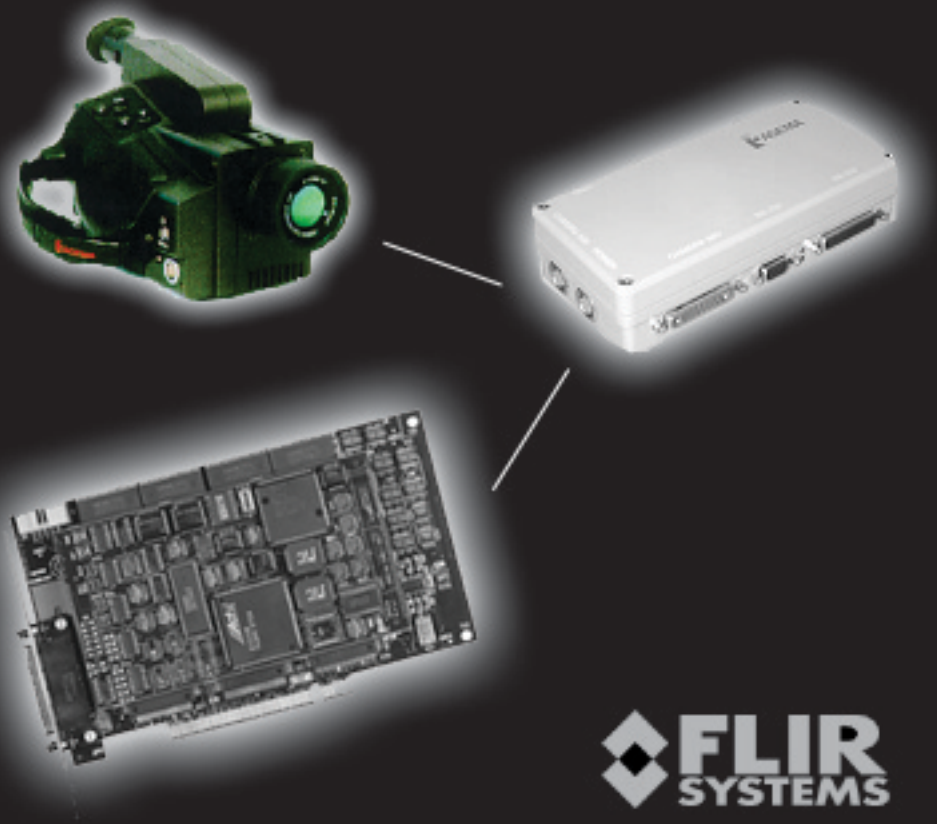
Toolkit IC2 Dig16

Developers Guide 1.01

Agema 550/570, ThermaCAM PM 5x5
and the ThermoVision Family



FLIR SYSTEMS

Toolkit IC2 Dig 16

Developers Guide 1.01

for

AGEMA 550

AGEMA 570

ThermaCAM PM 5x5

ThermoVision family

FLIR Publication number: 557 344 version B

Contents

1. GENERAL 6

1.1 Introduction 6
1.2 Copyright 6
1.3 Warranty 7
1.4 Quality assurance 7
1.5 Trademarks 7

2. SYSTEM PARTS 8

2.1 Toolkit parts 8

3. THE PARALLEL INTERFACE (PI) 9

4. FRAME GRABBER REQUIREMENTS 11

5. ACQUIRE IMAGES FROM THE FRAME GRABBER 12

5.1 General 12
5.2 Enable digital output 12
5.3 Get geometrical information 12
5.4 Load configuration file 12
5.5 Get an image from the frame grabber 13
5.6 Adjust pixels 13
5.7 Saving images to file 13

6. EXTERNAL TRIG 14

6.1 Trig control register 14

7. CODE EXAMPLE 16

8. CAMERA COMMUNICATION 18

8.1 General 18
8.2 Camera connector 19
8.3 Image and image header commands 20
8.3.1 Transfer a 16-bit image including header 20
8.3.2 Transfer an 8-bit image including header 20
8.3.3 Transfer image header 21
8.3.4 Transfer an 8-bit image 21
8.3.5 Transfer a 16-bit image 21
8.3.6 Transfer an image according to AFF 21

8.3.7 Transfer of image sub-block ... 21
8.3.8 Poll for changes in image sub blocks ... 22
8.4 Object parameter commands ... 22
8.4.1 Emissivity ... 22
8.4.2 Object distance ... 22
8.4.3 Ambient temperature ... 23
8.4.4 Atmospheric temperature ... 23
8.4.5 Relative humidity ... 23
8.5 Camera control commands ... 23
8.5.1 Digital output ... 23
8.5.2 Zoom ... 24
8.5.3 Pan ... 24
8.5.4 Auto zoom ... 24
8.5.5 Noise reduction ... 24
8.5.6 White/Black hot ... 25
8.5.7 Temperature range ... 25
8.5.8 Filter control ... 25
8.5.9 Available temperature ranges ... 25
8.5.10 Focus ... 26
8.5.11 Palette ... 26
8.5.12 Level ... 26
8.5.13 Span ... 27
8.5.14 Auto adjust ... 27
8.5.15 External auto adjust ... 27
8.5.16 Power off ... 27
8.5.17 Automatic shutter compensation ... 28
8.5.18 Stirling cooler ... 28
8.5.19 Battery ... 28
8.5.20 Viewfinder ... 28
8.6 Measurement commands ... 29
8.6.1 Switch measurement on/off ... 29
8.6.2 Run a measurement cycle ... 29
8.6.3 Controlling measurement areas ... 29
8.6.4 Read information from measurement areas ... 30
8.6.5 Object temperature ... 30
8.6.6 Object signal ... 30
8.6.7 Read pixel info from area ... 31
8.6.8 Spot meter ... 31
8.6.9 Isotherm ... 32
8.6.10 Profile ... 33
8.6.11 Reference temperature ... 34
8.6.12 Switch differential temperature on/off ... 34
8.6.13 Switch clipping in measurement functions on/off ... 34
8.7 Miscellaneous commands ... 34
8.7.1 Echo on or off ... 34
8.7.2 Version information ... 35

8.7.3 Set baud rate ... 35
8.7.4 Date and time ... 35

9. IMAGE DEFINITION ... 36

9.1 General ... 36
9.2 Image data ... 36
9.2.1 Geometric info ... 37
9.2.2 Object parameters ... 39
9.2.3 Calibration parameters ... 39
9.2.4 Adjust parameters ... 41
9.2.5 Presentation parameters ... 43
9.2.6 Image information ... 45
9.2.7 Measure function info ... 46
9.2.8 Measurement definition, common part ... 47
9.2.9 Measurement definition, spot ... 47
9.2.10 Measurement definition, rectangle ... 48
9.2.11 Measurement definition circle ... 48
9.2.12 Measurement definition, line ... 48
9.2.13 Measurement definition, isotherm ... 49
9.2.14 Distribution information ... 49
9.3 Image pixels ... 50
9.4 Text comments ... 51
9.5 The file header ... 51
9.5.1 General part ... 51
9.5.2 Index part ... 52

10. TEMPERATURE CALCULATION ON IR IMAGES ... 54

11. INDEX ... 55

1. General

1.1 Introduction

This document describes the necessary steps to acquire an image from an IC-PCI™ image capture board. The image source is an AGEMA 550, AGEMA 570 (4), ThermaCAM PM 5x5 camera or belongs to the ThermoVision family. Please note that all information in this document are subject to the conditions stated in a non-disclosure agreement with FLIR Systems AB.

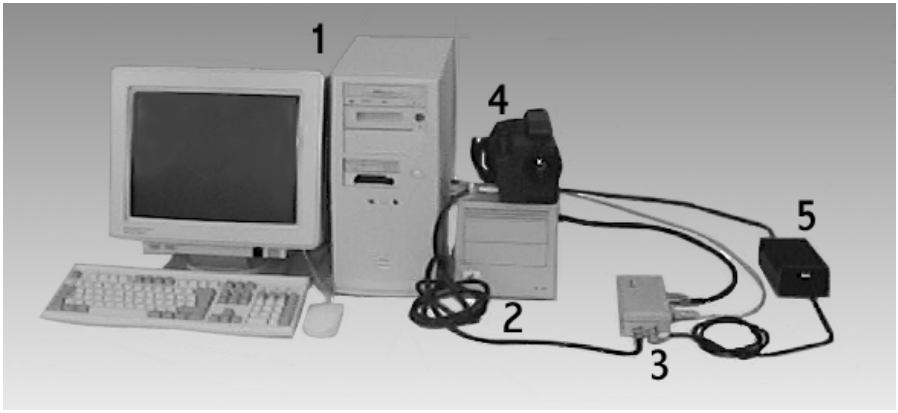


Figure 1 AGEMA 500 parallel interface system diagram

The **Parallel Interface** or **PI** (3) converts the digital image to a format suitable for the frame grabber (which resides inside the PC). The PI and the camera are powered by an external power supply (5). A serial cable from the desktop computer enables the application to communicate with the camera.

1.2 Copyright

1.3 Warranty

All products manufactured by FLIR Systems AB are warranted against defective materials and workmanship for a period of one (1) year from the delivery date of the original purchase, provided such products have been under normal storage, use and service, and in accordance with FLIR's instruction.

All products not manufactured by FLIR included in systems delivered by FLIR to the original purchaser carry the warranty, if any, of the particular supplier only and FLIR has no responsibility whatsoever for such products.

The warranty extends only to the original purchaser and is not transferable.

It is not applicable to any product, which has been subjected to misuse, neglect, accident or abnormal conditions of operation. Expendable parts are excluded from the warranty.

In the event of a defect in a product covered by this warranty the product shall not be further used in order to prevent additional damage. The purchaser shall promptly report any defect to FLIR or this warranty will not apply.

FLIR will, at its option, repair or replace any such defective product without charge if it, upon inspection, proves to be defective in material or workmanship and provided that it is returned to FLIR within the said one-year period.

FLIR has no other obligation or liability for defects than above set forth.

No other warranty is expressed or implied. 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 loss or damage, whether based on contract, tort or any other legal theory.

1.4 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.

1.5 Trademarks

Thermovision™ is a registered trademark of FLIR Systems AB.

ITEX™ and IC-PCI™ are registered trademarks of Imaging Technology Inc.

2. System parts

2.1 Toolkit parts

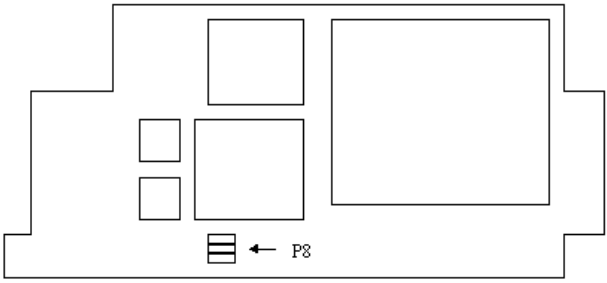
- **IC2-DIG16 frame grabber** (inside the PC) from Imaging Technology Inc. in USA (not included in this package). A PCI-standardised card.
- A **Parallel Interface (PI)** AGEMA 500/AGEMA 900 box. There is a LEMO connector on the short side to which a **Power Supply AGEMA 500/AGEMA 900** unit must be connected (the power supply is not included in the toolkit).
- A 44-pin DSUB cable that connects the PCI card to the **Parallel Interface**.
- A serial cable from the desktop computer, which fits into the "RS232" connector on the **Parallel Interface.**
- The PI - AGEMA 500 Cable fits into the "Camera 500" connector of the **PI**.
- The **Configuration files** diskette (not shown) with frame grabber configuration files. These files match IC2-DIG16 frame grabber revision A4 or higher. The configuration files have been tested with Imaging Technology's software driver version 2.8.0.0.

3. The parallel interface (PI)

There are two green LED:s (6) between the RS232 connector (5) and the RS422 connector (7). When power is applied to the parallel interface, the bottom LED will flash for approximately 15-20 seconds indicating a boot phase. The parallel interface is ready when the bottom LED stops flashing. If the flashing continues a boot error has occurred. The top LED will start flashing (camera frame frequency divided by 8) when it detects an input signal on either Camera 900 or Camera 500 connector. If an input signal is detected on both connectors, then the Camera 900 input has priority.

The Parallel Interface is equipped with a trig input connector (DSUB-9 Male) which can be configured by modifying jumpers on the interface circuit board. The trig inputs are not debounced. The easiest way to test the TTL trig input is simply to ground the TTL input pin, which will create at least one negative flank and a trig signal.

Top view of the parallel interface circuit board.



Jumper	P8:1 Trig type	P8:2 Trig slope	P8:3
ON	TTL input *)	Negative *)	Not used
OFF	OPTO coupled	Positive	Not used

*) default setting

Pin description for the TRIG connector

Pin	Name	Direction	Description
1	OPTO+	Input	OPTO isolated trig input. Threshold 0.1 mA at 1.4 V >10 µs duration 1.5 mA at 5V / 15 mA at 24V Max 24 V cont. Used together with pin 6.
2	TTL	Input	TTL trig input. Threshold 1.2 V >1 µs duration Max 24 V cont. (12 mA)
3	VSYNC	Output	TTL synch. output. Pulse on first image pixel to frame grabber. 0 - 5V max 5 mA. Pulse width 62.5 ns (AGEMA 500) or 125 ns (AGEMA 900)
4	EXSYNCB	Output	TTL buffered synch. signal from AGEMA 900 scanner. 0 - 5V max 5 mA
5	EXSYNC	In/Output	Synch. signal to/from AGEMA 900 scanner
6	OPTO-	Input	OPTO return
7	GND		Ground for TTL signals
8	GND		Ground for TTL signals

Trig information is included in the images sent to the frame grabber, see External trig on page 14.

4. Frame grabber requirements

The Parallel Interface has been tested with an IC-PCI™ frame grabber from Imaging Technology Inc. The board must have revision A4 or later. The frame grabber must be combined with an AM-DIG board. An image memory of 2 MB is sufficient.

FLIR has successfully used the ITEX™ software library revision 2.8.0.0. The PI has been tested both on Windows NT 4.0 and Windows 95 platforms. The frame grabber and the ITEX™ software library are not included in this package.

5. Acquire images from the frame grabber

5.1 General

In order to acquire images from the frame grabber you have communicate with the camera through a serial link. You will need to issue some commands to the camera and to retrieve some vital information concerning the image format and image temperature compensation. For a protocol description see Camera communication on page 18.

5.2 Enable digital output

Use the camera command DIGITAL to enable or disable the camera digital output. See Digital output on page 23.

5.3 Get geometrical information

Use the camera command IMAGESUB to retrieve geometrical information from the camera. See Transfer of image sub-block on page 21. The image sub block is returned as a data block that corresponds to the data structure described in the section Geometric info on page 37. The fields firstValidX and firstValidY together with the lastValidX and lastValidY will be used to extract the valid pixels from the frame grabber. Please note that all lines and columns are not valid pixel data when the image is retrieved from the frame grabber.

5.4 Load configuration file

Use the camera command VERSIONS to find out the revision of the HITI board. See Version information on page 35.

On the diskette there are several different camera configuration files, which are used to configure the IC2 Dig16 frame grabber (rev A4 or higher). Each configuration file corresponds to a specific camera type.

- THV570.CNF - Configuration file for AGEMA 570 cameras equipped with digital output hardware HITI rev. C

- THV550A.CNF – Configuration file for AGEMA 550 cameras equipped with

digital output hardware HITI rev. A

- THV550C.CNF – Configuration file for AGEMA 550 cameras equipped with digital output hardware HITI rev. C

It is the responsibility of the application program to load the configuration file.

5.5 Get an image from the frame grabber

This is best illustrated by a code example, see page 16.

5.6 Adjust pixels

The image pixels are not compensated for the internal camera temperature, and the non-linearity in the detector signal. This should be done as an image post processing operation. The global offset and gain gives the compensation for camera internal temperatures. They are continuously recalculated and updated by the camera software. Use the camera command POLL to retrieve the current global offset and gain from the camera. See command Poll for changes in image sub blocks on page 22.

The formula for adjusting the pixels is described in the section Adjust parameters on page 41. Temperature calculation formulas are described on page 54.

5.7 Saving images to file

You probably want to save your images on file later processing or analysis. The file format is described in this document on page 51. This format is required in case you want to export your images to any ThermaCAM PC software application. If you choose to use the this file format, then the pixels should not be adjusted as described in the previous section.

6. External trig

When an image buffer is read from the parallel interface the last line contains trigger information. The application can use this information to determine when to store an image.

This part describes the trig information data structure found on the last line.

Offset (from line start, hex)

+00h	Unsigned long	PI part no.
+04h	Unsigned short	PI version
+06h	Unsigned long	PI micro code part no.
+0Ah	Unsigned short	PI micro code version
+0Ch	Unsigned short	Trig control register
+0Eh	Unsigned short	Trig count. Incremented on each trig.
+10h	Unsigned short	Image count
+12h	Unsigned short	Hit count. Number of microseconds from the trig reference.
+3E	Unsigned short	Checksum

The checksum is the sum of the first 30 words.

6.1 Trig control register

Bit	Description
0	Trig input 0=TTL, 1=OPTO
1	Trig slope 0=Negative, 1=Positive
2	0=Jumper controlled 1=SW controlled
3-7	Not used
8	Jumper 1 position 0=On, 1=Off

9	Jumper 2 position 0=On, 1=Off
10	Jumper 3 position 0=On, 1=Off
11	1=Image is triggered 0=Image is not triggered
12-15	Not used

7. Code example

```
// Load frame grabber configuration file
if (itx_load_cnf(“THV570.CNF”) != ITX_NO_ERROR)
  return;


// Set the error level to display all errors
itx_err_level(WARNING);
itx_emsg_state(ITX_DISP_EMSG);

// Initialize system 0 boards and their daughter boards
if (itx_init_sys(0) != ITX_NO_ERROR)
  return;

// Get a MODCNF pointer to the ICP board
MODCNF* icpmod = itx_get_modcnf(0,"ICP",SEQ0);
if (!icpmod)
  return;

// Check for presence of acquisition module (AM)
MODCNF* ammod = icp_get_slot0(icpmod);
if(!ammod)
  return;

// ICP module can become PCI bus master
icp_bm_mode(icpmod, ICP_ENABLE);

// Create frame
unsigned short dx, dy, frame;
icp_get_acq_dim(icpmod, &dx, &dy);
frame = icp_create_frame(icpmod, dx, dy, ICP_PIX16,
                         ICP_MONO);

if (frame == ICP_BAD_ARG)
  return;

// Initiate interrupt
ITXINTR_ID vb;
itx_intr_connect(icpmod, ICP_INTR_VB, &vb);
itx_intr_set_timeout(vb, 1); // Time out 1 second

icp_grab(icpmod, frame);
```

```
// Image processing loop
while (TRUE)
{
  //Wait for image
  if (itx_intr_wait(vb) == ITXINTR_OK)
  {
    // Allocate image buffer
    buffer = malloc(dx * dy * sizeof(short));
    // Get image from frame grabber
    icp_read_area(icpmod, frame, 0, 0, dx, dy, buffer);

    // Process your image – break when done
  }
}

itx_intr_disconnect(vb);
free(buffer);
icp_delete_all_frames(icpmod);
```

8. Camera communication

8.1 General

There are two types of commands, control commands and question commands. The command syntax is an alphanumeric string, 1 – 10 characters and the first character must always be a letter. The command ends with ":" for control commands and "?" for question commands. Commands are not case sensitive and there are maximum 80 characters/row. All rows have to end with carriage return (CR). The camera considers CR or line feed (LF) as CR. Many commands can be used both as control commands and question commands.

Answers from the camera on question commands always begin with the command and then an exclamation mark “!”. The answer will then follow after the exclamation mark and can consist of various number of arguments separated by commas “,”. The arguments can be numerical or strings.

The string argument consists only of printable characters. Numerical argument can be integers, hex numbers or floating point numbers. Numerical argument is normally integers if nothing else is stated. Space is ignored in arguments and there are maximum 6 arguments in one command.

Commands are not echoed as default but echo is possible to switch on and off.

Some commands have to transfer a lot of data, which will then be transmitted in hex string form, most significant byte first (Big endian) and two characters per byte on separate rows after the command row. Maximum 80 characters per row. When big amount of data is transmitted XON/XOFF (ctrl-S/ctrl-Q) protocol is used.

Transmitting "quit:" during transfer from the camera will abort the transfer. If the transfer is aborted an error message (ERR!<no>,<aborted command>) will be transmitted from the camera. After “quit:” has been sent the camera will accept commands but the camera has to rest for two seconds and an OK has to be received from the camera before a new transfer can start.

Control commands are always acknowledged with OK by the camera. Wrong commands will give an error message ERR!<no>,<the command> and OK. An empty command (only CR) will receive OK as answer.

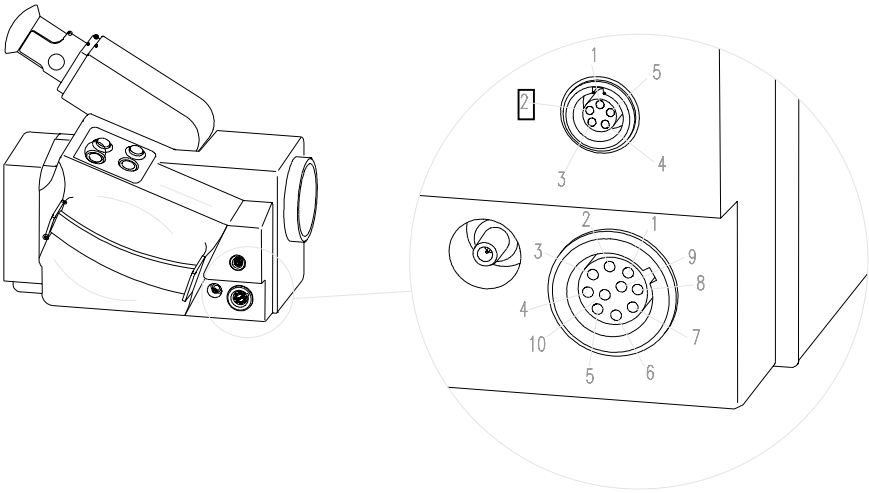
Question commands normally do not have arguments, but if an argument is given the answer is an echo of the result.

Some commands will give a “bytecount”, which is the number of data bytes that is going to be transferred and because of that, each data byte is two hex numbers the total number of characters to transfer will be the double.

The camera will normally not transmit anything by it self, except at start up when the camera present itself with the following:

BOOT THV 500 version X.XX, edit X date DD-Mmm-19XX
BOOT:Starting main program
MAIN Hilda version x.xx, edit x date DD-Mmm-19YY
Maybe also some other information.
OK

8.2 Camera connector



12 V socket:

Pin

1	RS-232, TxD (Output Data)
2	RS-232, RxD (Input Data)
3	S-Video Y (Optional 12 bit digital output).
4	S-Video C (Optional 12 bit digital output).
5	S-Video Ground
6	Not used
7	Power in 11-16 V

8 Ground Power
9 Ground Power
10 Power in 11-16 V

If extension cables are used they should be designed to have a maximum cable resistance of less than 0.18 ohm. The Power pins 7 and 10 and the ground pins 8 and 9 should always be used in parallel.

Headset socket

Pin

1 Headphone, +. Impedance >350 ohm. Max swing 2.3 Vpp
2 Head phone, -.
3 Microphone, +. Electorate type, Phantom feed 5 V DC, 1 mA, Max swing 30 mVpp
4 Microphone, -.
5 Not used

Video socket

Composite video, either PAL or NTSC SMB type connector 75-ohm cable.

8.3 Image and image header commands

8.3.1 Transfer a 16-bit image including header

Command image16?

Answer image16!<number of byte to transfer>,<image rev>

N number of rows with hex numbers until all data is transferred.

2 byte/pixel (= 4 hex characters/pixel)

<image rev> Image revision. 100 should be interpreted as 1.00

8.3.2 Transfer an 8-bit image including header

Command image8?

Answer image8!< number of byte to transfer >,<image rev>

N number of rows with hex numbers until all data is transferred.

1 byte/pixel (=2 hex characters/pixel)

<image rev> Image revision. 100 should be interpreted as 1.00

8.3.3 Transfer image header

Command imagehead?

Answer imagehead!< number of byte to transfer >,<image rev>

N number of rows with hex numbers until all data is transferred.

<image rev> Image revision. 100 should be interpreted as 1.00

8.3.4 Transfer an 8-bit image

Command imagepix8?

Answer imagepix8!<global offset(int)>,<global gain(float)>,<number of byte 8-bitpixels.> ,<x-size>,<y-size>

N number of rows with hex numbers until all data is transferred.

1 byte/pixel (=2 hex characters/pixel)

8.3.5 Transfer a 16-bit image

Command imagepix16?

Answer imagepix16!<global offset(int)>,<global gain(float)>,<number of byte 16-bit pixels.>,<x-size>,<y-size>

N number of rows with hex numbers until all data is transferred.

2 byte/pixel (= 4 hex characters/pixel)

8.3.6 Transfer an image according to AFF

Command affimage?[<flags>]

Answer affimage!<number of byte to transfer>

N number of rows with hex numbers until all data is transferred.

<flags> bit 0 (1 as default) Set to 1 for image including header.
bit 1 (0 as default) Set to 1 to include sound.

8.3.7 Transfer of image sub-block

Command imagesub?<id>

Answer imagesub!<id>, < number of byte to transfer>,<image rev>

	N number of rows with hex numbers until all data is transferred.
<id>	Image header sub-block id 1 - geomInfo 2 - objectPars 3 - calibPars 4 - adjustPars 5 - presentPars 6 - imageInfo 7 - measureData 8 - distrInfo
<image rev>	Image revision. 100 should be interpreted as 1.00

8.3.8 Poll for changes in image sub blocks

Command poll?

Answer poll!<global offset>,<global gain>,<object subcounter>,
<calib subcounter>,<present subcounter>,<measure subcounter>

8.4 Object parameter commands

8.4.1 Emissivity

Command emissivity:<emiss>

Answer OK

Command emissivity?[<emiss>]

Answer emissivity!<emiss>

<emiss> Float, Between 0.01 – 1.00

8.4.2 Object distance

Command dobj:<dist>

Answer OK

Command dobj?[<dist>]

Answer dobj!<dist>

<dist> Float, Distance in meters.

8.4.3 Ambient temperature

Command tamb:<amb>

Answer OK

Command tamb?[<amb>]

Answer tamb!<amb>

<amb> Float, Temperature in Kelvin.

8.4.4 Atmospheric temperature

Command tatm:<atm>

Answer OK

Command tatm?[<atm>]

Answer tatm!<atm>

<atm> Float, Temperature in Kelvin

8.4.5 Relative humidity

Command relhum:<hum>

Answer OK

Command relhum?[<hum>]

Answer relhum!<hum>

<hum> Float, Relative humidity between 0.00 – 1.00

8.5 Camera control commands

8.5.1 Digital output

Command digital:on/off

Answer OK

Command digital?[on/off]]

Answer digital![on/off]

8.5.2 Zoom

With a second argument = a the zoom will be centered around the image center.

Command zoom:<zoomfactor>[,a]

Answer OK

Command zoom?[<zoomfactor>[,a]]

Answer zoom!<zoomfactor>

<zoomfactor> Float, Zoom factor between 0.5 - 128

8.5.3 Pan

Command pan:<x>,<y >

Answer OK

Command pan?[<x>,<y>]

Answer pan!<x>,<y>

<x> and <y> Integer, x and y offset in pixels

8.5.4 Auto zoom

Command autozoom:up/down/off[,<speedvalue>]

Answer OK

8.5.5 Noise reduction

Command noise:<noisereduction>,<level>

Answer OK

Command noise?[<noisereduction>,<level>]

Answer noise!<noisereduction>,<level>

<noisereduction> Integer, Between 0 – 7, 0 is off and 7 is maximum

8.5.6 White/Black hot

Command vidinv:on/off

Answer OK

8.5.7 Temperature range

Command temprange:<range>

Answer OK

Command temprange?[<range>]

Answer temprange!<range>

<range> Integer, Between 0 - 3

8.5.8 Filter control

Command filter:on/off

Answer OK.

Command filter?[on/off]

Answer filter!on/off

8.5.9 Available temperature ranges

Command rangelims?

Answer rangelims!<low temp>,<high temp>,<range no>,<filter used>

	One line for each temperature range.
<low temp>	Float, Kelvin
<high temp>	Float, Kelvin
<range no>	Integer, Range number.
<filter used>	ON or OFF

8.5.10 Focus

Command	focusmov:up/down/off
Answer	OK

Command	focusmov?up/down/off
Answer	focusmov!up/down/off

8.5.11 Palette

Command	palette:<palette name>
Answer	OK

Command	palette?[<palette name>]
Answer	palette!<palette name>
<palette name>	String, Palette in use

Command	palettes?
Answer	palettes!<palette name>
<palette name>	String, Available palettes one per row.

8.5.12 Level

Command	level:<level>
Answer	OK

Command level?[<level>]

Answer level!<level>

<level> Integer, absolute pixel. See Adjust parameters on page 41.

8.5.13 Span

Command span:

Answer OK

Command span?[ ]

Answer span!

 Integer, absolute pixel. See Adjust parameters on page 41.

8.5.14 Auto adjust

Adjust level and span for best value, compare min- and max-pixel in image.

If activated: Histogram adjustment of image.

Command autoadj:[<flags>]

Answer OK

<flags> Integer, bit-mask 1 = level, 2 = span, 4 = histogram, 8 No offset map against the shutter. 3 is default (level and span).

8.5.15 External auto adjust

Correct image pixels to a neutral even reference surface. This function calculates an offset map using 2^<n> images.

Command omapcalc:[<n>]

Answer OK

8.5.16 Power off

Command poweroff:

Answer No answer

8.5.17 Automatic shutter compensation

Control the automatic image compensation.

Command tcompadj:on/off

Answer OK

Command tcompadj?[on/off]

Answer tcompadj!on/off

8.5.18 Stirling cooler

Command cooler:on/stdby/off

Answer OK

Command cooler?[on/stdby/off]

Answer cooler!on/cooling/stdby/off

8.5.19 Battery

Command battery?

Answer battery!ok/low/empty,internal/external

8.5.20 Viewfinder

Command viewfinder:on/off

Answer OK

Command viewfinder?[<on/off>]

Answer viewfinder!<on/off>

8.6 Measurement commands

When calculating temperature/object signal a status on the result is given. The status will be included in the answers related to the question commands and will be defined as a character accordingly. See Temperature Calculation on IR Images on page 54.

Character	Meaning
=	Valid value
U	undefined value
>	More than
<	Less than
O	Value is outside the image
E	Error

8.6.1 Switch measurement on/off

Command measure:on/off

Answer OK

Command measure?[on/off]

Answer measure!on/off

on switch on earlier defined measurements.

8.6.2 Run a measurement cycle

Command mwait:

Answer OK or Error

Error if measurement is off or OK when measurement cycle is finished.

8.6.3 Controlling measurement areas

If <areanr> is stated in the question command the answer will be for that area otherwise all active areas will be given in the answer.

Command area:<areanr> ,off/on/vis [,mm/avg/std]

Answer OK

Command area?[<areanr> [,off/on/vis [,mm/avg/std]]]

Answer area! <areanr>,off/on/vis, mm/avg/std

Second argument off = inactive, on = active, vis = active and visible in camera viewfinder. Third argument mm = min & max, avg =min & max & avg, std = min & max & avg & std

Command areapos: <areanr>,r,<x-size>,<y-size>[,<x-pos>,<y-pos>]

areapos: <areanr>,c,<radius>[,<x-center>,<y-center>]

Answer OK

If x-,y- pos or x-, y- center is not stated the area will be centered round earlier center position. The question command will be answered even if the area is inactive.

Command areapos?[<areanr> [, r, <x-size>, <y-size> [, <x-pos>, <y-pos>]]]

areapos? [<areanr> [, c, <radius> [, <x-center>, <y-center>]]]

Answer areapos! <areanr>, r, <x-size>, <y-size>, <x-pos>, <y-pos>

areapos! <areanr>, c, <radius>, <x-center>, <y-center>

8.6.4 Read information from measurement areas

If <areanr> is stated in the question command the answer will be for that area if it is active otherwise an error message will be given. If no argument is given the answer will include all active areas, if measure = on, otherwise an error message will be given.

8.6.5 Object temperature

Command readareak? [<areanr>]

Answer readareak! <areanr> [, <min, max> [, <avg> [, <std>]],status]

<avg>, <min>, <max> Float, Kelvin

8.6.6 Object signal

Command readarea? [<areanr>]

Answer readarea! <areanr> [, <min, max> [, <avg> [, <std>]],status]

8.6.7 Read pixel info from area

Command areapixel?<areanr>

Answer areapixel!<areanr>,<global offset>,<global gain>,<number of bytes.>,<x-size>,<y-size>

N number of rows with hex numbers until all data is transferred.

2 bytes/pixel (= 4 hex numbers/pixel)

8.6.8 Spot meter

Command spot:on/off/vis

Answer OK

Command spot?[on/off/vis]

Answer spot!on/off/vis

Command spotpos:<x-pos>,<y-pos>

Answer OK

Command spotpos?[<x-pos>,<y-pos>]

Answer spotpos!<x-pos>,<y-pos>

Command readspotk?

Answer readspotk!<object temperature>,<status>

<object temperature> Float, Kelvin

Command readspot?

Answer readspot!<object signal>,<status>

8.6.9 Isotherm

Command iso:on/off/vis/below/above

Answer OK

Command iso?on/off/vis/below/above

Answer iso!on/off/vis/below/above

The argument off switches the isotherm off. The argument on/vis sets the isotherm type to normal and makes it visible. The argument below/above sets the isotherm accordingly and makes the isotherm visible.

Command isocol: col/black/white

Answer OK

Command isocol?[col/black/white]

Answer isocol! col/black/white

Command isopos: <low>,<high>

Answer OK

Command isopos?[<low>,<high>]

Answer iso! <high>,<low>

<low> Integer, In absolute pixel

<high> Integer, In absolute pixel

Command readisok?

Answer readisok!<low iso temperature>,<high iso temperature>,<status>

<low iso temperature> Float, Kelvin

<high iso temperature> Float, Kelvin

Command readiso?

Answer readiso!<low object signal>,<high object signal><status>

<low object signal> Unsigned long

<high object signal> Unsigned long

8.6.10 Profile

Command prof: on/off/vis

Answer OK

Command prof?[on/off/vis]

Answer prof: on/off/vis

Command profpos: v/h [, <x-pos>,<y-pos>]

Answer OK

v/h Vertical or horizontal

Command profpos?[v/h [, <x-pos>,<y-pos>]]

Answer profpos! v/h, <x-pos>,<y-pos>

Command readprofk?

Answer readprofk!<object temperature>

<object temperature> Float, Kelvin

Command readprof?

Answer readprof!<object signal>

8.6.11 Reference temperature

Command tref:<temperature>

Answer OK

Command tref?[<temperature>]

Answer tref!<temperature>

<temperature> Float, Kelvin

8.6.12 Switch differential temperature on/off

Command diff: on/off

Answer OK

Command diff?[on/off]

Answer diff: on/off

8.6.13 Switch clipping in measurement functions on/off

Command measclip: on/off

Answer OK

Command measclip?[on/off]

Answer measclip: on/off

8.7 Miscellaneous commands

8.7.1 Echo on or off

Command echo:on/off

Answer OK

8.7.2 Version information

Command versions?[<type A/type B/mod>]

Answer versions!<type A>,<part no>,<serial no>,<revision>
versions!<type B>,<swversion>,<swedit>,<swdate>
versions!mod,basic/standard,550/570

<type A> camera/hisp/hipo/hiad/det/cooler/optunit/hico/hiti
<type B> boot/main/hispsw/parttab

8.7.3 Set baud rate

The camera talks using the baud rate 19200 after power-on boot. It is possible to change the baud rate once the OK prompt is shown. New baud rate is active until next Power On.

Command baudrate:9600/19200/38400

Answer OK (using new baud rate)

8.7.4 Date and time

Command date:yy,mm,dd,hh,mm,ss

Answer OK

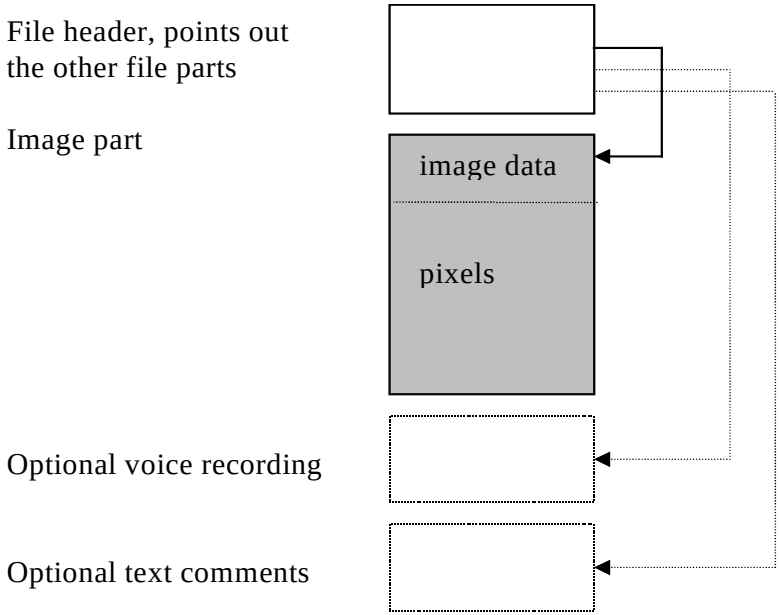
Command date?[yy,mm,dd,hh,mm,ss]

Answer date!yy,mm,dd,hh,mm,ss

9. Image definition

9.1 General

This part describes the internal data structures in a AGEMA 550/570 type of image. When images are stored in ".IMG" files, they are embedded in a tagged environment, like:



The format of the tagged environment is called "AFF" (AGEMA File Format) and is only briefly described in this section. This chapter concentrates on the embedded image structure, and how to convert to object temperature from the image pixels and data.

9.2 Image data

The format revision of image data described in this document is 1.03. Image data is the first part of the image. It consists of 8 sub structures.

1.	geomInfo	geometric information	44 bytes

2.	objectPars	object parameters	24 bytes
3.	calibPars	calibration parameters	200 bytes
4.	adjustPars	adjustment parameters	80 bytes
5.	presentPars	presentation parameters	44 bytes
6.	imageInfo	image information	36 bytes
7.	measureData	measurement functions definitions	116 bytes
8.	distrInfo	image distribution information	24 bytes

Total 568 bytes

Basic data types are:

char	8 bit	Often represents ASCII characters, but may represent an 2's complement 8 bit integer (-128 - +127)
unsigned char	8 bit	8 bit integer number (0 - 255)
short	16 bit	16 bit integer (2's complement)
unsigned short	16 bit	16 bit integer
Long	32 bit	32 bit integer (2's complement)
unsigned long	32 bit	32 bit integer
Float	32 bit	IEEE floating point number, sign + 23 bit mantissa + 8 bit exponent, representing numbers in the range +/- 10^{38}
char[<len>]	len * 8 bit	ASCII character string, most certainly terminated with the NUL character (=0)

9.2.1 Geometric info

This block contains information of the geometric construction of the image. The image format allows exclusion of the leftmost and/or rightmost pixel columns. Also lines in the beginning and/or end may be excluded

Offset (from block start, hex) Block start offset is +00h

+00h	unsigned short pixelSize;	Size of one pixel in bytes, normally 2 (pixels as unsigned short). Allowed values are 1,2 or 4 for 8,16 or 32 bits/pixel

+02h	unsigned short imageWidth	Image width in pixels
+04h	unsigned short imageHeight;	Image height in pixels
+06h	unsigned short upperLeftX;	X and Y co-ordinates for upper left corner in
+08h	unsigned short upperLeftY;	case this image is a cut-out, normally 0

The following four number idenitifies the valid pixels area within the image. Image format allows that the row(s) and column(s) near edges only contains calibration pixels that should not be considered as real pixels

+0Ah	unsigned short firstValidX	Normally 0
+0Ch	unsigned short lastValidX;	Normally imageWidth - 1
+0Eh	unsigned short firstValidY;	Normally 0
+10h	unsigned short lastValidY;	Normally imageHeight - 1
+12h	unsigned short detectorDeep;	Type of detector, original meaning was the number of bits from the detector A/D converter. Normal values are: 12 - InSb/ThermaCAM 550 15 - uncooled FPA This variable may be used to switch between different algorims for different camera types

The following arrays describes geometry for a square division of the image. These squares were intended for a future local stray light compensation of pixels, see adjust pars.

+14h	unsigned short slSquareLimsX[6];	
+20h	unsigned short slSquareLimsY[5];	
		slSquareLims describes the squares defined as 'first used x/y for this square' For an image 320 x 240 this should

		mean: slSquareLimsX = {0, 64, 128, 192, 256,320} slSquareLimsY = {0, 60, 120, 180, 240} if squares 64 x 60 are used
+2Ah	unsigned char stripeHeight	0 = not striped
+2Bh	unsigned char stripeStart	Striped image's first line from top of original image

9.2.2 Object parameters

This block contains the parameters for object temperature calculation

Offset (from block start, hex) Block start offset is +2Ch

+00h	float emissivity;	Object emissivity, 0.01 - 1.0
+04h	float objectDistance;	Distance to object in meters
+08h	float ambTemp;	Ambient (surrounding) temperature in Kelvin
+0Ch	float refTemp;	Reference temperature in Kelvin, used for delta temperature presentation
+10h	float atmTemp;	Atmospheric temperature in Kelvin. Note that atmTemp is set in parallell with ambTemp from the camera user interface. Different values can only be set using commands from the serial line
+14h	float relHum;	Relative humidity of atmosphere, 0.0 - 0.99

9.2.3 Calibration parameters

This structure defines the constants necessary for temperature calculation from pixel values.

Offset (from block start, hex) Block start offset is +44h

+00h	float R;	Calibration constant R
+04h	float B;	Calibration constant B
+08h	float F;	Calibration constant F
+0Ch	float alpha1;	Attenuation for atmosphere without water vapour
+10h	float alpha2;	Attenuation for atmosphere without water vapour
+14h	float beta1;	Attenuation for water vapour
+18h	float beta2;	Attenuation for water vapour
+1Ch	float X;	Scaling factor for attenuation
+20h	float tmax;	Upper temp limit [K] when calibrated for current temp range
+24h	float tmin;	Lower temp limit [K] when calibrated for current temp range
+28h	short notation;	Presentation precision, HIGH=1 / LOW=0
+2Ah	char imageFrq;	Image frequence. 0 = unknown
+2Bh	char saturationLimit;	% of rawpixel max value to be considered as over saturated (= too hot for current temperature range). Values 0 – 100.Used by scanner for calculation of adjust par – ipixOverflow
+2Ch	char scanner_name[21];	Scanner type (name)
+41h	char scanner_artn[10];	Scanner article number, for instance "990104"
+4Bh	char scanner_sn[10];	Scanner serial number, for instance "656012"
+55h	char lens_name[11];	Name of lens type, for instance "FOV 20"
+60h	char lens_artn[10];	Lens article number, for instance "0"

+6Ah	char lens_sn[10];	Lens serial number
+74h	float lens_fov;	Lens FOV, 0 if undefined
+78h	char filter_name[11];	Filter type, for instance "NOF" (No Filter)
+83h	char filter_artn[10];	Filter article number, for instance "0"
+8Dh	char filter_sn[10];	Filter serial number, for instance "0"
+97h	char pad2;	Currently unused entry
+98h	unsigned short calib_mode;	Bit field for calibration mode Defined values are: 00h - Combination calibrated 01h - Combination estimated 02h - Combination type estimated 03h - Combination invalid
+9Ah	unsigned short e_tao;	Estimated transmission * 1000, 0 if none
+9Ch	char calib_title[16];	Calibration title, for instance "CalRev: Org"
+ACh	char arc_ver[4];	Arc file version, for instance "1.0"
+B0h	char arc_date[16];	Arc date/time, for instance "96 Sep 02 13:13"
+C0h	char arc_sign[4];	Arc signature
+C4h	char country_code[4];	In which country the calibration was made

9.2.4 Adjust parameters

Adjustment parameters are vital for correct image pixel handling. Adjustment parameters includes parameters used for conversion of the raw (uncompensated) pixel values to camera state independent pixel values proportional to the radiation reaching the corresponding detector pixel. Compensation is made for internal camera temperature, and non-linearity in the detector signal. The values, global offset and gain gives the compensation for camera internal temperatures. They are

continuously recalculated and updated by the camera software. To get absolute pixel value from image raw (FPA) pixel value, use the following formula:

$$absPixel(imgPixel) = globalGain \cdot Lfunc(imgPixel) + globalOffset$$

where

$$Lfunc(p) = \frac{p - Obas}{1 - L \cdot (p - Obas)}$$

Offset (from block start, hex) Block start offset is +10Ch

+00h	long globalOffset;	Offset and gain latched at frame grab. Use these for calculations
+04h	float globalGain;	on frame grabbed (stored) images
+08h	long curGlobalOffset;	Offset and gain continously recalculated by temperature
+0Ch	float curGlobalGain;	compensation.Use these for live image compensation
+10h	float L;	Linearization constant for absPixel calculation - calibration constant
+14h	unsigned short ipixUnderflow;	Image pixel underflow limit - detector type dependent
+16h	unsigned short ipixOverflow;	Image pixel overflow limit - detector type dependent
+18h	short autoControlOverride;	Flag that is set when new temperatures from sensors are ignored. Continous update of global offset/gain turned off. globalOffset, globalGain are only changed via command
+1Ah	short Obas;	Base offset used in non-linearity conversion
+1Ch	long focusPos;	Focus position as counter value - NOT implemented (=0)
+20h	float focusDistance;	focusDistance converted to meters. 0 means not defined.

NOT implemented

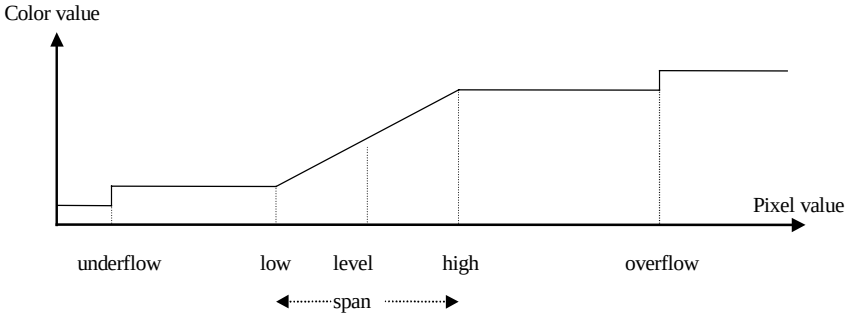
The following values were intended for a geometric stray light compensation. It has NOT been implemented, and may therefore be ignored for the time beeing.

+24h	long strayLightCompUsed;	Flag set if slVals should be used in temperature calculation, cleared otherwise
+28h	short slVals[14];	Array with stray light compensation values. To be subtracted from imgpixels in the corresponding square (defined by geometric info) before converting to absPixel
+44h	unsigned short striped_freq	Frequency in stripe mode
+46h	unsigned short trig_flags	Trig flags Bit 15: 1=This trig info is relevant 0=Look for trig info in pixel data instead Bit 3: 0=Par.Intf. 1=Serial port trig (or LPT) Bit 2: 0=TTL 1=OPTO Bit 1: 0=Negative 1=Positive Bit 0: 0=No trig 1=Trigged
+48h	unsigned short trig_count	Trig counter
+4Ah	unsigned short trig_hit	Hit count - microsecs from image trig reference
+4Ch	unsigned short ext_optTemp	External optics temp in deciKelvin.
+4Eh	unsigned short ext_optTao	External optics tao in 1/1000:ths. 0 if none.

9.2.5 Presentation parameters

Presentation parameters are intended for conversion from temperature compensated, (absolute) pixel values to colour values in the same way as the camera does.

Offset (from block start, hex) Block start offset is +15Ch



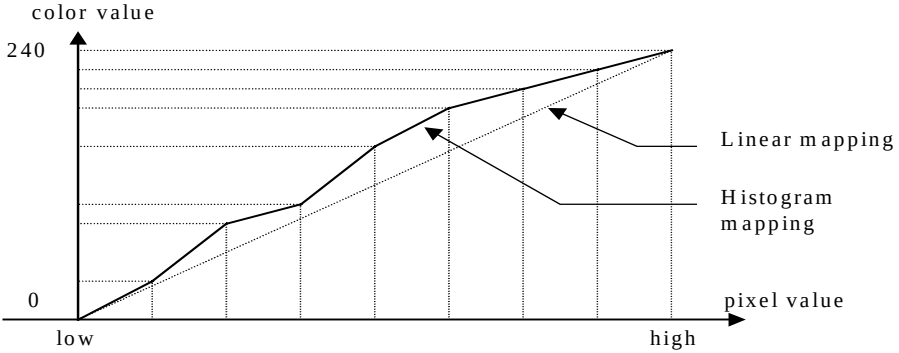
Level and span describes the low and high limit for pixel values to present. Observe that level and span is given as teperature compensated 'absolute pixel' values. If they should be used in a LUT construction, they must be converted to image pixels first, using adjust parameters

+00h	long level;	Level as normalized pixel value (abspixel), level is defined as middle of span (in pixel units)
+04h	long span;	Span as normalized pixel value (abspixel)
+08h	short useHistogram;	Histogram color settings if true, linear color settings else
+0Ah	short IsoColour;	Color to use for isotherm; 0=color, 1=black, 2=white

If histogram (autopalette) should be used for presentation, the linear mapping pixels values to color values between low and high is modified. span is divided into 16 slots. colorDistr[i] describes how many colors should be assigned to the pixel values in slot[i]. Sum of colorDistr should be 240 (240 colors are used for image presentation in the camera).

+0Ch	unsigned char colorDistr[16];	Describes how to distribute colors

between low level and high level to build histogram LUT (if useHistogram is true).



+1Ch	char paletteName[16];	Name of palette to use. Recognised names are: *"IRONBOW", "RAINBOW", "GREY", "GREY_INV", "IRONBOW10", "RAINBOW10", "GREY10", "GREY_INV10"*

9.2.6 Image information

This part describes various information for a grabbed image

Offset (from block start, hex) Block start offset is +188h

+00h	unsigned long imageTime;	Time in seconds since 1970-01-01 00:00
+04h	unsigned long imageMilliTime;	Milliseconds since last second
+08h	short valid;	0 if image not valid, <>0 else
+0Ah	short noiseReductFactor;	0 - 7, 0 means no noise reduction

+0Ch	char imgName[16];	Name of image: <original file name> if saved image in camera "Live" if image from FPA (even if frozen) "No image" if image file not existing or bad "PATTERNx"are special names for test images
+1Ch	float zoomFactor;	Image zoom factor as value 0.5 - 128. Zoom factor is set in discrete steps. Possible values are given from the formula: ZoomFactor = 128 / (256 - i); where i is an integer between 0 and 255
+20h	short xpanVal;	X co-ordinate in original image for first visible column
+22h	short ypanVal;	Y co-ordinate in original image for first visible row

9.2.7 Measure function info

The image definition includes the definitions for max 7 concurrent measurement functions. The definition consists of type, activity level, geometry and expected result type for a function, not the measurement result itself.

Offset (from block start, hex) Block start offset is +1ACh

+00h	unsigned short bitField;	bitsfield with the defined bits: bit 0: Set if difference temperature should be used bit 1 - 15: Currently unused
+02h	unsigned short measureActive;	Global flag controlling all measurement. If set, measurement

		should be performed, if =0, no measurement should be done, even if individual measurement functions are marked as active
+04h	MEAS_DEF_T funcs[7];	7 data structures. The size of each MEAS_DEF_T is 16 bytes The interpretation of funcs[i] depends of the common part in the structure, see below.

9.2.8 Measurement definition, common part

Offset (from funcs[i] start, hex)

+00h	unsigned short activeLevel;	Defines if this measurement function should be calculated on 0 means "off", 1 means "active", 2 means "active and visible as graphic marker"
+02h	unsigned short funcType;	How to interpret the measurement type dependent data 0 - spot;1 - rectangle;2 - circle;3 - line;4 - isotherm
+04h	byte typeDependent[12];	See the type dependent parts below
+10h		

9.2.9 Measurement definition, spot

Offset (from funcs[i] start, hex)

+04h	short x;	Spot position, X coordinate relative image left column
+06h	short y;	Spot position, Y coordinate relative image upper (first) line
+08h	long filler1;	Unused data
+0Ch	unsigned short calcField;	Should be = 1, means calculate value

+0Eh	short filler2;	Unused data

9.2.10 Measurement definition, rectangle

Offset (from funcs[i] start, hex)

+04h	short x;	X co-ordinate of upper left corner
+06h	short y;	Y co-ordinate of upper left corner
+08h	short width;	Width of area
+0Ah	short height;	Height of area
+0Ch	unsigned short calcField;	Bit 0: Calculate min temperature
		Bit 1: Calculate max temperature
		Bit 2: Calculate average temperature
		Bit 3: Calculate standard deviation
		Bit 4 - 15: Undefined for lines
+0Eh	short filler2;	Unused data

9.2.11 Measurement definition circle

Offset (from funcs[i] start, hex)

+04h	short x;	X co-ordinate of centre
+06h	short y;	Y co-ordinate of centre
+08h	short radius;	radius of circle in pixels
+0Ah	short filler1;	Unused data
+0Ch	unsigned short calcField;	Same definition as rectangle above
+0Eh	short filler2;	Unused data

9.2.12 Measurement definition, line

Offset (from funcs[i] start, hex)

+04h	short x;	X coordinate of marker
+06h	short y;	Y coordinate of marker

+08h	short orientation;	Orientation of line relative marker 0 - vertical 1 - horizontal
+0Ah	short filler1;	Unused data
+0Ch	unsigned short calcField;	Bit 0: Calculate min temperature Bit 1: Calculate max temperature Bit 2: Calculate average temperature Bit 3: Calculate standard deviation Bit 4: Marker temperature Bit 5 - 15: Undefined for rectangles
+0Eh	short filler2;	Unused data

9.2.13 Measurement definition, isotherm

Offset (from funcs[i] start, hex)

Isotherm limits have a somewhat forced definition to be compatible with an earlier image format version

+04h	unsigned short wLowLSW;	Least significant word of low isotherm limit
+06h	unsigned short wLowMSW;	Most significant word of low isotherm limit
+08h	unsigned short wHighLSW;	Least significant word of high isotherm limit
+0Ah	unsigned short wHighMSW;	Most significant word of high isotherm limit
+0Ch	unsigned short calcField;	0 – Isotherm not defined 1 – Isotherm below high limit 2 – Isotherm above high limit 3 – Isotherm between high and low limits
+0Eh	short filler2;	Unused data

9.2.14 Distribution information

Distribution information is given to let image subscribers more easily detect changes in different sub structures of the image data. Counters are incremented by

1 when the corresponding structure is updated. All counters start at 0 on camera reset. Observe that some structures do not have sub counters.

Offset (from block start, hex) Block start offset is +220h

+00h	unsigned long imageCount;	Updated for every new "system image"
+04h	unsigned long calcdataCount;	Updated when calcData (except distrInfo) is updated
+08h	unsigned long pixelsCount;	Updated when pixels is updated
+0Ch	unsigned short objectSubCount;	Updated when objectPars is updated
+0Eh	unsigned short calibSubCount;	Updated when calibPars is updated
+10h	unsigned short presentSubCount;	Updated when presentPars is updated
+12h	unsigned short measureSubCount;	Updated when measureData is updated
+14h	unsigned long distrLive;	Set when image distribution is 'LIVE'
		0 otherwise ('freeze' or 'recalled')

9.3 Image pixels

Image pixels are represented as unsigned short (normally). Number of expected pixels in the image are given by the geomInfo structure imageWidth and imageHeight pixels should be seen from left to right and with rows from upper to lower. The normal image size is 320 x 240 pixels. Image pixel values are allowed to vary between 0 and 65535, with 0 as special, 'undefined pixel' value

Actual pixel value range depends of detector type and range. Use image data for actual limits ipixUnderflow and ipixOverflow.

Note that image pixels are uncompensated for camera internal temperature and detector non-linearity. See "Adjust parameters" for a formula how pixels should be compensated.

9.4 Text comments

Text comments come in pairs of Label and Value. The number and size of Labels and Values are not fixed or defined anywhere, but each Label and Value is separated by a NULL character.

Label1<NULL>Valule1<NULL>Label2<NULL>Value2<NULL>
…LabelN<NULL>ValueN<NULL>

The text comments are stored in a separate block of data usually following the pixels and the voice recording. In order to find them you have to read the file header and find an index entry with wMainType 4. It will give you the size and offset (from the beginning of the file) of the text comment block.

9.5 The file header

The camera embeds the image data and pixels when building an image file. This chapter describes the file header. The header contains of two parts. One general part with fixed size 64 bytes and one index table which size may vary. The header data are always represented as "big endian".

9.5.1 General part

Offset (from block start, hex)

+00h	char szFormatID[4];	Fileformat ID; Identifies file as 'Agema File Format' Should be "AFF"
+04h	char szOrigin[16];	File origin, "THV550" or “THV570”
+14h	unsigned long dwVersion	File format version. Should be 100 for this definition
+18h	unsigned long dwIndexOff;	Offset to index part relative file start
+1Ch	unsigned long dwAntIndex;	Total number of indexes in index table (used + free)
+20h	unsigned long dwNextIndexID;	Next free index ID, starting on 1
+24h	char szSpare[16];	Spare for future usage
+34h	unsigned long dwProtection;	R /W protection. 0 - no protection

		1 - read protection, 2 - write protection
+38h	unsigned long dwPWChecksum;	Password checksum.
+3Ch	unsigned long dwChecksum;	Checksum calculated on general header and index table Checksum should be calculated as XOR of all longwords in header

9.5.2 Index part

The index part contains a number of equally formatted descriptors pointing out the actual data blocks in the file. Each descriptor is 32 bytes. Total number of descriptors is given by dwAntIndex. One descriptor looks like:

Offset (from block start, hex)

+00h	unsigned short wMainType;	A registered number that uniquely defines one data type, for instance: 'image' or 'voice comment'. The following main types are registered: 0 – this entry is 'not used' 1 – image data + pixels (described in this section) Pixels are stored immediately after data 2 – voice comment 3 – IRwin External analysis 4 – text comments 5 – 7 - for future use in camera 8 – 15 - IRwin use
+02h	unsigned short wSubType;	Some main types may have variants of data interpretation. For main type 1 (image) there are: Data represents an image with data+pixels pixel format is given by first data word (geometric info, pixelSize), which may take the values 1,2 or 4. Normal images have pixelSize=2 (unsigned short). **Observe** that data may be stored as big endian or little endian. First word (pixelSize) shows if data has the expected endian format, or must be structure swapped on reading (pixelSize 0200h, 0100h or 0400h) Images stored in camera are stored in big endian. Images stored in some ThermaCAM PC programs

		may store image data and pixels in little endian for performance reasons (PCs are little endian). But as mentioned in the beginning of this chapter, the header (general + index table) is always big endian Data is a sequence of images stacked directly after each other (image = data+pixels). Endian order as given by first data word (as for single image), number of images stacked is given by dwObjectNr below
+04h	unsigned long dwVersion;	Version of the data structure. This document describes version 103
+08h	unsigned long dwIndexID;	Index ID of actual data. IDs are taken from dwNextIndexID when inserting a new data structure in the file. Structures that should be treated as a group (image and voice comment) have the same dwIndexID.
+0Ch	unsigned long dwDataPtr;	Offset to the actual data structure from the beginning of the file
+10h	unsigned long dwDataSize;	Size of the actual data structure
+14h	unsigned long dwParent;	Not used for IR images
+18h	unsigned long dwObjectNr;	Not used for single IR images (wMainType=1, wSubType=0) number of images in sequence (wMainType=0, wSubType=3)
+1Ch	unsigned long dwChecksum;	Not used for IR images. Calculated as XOR of all longwords in the actual data structure. =0 if not used. If checksum is used and is calculated to 0, FFFFFFFFh is written as dwChecksum

10. Temperature calculation on IR images

To convert an image pixel value to a corresponding object temperature, one must perform some calculations using the data part of the image. The following formulas should be applied

1. Convert from image pixel value to a temperature compensated, linearized pixel value, here called 'absPixel'. See discussion in chapter Adjust parameters on page 41 how to do this.
2. Calculate some temporary results to use (parameters here are from image "Object parameters" and "Calibration parameters").

   tatmC = atmTemp - 273.15

   $$H2O = relHum \cdot e^{(1.5587+6.939\cdot10^{-2}\cdot tatmC-2.7816\cdot10^{-4}\cdot tatmC^2+6.8455\cdot10^{-7}\cdot tatmC^3)}$$

   $$\tau = X \cdot e^{\left[-\sqrt{objectDistance}\cdot(alpha1+beta1\cdot\sqrt{H2O})\right]} + (1-X)\cdot e^{\left[-\sqrt{objectDistance}\cdot(alpha2+beta2\cdot\sqrt{H2O})\right]}$$

   $$K1 = \frac{1}{emissivity \cdot \tau}$$

   $$K2 = \left[\frac{1-emissivity}{emissivity}\cdot\frac{R}{\left(e^{\frac{B}{ambTemp}}-F\right)} + \frac{1-\tau}{emissivity\cdot\tau}\cdot\frac{R}{\left(e^{\frac{B}{ambTemp}}-F\right)}\right]$$

3. object signal (the corresponding signal level on the object) is calculated with the formula:

   $$objectSignal(absPixel) = \frac{K1}{2}\cdot absPixel - K2$$

4. The corresponding temperature (in Kelvin) is given by the formula:

   $$objectTemp(objectSignal) = \frac{B}{\ln\left(\frac{R}{objectSignal}+F\right)}$$

11. Index

A

Acquire images 12
Adjust parameters 41
affimage 21
area 29
areapixel 31
areapos 30
autoadj 27
autozoom 24

B

battery 28
baudrate 35

C

Camera communication 18
Camera connector 19
Configuration file 12
cooler 28

D

date 35
diff 34
digital 23
Distribution information 49
dobj 22

E

echo 34
emissivity 22

F

File format, general part 51
File format, index part 52
File header 51
filter 25
focusmov 26

G

Geometric info 37

I

image16 20
image8 20
imagehead 21
imagepix16 21
imagepix8 21
imagesub 21
Introduction 6
iso 32
isocol 32
isopos 32

L

level 26

M

measclip 34
measure 29
mwait 29

N

noise 24

O

Object parameters 39
omapcalc: 27

P

palette 26
pan 24
poll 22
poweroff 27
Presentation parameters 43
prof 33
profpos 33

Q

quit 18

R

rangelims 25
readarea 30
readareak 30
readiso 33
readisok 32
readprof 33
readprofk 33
readspot 31
readspotk 31
relhum 23
Requirements 11

S

shutter 28
span 27
spot 31
spotpos 31
System parts 8

T

tamb 23
tatm 23
Temperature calculations 54
temprange 25
Text comments 51
tref 34

V

versions 35
vidinv 25
viewfinder 28

Z

zoom 24

USA (Portland)	Tel: +1 (503) 684-3731	Fax: +1 (503) 684-3207
USA (Billerica)	Tel: +1 (978) 901-8000	Fax: +1 (978) 901-8887
Sweden	Tel: +46 (0) 8 753 27 50	Fax: +46 (0) 8 755 07 52
France	Tel: +33 (0) 1 41 33 97 97	Fax: +33 (0) 1 47 36 18 32
Germany	Tel: +49 (0) 6995 0090-0	Fax: +49 (0) 6995 0090-40
Italy	Tel: +39 (0) 2-3909121	Fax: +39 (0) 2-39005185
UK	Tel: +44 (0) 1732 220011	Fax: +44 (0) 1732 220014

(April 2001)