UART-Interface:

Transfer rate:

HTPA8x8:	Send/receive:	460800 baud
HTPA16x16:	Send/receive:	115200 baud
HTPA32x31:	Send/receive:	460800 baud
HTPA32x31:	Send/receive:	460800 baud

Protocol Specifications:

Data format:	8N1 (8 data bits, no parity, 1 stop bit)
Flow Control:	NONE
Local Options:	Raw Input Voltage and Temperature Output

Electrical Specifications:

VDD:	Supply (+3.3V DC)
UART Transmit/Receive:	TTL
VSS	GND
Power Supply:	**3.3** VDC +/- 5%, 300mA
IDD (Idle mode)	75 mA
IDD (Operating mode)	135 mA

Pinout:

No.	Function	Type
1	VDD	Power
2	VDD	Power
3	VSS	Power
4	VSS	Power
5	SS#	Input
6	UTX/SDO **	Output
7	URX/SDI **	Input
8	SCK	Input
9	CTX*	Output
10	CRX*	Input

* Pins planned for CAN Bus.

** URX/UTX is used for the UART Module

SDI/SDO/SS#/SCK pins support the Module with SPI Interface.

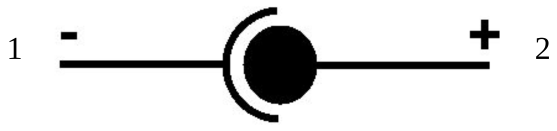
Connector layout: see drawings.

Ethernet-Interface:

Protocol Specifications:

Protocol type:	UDP
All communication on Port:	30444

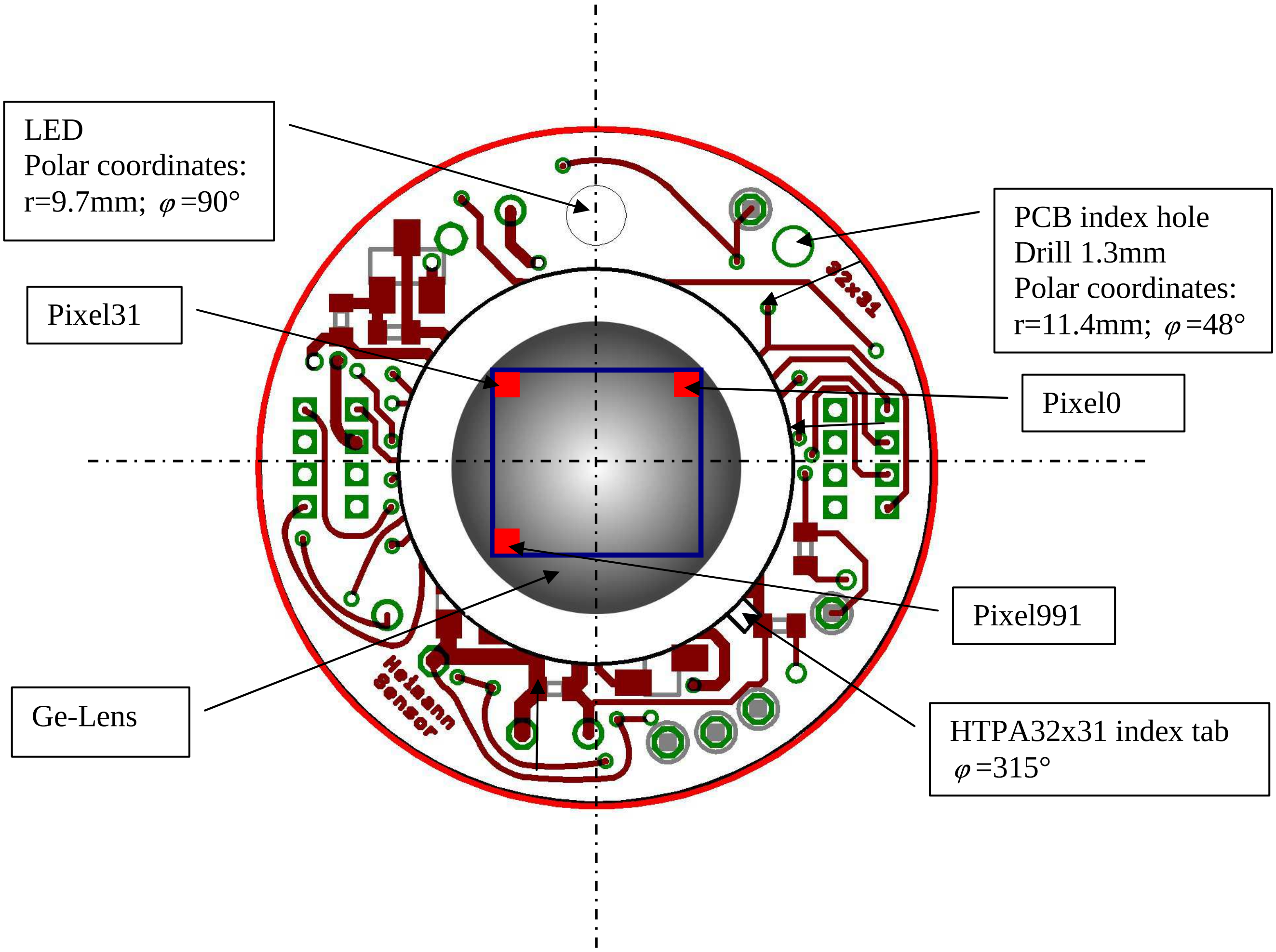
Power connection at Ethernet device:



1	VSS	(-)	GND
2	VDD	(+)	Supply (+3.3V DC)

Power Supply: **3.3** VDC +/- 5%, 300mA

HTPA32x31 UDP Module Optical Orientation of Pixels:



LED
Polar coordinates:
r=9.7mm; φ=90°

Pixel31

PCB index hole
Drill 1.3mm
Polar coordinates:
r=11.4mm; φ=48°

Pixel0

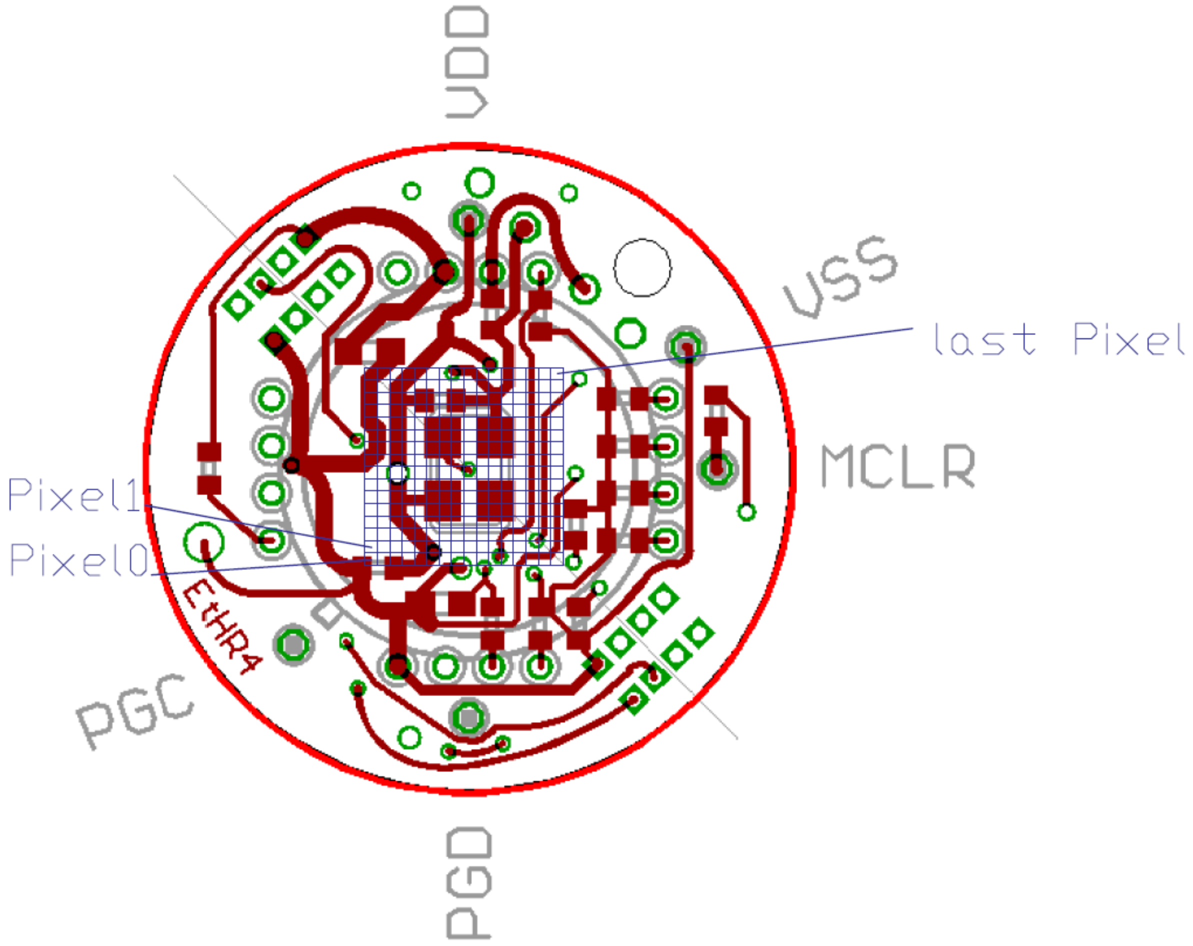
Pixel991

Ge-Lens

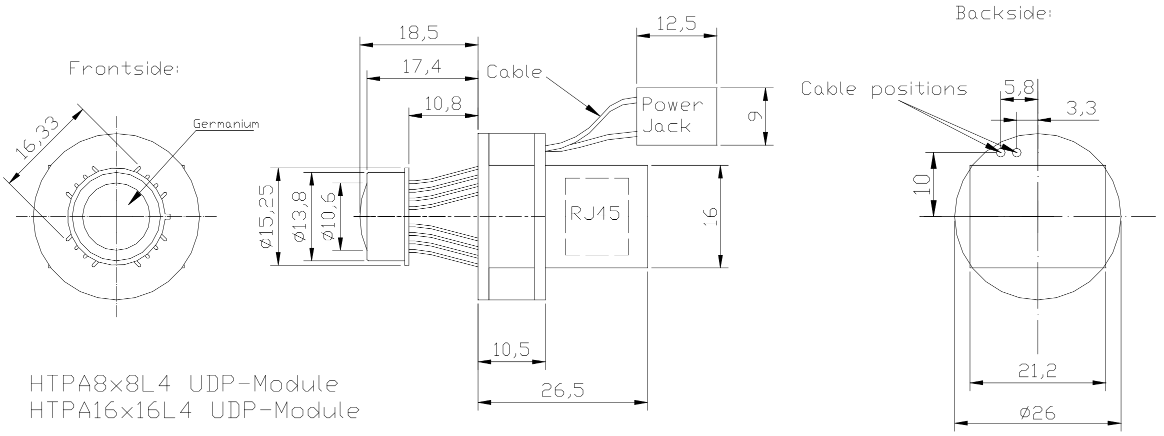
HTPA32x31 index tab
φ=315°

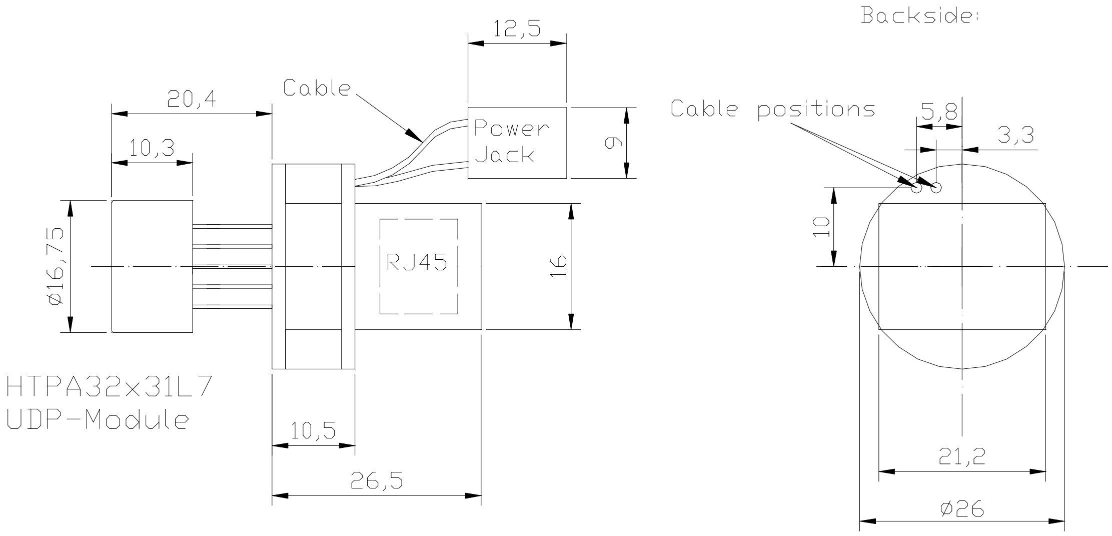
Top View

Optical Orientation HTPA16x16 UDP Module:



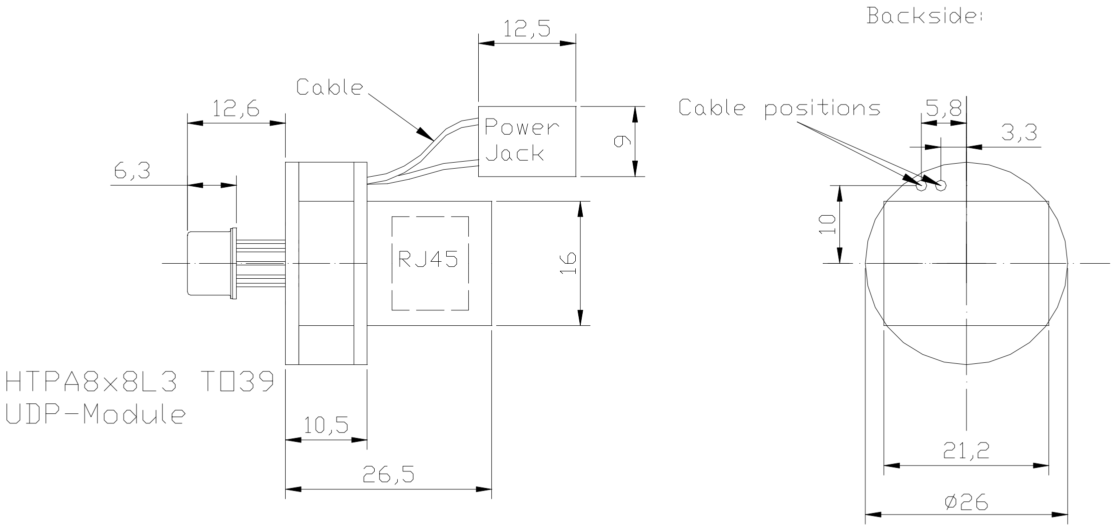
Module dimensions:

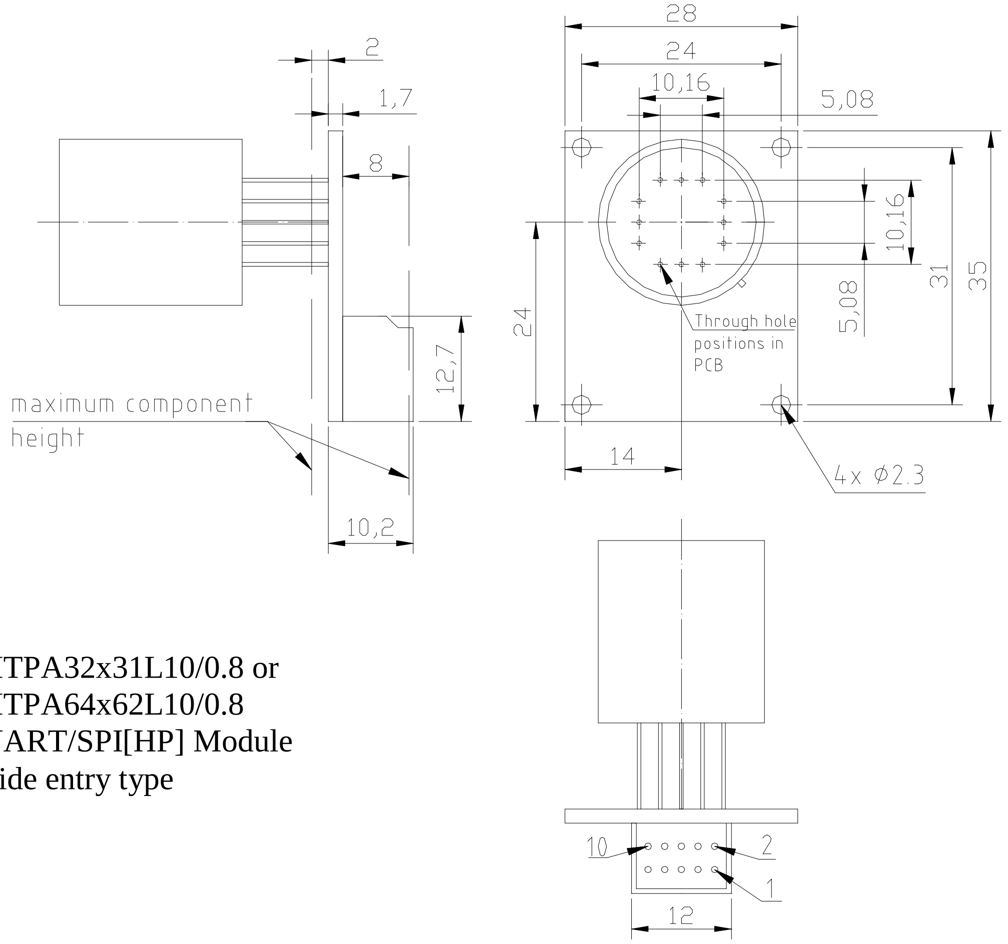




In these drawings always only one optic option is shown for each HTPA type. If another optic is built in, refer to the datasheet of the HTPA sensor. PCB size, pin length of the HTPA and positions on the PCB remain the same, only the cap of the HTPA changes.

Module dimensions (continued):

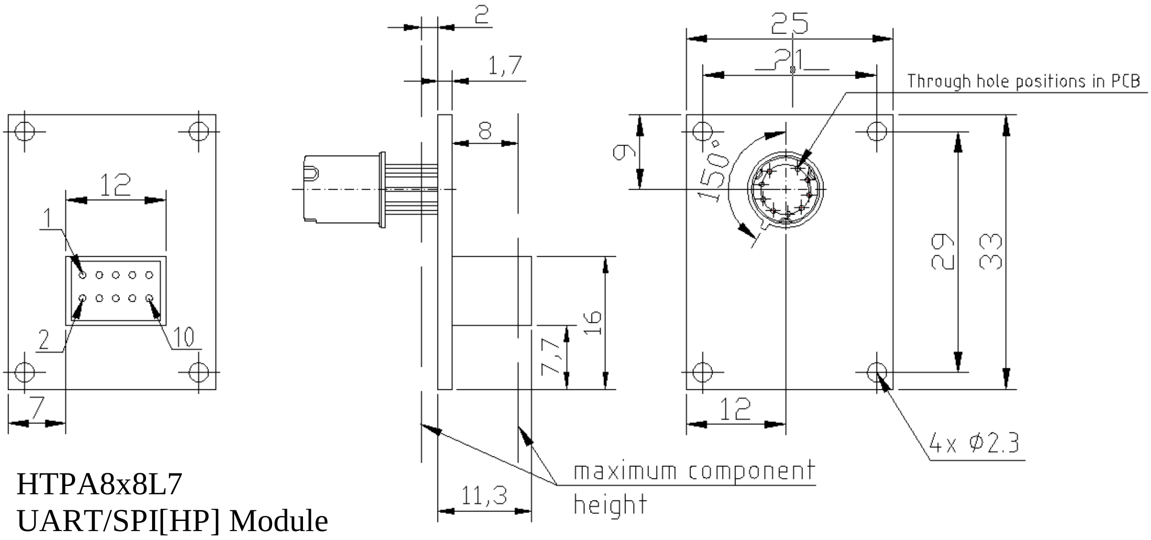




HTPA32x31L10/0.8 or
HTPA64x62L10/0.8
UART/SPI[HP] Module
Side entry type

In these drawings always only one optic option is shown for each HTPA type. If another optic is built in, refer to the datasheet of the HTPA sensor. PCB size, pin length of the HTPA and positions on the PCB remain the same, only the cap of the HTPA changes.

Module dimensions (continued):



HTPA8x8L7
UART/SPI[HP] Module
Top entry type

In these drawings always only one optic option is shown for each HTPA type. If another optic is built in, refer to the datasheet of the HTPA sensor. PCB size, pin length of the HTPA and positions on the PCB remain the same, only the cap of the HTPA changes.

Communication:

<table>
<tr><th colspan="5">*Communication via Terminal / UDP*</th></tr>
<tr><th>Sent Char</th><th>HTPA8x8</th><th>HTPA16x16</th><th>HTPA32x31
HTPA64x62</th><th>Result/Received message</th></tr>
<tr><td>'a'</td><td>X</td><td>X</td><td>X</td><td>Decreases the operating frequency of the array</td></tr>
<tr><td>'A'</td><td>X</td><td>X</td><td>X</td><td>Increases the operating frequency of the array</td></tr>
<tr><td>'b'</td><td>X</td><td>X</td><td>X</td><td>Measure VDD (referenced to VREF1225)</td></tr>
<tr><td>'C'</td><td>X</td><td>X</td><td></td><td>Capture single voltage frame. Use ADC of ASIC. Output via ASCII if sent via UART, binary if sent via UDP.</td></tr>
<tr><td>'c'</td><td>X</td><td>X</td><td>X</td><td>Capture single voltage frame. Use ADC of µC. Output via ASCII if sent via UART, binary if sent via UDP.</td></tr>
<tr><td>'d'/'D'</td><td>X</td><td>X</td><td></td><td>Toggle POR_N</td></tr>
<tr><td>'f'</td><td>X</td><td>X</td><td>X</td><td>Toggle Resetbit</td></tr>
<tr><td>'F'</td><td>X</td><td>X</td><td></td><td>Analog operating point is at start of AD-range, only positive signals convertable</td></tr>
<tr><td>'G'</td><td>X</td><td>X</td><td></td><td>Analog operating point is in the middle of AD-range, positive and negative signals convertable</td></tr>
<tr><td>'g'</td><td>X</td><td>X</td><td></td><td>Analog operating point is at end of AD-range, only negative signals convertable</td></tr>
<tr><td>'h'</td><td>X</td><td>X</td><td>X</td><td>pushes binary EEDATA out</td></tr>
<tr><td>'i'</td><td></td><td></td><td>X</td><td>Read single voltage frame. Output in ASCII format. Serial order: Pixeldata[K*10], el. Offsets, Ambient Temperature</td></tr>
<tr><td>'I'</td><td></td><td></td><td>X</td><td>Read single temperature frame. Output in ASCII format. Serial order: Pixeldata[K*10], el. Offsets, Ambient Temperature</td></tr>
<tr><td>'J'</td><td>X</td><td>X</td><td>X</td><td>Toggle Amplification</td></tr>
<tr><td>'k'</td><td>X</td><td>X</td><td>X</td><td>Read single temperature frame. Output in binary format.</td></tr>
<tr><td>'K'</td><td>X</td><td>X</td><td>X</td><td>send continous binary temperature datastream(µC-ADC)[K*10]
Output of a complete cycle in this order:

HTPA 8x8 and HTPA16x16: Pixel0,Pixel1, ….PixelX, el.Offset0, el.Offset1,…, el.OffsetY,PTAT0,PTAT1,…,PTATZ
HTPA32x31: see Table2.
For a detailed Description of the serial order see Table2.

16x16 Array: X=255; Y=7; Z=7
8x8 Array: X=63; Y=4; Z=4

One dataset has exactly 2 bytes: first the low-Byte is send, then the high-byte. Each Dataset contains the measured Temperature in Kelvin*10. The first 4 datasets *el.Offset0…el.Offset3* after the last Pixel voltage *PixelX* transmit additional the current VDD in the MSB's:

VDD and TAmb for HTPA8x8 and HTPA16x16:
<table>
<tr><th>Dataset</th><th>Bit15</th><th>Bit14</th><th>Bit13</th><th>Bit12</th><th>Bit11</th><th>Bit10</th><th>…</th><th>Bit1</th><th>Bit0</th></tr>
<tr><td>elOff0</td><td>MSB VDD</td><td>…</td><td>…</td><td>Bit12 VDD</td><td>MSB elOff0</td><td>…</td><td>…</td><td>…</td><td>LSB elOff0</td></tr>
<tr><td>elOff1</td><td>Bit11 VDD</td><td>…</td><td>…</td><td>Bit8 VDD</td><td>MSB elOff1</td><td>…</td><td>…</td><td>…</td><td>LSB elOff1</td></tr>
<tr><td>elOff2</td><td>Bit7 VDD</td><td>…</td><td>…</td><td>Bit4 VDD</td><td>MSB elOff2</td><td>…</td><td>…</td><td>…</td><td>LSB elOff2</td></tr>
<tr><td>elOff3</td><td>Bit3 VDD</td><td>…</td><td>…</td><td>LSB VDD</td><td>MSB elOff3</td><td>…</td><td>…</td><td>…</td><td>LSB elOff3</td></tr>
</table>
The Sensor temperature is available in the datasets after *el.Offset3* :
<table>
<tr><th>Dataset</th><th>Bit15</th><th>Bit14</th><th>Bit13</th><th>Bit12</th><th>Bit11</th><th>Bit10</th><th>…</th><th>Bit1</th><th>Bit0</th></tr>
<tr><td>elOff3+1</td><td>MSB TAmb</td><td>…</td><td>…</td><td>Bit12 TAmb</td><td>MSB elOff3+1</td><td>…</td><td>…</td><td>…</td><td>LSB elOff3+1</td></tr>
<tr><td>elOff3+2</td><td>Bit11 TAmb</td><td>…</td><td>…</td><td>Bit8 TAmb</td><td>MSB elOff3+2</td><td>…</td><td>…</td><td>…</td><td>LSB elOff3+2</td></tr>
<tr><td>elOff3+3</td><td>Bit7 TAmb</td><td>…</td><td>…</td><td>Bit4 TAmb</td><td>MSB elOff3+3</td><td>…</td><td>…</td><td>…</td><td>LSB elOff3+3</td></tr>
<tr><td>elOff3+4</td><td>Bit3 TAmb</td><td>…</td><td>…</td><td>LSB TAmb</td><td>MSB elOff3+4</td><td>…</td><td>…</td><td>…</td><td>LSB elOff3+4</td></tr>
<tr><td>elOff3+5</td><td>0</td><td>0</td><td>0</td><td>0</td><td>MSB elOff3+5</td><td>…</td><td>…</td><td>…</td><td>LSB elOff3+5</td></tr>
</table>
</td></tr>
<tr><td>'l'</td><td>X</td><td>X</td><td>X</td><td>Get Ambient Temperature (Calculates the Ambient Temperature from the **last measured** Frame)</td></tr>
<tr><td>'m'</td><td>X</td><td>X</td><td>X</td><td>Toggle usage of µC-Buffer for el. Offsets (Stack depth = 64 for HTPA8x8 and HTPA16x16; Stack depth = 32 for HTPA32x31)</td></tr>
<tr><td>'M'</td><td>X</td><td>X</td><td>X</td><td>Shows current and calibration settings. Device prints the following stream:
"HTPA series responsed! I am Arraytype X" Possible values for X: "0"=HTPA8x8, "1"=HTPA16x16, "3"=HTPA32x31
"Firmware v.X.XX written by B.Forg; Heimann Sensor GmbH; YYYY-MM-DD" Version information.
"I am running on XXXX.X kHz" Actual MCLK-setting in kHz
"Amplification is X" Actual set amplification. Possible strings for X: "low" or "high"
"MAC-ID: X IP: Y DevID: Z\r\n" (Only Ethernet devices show a MAC-ID, DevID is shown in any case)
X= MAC-ID of the device, i.e. "00.97.FF.00.10.08"; Y=current IP of the device, Z=user setable ID, range 00000…65535
"PIXCvsTA *X* , BFL3 *X* , F8_14 *X* , THvsTA *X* IGNORE_ELOFF *X* ELOFF32 *X* SBY *Y* FC *X* EXP *Z* "</td></tr>
</table>

Table1: Control Characters

Table1, continuation:

<table>
<tr><th colspan="5"><i>Communication via Terminal / UDP</i></th></tr>
<tr><th>Sent Char</th><th>HTPA8x8</th><th>HTPA16x16</th><th>HTPA32x31
HTPA64x62</th><th>Result/Received message</th></tr>
<tr><td>'o'</td><td></td><td>X</td><td>X</td><td>Use external reference voltages</td></tr>
<tr><td>'O'</td><td></td><td>X</td><td>X</td><td>Use internal reference voltages</td></tr>
<tr><td>'q'/'Q'</td><td>X</td><td>X</td><td>X</td><td>Allow Changes (required for Calibration)</td></tr>
<tr><td>'t'</td><td>X</td><td>X</td><td>X</td><td>Continuous binary voltage data of the µC-ADC is transmitted.
Output of a complete cycle in this order:

<i>HTPA 8x8 and HTPA16x16: Pixel0,Pixel1, ….PixelX, el.Offset0, el.Offset1,…, el.OffsetY,PTAT0,PTAT1,…,PTATZ</i>
<i>HTPA32x31: see Table2.</i>
<b><i>For a detailed Description of the serial order see Table2.</i></b>

16x16 Array: 8x8 Array:
X=255; Y=7; Z=7 X=63; Y=4; Z=4

One dataset has exactly 2 bytes: first the low-Byte is send, then the high-byte. Each Dataset contains the ADC-Data in digits and The first 4 datasets <i>el.Offset0…el.Offset3</i> after the last Pixel voltage <i>PixelX</i> transmit additional the current VDD in the MSB’s:

<b>VDD for HTPA8x8 and HTPA16x16:</b>
<table>
<tr><th>Dataset</th><th>Bit15</th><th>Bit14</th><th>Bit13</th><th>Bit12</th><th>Bit11</th><th>Bit10</th><th>…</th><th>Bit1</th><th>Bit0</th></tr>
<tr><td>elOff0</td><td>MSB VDD</td><td>…</td><td>…</td><td>Bit12 VDD</td><td>MSB elOff0</td><td>…</td><td>…</td><td>…</td><td>LSB elOff0</td></tr>
<tr><td>elOff1</td><td>Bit11 VDD</td><td>…</td><td>…</td><td>Bit8 VDD</td><td>MSB elOff1</td><td>…</td><td>…</td><td>…</td><td>LSB elOff1</td></tr>
<tr><td>elOff2</td><td>Bit7 VDD</td><td>…</td><td>…</td><td>Bit4 VDD</td><td>MSB elOff2</td><td>…</td><td>…</td><td>…</td><td>LSB elOff2</td></tr>
<tr><td>elOff3</td><td>Bit3 VDD</td><td>…</td><td>…</td><td>LSB VDD</td><td>MSB elOff3</td><td>…</td><td>…</td><td>…</td><td>LSB elOff3</td></tr>
</table></td></tr>
<tr><td>'T'</td><td>X</td><td>X</td><td></td><td>Continuous binary data of the ASIC-ADC is transmitted.
Output order is equal to ‘t’.</td></tr>
<tr><td>'u'</td><td>X</td><td>X</td><td></td><td>Continuous binary data of the ASIC-ADC is transmitted. PTAT-Voltages are sampled with the uC-ADC.
Output order is equal to ‘t’.</td></tr>
<tr><td>'U'</td><td>X</td><td>X</td><td></td><td>Capture single frame. Use ADC of ASIC. Output via ASCII. PTAT-Voltages are sampled with the uC-ADC.</td></tr>
<tr><td>'v'</td><td>X</td><td>X</td><td>X</td><td>Announce IP (Only Ethernet devices)</td></tr>
<tr><td>'V'</td><td>X</td><td>X</td><td>X</td><td>Device awaits control message (only non-Ethernet devices)</td></tr>
<tr><td>'w'</td><td>X</td><td>X</td><td>X</td><td>shows Calibration-constants</td></tr>
<tr><td>'W'</td><td>X</td><td>X</td><td>X</td><td>Calibration. ATTENTION! Old Dataset cannot be restored!</td></tr>
<tr><td>'x'</td><td>X</td><td>X</td><td>X</td><td>Stops Stream without prompt.</td></tr>
<tr><td>'X'</td><td>X</td><td>X</td><td>X</td><td>Stops Stream by sending “STOP!\r\n”</td></tr>
<tr><td>'y'</td><td>X</td><td>X</td><td>X</td><td>switch off ASIC-Supply (5V)</td></tr>
<tr><td>'Y'</td><td>X</td><td>X</td><td>X</td><td>switch on ASIC-Supply (5V)</td></tr>
</table>

Table1 (continuation): Control Characters

Serial order in Frame:

HTPA8x8 Temperature Mode	
Dataset	Value
0	Temperature of Pixel0 in K*10
1	Temperature of Pixel1 in K*10
2	Temperature of Pixel2 in K*10
3	Temperature of Pixel3 in K*10
…	…
63	Temperature of Pixel63 in K*10
64	4 bits of VDD and elOff0 in digits (refer to Table1)
65	4 bits of VDD and elOff1 in digits (refer to Table1)
66	4 bits of VDD and elOff2 in digits (refer to Table1)
67	4 bits of VDD and elOff4 in digits (refer to Table1)
68	4 bits of TAmb and PTAT0 in digits (refer to Table1)
68	4 bits of TAmb and PTAT1 in digits (refer to Table1)
68	4 bits of TAmb and PTAT2 in digits (refer to Table1)
68	4 bits of TAmb and PTAT3 in digits (refer to Table1)

HTPA8x8 Voltage Mode	
Dataset	Value
0	absolute Voltage of Pixel0 in digits
1	absolute Voltage of Pixel1 in digits
2	absolute Voltage of Pixel2 in digits
3	absolute Voltage of Pixel3 in digits
…	…
63	absolute Voltage of Pixel63 in digits
64	4 bits of VDD and elOff0 in digits (refer to Table1)
65	4 bits of VDD and elOff1 in digits (refer to Table1)
66	4 bits of VDD and elOff2 in digits (refer to Table1)
67	4 bits of VDD and elOff4 in digits (refer to Table1)
68	4 bits of TAmb and PTAT0 in digits (refer to Table1)
68	4 bits of TAmb and PTAT1 in digits (refer to Table1)
68	4 bits of TAmb and PTAT2 in digits (refer to Table1)
68	4 bits of TAmb and PTAT3 in digits (refer to Table1)

HTPA16x16 Temperature Mode	
Dataset	Value
0	Temperature of Pixel0 in K*10
1	Temperature of Pixel1 in K*10
2	Temperature of Pixel2 in K*10
3	Temperature of Pixel3 in K*10
…	…
255	Temperature of Pixel255 in K*10
256	4 bits of VDD and elOff0 in digits (refer to Table1)
257	4 bits of VDD and elOff1 in digits (refer to Table1)
258	4 bits of VDD and elOff2 in digits (refer to Table1)
259	4 bits of VDD and elOff3 in digits (refer to Table1)
260	elOff4 in digits
261	elOff5 in digits
262	elOff6 in digits
263	elOff7 in digits
264	4 bits of TAmb and PTAT0 in digits (refer to Table1)
265	4 bits of TAmb and PTAT1 in digits (refer to Table1)
266	4 bits of TAmb and PTAT2 in digits (refer to Table1)
267	4 bits of TAmb and PTAT3 in digits (refer to Table1)
268	PTAT4 in digits
…	…
271	PTAT7 in digits

HTPA16x16 Voltage Mode	
Dataset	Value
0	absolute Voltage of Pixel0 in digits
1	absolute Voltage of Pixel1 in digits
2	absolute Voltage of Pixel2 in digits
3	absolute Voltage of Pixel3 in digits
…	…
255	absolute Voltage of Pixel255 in digits
256	4 bits of VDD and elOff0 in digits (refer to Table1)
257	4 bits of VDD and elOff1 in digits (refer to Table1)
258	4 bits of VDD and elOff2 in digits (refer to Table1)
259	4 bits of VDD and elOff3 in digits (refer to Table1)
260	elOff4 in digits
261	elOff5 in digits
262	elOff6 in digits
263	elOff7 in digits
264	4 bits of TAmb and PTAT0 in digits (refer to Table1)
265	4 bits of TAmb and PTAT1 in digits (refer to Table1)
266	4 bits of TAmb and PTAT2 in digits (refer to Table1)
267	4 bits of TAmb and PTAT3 in digits (refer to Table1)
268	PTAT4 in digits
…	…
271	PTAT7 in digits

Table2: Serial order

Each dataset consists of a 16 bit value. If a frame consists out of more than one packet, packets are appended. For UART devices the 16 bit values are transmitted with LSB first.

HTPA32x31 Temperature Mode

Dataset	Value
0	Temperature of Pixel0 in K*10
1	Temperature of Pixel16 in K*10
2	Temperature of Pixel1 in K*10
3	Temperature of Pixel17 in K*10
…	…
30	Temperature of Pixel15 in K*10
31	Temperature of Pixel31 in K*10
32	Temperature of Pixel32 in K*10
33	Temperature of Pixel48 in K*10
…	…
991	Temperature of Pixel991 in K*10
992	elOff0 in digits
993	elOff16 in digits
994	elOff1 in digits
995	elOff17 in digits
…	…
1022	elOff15 in digits
1023	elOff31 in digits
1024	least significant 12 bits of VDD
1025	most significant 4 bits of VDD
1026	least significant 12 bits of TAmb
1027	most significant 4 bits of TAmb
1028	no value, ignore
1029	no value, ignore
…	…
1039	no value, ignore
1040	PTAT0 in digits
1041	no value, ignore
1042	PTAT1 in digits
…	…
1053	no value, ignore
1054	PTAT7 in digits
1055	no value, ignore

HTPA32x31 Voltage Mode

Dataset	Value
0	absolute Voltage of Pixel0 in digits
1	absolute Voltage of Pixel16 in digits
2	absolute Voltage of Pixel1 in digits
3	absolute Voltage of Pixel17 in digits
…	…
30	absolute Voltage of Pixel15 in digits
31	absolute Voltage of Pixel31 in digits
32	absolute Voltage of Pixel32 in digits
33	absolute Voltage of Pixel48 in digits
…	…
991	absolute Voltage of Pixel991 in digits
992	elOff0 in digits
993	elOff16 in digits
994	elOff1 in digits
995	elOff17 in digits
…	…
1022	elOff15 in digits
1023	elOff31 in digits
1024	least significant 12 bits of VDD
1025	most significant 4 bits of VDD
1026	no value, ignore
1027	no value, ignore
1028	no value, ignore
1029	no value, ignore
…	…
1039	no value, ignore
1040	PTAT0 in digits
1041	no value, ignore
1042	PTAT1 in digits
…	…
1053	no value, ignore
1054	PTAT7 in digits
1055	no value, ignore

HTPA64x62 Temperature Mode

Dataset	Value
0	Temperature of Pixel0 in K*10
1	Temperature of Pixel32 in K*10
2	Temperature of Pixel1 in K*10
3	Temperature of Pixel33 in K*10
…	…
62	Temperature of Pixel31 in K*10
63	Temperature of Pixel63 in K*10
64	Temperature of Pixel64 in K*10
65	Temperature of Pixel96 in K*10
…	…
3967	Temperature of Pixel3967 in K*10
3968	elOff0 in digits
3969	elOff32 in digits
3970	elOff1 in digits
3971	elOff33 in digits
…	…
4030	elOff31 in digits
4031	elOff63 in digits
4032	least significant 12 bits of VDD
4033	most significant 4 bits of VDD
4034	least significant 12 bits of TAmb
4035	most significant 4 bits of TAmb
4036	no value, ignore
4037	no value, ignore
…	…
4047	no value, ignore
4048	PTAT0 in digits
4049	PTAT1 in digits
4050	PTAT2 in digits
…	…
4063	PTAT15 in digits
4064	no value, ignore
…	no value, ignore
4095	no value, ignore

HTPA64x62 Voltage Mode

Dataset	Value
0	absolute Voltage of Pixel0 in digits
1	absolute Voltage of Pixel32 in digits
2	absolute Voltage of Pixel1 in digits
3	absolute Voltage of Pixel33 in digits
…	…
62	absolute Voltage of Pixel31 in digits
63	absolute Voltage of Pixel63 in digits
64	absolute Voltage of Pixel64 in digits
65	absolute Voltage of Pixel96 in digits
…	…
3967	absolute Voltage of Pixel3967 in digits
3968	elOff0 in digits
3969	elOff32 in digits
3970	elOff1 in digits
3971	elOff33 in digits
…	…
4030	elOff31 in digits
4031	elOff63 in digits
4032	least significant 12 bits of VDD
4033	most significant 4 bits of VDD
4034	no value, ignore
4035	no value, ignore
4036	no value, ignore
4037	no value, ignore
…	…
4047	no value, ignore
4048	PTAT0 in digits
4049	PTAT1 in digits
4050	PTAT2 in digits
…	…
4063	PTAT15 in digits
4064	no value, ignore
…	no value, ignore
4095	no value, ignore

Table2 (continuation): Serial order

Each dataset consists of a 16 bit value. If a frame consists out of more than one packet, packets are appended. For UART devices the 16 bit values are transmitted with LSB first.

Packets (UDP, only Ethernet device):

Number of packets	Packet size [byte]	HTPA type	Comments
1	144	HTPA8x8	-
1	544	HTPA16x16	-
2	1058+1054	HTPA32x31	see below for details
8	1101+621	HTPA64x62	see below for details

	Packet details for HTPA32x31	
Packet No.	Packet size	Packet contains
1	1058	Data of Pixel0 - Pixel528
2	1054	Data of Pixel529 to end of frame

	Packet details for HTPA64x62	
Packet No.	Packet size	Packet contains
1	1101	Packet index 1 (8bit), data of Pixel0-Pixel550
2	1101	Packet index 2 (8bit), data of Pixel551-Pixel1101
3	1101	Packet index 3 (8bit), data of Pixel1102-Pixel1652
4	1101	Packet index 4 (8bit), data of Pixel1653-Pixel2203
5	1101	Packet index 5 (8bit), data of Pixel2204-Pixel2754
6	1101	Packet index 6 (8bit), data of Pixel2755-Pixel3305
7	1101	Packet index 7 (8bit), data of Pixel3306-Pixel3856
8	621	Packet index 8 (8bit), data of Pixel3857 to end of frame

Each dataset (except of packet index) consists out of a 16 bit value. For serial order of the datasets refer to section “serial order in Frame”.

Control Messages:

In the set of control messages, expressions in angled braces have to be substituted by following strings:

[IP] insert IP in ASCII format, i.e.: *“192.168.240.122”*

[MACID] insert MAC ID in ASCII format and hexadecimal, i.e.: *“00.1A.22.33.44.55”*

[AT] insert index of array types in ASCII format

Array type	Index
HTPA 8x8	*“0”*
HTPA 16x16	*“1”*
HTPA 32x31	*“3“*
HTPA 64x62	*“5“*

[MCLK] insert Frequency of MCLK in ASCII format and kHz, i.e.: *“1050.1”*

[AMP] insert state of amplification in ASCII format:

State	String
Low	*“low”*
High	*“high”*

[MSK] insert subnet mask in ASCII format, i.e.: *“255.255.255.000”*

[DEVID] insert 5 digit device ID in ASCII format, i.e. *“00197”* Range: 00000… 65535

Set of control messages:

Message1: "Calling HTPA series devices" (only Ethernet device)
Conditions: Can be sent as Broadcast, or if device already known as normal packet.
Answer: "HTPA series responsed! I am Arraytype **[AT]**"
Firmware version, date and author information.
"I am running on **[MCLK]** kHz"
"Amplification is **[AMP]**\r\n"
“MAC-ID: **[MACID]** IP: **[IP]**\r\n”
A second packet with calibration depending information is send.

Message2: "x Release HTPA series device" (only Ethernet device)
Result: Device disables hardware IP filter. All packets except ARP’s, DHCP requests, Broadcasts, Message1, Message3 and Message4 are discarded.
Answer: "HW-Filter released\r\n"

Message3: "HTPA device IP change request to **[IP]**.**[MSK]**." (only Ethernet device)
Result: The device changes the IP and the subnet mask to the given value and writes it to EEPROM. The IP becomes the default IP, therefore the device will use it at the next reset, if no DHCP is found.
Answer: "Device changed IP to **[IP]**. and Subnet to **[MSK]**.\r\n"

Message4: "Bind HTPA series device" (only Ethernet device)
Result: Device enables hardware IP filter. Only packets from sender IP, ARP’s, DHCP requests and Broadcasts are accepted. Device accepts now the control characters listed in **Table 1.**
Answer: "HW Filter is **[IP]** MAC **[MACID]**\n\r""
Insert in the above string the IP and MAC-ID of the Sender from Message4.

Control Messages [continued]:

Message5: "Set EEPROM data"
Conditions: Only possible if Message 4 already successful sent.
ATTENTION! Calibration data is overwritten!!!
Result: Writes the next received packets into EEPROM, if packet size is equal to 1024 bytes. Device writes to EEPROM, until EEPROM is completely filled. EEPROM size depends on Device type: HTPA8x8, HTPA16x16 and HTPA32x31: 16384 byte; HTPA64x62: 65536 byte.
Answer: "Write was successful.\n\r"

Message6: "Set DeviceID to **[DEVID]**"
Result: The given Device ID **[DEVID]** is written to EEPROM. This ID is shown on receive of ‘M’. The eDevice ID can be used for customer specific purposes.
Answer: "DeviceID changed to **[DEVID]**\r\n"

Temperature calculation:

To get the calibration settings for your device, request a single temperature frame, by sending ‘k’. Now the device automatically loaded the settings which were used during calibration. After the receive of the temperature frame request the actual configuration of the device by sending ‘M’. Store the information in the line:
"PIXCvsTA *X*, BFL3 *X*, F8_14 *X*, THvsTA *X* IGNORE_ELOFF *X* ELOFF32 *X* SBY *Y* FC *X* EXP *Z*". All cursive letters are device dependent constants. Possible strings: *X*="true" or "false", *Y*="1" or "0", *Z* is the string of a 2 digit decimal value, i. e. "3.47". Store those constants, the calculation algorithm is depending on those.
Now load the calibration constants by sending ‘w’. The module will send the calibration information, according to Table 3. Store the coloured constants. All stored constants will be labelled according to the value in “Name of constant”. If there is no coloured mark in the example string and no insertion in “Name of constant”, it is not necessary to store it.

Header of calibration data		
Example String	**Name of constant**	**Unit(s)**
"ASIC-Register @ Calib was 0x402\r\n"		
"Written back 0x402 to ASIC.\r\n"		
"PTAT-gradient 0.569000 dK/dig PTAT-Offset@0V 1973.000000\r\n"	$PTAT_G$, $PTAT_O$	[dK/dig], [dig]
"Ambient 1: 285.2, Ambient 2: 295.2 Ambient 3: 310.2 Ambient 4: 324.3 at Calibration\r\n"	T_{cT1}, T_{cT2}, T_{cT3}, T_{cT4}	[K]
"TObj1: 373.2, TObj2: 373.2 TObj3: 373.2 TObj4: 373.2 at Calibration\r\n"		
"TObjcal1: 285.9,, TObjcal2: 297.2 TObjcal3: 311.2 TObjcal4: 323.3 at Calibration\r\n"	T_{cO1}, T_{cO2}, T_{cO3}, T_{cO4}	[K]
"Arraytype is 3\r\n"		
"Nr. Thermaloff(1) PixC(1) Thermaloff(2) PixC(2) Thermaloff(3) PixC(3) Thermaloff(4) PixC(4) ThGrad ThOff\r\n"		

Table3: Header of calibration data

Constant:	Meaning:
$PTAT_G$	Gradient of the PTAT circuit in tenths of Kelvin per digits.
$PTAT_O$	Offset of the PTAT circuit in digits.
T_{cT1}	Absolute ambient temperature during calibration of the thermal offsets in calibration point 1.
T_{cT2}	Absolute ambient temperature during calibration of the thermal offsets in calibration point 2.
T_{cT3}	Absolute ambient temperature during calibration of the thermal offsets in calibration point 3.
T_{cT4}	Absolute ambient temperature during calibration of the thermal offsets in calibration point 4.
T_{cO1}	Absolute ambient temperature during calibration of the pixel constants in calibration point 1.
T_{cO2}	Absolute ambient temperature during calibration of the pixel constants in calibration point 2.
T_{cO3}	Absolute ambient temperature during calibration of the pixel constants in calibration point 3.
T_{cO4}	Absolute ambient temperature during calibration of the pixel constants in calibration point 4.

Table4: Meaning of constants (1)

After the header the pixel dependent constants are immediately transmitted. The number of packets (UDP) or lines (UART) depends on the device type and is equal to the number of sensitive pixels. Each packet / line consists of:

Pixelnumber, Thermal1(X), Pixelconstant1(X), Thermal2(X), Pixelconstant2(X), Thermal3(X), Pixelconstant3(X), Thermal4(X), Pixelconstant4(X).

X represents in the above line the corresponding pixel number.

Store all the pixel dependent constants.

Pixel dependent calibration data			
No.	**Example String**	**Name of constant**	**Comments**
0	"0 127 1316732 112 1182251 98 1126390 72 849857"	$Th_1(0)$, $P_1(0)$, $Th_2(0)$, $P_2(0)$, $Th_3(0)$, $P_3(0)$, $Th_4(0)$, $P_4(0)$,	Number of dataset grey marked
1	"1 137 1396731 132 1482251 128 1516391 127 1549867"	$Th_1(1)$, $P_1(1)$, $Th_2(1)$, $P_2(1)$, $Th_3(1)$, $P_3(1)$, $Th_4(1)$, $P_4(1)$,	Number of dataset grey marked
…	…	…	…
N	"N 127 1516732 112 1182251 98 1126390 72 849857"	$Th_1(N)$, $P_1(N)$, $Th_2(N)$, $P_2(N)$, $Th_3(N)$, $P_3(N)$, $Th_4(N)$, $P_4(N)$,	N equals the number of pixels-1. HTPA8x8→N=63 HTPA16x16→N=255 HTPA32x31→N=991

Table5: Pixel dependent calibration data

Constant:	**Meaning:**
$Th_1(X)$	Thermal offset of Pixel X in digits at the thermal calibration point 1 → T_{cT1}
$Th_2(X)$	Thermal offset of Pixel X in digits at the thermal calibration point 2 → T_{cT2} .
$Th_3(X)$	Thermal offset of Pixel X in digits at the thermal calibration point 3 → T_{cT3}
$Th_4(X)$	Thermal offset of Pixel X in digits at the thermal calibration point 4 → T_{cT4}
$P_1(X)$	Pixel constant of Pixel X at the object calibration point 1 → T_{cO1}
$P_2(X)$	Pixel constant of Pixel X at the object calibration point 2 → T_{cO2}
$P_3(X)$	Pixel constant of Pixel X at the object calibration point 3 → T_{cO3}
$P_4(X)$	Pixel constant of Pixel X at the object calibration point 4 → T_{cO4}

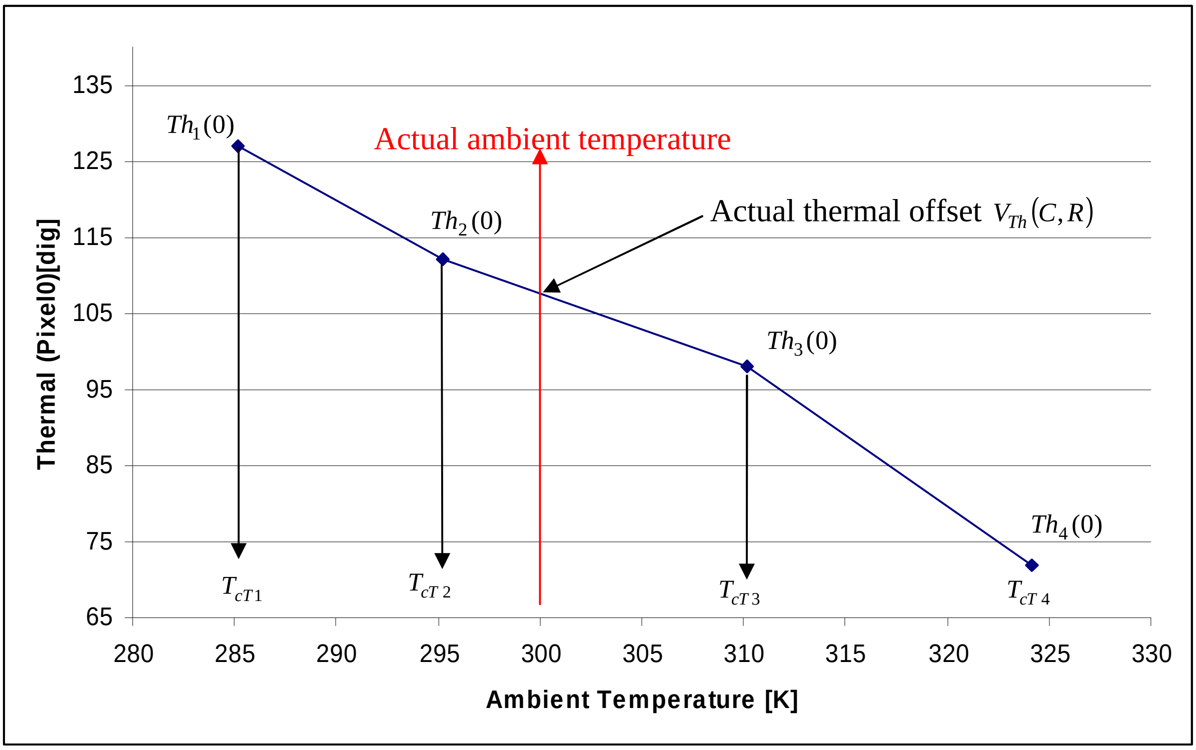
Table6: Meaning of constants (2)

The calibration is done with four different ambient temperature calibration points for Thermals and pixel constants, each.

Therefore, it is useful to calculate a gradient and an offset (each for thermal offsets and pixel constants) between two calibration points and to do a linear interpolation between the two calibration points, which are the closest to the actual ambient temperature. This will provide the highest accuracy.

For the following example we took the data of Pixel0, provided in Table 3 and Table 5. The linear interpolation is shown at the example of the thermal offset. Assume an actual ambient temperature of 300 K. Refer to chapter "Ambient temperature" for ambient temperature calculation.

Calculation example for actual thermal offset of Pixel 0 at 300 Kelvin:



To calculate the actual thermal offset $V_{Th}(C,R)$ a linear interpolation is sufficient. *(C,R)* represents the column and row dependency of the value. The same proceeding for the pixel constants is recommended.

Ambient Temperature:

Calculate the ambient temperature by following equation:

$$T_{Amb} = \frac{\sum_{i=0}^{N} PTAT_i}{N+1} \cdot PTAT_G + PTAT_O \ \text{[dK]}$$

T_{Amb} is the actual ambient temperature in tenths of Kelvin.
$PTAT_i$ represents the *N* current PTAT-values, refer to Table2. *N* is device dependent.

Object Temperature:

In the followings dependency of the column is displayed as *variable-name (C).*
Dependency of row and column (pixel dependent) is displayed as *variable-name(C,R).*

The transmitted absolute pixel voltage consist of the electrical offset of the amplifier $V_{elOff}(C)$, the thermal offset of the pixel $V_{Th}(C,R)$ and the amplified pixel voltage $V_{Pix}(C,R)$:

$$V_{Abs}(C,R) = V_{Pix}(C,R) + V_{Th}(C,R) + V_{elOff}(C)$$

For the HTPA8x8 and the HTPA16x16 the corresponding electrical offset to a pixel voltage can be determined by the modulo-n-check. Divide the pixel number by the number of amplifiers (refer to the datasheet of the sensor), the residue is the corresponding electrical offset. For the HTPA32x31 the column address of the pixel is equal to the number of the electrical offset value.

If your device has a calibration setting of “IGNORE_ELOFF false” it is necessary to subtract the corresponding electrical offset of the amplifier $V_{elOff}(C)$ from the transmitted absolute Voltage of the pixel, else the electrical offset can be set to zero.

IGNORE_ELOFF false→ $V_{Abs}(C,R) - V_{elOff}(C) = V_{Pix}(C,R) + V_{Th}(C,R)$
IGNORE_ELOFF true→ $V_{Abs}(C,R) = V_{Pix}(C,R) + V_{Th}(C,R)$

The next step is to subtract the corresponding thermal offset to get the amplified pixel voltage:
IGNORE_ELOFF false→ $V_{Pix}(C,R) = V_{Abs}(C,R) - V_{elOff}(C) - V_{Th}(C,R)$
IGNORE_ELOFF true→ $V_{Pix}(C,R) = V_{Abs}(C,R) - V_{Th}(C,R)$

As described before, the value of $V_{Th}(C,R)$ should be determined by a linear interpolation between the two thermal offset values out of $Th_1(C,R), Th_2(C,R), Th_3(C,R), Th_4(C,R)$, which are as closest to the actual ambient temperature.

Next step is the linear interpolation to get the actual pixel constant $PixC(C,R)$, in dependency of the current ambient temperature T_{Amb}.

Now the object temperature T_O in tenths of Kelvin can be calculated by:

$$T_O = \sqrt[X]{\frac{V_{Pix}(C,R)\cdot PixC(C,R)\cdot VDM}{\varepsilon} + T_{Amb}^X} \qquad [\mathrm{dK}]$$

ε is the emission coefficient of the surface the pixel is receiving the radiation from.
VDM is a multiplier, which is equal to the VDM constant.
X is an exponent, which is equal to the EXP constant. VDM and EXP can be both obtained with the control character ‘M’.