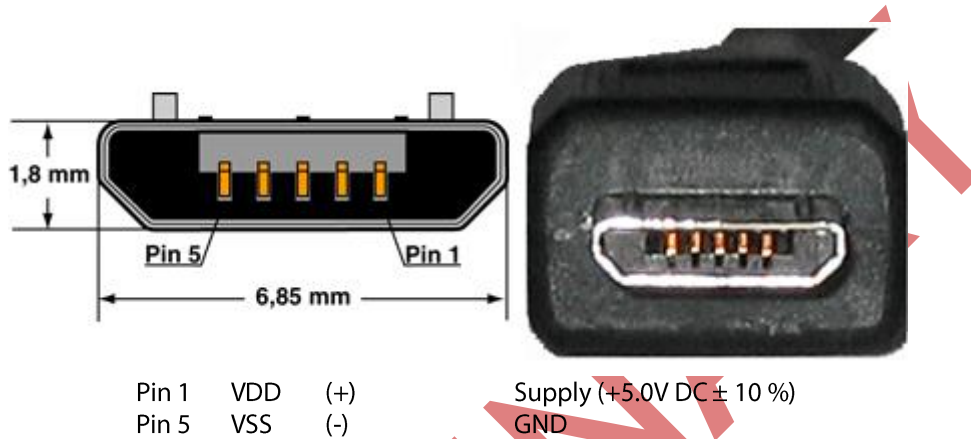


Electrical Specifications:

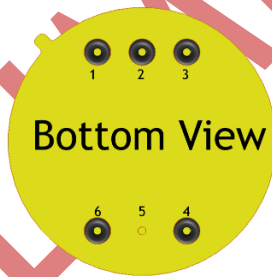
VDD:	Supply (+5.0 V DC via USB)
VSS:	GND
Power Supply:	5.0 V DC $\pm 10\%$
IDD (Operating mode):	180 mA

Power connection at Ethernet device:



Pinout

No.	Function	Type
1	MISO	Input/Output
2	MOSI	Input/Output
3	SCL	Input
4	VDD	Power
5	VSS	Power
6	EE_Enable	Input/Output

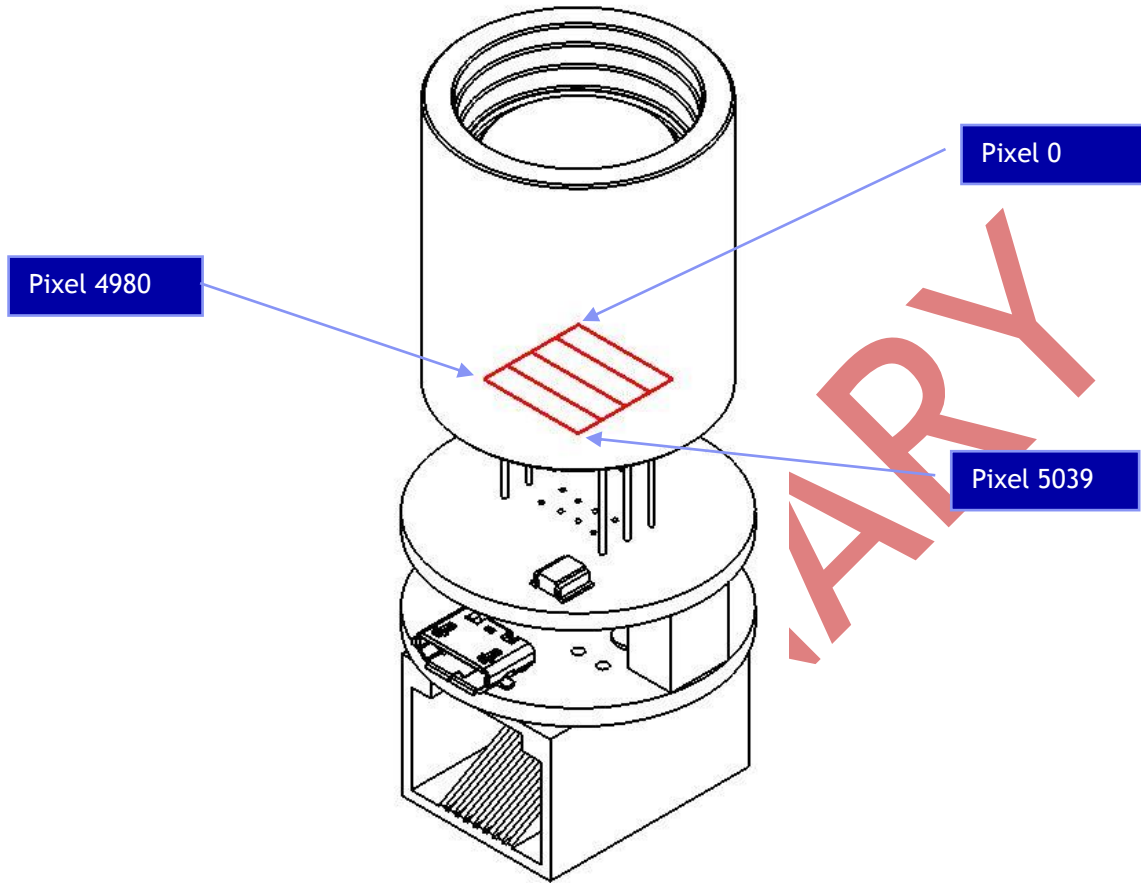


Ethernet-Interface:

Protocol Specifications:

Protocol type:	UDP
All communication on Port:	30444

HTPA120x84d UDP Module Optical Orientation of Pixels:

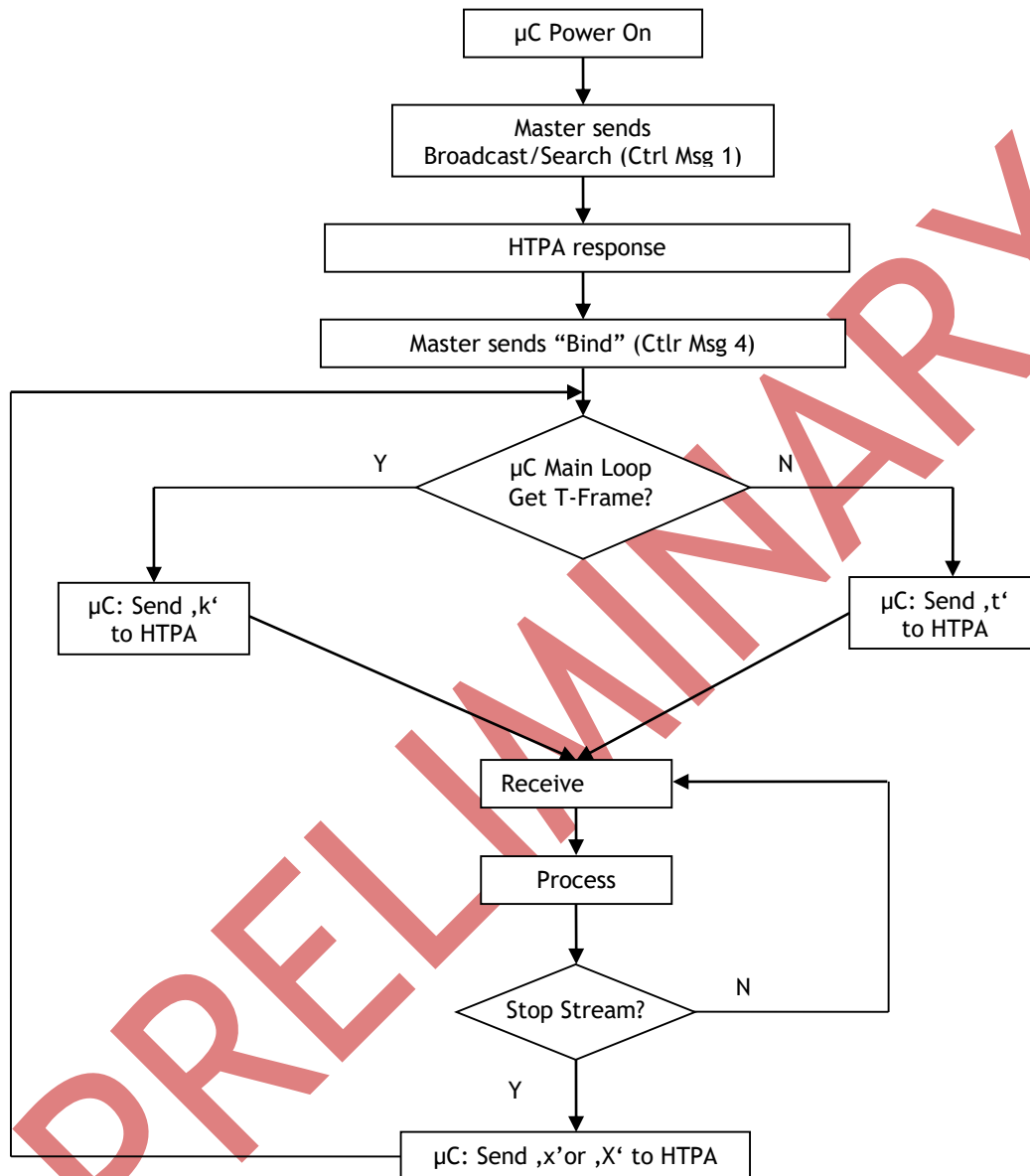


Pixelmap:

0	1	2	3	4	5	6	...	117	118	119
120	121	122	123	124	125	126	...	237	238	239
.										.
.										.
.										.
9960	9961	9962	9963	9964	9965	9966	...	10077	10078	10079

Communication and Timings:

Proposed flow chart of communication. (Master is referred as μC , Slave as HTPA module)



Communication:

<i>Communication via UDP</i>	
Sent Char	Result/Received message
'a'/'A'	decreases / increases CLK setting
'b'	Measure VDD (referenced to VREF1225)
'c'	Capture single voltage frame. Use ADC of μ C.
'f'	Toggle between sending raw and compensated voltages when sending 't'
'h'	pushes binary EEDATA out
'i'/'I'	decreases / increases BIAS setting
'j'/'J'	decreases / increases BPA setting
'G'	Shows current configuration settings (CLK, BIAS, BPA,...)
'k'	Read single temperature frame. Output in binary format.
'K'	send continous binary temperature datastream(μ C-ADC)[K*10] Output of a complete cycle <i>For a detailed Description of the serial order see Table2.</i>
'M'	Shows current and calibration settings. Device prints the following stream: "HTPA series responded! I am Arraytype 16 MODTYPE 5" "HTPA120x84d v.X.XX Heimann Sensor GmbH; written by M. Schnorr YYYY-MM-DD" Version information. "I am running on XXXX.X kHz" Actual MCLK-setting in kHz "MAC-ID: X IP: Y DevID: Z\r\n" X=MAC-ID of the device, i.e. "00.97.FF.00.10.08"; Y=current IP of the device, Z=DeviceID, range
'p'	toggle PU (pull-up) setting
'q'/'Q'	Allow Changes (required for Calibration)
'r'/'R'	decreases / increases resolution
't'	Continuous binary voltage data of the sensor is transmitted. Output of a complete cycle <i>For a detailed Description of the serial order see Table2.</i>
'v'	Announce IP (Only Ethernet devices)
'W'	Calibration. ATTENTION! Old Dataset cannot be restored!
'x'	Stops Stream without prompt.
'X'	Stops Stream by sending "STOP!\r\n"

Please be aware, that the source and destination port has to be 30444.

Serial order of data in stream:

HTPA120x84dR2 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
...	...
10079	Temperature of Pixel10079 in K*10
10080	el. Offset 0
10081	el. Offset 1
...	...
11759	el. Offset 1679
11760	VDD
11761	TAmb in K*10
11762	PTAT0
11763	PTAT1
11764	PTAT2
11765	PTAT3
11766	PTAT4
11767	PTAT5
11768	PTAT6
11769	PTAT7
11770	PTAT8
11771	PTAT9
11772	PTAT10
11773	PTAT11
11774	ATC0
11775	ATC1

HTPA120x84dR2 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
...	...
10079	absolute Voltage of Pixel10079 in digits
10080	el. Offset 0
10081	el. Offset 1
...	...
11759	el. Offset 1679
11760	VDD
11761	TAmb in K*10
11762	PTAT0
11763	PTAT1
11764	PTAT2
11765	PTAT3
11766	PTAT4
11767	PTAT5
11768	PTAT6
11769	PTAT7
11770	PTAT8
11771	PTAT9
11772	PTAT10
11773	PTAT11
11774	ATC0
11775	ATC1

Each dataset consists of a 16-bit value, first the Low-Byte is send, then the High-Byte.

Packets (UDP, only Ethernet device):

Number of packets	Packet size [byte]	HTPA type
1	262	HTPA8x8
1	780	HTPA16x16
2	1058+1054	HTPA32x31
2	1292+1288	HTPA32x32d
5	1159+1157	HTPA60x40d
9	1285+1281	HTPA84x60d
10	1283	HTPA80x64d
17	1401+1149	HTPA120x84dR0
17	1401+1153	HTPA120x84dR2
30	1401 + 1057	HTPA160x120d

Packet details for HTPA120x84dR2, with ATC, AT=15		
Packet No	Packet size	Packet contains
1	1401	Packet index 1 (8bit), data of Pixel0-Pixel699
2	1401	Packet index 2 (8bit), data of Pixel700-Pixel1399
3	1401	Packet index 3 (8bit), data of Pixel1400-Pixel2099
4	1401	Packet index 4 (8bit), data of Pixel2100-Pixel2799
5	1401	Packet index 5 (8bit), data of Pixel2800-Pixel3499
6	1401	Packet index 6 (8bit), data of Pixel3500-Pixel4199
7	1401	Packet index 7 (8bit), data of Pixel4200-Pixel4899
8	1401	Packet index 8 (8bit), data of Pixel4900-Pixel5599
9	1401	Packet index 9 (8bit), data of Pixel5600-Pixel6299
10	1401	Packet index 10 (8bit), data of Pixel6300-Pixel6999
11	1401	Packet index 11 (8bit), data of Pixel7000-Pixel7699
12	1401	Packet index 12 (8bit), data of Pixel7700-Pixel8399
13	1401	Packet index 13 (8bit), data of Pixel8400-Pixel9099
14	1401	Packet index 14 (8bit), data of Pixel9100-Pixel9799
15	1401	Packet index 15 (8bit), data of Pixel9800-el.Offset419
16	1401	Packet index 16 (8bit), data of el.Offset420-el.Offset1119
17	1153	Packet index 17 (8bit), data of el.Offset1120 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 32x16	"2"
HTPA 32x32d	"10"
HTPA 80x64d	"11"
HTPA 120x84d	"12"
HTPA 84x60d	"13"
HTPA 60x40d	"14"
HTPA 160x120d	"15"
HTPA 120x84dR2	"16"

[MCLK] insert Frequency of MCLK in ASCII format and kHz, i.e.: "1050.1"
[MSK] insert subnet mask in ASCII format, i.e.: "255.255.255.000"
[DEVID] insert 10 digit device ID in ASCII format, i.e. "0123456789" Range: 0000000000...4294967295
[MODT] insert index of Modulertype in ASCII format, i.e.: 005
[ADCRES] insert ADC resolution in ASCII format, i.e.: "16" Range: 08...16

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 responded! I am Arraytype [AT] MODTYPE [MODT]\r\n
ADC: [ADCRES]\r\n"
Firmware version, date and author information.
"I am running on [MCLK] kHz\r\n"
"MAC-ID: [MACID] IP: [IP] DevID: [DEVID]\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"

Control Messages [continued]:

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.	

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 HTPA32x31/32x32: 16384 byte, HTPA64x62: 65536 byte, HTPA80x64: 32768 byte.	
Answer:	"Write was successful.\n\r"	

Message6:	"Set Emission to [EPSILON]"	
Result:	The given emissivity [EPSILON] is written to the EEPROM. The emissivity can be used for customer specific purposes to compensate the radiation factor of different materials.	
Answer:	"Emission changed to [EPSILON]%\n\r"	

Temperature calculation:

The module is already transmitting calculated temperatures if character "k/K" was sent from the master. For details about the temperature calculation please see the datasheet of the sensor.

Module Dimensions:

PRELIMINARY