line cursor, counted from the beginning of the line.
.... emissivity	double	rws	Object emissivity (0.001 to 1.0) Will only be used with this circle, and if bit 0 in parMask is set.

.... isoCoverage	double	rs	Percentage (0-100) of the line pixels covered by isotherm 1.
.... maxS	double	rs	Maximum pixel value on the line
.... maxT	double	rs	Maximum Kelvin value on the line
.... maxValid	ascii	rs	Status of the maximum, see spotmeter valueValid.
.... maxX,maxY	int	rs	Image pixel coordinates of the maximum. Top left corner is 0,0.
.... medianS	double	rs	Median of the pixel values on the line.
.... medianT	double	rs	Median of the Kelvin temperatures on the line
.... medianValid	ascii	rs	Status of the median, see spotmeter valueValid.
.... minS	double	rs	Minimum pixel value on the line.
.... minT	double	rs	Minimum Kelvin value on the line.
.... minValid	ascii	rs	Status of the minimum, see spotmeter valueValid.
.... minX, minY	int	rs	Image pixel coordinates of the minimum. Top left corner is 0,0.
.... objectDistance	double	rws	Distance to the object in m. Will only be used on the line, and if bit 1 in parMask is set.
.... parMask	int	rws	Bitmask indicating what extra parameters to include in the measurement. Only change the bits described in this table! The other ones are reserved!
.... sdevS	double	rs	Standard deviation of the pixel values of the line.
.... sdevT	double	rs	Standard deviation of the Kelvin temperatures of the line.
.... sdevValid	ascii	rs	Status of the standard deviation, see spotmeter "valueValid".
.... x1,y1,x2,y2	int	rws	Image pixel coordinates of the ends of the line.

Table 64

4.2.5.7 Reference Temperature function

Name	Type	Attr.	Explanation
.image.sysimg.measureFuncs.reftemp.1	entry	-	Branch for the reference temperature

.... refT	double	rw	Input value of the reference temperature in Kelvin
.... valueT	double	rs	Output value of the reference temperature in Kelvin

Table 65

The reason this function exists, is for it to be available to the Temperature Difference functions.

4.2.5.8 Temperature Difference functions

Name	Type	Attr.	Explanation
.image.sysimg.measureFuncs.diff.X	entry	-	Branch for measurement diff X
.... active	bool	rws	Set to true to activate this diff.
.... id0, id1	int	rws	Index of the two measurement functions. For example spot 2 and mbox 1
.... res0, res1	ascii	rws	Results to compute the difference from. For example spot value and mbox min.
.... type0, type1	ascii	rws	Measurement function types. For example spot and mbox.
.... valueT	double	rs	The computed result in Kelvin The result is res0 - res1.
.... valueValid	ascii	rs	Status of the result, see spotmeter valueValid.

Table 66

4.2.6 Alarm resources

4.2.6.1 Principles

There can be three types of alarms: measurement result alarms, digital input alarms and camera internal temperature sensor alarm. Every alarm can be set-up to trigger one or more actions, such as adding an entry to the alarm log, saving an image, sending an e-mail etc.

Two resource tree branches are used to control the alarms: .image.sysimg.alarms.measfunc and .resmon.items.

For each entry in .resmon.items (such as .resmon.items.1), there is a matching entry in .image.sysimg.alarms.measfunc (such as .image.sysimg.alarms.measfunc.1). The total number of alarms is a camera configuration parameter.

The last alarm does not have a number. The name is .resmon.items.batch and it corresponds to .image.sysimg.alarms.measfunc.9 (if the camera has 8 alarms). This "batch" alarm can only be used to enable and disable the output of the other active alarms. This feature allows you to, for instance, let the digital input switch on and off the alarm actions of the camera.

All the conditions for the digital input and temperature sensor alarms are specified in the

`.resmon.items.X.settings` parameters and the corresponding `measfunc.X` parameters must be marked with active=false.

The general configuration parameters for all the `.resmon.item` alarms are kept in the `.resmon.config` branch.

When you have made changes to the alarm configurations, you have to set the resource `.resmon.reinit` to true to apply and (after a while) save the changes.

4.2.6.2 Resmon branches

These parameters describe the basic general and specific alarm settings (=resource monitoring). There are corresponding parameters in the `.image.sysimg.alarms.measfunc` branch.

Name	Type	Attr.	Explanation
`.resmon.reinit`	bool	w	To be set to true, when changes have been made to the alarm settings, in order to apply and save them.
`.resmon.config`	entry	-	Branch for general alarm parameters.
`...email`	string	rw	ReceiverName@domain.com:nnn.nnn.nnn.nnn where nnn.nnn.nnn.nnn is the mail server ip number.
`...emailHelo`	string	rw	Mail client smtp Helo string with the host domain name for the camera.
`...emailSender`	string	rw	Reply address provided by the camera. (Default Alarm@FlirCam). Information about the specific alarm is present in the body of the e-mail.
`...ftp`	string	rw	userid:password@ftpserver ip address where nnn.nnn.nnn.nnn is the ftp server ip number.
`...ftpMode`	string	rw	`active` or `passive` (default)
`...imageDirectory`	string	rw	Where, in the camera, alarm images/movies are stored. The file names are created according to this template: img YYYYMMDD HHMMSSS mmm.ext When transmitted via e-mail or FTP, the name format becomes: camid img YYYYMMDD HHMMSSS mmm.ext
`...maxImages`	int	r	Maximum number of images stored.
`...imageFormat`	string	rw	JPEG/FFF/PNG/FFF+JPEG/PNG+JPEG
`...movieFormat`	string	rw	MPEG4
`...hideGraphics`	bool	rw	Shadow of `.gui.system.hideGraphics`. Hides the image scale and measurement graphics, if set to true.

Table 67

Name	Type	Attr.	Explanation
.resmon.items.X+ .resmon.items.batch	entry	-	Branch for specific alarm control.
....type	string	rw	Type of monitoring (measfunc/input/sensor)
....active	bool	rw	True if active. For measurement result alarms, the corresponding .image.sysing.alarm.measurefunc entry also has to be active.
....sysimgRef	string	r	For instance image.sysimg.alarms.measurefunc.3
....trigged	bool	rs	Alarm condition state. True if alarm.
....description	string	rw	Descriptive text

Table 68

Name	Type	Attr.	Explanation
.resmon.items.X.settings+ .resmon.items.batch.settings	entry	-	Branch for specific alarm settings for digital input and camera internal temperature sensor.
....condition	string	rw	ABOVE/BELOW (temperature sensor only)
....hysteresis	bool	rw	Hysteresis in Kelvin (temperature sensor only).
....duration	bool	rw	Minimum duration for the condition
....id	string	rw	Digital Input number (1 or 2)
....value	string	rw	Temperature value in Kelvin (temperature sensor only) true (high voltage trig, digital input only) false (low voltage trig, digital input only)

Table 69

Name	Type	Attr.	Explanation
.resmon.items.X.actions	entry	-	Branch for specific alarm actions. All output is automatically disabled if the batch alarm is active but not triggered.
....disableNuc	bool	rw	Automatic Non-Uniformity correction disable.

....enableOverlay	bool	rw	Video graphic overlay display enable.
....storeImage	bool	rw	Image storage enable. (Directory etc. in the config branch)
....mailImage	bool	rw	Image mailing enable. (Address etc. in the config branch)
....mailMovie	bool	rw	Movie mailing enable. (Address etc. in the config branch)
....markImage	string	rw	"start", "stop" or "off". (Image tags)
....sendImage	bool	rw	Image ftp transfer enable. (Address etc. in the config branch)
....sendMovie	bool	rw	Movie ftp transfer enable. (Address etc. in the config branch)
....setOutput	int	rw	Digital output activation (1 or 2). 0 deactivates.
....pulseTime	int	rw	Pulse length (in milliseconds) for the digital output. 0 = no pulse, constant high level at alarm.

Table 70

4.2.6.3 Measfunc branch

These parameters describe the settings for the measurement result alarms. There are corresponding parameters in the `.resmon.items` branch.

Name	Type	Attr.	Explanation
`.image.sysimg.alarms.measfunc.X`	entry	-	Branch for measurement result alarm settings.
.... `active`	bool	rw	true for active.
.... `duration`	int	rw	Noise filter in milliseconds.
.... `hysteresis`	double	rw	Result hysteresis in Kelvin
.... `label`	string	rw	Alarm name
.... `measFuncId`	int	rw	Measurement function identity
.... `measFuncResType`	string	rw	Result type (valueT/ maxT/minT/avgT/sdevT/medianT)
.... `measFuncThresholdTemp`	double	rw	Threshold temperature in Kelvin
.... `measFuncType`	string	rw	Function type (spot/mbox/mcircle/mline/reftemp/diff)

.... trig	bool	rs	Alarm state. True if alarm.
.... type	string	rw	type of condition (BELOW/ABOVE/MATCH)

Table 71

4.2.6.4 Examples

4.2.6.4.1 Alarm on spot 1 > 55 C

This alarm will store an image when spotmeter 1 rises above 55 Celsius.

Resource	Set to
.resmon.items.1.type	measfunc
.resmon.items.1.active	true
.resmon.items.1.actions.storeImage	true
.resmon.config.imageFormat	JPEG
.image.sysimg.alarms.measfunc.1.measFuncID	1
.image.sysimg.alarms.measfunc.1.measFuncResType	valueT
.image.sysimg.alarms.measfunc.1.measFuncType	spot
.image.sysimg.alarms.measfunc.1.measFuncThresholdTemp	328.15
.image.sysimg.alarms.measfunc.1.duration	0
.image.sysimg.alarms.measfunc.1.hysteresis	0.5
.image.sysimg.alarms.measfunc.1.type	ABOVE
.image.sysimg.alarms.measfunc.1.active	true
.resmon.reinit	true

Table 72

4.2.6.4.2 Alarm on digital input 1 high

This alarm will pulse Output 2 when Input 1 becomes high.

Resource	Set to
.resmon.items.1.type	input
.resmon.items.1.active	true
.resmon.items.1.actions.setOutput	2
.resmon.items.1.actions.pulseTime	1000

.image.sysimg.alarms.measfunc.1.active	false
.resmon.items.1.settings.id	1
.resmon.items.1.settings.value	true
Resource	Set to
.resmon.reinit	true

Table 73

4.2.7 Scheduled transfer resources

4.2.7.1 Principles

Transfer of measurement data and images can be set-up on a regular basis, if desired.

This is controlled by the `.resmon.schedule` branch. Transfers can be made at different times on different days of the week.

These images are not stored in the camera, only transferred. They can't be retrieved later from the camera.

When you have made changes to the scheduled transfer settings, you have to set the resource `.resmon.schedule.reinit` to true to apply and (after a while) save the changes.

4.2.7.2 Resmon branches

These parameters describe the settings for periodic transfer of measurement data and images.

Name	Type	Attr.	Explanation
`.resmon.schedule.active`	bool	rw	Activates the scheduled transfer functions.
`.resmon.schedule.reinit`	bool	w	To be set to true, when changes have been made to these settings, in order to apply and save them.
`.resmon.schedule.config`	entry	-	Branch for general transfer parameters.
`....email`	string	rw	ReceiverName@domain.com:nnn.nnn.nnn.nnn where nnn.nnn.nnn.nnn is the mail server ip number.
`....emailHelo`	string	rw	Mail client smtp Helo string with the host domain name for the camera.
`....emailSender`	string	rw	Reply address provided by the camera. (Default Alarm@FlirCam). Information about the specific alarm is present in the body of the e-mail.
`....ftp`	string	rw	userid:password@ftpserver ip address where nnn.nnn.nnn.nnn is the FTP server IP address.
`....ftpMode`	string	rw	`active` or `passive` (default)

....imageFormat	string	rw	JPEG/FFF/PNG/FFF+JPEG/PNG+JPEG

Table 74

Name	Type	Attr.	Explanation
.resmon.schedule.actions	entry	-	Branch for scheduled transfer actions.
....mailResult	bool	rw	Result transfer enable. (Name etc. in the .resmon.schedule.results.X branches)
....mailImage	bool	rw	Image mailing enable. (Address etc. in the .resmon.schedule.config branch)
....sendImage	bool	rw	Image ftp transfer enable. (Address etc. in the .resmon.schedule.config branch)
.resmon.schedule.results.X	entry	-	Branch for results transfer (max 5)
....active	bool	rw	Activation
....type	string	rw	spot or mbox
....number	int	rw	The spot or mbox number

Table 75

Name	Type	Attr.	Explanation
.resmon.schedule.sunday to saturday	entry	-	7 branches for weekday scheduling
....active	bool	rw	Activation
....mode	string	rw	once or repeat
....start	string	rw	"HH:MM" (24 hour clock)
....stop	string	rw	"HH:MM" (24 hour clock)
....interval	string	rw	"HH:MM" repetition interval

Table 76

4.2.8 G300a resources

Set High Sensitivity Mode (HSM) level using the following resources. The HSM level can be set to 8 different combinations. HSM Level range is 0-7 where the level index refers to this vector:

```
ushort[] convTab = { 0x0600, 0x1000, 0x2000, 0x3000, 0x4000, 0x5A00, 0x5A01,
0x5A02};
```

Name	Type	Attr.	Explanation

.image.sysimg.basicImgData.extendedPresentParams	entry	-	Branch for High Sensitivity Mode (G300a)
....gasDetectDynRange	int	rw	Most significant byte from convTab vector
....diffImageDelta	int	rw	Least significant byte from convTab vector

4.2.9 Utility resources

Name	Type	Attr.	Explanation
.system.tempsens	entry	-	Branch for camera temperature sensor values
...TSAperture	double	r	Temperature of the aperture, in Kelvin.
...TSCapsule	double	r	Temperature of the detector capsule, in Kelvin.
...TSLens	double	r	Temperature of the lens, in Kelvin.
...TSShutter	double	r	Temperature of the shutter, in Kelvin.
...TSVcam	double	r	Temperature of the video camera, if any, in Kelvin.
.stats	entry	-	Branch for camera statistics.
..coldStarts	int	r	Number of camera restarts.
..maxOpTemp	double	r	Highest camera temperature during normal operation
..minOpTemp	double	r	Lowest camera temperature during normal operation
..operatingTime	int	r	Camera total operating time (seconds).
..upTime	int	r	Camera running time since last restart (seconds).
.system.vcam	entry	-	Branch for camera visual sensor, if any
..torch	bool	rw	Control the LED lamp (AX8 camera).

Table 77

4.2.10 Digital Input/Output resources

Digital input and digital output can be controlled in two ways, either via alarms or directly via their resources.

Name	Type	Attr.	Explanation

.power.settings	entry	-	Branch for power related settings
...diginFunc1	ascii	rw	Config of the digital input 1, see below.
...diginFunc2	ascii	rw	Config of the digital input 2, see below.
...digoutFunc1	ascii	rw	Config and state of the digital output 1, see further below.
...digoutFunc2	ascii	rw	Config and state of the digital output 2, see further below.
.power.state	entry	-	Branch for power related settings
...digin1	bool	rs	Current state of digital input 1
...digin2	bool	rs	Current state of digital input 2

Table 78

diginFuncX config	Explanation
off	General purpose input. The state can be read from .power.state.diginX.
markRise	Set a Mark flag in one image of a GigE stream on each rising edge.
markFall	Set a Mark flag in one image of a GigE stream on each falling edge.
markStartRise	Set a Start Mark in one image of a GigE stream on each rising edge.
markStartFall	Set a Start Mark in one image of a GigE stream on each falling edge.
markStopRise	Set a Stop Mark in one image of a GigE stream on each rising edge.
markStopFall	Set a Stop Mark in one image of a GigE stream on each falling edge.
markStartRiseStopFall	Set a Start Mark in one image of a GigE stream on each rising edge and a Stop Mark in one image of a GigE stream on each falling edge.
markStartFallStopRise	Set a Start Mark in one image of a GigE stream on each falling edge and a Stop Mark in one image of a GigE stream on each rising edge.
sendStartRise	Enable a GigE image stream on the next rising edge, if it is disabled.
sendStartFall	Enable a GigE image stream on the next falling edge, if it is disabled.
sendStopRise	Disable a GigE image stream on the next rising edge, if it is enabled.
sendStopFall	Disable a GigE image stream on the next falling edge, if it is enabled..
diginFuncX config	Explanation

sendStartRiseStopFall	Enable a GigE image stream on the next rising edge, if it is disabled, and disable it again on the following falling edge.
sendStartFallStopRise	Enable a GigE image stream on the next falling edge, if it is disabled, and disable it again on the following raising edge.

Table 79

digoutFuncX config	Explanation
notUsed	Undefined output.
high	High output.
low	Low output.
frameStartRise	Low output with a short high pulse (about 150 ps) with a positive edge when each image starts.
frameStartFall	High output with a short low pulse (about 150 ps) with a negative edge when each image starts.
digin	Reflection of the corresponding digital output state

Table 80

4.2.11 Capability resources (read only)

Name	Type	Attr.	Explanation
.caps.config.image.services.store	entry	-	Branch for image related capabilities
....enabled	bool	r	True, if images can be stored.
....radiometric.enabled	bool	r	True, if radiometric images can be stored.
.caps.config.image.sysimg.measureFuncs			Branch for measurement function related capabilities.
.... enabled	bool	r	True, if measurement functions can be used.
.... diff.enabled	bool	r	True, if measurement function diff can be used.
.... diff.maxCount	int	r	Number of diff measurement functions allowed.
.... mbox.enabled	bool	r	True, if measurement function box can be used.
.... mbox.maxCount	int	r	Number of box measurement functions allowed.
.... mcircle.enabled	bool	r	True, if measurement function circle can be used.

.... mcircle.maxCount	int	r	Number of circle measurement functions allowed.
.... spot.enabled	bool	r	True, if measurement function circle can be used.
.... spot.maxCount	int	r	Number of spot measurement functions allowed.
.caps.config.image.sysimg.alarms.measfunc.maxcount	int	r	Number of measurement alarms allowed. Also the number of resmon alarms allowed.

Table 81