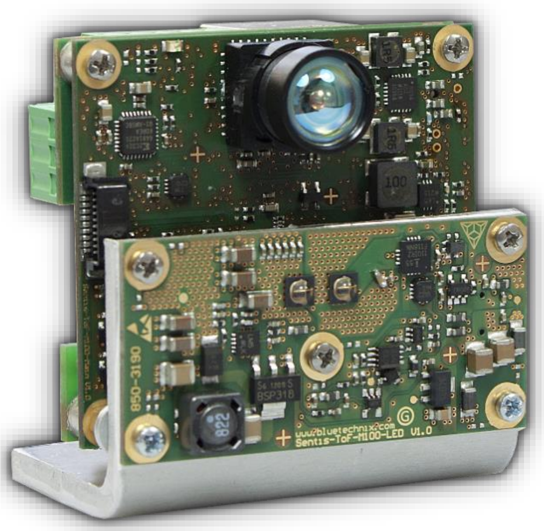
BLUETECHNIX
Embedding Ideas

Sentis-ToF-M100



User Manual

Version 0.4

Bluetechnix GmbH

Waidhausenstraße 3/19
A-1140 Vienna
AUSTRIA

office@bluetechnix.com
www.bluetechnix.com

Sentis-ToF-M100 – User Manual

Document No.: 900-308 / A

Publication date: September 12, 2013

Subject to change without notice. Errors excepted.

This document is protected by copyright. All rights reserved. No part of this document may be reproduced or transmitted for any purpose in any form or by any means, electronically or mechanically, without expressly written permission by Bluetechnix GmbH.

Windows is a registered trademark of Microsoft.

Table of Contents

© Bluetechnix 2013

© Bluetechnix 2013

© Bluetechnix 2013

© Bluetechnix GmbH 2013

All Rights Reserved.

The information herein is given to describe certain components and shall not be considered as a guarantee of characteristics.

Terms of delivery and rights of technical change reserved.

We hereby disclaim any warranties, including but not limited to warranties of non-infringement, regarding circuits, descriptions and charts stated herein.

Bluetechnix makes and you receive no warranties or conditions, express, implied, statutory or in any communication with you. Bluetechnix specifically disclaims any implied warranty of merchantability or fitness for a particular purpose.

Bluetechnix takes no liability for any damages and errors causing of the usage of this board. The user of this board is responsible by himself for the functionality of his application. He is allowed to use the board only if he has the qualification. More information is found in the General Terms and Conditions (AGB).

Information

For further information on technology, delivery terms and conditions and prices please contact Bluetechnix (http://www.bluetechnix.com).

Warning

Due to technical requirements components may contain dangerous substances.

1 General Information

This guide applies to the Sentis-ToF-M100 camera platform from Bluetechnix GmbH. Follow this guide chapter by chapter to set up and understand your product. If a section of this document only applies to certain camera parts, this is indicated at the beginning of the respective section.

The document applies to X-Grade product V0.0.0 with firmware version 0.1.0.

1.1 Symbols Used

This guide makes use of a few symbols and conventions:

Warning

Indicates a situation which, if not avoided, could result in minor or moderate injury and/or property damage or damage to the device.

Caution

Indicates a situation which, if not avoided, may result in minor damage to the device, in malfunction of the device or in data loss.

Note

Notes provide information on special issues related to the device or provide information that will make operation of the device easier.

Procedures

A procedure always starts with a headline

1. The number indicates the step number of a certain procedure you are expected to follow. Steps are numbered sequentially.

This sign ➤ indicates an expected result of your action.

References

This symbol indicates a cross reference to a different chapter of this manual or to an external document.

1.2 Certification

!

X-Grade Version

X-Grade version of the products are not intended for sale and have therefore no certifications. The user is responsible for a correct usage in order with federal laws.

2 Overview

2.1 Components

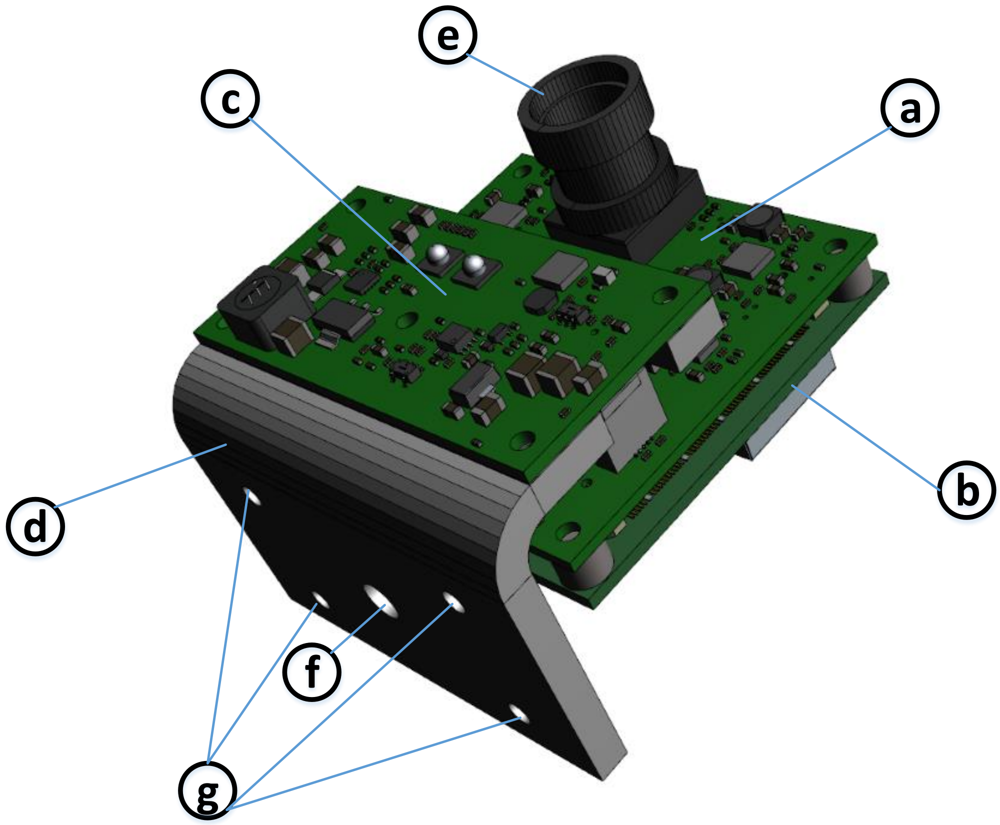


Figure 2-1: Sentis-ToF-M100 components

a. Main-Board
b. Interface-Board
c. LED-Board
d. Cooling Plate
e. Sensor Lens
f. Tripod Socket
g. Mounting Holes (Use M3 screws for mounting the device to an additional cooling plate)

2.2 Interfaces and Connectors

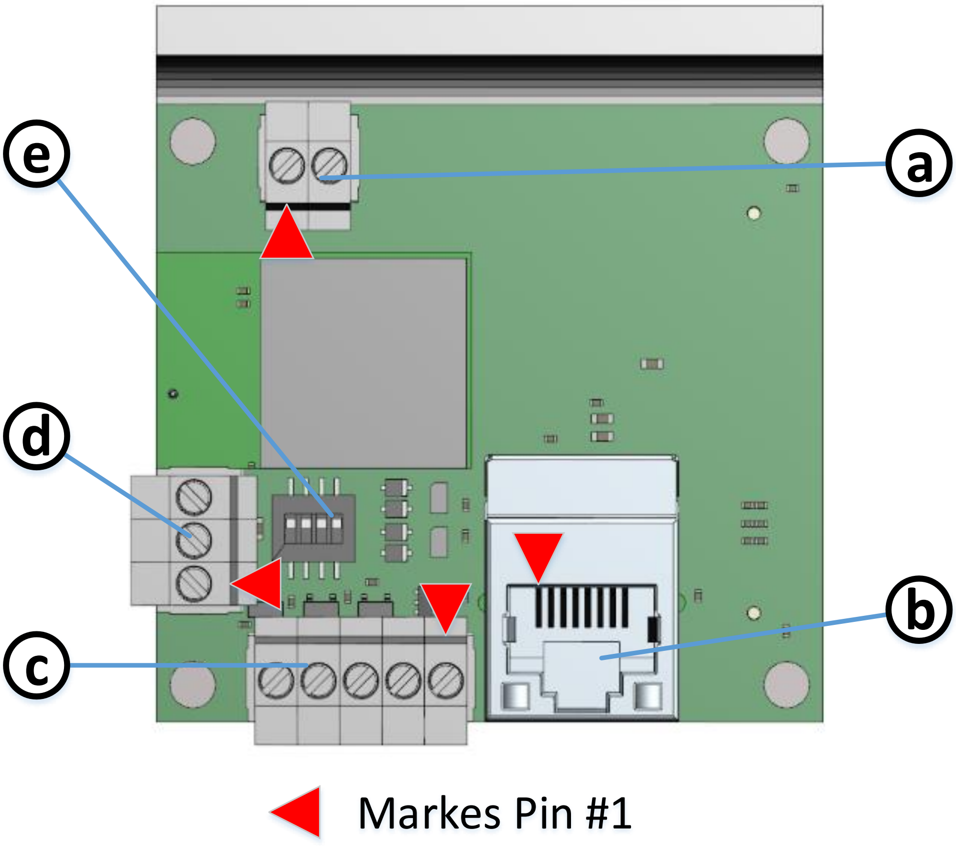


Figure 2-2: Sentis-ToF-M100 connectors and interfaces

a. Power Supply
b. Ethernet (RJ45) 10/100Base-T
c. GPIOs
d. RS232/485
e. Configuration DIP-Switch

3 Hardware Installation

3.1 Mounting

Caution

Cooling plate may become hot!

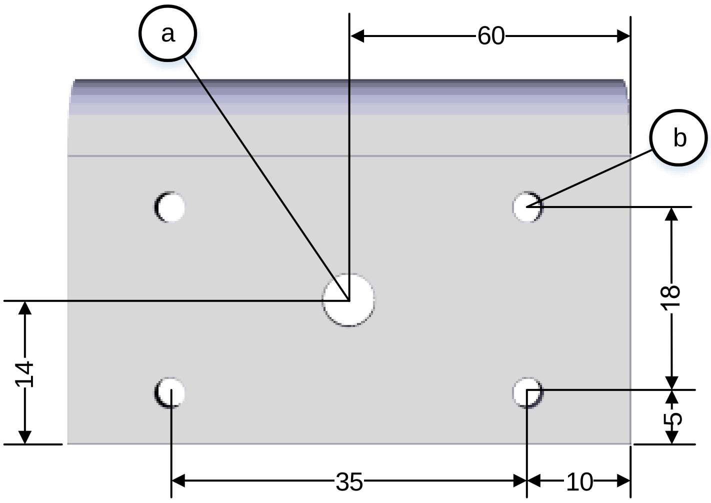


Figure 3-1: Mounting Holes for the Cooling Plate

3.1.1 Tripod Socket (a)

This is a ¼"-20 screw thread, which allows mounting the Sentis-ToF-M100 to any standard camera tripod.

3.1.2 Mounting Holes (b)

The cooling plate has four M3 screw threads that allows mounting the Sentis-ToF-M100 to a heat spreader.

3.1.3 Mount Spacing

If the Sentis^{ToF} - M100 is used without any additional heat sink attached to the cooling plate, the recommended minimum spacing between hardware and surrounding is 10mm in each direction.

Note

By mounting the camera onto a heat sink, it's allowed to decrease the recommended minimum spacing. In this case the customer is responsible for an adequate cooling.

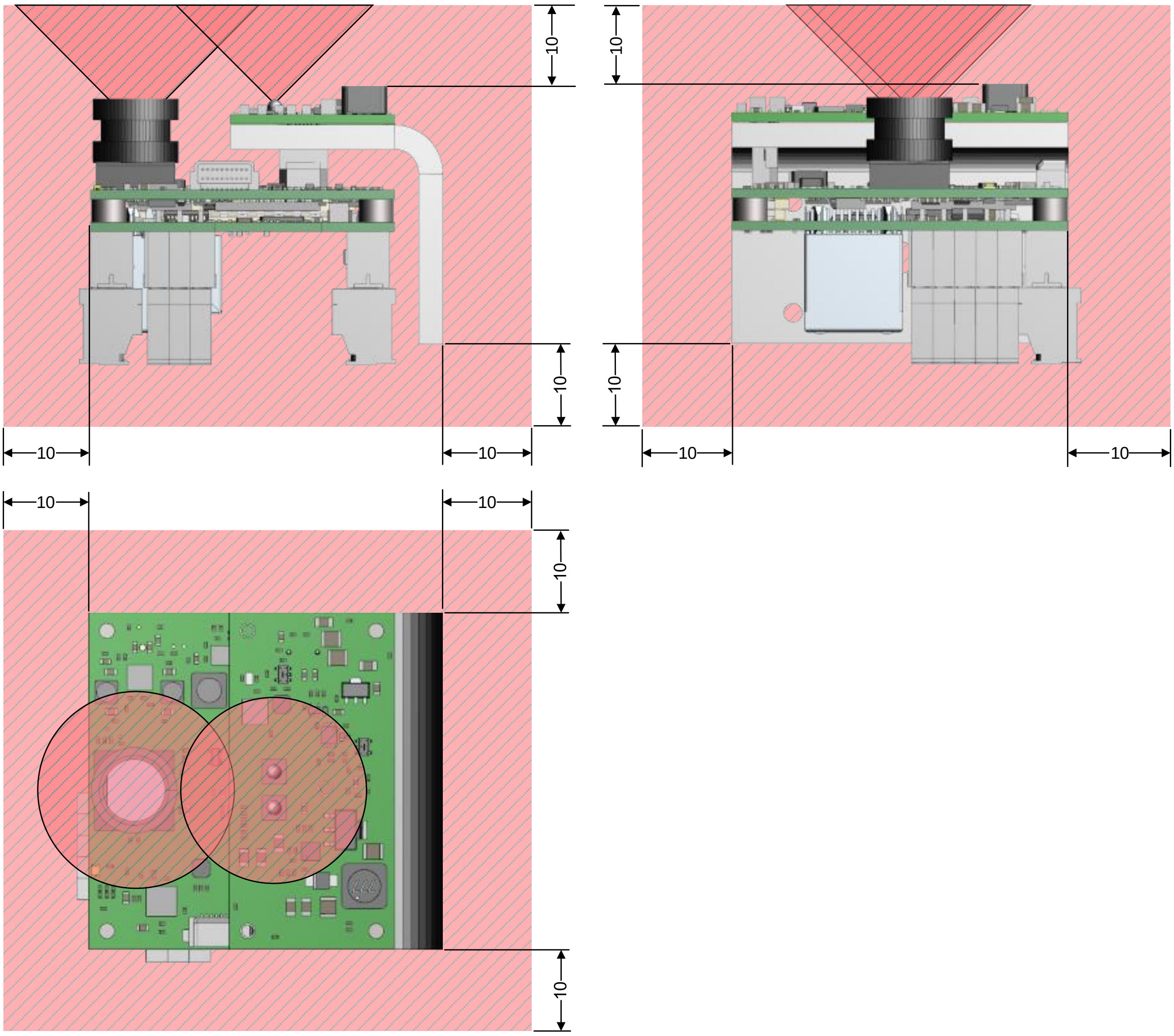


Figure 3-2: Recommended minimum spacing for air circulation

3.2 Processor cooling

In harsh environment or when the Sentis^{ToF} - M100 is used within a case without appropriate cooling it may be necessary to provide a head sink for the processor. Therefore the Interface-Board has a cut-out for gluing an appropriate heat sink onto the processor as shown in the Figure 3-3.

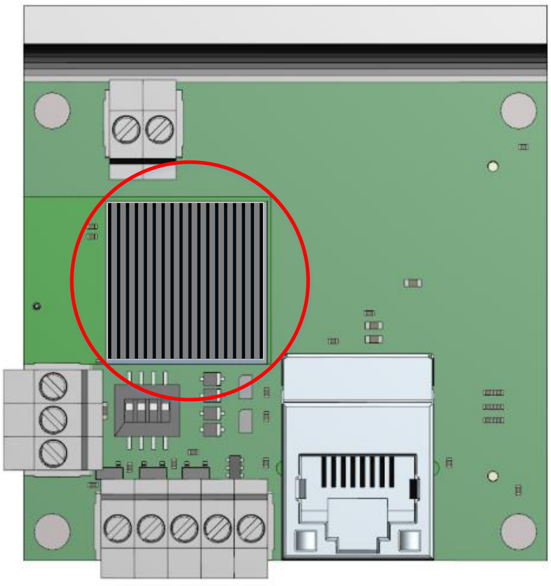


Figure 3-3: Processor with glued heat sink

3.3 Optical Isolation (a)

To prevent direct irradiation from the LED into the camera lens, an optical barrier has to be applied. The following pictures shows the maximum height of two types of such an isolation, dependent of the mounting position.

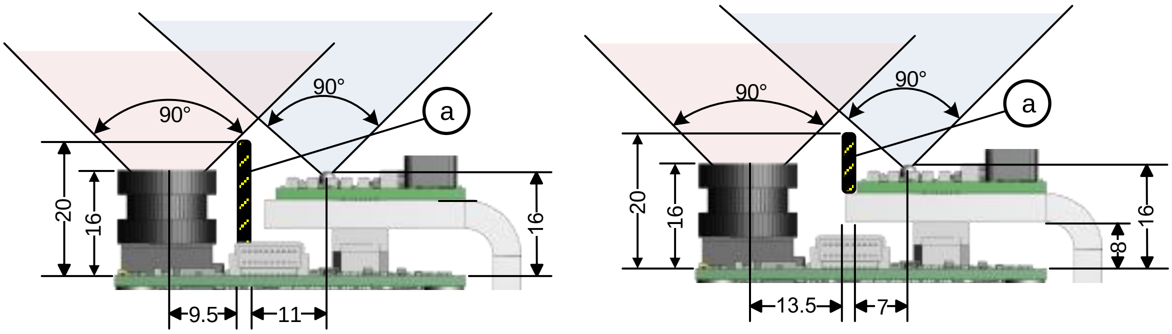


Figure 3-4: Optical Barrier

The isolation can be implemented as a straight barrier (Figure 3-4), or a surrounding rectangle barrier. Figure 3-5 shows the recommended dimension of such an optical isolator. On the PCB in the rectangle shown there are no components placed, therefore the barrier can go down to the PCB.

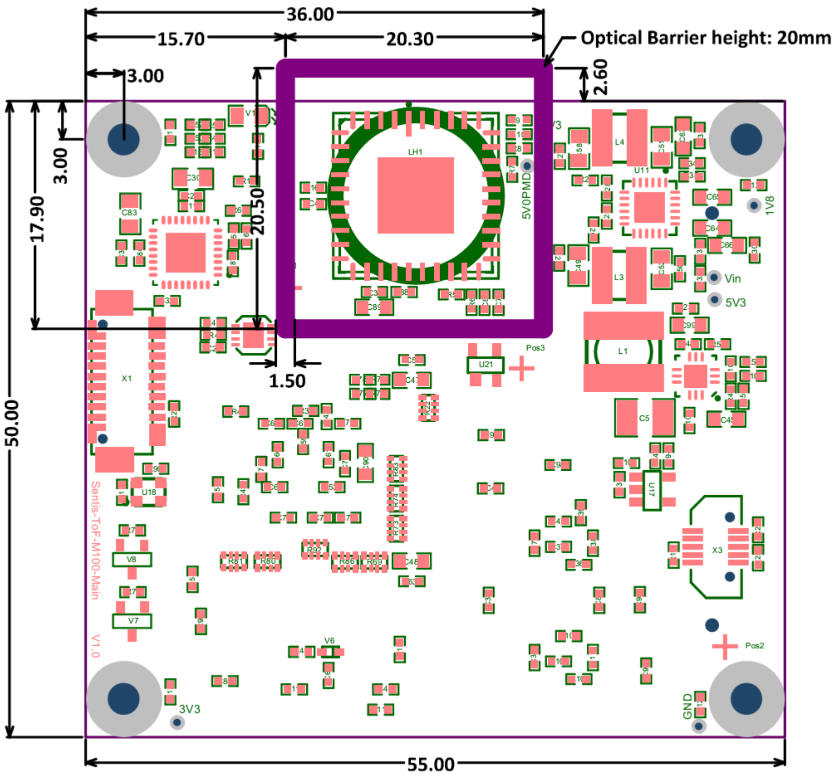


Figure 3-5: Mechanical dimensions for a sensor surrounding optical barrier

3.4 Lens and focus

The M12 lens is not glued onto the lens holder so you can use other lenses or change the focus. If you use other lenses they should be of type “fast lens”. Be aware that in that case you may have to recalibrate the sensor.

The sensor array is 7.2mm x 5.4mm.

3.5 External ToF-Flash

In case you want to use the external ToF-Flash or another external light source you can connect them to the Sentis-ToF-M100 by mounting an appropriate adapter board. Please refer to 4.4.1 to see how to mount the adapter board.

4 Board Description

4.1 Signal naming

Signal names are usually written in capital letters. They are noted in positive logic (positive asserted). If the signal is negative asserted an “n” will be added as prefix to the signal name.

Type:

The type describes the electrical characteristics of the signal. The following types are available:

- I Input
- O Output
- DN Negative Differential Output
- DP Positive Differential Output
- P Power supply
- 3.3V TTL TTL compatible signal with 3.3V high level and 0V low level
- 5V tolerant Accepts 5V input level

4.2 Connector Numbering

All pins no. 1 of each connector are marked in the figures with a red arrow. The connector numbering always starts at this pin, continuing in this row, and going backwards at the opposite side.

4.3 Main-Board

The Main-Board can be used in connection with a customized daughter-board. In that case you have the possibility to develop a customized board with the interfaces of your need. The SentisToF - M100 Main-Board in that case must be connected with the daughter-board using the 100pol board to board connector.

4.3.1 Interface-Board Connector

Mating connector: FX10A-100P/10-SV from Hirose

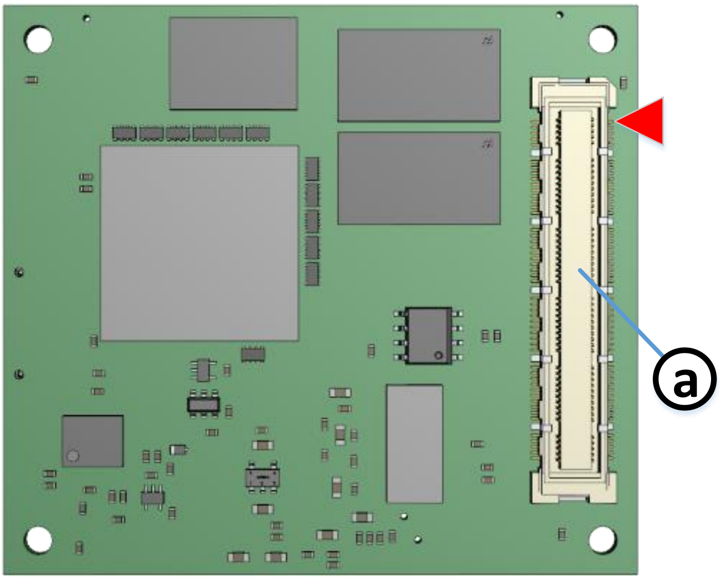


Figure 4-1: Main Board Interface Connector Location

Note

This mezzanine connectors are not reverse polarity protected.

No	Signal	Type	Description
1	GND	P	Power Ground
2	GND	P	Power Ground
3	GND	P	Power Ground
4	GND	P	Power Ground
5	GND	P	Power Ground
6	A1	O	Asynchronous Memory Address 1
7	A2	O	Asynchronous Memory Address 2
8	A3	O	Asynchronous Memory Address 3
9	A4	O	Asynchronous Memory Address 4
10	A5	O	Asynchronous Memory Address 5
11	A6	O	Asynchronous Memory Address 6
12	A7	O	Asynchronous Memory Address 7
13	A8	O	Asynchronous Memory Address 8
14	A9	O	Asynchronous Memory Address 9
15	A10	O	Asynchronous Memory Address 10
16	GND	P	Power Ground
17	nAMS1	O	Asynchronous Memory Select 1
18	nAMS2	O	Asynchronous Memory Select 2
19	GND	P	Power Ground
20	SPI.MOSI	O	Serial Peripheral Interface Data Output
21	SPI.MISO	I	Serial Peripheral Interface Data Input
22	SPI.SCLK	O	Serial Peripheral Interface Serial Clock
23	SPI.SS	O	Serial Peripheral Interface Slave Select
24	GND	P	Power Ground
25	I2C.A0	I	I²C Slave Address Selection 0

No	Signal	Type	Description
26	I2C.A1	I	I²C Slave Address Selection 1
27	Reserved	I	Do not connect
28	I2C.SDA	IO	I²C Data IO
29	I2C.SCL	I	I²C Clock Input
30	GND	P	Power Ground
31	PPI.CLK	I	Parallel Peripheral Interface Clock Input
32	PPI.SYNC3	IO	Parallel Peripheral Interface Frame Sync 3
33	PPI.SYNC1	IO	Parallel Peripheral Interface Frame Sync 1
34	PPI.SYNC2	IO	Parallel Peripheral Interface Frame Sync 2
35	PPI.D7	IO	Parallel Peripheral Interface Data 7
36	PPI.D6	IO	Parallel Peripheral Interface Data 6
37	PPI.D5	IO	Parallel Peripheral Interface Data 5
38	PPI.D4	IO	Parallel Peripheral Interface Data 4
39	PPI.D3	IO	Parallel Peripheral Interface Data 3
40	PPI.D2	IO	Parallel Peripheral Interface Data 2
41	PPI.D1	IO	Parallel Peripheral Interface Data 1
42	PPI.D0	IO	Parallel Peripheral Interface Data 0
43	IV2	I	Interface Board Version Control 2
44	IV1	I	Interface Board Version Control 1
45	IV0	I	Interface Board Version Control 0
46	3V3	P	3.3V Power output (max. 200mA)
47	3V3	P	3.3V Power output (max. 200mA)
48	GND	P	Power Ground
49	GND	P	Power Ground
50	GND	P	Power Ground
51	VIN	P	Main Supply Voltage (12V to 30V)
52	VIN	P	Main Supply Voltage (12V to 30V)
53	VIN	P	Main Supply Voltage (12V to 30V)
54	GND	P	Power Ground
55	GND	P	Power Ground
56	SSDI.MnS	I	Synchronous Serial Data Interface Master/Slave selection
57	SSDI.nSS	IO	Synchronous Serial Data Interface Slave Select
58	SSDI.DIN	I	Synchronous Serial Data Interface Data Input
59	SSDI.CLK	IO	Synchronous Serial Data Interface Clock
60	SSDI.DOUT0	O	Synchronous Serial Data Interface Data Output 0
61	SSDI.DOUT1	O	Synchronous Serial Data Interface data Output 1
62	CLKOUT	O	25MHz Clock Output
63	PEN	I	Power Enable Input
64	nRST_OUT	O	Reset Output
65	nRST_IN	I	Reset Input
66	PF40	IO	GPIO
67	PF39	IO	GPIO
68	PF38	IO	GPIO
69	PF37	IO	GPIO
70	PF36	IO	GPIO
71	PF11	IO	GPIO
72	UART.TX	O	UART transmit data
73	UART.RX	I	UART receive data
74	GND	P	Power Ground
75	nARE	O	Asynchronous Memory Read Enable

No	Signal	Type	Description
76	nAWE	O	Asynchronous Memory Write Enable
77	nAOE	O	Asynchronous Memory Output Enable
78	FLASH_WP	I	Flash Write Protection
79	GND	P	Power Ground
80	D15	IO	Asynchronous Memory Data 15
81	D14	IO	Asynchronous Memory Data 14
82	D13	IO	Asynchronous Memory Data 13
83	D12	IO	Asynchronous Memory Data 12
84	D11	IO	Asynchronous Memory Data 11
85	D10	IO	Asynchronous Memory Data 10
86	D9	IO	Asynchronous Memory Data 9
87	D8	IO	Asynchronous Memory Data 8
88	D7	IO	Asynchronous Memory Data 7
89	D6	IO	Asynchronous Memory Data 6
90	D5	IO	Asynchronous Memory Data 5
91	D4	IO	Asynchronous Memory Data 4
92	D3	IO	Asynchronous Memory Data 3
93	D2	IO	Asynchronous Memory Data 2
94	D1	IO	Asynchronous Memory Data 1
95	D0	IO	Asynchronous Memory Data 0
96	VLED	P	LED Supply Voltage (12V to 30V)
97	VLED	P	LED Supply Voltage (12V to 30V)
98	VLED	P	LED Supply Voltage (12V to 30V)
99	VLED	P	LED Supply Voltage (12V to 30V)
100	VLED	P	LED Supply Voltage (12V to 30V)

Table 4-1: 100-pole Main Board Connector Description

4.3.2 Debug Interface

The debug interface is only needed if you want to develop your own application running on the Sentis^{ToF}-M100. Please refer to 7.3.

The mating connector types are: IL-WX-20PB-VF-BE (straight)
IL-WX-20PB-HF-HD-S-BE (right angle)

!

Note

This connector is not reverse polarity protected.

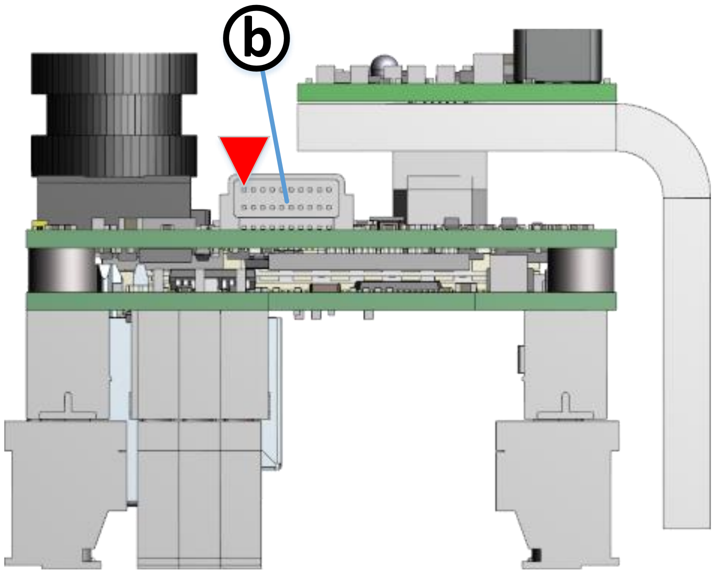


Figure 4-2: Debug Connector Location

No	Signal	Type	Description
1	SSDI.TCLK	IO	Synchronous Serial Data Interface Clock
2	SSDI.DTPRI	O	Synchronous Serial Data Interface Clock
3	SSDI.DRPRI	I	Synchronous Serial Data Interface Clock
4	SSDI.TFS	IO	Synchronous Serial Data Interface Frame Synchronization
5	JTAG.EMU	O	Blackfin JTAG Interface
6	JTAG.TMS	I	Blackfin JTAG Interface
7	JTAG.TCK	I	Blackfin JTAG Interface
8	JTAG.TRST	I	Blackfin JTAG Interface
9	JTAG.TDI	I	Blackfin JTAG Interface
10	JTAG.TDO	O	Blackfin JTAG Interface
11	3V3	P	3.3V Power Output (max. 50mA)
12	GND	P	Power Ground
13	Reserved		
14	Reserved		
15	GND	P	Power Ground
16	Reserved		
17	Reserved		
18	PF38	IO	Blackfin GPIO
19	PF39	IO	Blackfin GPIO
20	GND	P	Power Ground

Table 4-2: Debug Connector Interface Description

4.3.3 LED-Board Interface

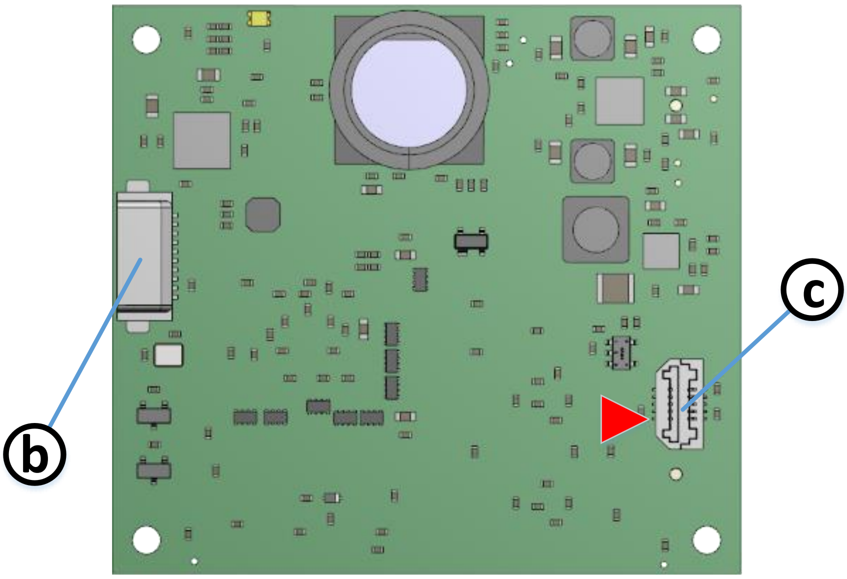


Figure 4-3: LED-Board Interface Connector Location

No	Signal	Type	Description
1	I2C.SDA	IO	I²C Data IO
2	I2C.SCL	O	I²C Clock Output
3	LED.EN0	O	LED Enable
4	LED.MOD_N	DN O	Negative LVDS Modulation Signal
5	LED.MOD_P	DP O	Positive LVDS Modulation Signal
6	VLED	P	LED Voltage Supply
7	VLED	P	LED Voltage Supply
8	GND	P	Power Ground
9	GND	P	Power Ground
10	3V3	P	3.3V Supply (max. 50mA)

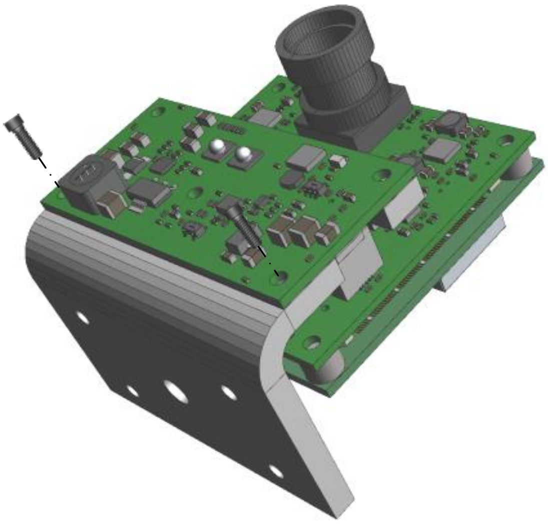
Table 4-3: LED-Board Interface Description

4.4 ToF-Flash Adapter

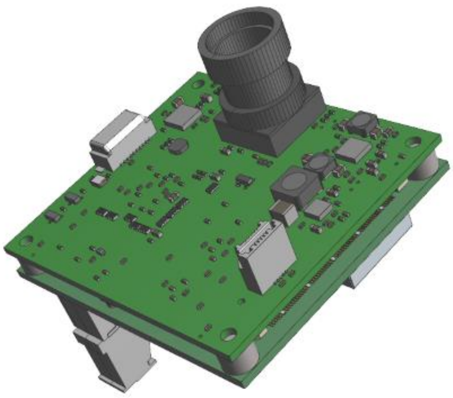
4.4.1 Adapter Assembling

In case it would be necessary to use an external illumination to extend the range of the Sentis^{ToF} - M100 you can use the modulation signal interface to synchronize this light source with the Sentis^{ToF} - M100. Perform the following steps to modify the Sentis^{ToF} - M100 for external illuminations:

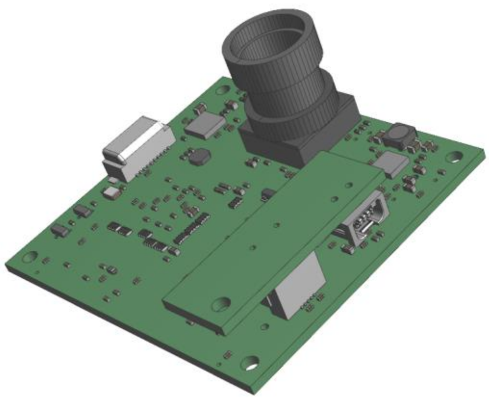
1. Remove the two mounting screws at the board edge as shown in the picture below



2. Remove the LED-Board with the Cooling Plate



3. Attach the modulation light interface adapter



4. Screw the adapter with one of the LED-Board screws

4.4.2 Modulation Signal Interface

Once the adapter is mounted you can connect an external illumination module using the “Modulation Signal Interface” connector (a) on the Adapter.

Mating Connector Type: **MQ172X-4SA-CV**

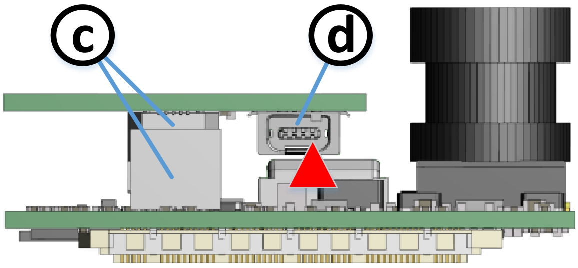


Figure 4-4: Modulation Light Connector Location

The Modulation Light Interface provides the modulation signal for an external illumination module (differential LVDS).

Caution

Overvoltage on the Modulation Light Interface will destroy the Sentis-ToF-M100.

Pin No.	Signal Name	Type	Description
1	DATA_EX[1)]	IO (3.3V TTL)	Data exchange signal
2	EXT.MOD_N	DN	Modulation signal output-
3	EXT.MOD_P	DP	Modulation signal output+
4	GND	P	Reference ground

Table 4-4: Modulation Signal Interface

1) Note
The usage of this pin may depend on the firmware version.

4.5 Interface-Board

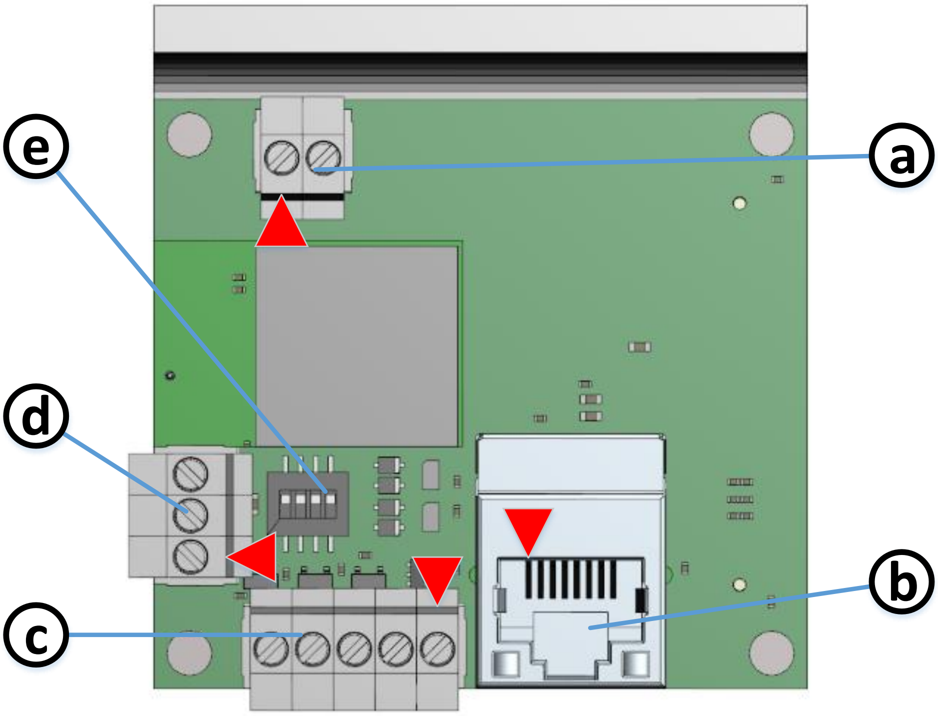


Figure 4-5: Interface Board Connector Location

4.5.1 Main-Board Connector

The Main-Board Connector is described in chapter 4.3.1.

4.5.2 Power Connector (a)

This 3.5mm terminal connector allows plugging a cable entry plug like: **691361100002** from Würth Elektronik. Compatible connectors from other manufacturers may be found as well.

Pin #1 is the positive supply voltage, pin #2 is power ground. This pins are protected against wrong polarity.

Voltage range: 18V to 30V.

4.5.3 Ethernet (b)

This is a standard straight RJ45 10/100 Base-T compatible Ethernet connector.

4.5.4 GPIO (c)

This 5 pole 3.5mm terminal connector allows plugging a cable entry plug like: **691361100005** from Würth Elektronik.

No.	BF561 Signal	Type	Description
1	GND	P	Signal Ground

© Bluetechnix 2013

No.	BF561 Signal	Type	Description
2	PF38	I	General Purpose Input
3	PF37 PF47	I O (open drain)	General Purpose Input and Output
4	PF36 PF40	I O (open drain)	General Purpose Input and Output
5	PF11 PF39	I O (open drain)	General Purpose Input and Output

Table 4-5: GPIO Connector Description

For the software usage of this GPIOs refer to chapter 5.3.

This GPIO pins have 30V tolerant inputs. If they are used as outputs, they must be externally pulled up to the used IO-voltage.

4.5.5 RS232/RS485 (d)

This 3 pole 3.5mm terminal connector allows plugging a cable entry plug like: **691361100003** from Würth Elektronik.

No.	Signal	Type	Description
1	GND	P	Signal Ground
2	RS232 RxD1) RS485 A/Y	IO DN	RS232 Receive Data RS485 Negative Differential Data
3	RS232 TxD1) RS485 B/Z	IO DP	RS232 Transmit Data RS485 Positive Differential Data

Table 4-6: GPIO Connector Description

! **1) Note**
The Interface-Mode can be selected with the DIP-Switch 1 (see table below).

Both interfaces are running in half duplex mode only.

4.5.6 DIP-Switch (e)

The DIP-Switch allows configuring the RS232/RS485 transceiver. The following table shows the functionality of each switch.

No.	Name	Description
1	RS485 Enable	ON: Transceiver works in RS485 mode OFF: Transceiver works in RS232 mode
2	RS485 High Slew-rate activation	ON: high slew rate active (20Mbps) OFF: high slew rate inactive (0.46 Mbps)
3	NC	Not Used
4	RS485 Termination	ON: Enables the 120Ω RS485 termination resistor OFF: No termination resistor is active

Table 4-7: DIP-Switch Description

!

Note
Make sure that the termination resistor is always disabled, if the driver runs in RS232 mode.

5 Interfacing

The Sentis[ToF] - M100 provides two types of interfaces. Ethernet over the Interface Board or I²C/SSDI available on the 100pol board to board connector of the Sentis Main-Board. Which interface will be supported depends on the firmware. The default firmware supports the Ethernet interface only. Please contact support@bluetechnix.com for information about I²C/SSDI support.

The interfaces are split into a control and data interface. The control interface is used to set and read the configuration of the Sentis over a set of registers. Refer to 8.6 for a detailed register description.

The data interface provides a continuous stream of the distance and amplitude values or the XYZ data depending on the configuration.

5.1 Control Interface

As control interface either the I²C or the Ethernet interface can be used. The I²C interface is only available on the 100 pol. board to board connector of the Main-Board.

5.1.1 Ethernet

The Sentis[ToF] - M100 can be configured using a TCP/IP connection. For the control interface the Sentis is listening to the following factory default IP settings:

- **IP-Address**: 192.168.0.10
- **Subnet mask**: 255.255.255.0
- **Network protocol**: TCP
- **TCP port**: 10001

! **Note**

The Ethernet IP settings can be configured using the *Eth0_* registers. The changes become active on the next reconnect or reboot.

Once a TCP connection has been established the Sentis can be configured using a dedicated set of command frames. The Sentis answers to each command frame with a dedicated response frame. The following table shows the currently supported command frames:

Command frame	Description
Register Read	Used to read one or more consecutive registers
Register Write	Used to write one or more consecutive registers
Reset	Used to reset/reboot the Sentis-ToF-M100
Flash Update	Used to either update the firmware or the bootloader or to flash a file to a specific location
Alive	Used to keep alive the control interface connection. It has to be send if no other command frame will be sent within two second and the connection should be still open. If no command will be send for a certain amount of time the Sentis closes the control interface connection and waits for a new incoming connection request.

Table 5-1: Supported command frames

The following section describes each command frame and the expected answer in detail. To be able to communicate with the Sentis-ToF-M100 the frame must be composed exactly as described.

The following types are used:

- **Uint8**: 8 bit unsigned integer
- **Uint16**: 16 bit unsigned integer
- **Uint32**: 32 bit unsigned integer

! **Note**
Values with ‘0x’ as prefix are hexadecimal values.

5.1.1.1 Register read

Command frame

Addr	Field	Type	Value	Description
0x00	Preamble	Uint16 (highbyte first)	0xA1EC	Unique identifier, start of header
0x02	ProtocolVersion	Uint8	0x03	This description refers to protocol version V3.0
0x03	Command	Uint8	0x03	Command code for read registers
0x04	SubCommand	Uint8		Ignored
0x05	Status	Uint8		Ignored
0x06	Flags	Uint16	Refer to table	Optional flags
0x08	Length	Uint32 (highbyte first)	<# of bytes to read>	Number of bytes to read. Must be a multiple of two. The length divided by two represents the # of registers to read.
0x0C	RegisterAddress	Uint16 (highbyte first)	<Register Address>	Start register address for read command
0x0E	HeaderData2	Uint8		Ignored
0x0F	HeaderData3	Uint8		Ignored
0x10	Reserved (42 bytes)	Uint8[]		Ignored
0x3A	DataCrc32	Uint32 (highbyte first)	0x0	No <Data> present 2)
0x3E	HeaderCrc16	Uint16 (highbyte first)	<CRC16 checksum>	Checksum over 60 bytes of Header: 0x02 – 0x3D 1)

Table 5-2: Register read command frame

Note 1): For the CRC16 calculation the CRC-CCITT is used (Polynom: 0x1021, start value: 0). Please ask the Bluetechnix support for an implementation example of the CRC-CCITT.

Note 2): For the CRC16 calculation the CRC-32 is used (Polynom: 0x04C11DB7, start value: 0xFFFFFFFF). Please ask the Bluetechnix support for an implementation example of the CRC-32.

Response frame

Addr	Field	Type	Value	Description

Addr	Field	Type	Value	Description
0x00	Preamble	Uint16 (highbyte first)	0xA1EC	Unique identifier, start of header
0x02	ProtocolVersion	Uint8	0x03	This document refers to version V3.0
0x03	Command	Uint8	0x03	Command code for read registers
0x04	SubCommand	Uint8		Ignore
0x05	Status	Uint8	Refer to table	Result code
0x06	Flags	Uint16	Refer to table	Optional flags
0x08	Length	Uint32 (highbyte first)	<# of bytes read>	The number of bytes read (length of <Data> in bytes). The length divided by two represents the # of registers read.
0x0C	RegisterAddress	Uint16 (highbyte first)	<Register Address>	Start register address of read data
0x0E	HeaderData2	Uint8		Ignored
0x0F	HeaderData3	Uint8		Ignored
0x10	Reserved (42 bytes)	Uint8[]		Ignored
0x3A	DataCrc32	Uint32 (highbyte first)	<CRC32 checksum>	Checksum over <Data> [2)]
0x3E	HeaderCrc16	Uint16 (highbyte first)	<CRC16 checksum>	Checksum over 60 bytes of Header: 0x02 – 0x3D [1)]
0x40	Data	byte[] (highbyte first)	<result data>	Result: One or more 16 bit values

Table 5-3: Register read response frame

Note 1): For the CRC16 calculation the CRC-CCITT is used (Polynom: 0x1021, start value: 0). Please ask the Bluetechnix support for an implementation example of the CRC-CCITT.

Note 2): For the CRC16 calculation the CRC-32 is used (Polynom: 0x04C11DB7, start value: 0xFFFFFFFF). Please ask the Bluetechnix support for an implementation example of the CRC-32.

Flags

Flags	Description
Bit 0	1: Ignore DataCrc32

Table 5-4: Register read flag description

Result codes

Please refer to Table 5-18.

5.1.1.2 Register write

Command frame

Addr	Field	Type	Value	Description
0x00	Preamble	Uint16 (highbyte first)	0xA1EC	Unique identifier, start of header
0x02	ProtocolVersion	Uint8	0x03	This document refers to version V3.0
0x03	Command	Uint8	0x04	Command code for write registers
0x04	SubCommand	Uint8		Ignored
0x05	Status	Uint8		Ignored
0x06	Flags	Uint16	Refer to table	Optional flags
0x08	Length	Uint32	<# of bytes	The number of bytes to write. Must be a multiple

Addr	Field	Type	Value	Description
		(highbyte first)	to write>	of two and match length of <Data> in bytes. The length divided by two represents the # of registers to write.
0x0C	RegisterAddress	Uint16 (highbyte first)	<Register Address>	Start register address for write command
0x0E	HeaderData2	Uint8		Ignored
0x0F	HeaderData3	Uint8		Ignored
0x10	Reserved (42 bytes)	Uint8[]		Ignored
0x3A	DataCrc32	Uint32 (highbyte first)	<CRC32 checksum>	Checksum over <Data> 2)
0x3E	HeaderCrc16	Uint16 (highbyte first)	<CRC16 checksum>	Checksum over 60 bytes of Header: 0x02 – 0x3D 1)
0x40	Data (highbyte first for each register value)	byte[]	<data to write>	One or more 16 bit values in a stream that should be written

Table 5-5: Register write command frame

Note 1): For the CRC16 calculation the CRC-CCITT is used (Polynom: 0x1021, start value: 0). Please ask the Bluetechnix support for an implementation example of the CRC-CCITT.

Note 2): For the CRC16 calculation the CRC-32 is used (Polynom: 0x04C11DB7, start value: 0xFFFFFFFF). Please ask the Bluetechnix support for an implementation example of the CRC-32.

Response frame

Addr	Field	Type	Value	Description
0x00	Preamble	Uint16 (highbyte first)	0xA1EC	Unique identifier, start of header
0x02	ProtocolVersion	Uint8	0x03	This document refers to version V3.0
0x03	Command	Uint8	0x04	Command code for write registers
0x04	SubCommand	Uint8		Ignored
0x05	Status	Uint8	Refer to table	Result code
0x06	Flags	Uint16	Refer to table	Optional flags
0x08	Length	Uint32 (highbyte first)	0	No <Data> present
0x0C	RegisterAddress	Uint8 (highbyte first)	<Register Address>	Same as in sent command
0x0E	HeaderData2	Uint8		Ignored
0x0F	HeaderData3	Uint8		Ignored
0x10	Reserved (42 bytes)	Uint8[]		Ignored
0x3A	DataCrc32	Uint32 (highbyte first)	0x0	No <Data> present 2)
0x3E	HeaderCrc16	Uint16 (highbyte first)	<CRC16 checksum>	Checksum over 60 bytes of Header: 0x02 – 0x3D 1)

Table 5-6: Register write response frame

Note 1): For the CRC16 calculation the CRC-CCITT is used (Polynom: 0x1021, start value: 0). Please ask the Bluetechnix support for an implementation example of the CRC-CCITT.

Note 2): For the CRC16 calculation the CRC-32 is used (Polynom: 0x04C11DB7, start value: Polynom: 0xFFFFFFFF). Please ask the Bluetechnix support for an implementation example of the CRC-32.

Flags

Flags	Description
Bit 0	1: Ignore DataCrc32

Table 5-7: Register write flag description

Result codes

Please refer to Table 5-18.

5.1.1.3 Reset

Command frame

Addr	Field	Type	Value	Description
0x00	Preamble	Uint16 (highbyte first)	0xA1EC	Unique identifier, start of header
0x02	ProtocolVersion	Uint8	0x03	This document refers to version V3.0
0x03	Command	Uint8	0x07	Command code for reset
0x04	SubCommand	Uint8		Ignored
0x05	Status	Uint8		Ignored
0x06	Flags	Uint16	Refer to table	Optional flags
0x08	Length	Uint32 (highbyte first)	0x0	No <Data> present
0x0C	HeaderData0	Uint8		Ignored
0x0D	HeaderData1	Uint8		Ignored
0x0E	HeaderData2	Uint8		Ignored
0x0F	HeaderData3	Uint8		Ignored
0x10	Reserved (42 bytes)	Uint8[]		Ignored
0x3A	DataCrc32	Uint32 (highbyte first)	0x0	No <Data> present [2)]
0x3E	HeaderCrc16	Uint16 (highbyte first)	<CRC16 checksum>	Checksum over 60 bytes of Header: 0x02 – 0x3D [1)]

Table 5-8: Reset command frame

Note 1): For the CRC16 calculation the CRC-CCITT is used (Polynom: 0x1021, start value: 0). Please ask the Bluetechnix support for an implementation example of the CRC-CCITT.

Note 2): For the CRC16 calculation the CRC-32 is used (Polynom: 0x04C11DB7, start value: 0xFFFFFFFF). Please ask the Bluetechnix support for an implementation example of the CRC-32.

Response frame

Addr	Field	Type	Value	Description
0x00	Preamble	Uint16 (highbyte first)	0xA1EC	Unique identifier, start of header
0x02	ProtocolVersion	Uint8	0x03	This document refers to version V3.0
0x03	Command	Uint8	0x07	Command code for reset
0x04	SubCommand	Uint8		Ignored
0x05	Status	Uint8	Refer to table	Result code
0x06	Flags	Uint16	Refer to table	Optional flags
0x08	Length	Uint32 (highbyte first)	0x0	No <Data> present

Addr	Field	Type	Value	Description
0x0C	HeaderData0	Uint8		Ignored
0x0C	HeaderData1	Uint8		Ignored
0x0E	HeaderData2	Uint8		Ignored
0x0F	HeaderData3	Uint8		Ignored
0x10	Reserved (42 bytes)	Uint8[]		Ignored
0x3A	DataCrc32	Uint32 (highbyte first)	0x0	No <Data> present [2)]
0x3E	HeaderCrc16	Uint16 (highbyte first)	<CRC16 checksum>	Checksum over 60 bytes of Header: 0x02 – 0x3D [1)]

Table 5-9: Reset response frame

Note 1): For the CRC16 calculation the CRC-CCITT is used (Polynom: 0x1021, start value: 0). Please ask the Bluetechnix support for an implementation example of the CRC-CCITT.

Note 2): For the CRC16 calculation the CRC-32 is used (Polynom: 0x04C11DB7, start value: 0xFFFFFFFF). Please ask the Bluetechnix support for an implementation example of the CRC-32.

Flags

Flags	Description
	Currently no flags defined for this command

Table 5-10: Reset flag description

Result codes

Please refer to Table 5-18.

5.1.1.4 Flash Update

Command frame

Addr	Field	Type	Value	Description
0x00	Preamble	Uint16 (highbyte first)	0xA1EC	Unique identifier, start of header
0x02	ProtocolVersion	Uint8	0x03	This document refers to version V3.0
0x03	Command	Uint8	0x0B, 0x0C or 0x0D	0x0B: Flash Bootloader 0x0C: Flash Application 0x0D: Flash generic file (or OTP)
0x04	SubCommand	Uint8	Refer to table	Indicates which flash to write to
0x05	Status	Uint8		Ignored
0x06	Flags	Uint16	Refer to table	Optional flags (remember: DataCrc32 is mandatory, the appropriate flag has been set to 0)
0x08	Length	Uint32 (highbyte first)	<# of bytes to write>	The size of the binary file to flash
0x0C	FlashAddress	Uint32 (highbyte first)	<Flash Address>	A generic file is flashed to this address. When Flashing a Bootloader or application it is ignored.
0x10	Reserved (42 bytes)	Uint8[]		Ignored
0x3A	DataCrc32	Uint32 (highbyte first)	<CRC32 checksum>	Checksum over <Data> [2)]

Addr	Field	Type	Value	Description
0x3E	HeaderCrc16	Uint16 (highbyte first)	<CRC16 checksum>	Checksum over 60 bytes of Header: 0x02 – 0x3D [1)]
0x40	Data	byte[]	<binary loader file>	The file to flash as a binary byte stream

Table 5-11: Flash update command frame

Note 1): For the CRC16 calculation the CRC-CCITT is used (Polynom: 0x1021, start value: 0). Please ask the Bluetechnix support for an implementation example of the CRC-CCITT.

Note 2): For the CRC16 calculation the CRC-32 is used (Polynom: 0x04C11DB7, start value: 0xFFFFFFFF). Please ask the Bluetechnix support for an implementation example of the CRC-32.

<u>Response frame</u>

Addr	Field	Type	Value	Description
0x00	Preamble	Uint16 (highbyte first)	0xA1EC	Unique identifier, start of header
0x02	ProtocolVersion	Uint8	0x03	This document refers to version V3.0
0x03	Command	Uint8	0x0B, 0x0C or 0x0D	0x0B: Flash Bootloader 0x0C: Flash Application 0x0D: Flash generic file (or OTP)
0x04	SubCommand	Uint8	Refer to table	Indicates which flash to write to
0x05	Status	Uint8	Refer to table	Result code
0x06	Flags	Uint16	Refer to table	Optional flags
0x08	Length	Uint32 (highbyte first)	0x0	No <Data> present
0x0C	HeaderData0	Uint8		Ignored
0x0D	HeaderData1	Uint8		Ignored
0x0E	HeaderData2	Uint8		Ignored
0x0F	HeaderData3	Uint8		Ignored
0x10	Reserved (42 bytes)	Uint8[]		Ignored
0x3A	DataCrc32	Uint32 (highbyte first)	0x0	No <Data> present [2)]
0x3E	HeaderCrc16	Uint16 (highbyte first)	<CRC16 checksum>	Checksum over 60 bytes of Header: 0x02 – 0x3D [1)]

Table 5-12: Flash update response frame

Note 1): For the CRC16 calculation the CRC-CCITT is used (Polynom: 0x1021, start value: 0). Please ask the Bluetechnix support for an implementation example of the CRC-CCITT.

Note 2): For the CRC16 calculation the CRC-32 is used (Polynom: 0x04C11DB7, start value: 0xFFFFFFFF). Please ask the Bluetechnix support for an implementation example of the CRC-32.

<u>Subcommand</u>

SubCommand	Description
0	SPI-flash
1	Parallel flash
2	OTP

Table 5-13: Flash update subcommand description

Flags

Flags	Description
Bit 0	1: Ignore DataCrc32

Table 5-14: Flash update flag description

Result codes

Please refer to Table 5-18.

5.1.1.5 Alive

Command frame

Addr	Field	Type	Value	Description
0x00	Preamble	Uint16 (highbyte first)	0xA1EC	Unique identifier, start of header
0x02	ProtocolVersion	Uint8	0x03	This document refers to version V3.0
0x03	Command	Uint8	0xFE	Command code for 'Alive message
0x04	SubCommand	Uint8		Ignored
0x05	Status	Uint8		Ignored
0x06	Flags	Uint16	Refer to table	Optional flags
0x08	Length	Uint32	0x0	No <Data> present
0x0C	HeaderData0	Uint8		Ignored
0x0D	HeaderData1	Uint8		Ignored
0x0E	HeaderData2	Uint8		Ignored
0x0F	HeaderData3	Uint8		Ignored
0x10	Reserved (42 bytes)	Uint8[]		Ignored
0x3A	DataCrc32	Uint32 (highbyte first)	0x0	No <Data> present 2)
0x3E	HeaderCrc16	Uint16 (highbyte first)	<CRC16 checksum>	Checksum over 60 bytes of Header: 0x02 – 0x3D 1)

Table 5-15: Alive command frame

Note 1): For the CRC16 calculation the CRC-CCITT is used (Polynom: 0x1021, start value: 0). Please ask the Bluetechnix support for an implementation example of the CRC-CCITT.

Note 2): For the CRC16 calculation the CRC-32 is used (Polynom: 0x04C11DB7, start value: 0xFFFFFFFF). Please ask the Bluetechnix support for an implementation example of the CRC-32.

Response frame

Addr	Field	Type	Value	Description
0x00	Preamble	Uint16 (highbyte first)	0xA1EC	Unique identifier, start of header
0x02	ProtocolVersion	Uint8	0x03	This document refers to version V3.0
0x03	Command	Uint8	0xFE	Command code for 'Alive message'
0x04	SubCommand	Uint8		Indicates which flash to write to
0x05	Status	Uint8	Refer to table	Result code
0x06	Flags	Uint16	Refer to table	Optional flags
0x08	Length	Uint32 (highbyte first)	0x0	No <Data> present
0x0C	HeaderData0	Uint8		Ignored

Addr	Field	Type	Value	Description
0x0D	HeaderData1	Uint8		Ignored
0x0E	HeaderData2	Uint8		Ignored
0x0F	HeaderData3	Uint8		Ignored
0x10	Reserved (42 bytes)	Uint8[]		Ignored
0x3A	DataCrc32	Uint32 (highbyte first)	0x0	No <Data> present [2)]
0x3E	HeaderCrc16	Uint16 (highbyte first)	<CRC16 checksum>	Checksum over 60 bytes of Header: 0x02 – 0x3D [1)]

Table 5-16: Alive response frame

Note 1): For the CRC16 calculation the CRC-CCITT is used (Polynom: 0x1021, start value: 0). Please ask the Bluetechnix support for an implementation example of the CRC-CCITT.

Note 2): For the CRC16 calculation the CRC-32 is used (Polynom: 0x04C11DB7, start value: 0xFFFFFFFF). Please ask the Bluetechnix support for an implementation example of the CRC-32.

Flags

Flags	Description
	Currently no flags defined for this command

Table 5-17: Alive flag description

Result codes:

Status	Description
0x00	Ok
0x0D	Invalid handle (internal error)
0x0F	Illegal write: The Address is not valid or the register is not write-enabled
0x10	Illegal read: The Address is not valid (deprecated, replaced by 17)
0x11	Register end reached
0xFA	Length exceeds maximum filesize (not enough memory for file download)
0xFB	HeaderCrc16 mismatch
0xFC	DataCrc32 mismatch
0xFD	Length invalid: Cannot be equal 0
0xFE	Length invalid: Cannot be grater 0
0xFF	Unknown command

Table 5-18: Result codes

5.1.2 I²C

The I²C is used as control interface for reading or writing the registers of the sensor module. The interface is compatible to the I²C specification Rev.4 February 2012 from NXP formally Phillips. The supported data rates are up to the 250 kHz. The default **hardware address** is **0x40.** The two LSBs of the hardware address can be selected by using the two address lines A0 and A1. These results in the following hardware address composition for read and write:

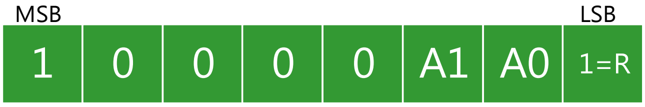


Figure 5-1: I2C read address

Figure 5-2: I2C write address

Note

Be aware that the I²C interface is only available on the 100pol board to board connector of the Sentis Main-Board. Please refer to 4.3 for information about the connector.

The following table shows the signals of the I²C control interface.

#	Name	Description	Type (Sensor view)	Status
1	SCL	Clock signal	I/O, 3.3V OD (pullup required)	
2	SDA	Data signal	I/O, 3.3V OD (pullup required)	
5	A0	Address line 0	I, 3.3V TTL	
6	A1	Address line 1	I, 3.3V TTL	

Table 5-19: I2C signal description

5.1.2.1 Timing characteristics

The following figure shows an I²C transfer for two bytes.

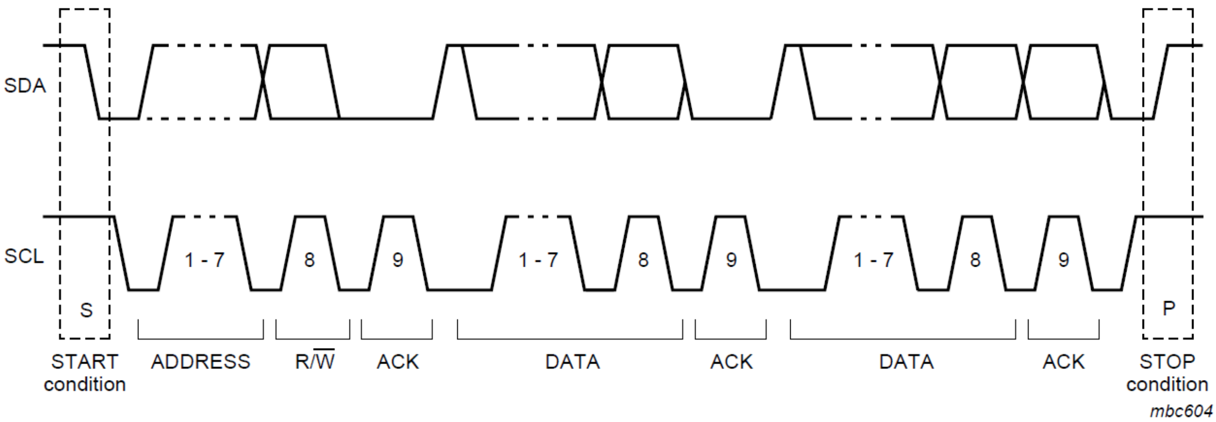


Figure 5-3: I²C timing characteristics for a two byte transfer

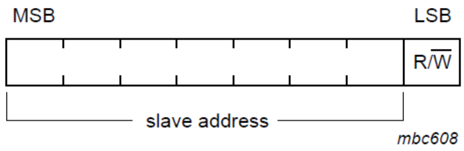


Figure 5-4: The first byte after the START procedure

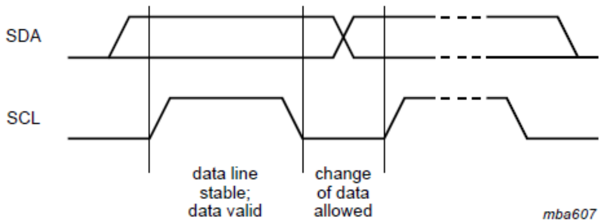


Figure 5-5: Clock to data reference

The minimum pulse width of the clock signal is 1µs. The minimum period of the clock signal is 2.5µs. The timing diagrams are taken from the I2C-bus specification and user manual, Rev. 4 – 13 February 2012 from NXP Semiconductors.

5.1.2.2 Data format

Each I²C transfer starts with the “START” condition followed by the slave address. In case of a write procedure the two bytes register address must follow with the high byte first. This register address will be taken as the base address for future read cycles. The transmission can stop at this time by sending a “STOP” condition or can continue with the content of the register to write into. Additional byte pairs may send by the master which will be written into the register auto-incrementing the register address.

In case of a read procedure the first two bytes sent by the slave are representing the content of the register addressed by the last write cycle. The master can read further byte pairs containing the content of the next register following the base address register auto-incrementing the register address. The register address will be reset to the first one after the “STOP” condition.

Note

The maximum number of registers which can be read or written using the auto-increment feature is 64 (128bytes). If the master will continue to read, the data is invalid. If additional bytes will be written by the master, they will be ignored.

The following transaction diagrams show the data stream for a register write and a register read procedure. Each transferred byte starts with the MSB first.

S	Slave Address	0 (W)	A/nA	Register address (Bit[14-7])	A/nA	Register address (Bit[6-0])	0	A/nA	Register content (Bit[15-8])	A/nA	Register content (Bit[7-0])	A/nA	P

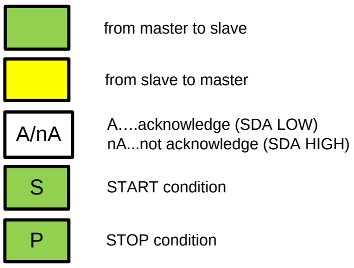


Figure 5-6: I2C register write procedure

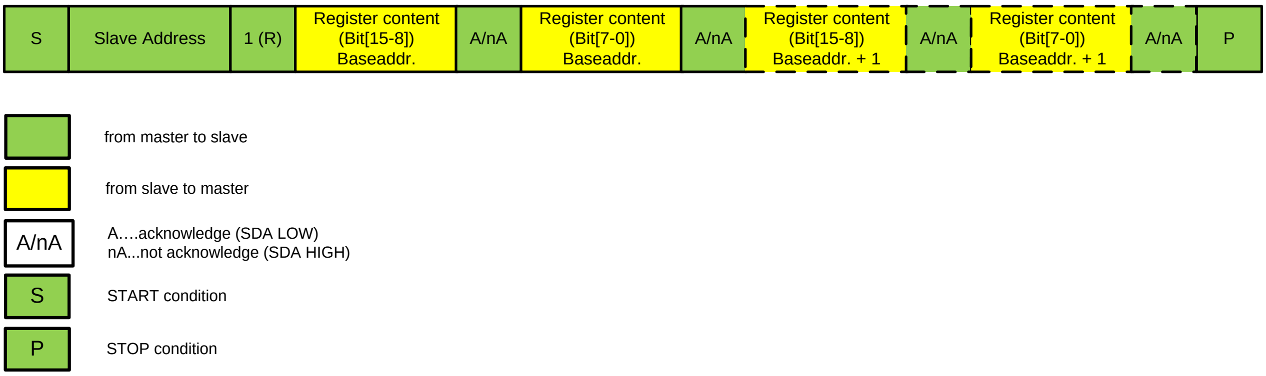


Figure 5-7: I2C register read procedure

5.2 Data Interface

For the data interface either a fast SPI (SSDI) or the Ethernet can be used. The Synchronous Serial Data Interface (SSDI) is only available on the 100 pol. board to board connector of the Main-Board.

!

Note

Different data interfaces uses different header formats. The data section (the frame content) is common for all data interfaces. Please refer to 6.3 for information on the available frame data formats.

5.2.1 Ethernet

On the Ethernet either a TCP or UDP stream can be used to read the depth and amplitude data from the Sentis. The factory default enables the UDP streaming but is also listening on the appropriate TCP port for incoming TCP connections requests. Therefore it is possible to establish a TCP and a UDP stream in parallel but this is not recommended as it may slow down the performance of the system and the achievable frame-rate.

Note

If you want to use TCP streaming disable the UDP streaming as otherwise it may slow down the performance.

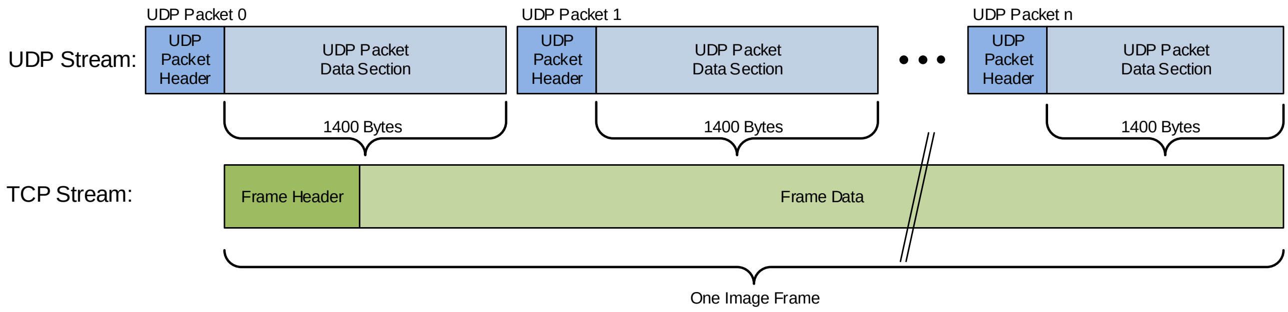


Figure 5-8: UDP and TCP streaming data format

The following types are used in the data streaming protocol:

- **Uint8**: 8 bit unsigned integer
- **Uint16**: 16 bit unsigned integer
- **Uint32**: 32 bit unsigned integer

Note
Values with '0x' as prefix are hexadecimal values.

Note
The Ethernet IP settings can be configured using the *Eth0_* registers. The changes become active on the next reconnect or reboot when the register map has been saved.

5.2.1.1 UDP

The UDP streaming is enabled by factory default. The Sentis streams to the following IP settings:

- **IP-Address**: Multicast address 224.0.0.1
- **UDP port**: 10002

As multicast is used more than one can receive the stream within the same subnet at the same time. The client has to join the appropriate multi cast group and open the port 10002 on his local network interface card (NIC) where the Sentis is connected to. The receiver should receive the stream and interpret it as the following protocol description shows.

Note

Be aware that a multicast stream may slow down the Ethernet network as the stream must be spread to all active links of switches/hubs and routers.

The **current protocol version** is **1**.

Each image transmitted on the UDP stream is split into packets of 1432 bytes length (except the last may be smaller). Each packet consists of a 32 byte packet header and 1400 bytes of image data section (refer to Figure 5-8).

Addr	Field	Type	Value	Description
0x00	Version	Uint16 (highbyte first)	0x0001	Protocol version
0x02	FrameCounter	Uint16 (highbyte first)		Continuous frame counter. On an overrun it restarts at 0.
0x04	PacketCounter	Uint16 (highbyte first)		Actual packet #. The frame data must be recomposed in order of the packet #.
0x06	DataLength	Uint16 (highbyte first)		Length of the image data section of the current packet.
0x08	FrameSize	Uint32 (highbyte first)		Size of the image data. It may be used to calculate the expected # of packets for a frame.
0x0C	PacketCRC32	Uint16 (highbyte first)		CRC32 checksum over the entire packet (pos 0 to pos n) [1)]
0x10	Flags	Uint32	Refer to table	Optional flags
0x14	Reserved			Reserved for future use
0x20	ImageData			Image data section

Table 5-20: UDP packet header

Note 1): For the CRC16 calculation the CRC-32 is used (Polynom: 0x04C11DB7, start value: 0xFFFFFFFF). Please ask the Bluetechnix support for an implementation example of the CRC-32.

<u>Flags</u>

Flags	Description
Bit 0	1: Ignore DataCrc32

Table 5-21: UDP packet header flag description

<u>Image data</u>

The image data itself is split into a 64 byte image header and the image data section. The format of the image data depends on the selected image format and is described in chapter 6.3. Below you can find the format of the 64 byte image header.

Addr	Field	Type	Value	Description
0x00	Reserved	Uint16	0xFFFF	
0x02	HeaderVersion	Uint16 (highbyte first)	0x0003	Current header version
0x04	ImageWidth	Uint16		Width of the image in pixels.

Addr	Field	Type	Value	Description
		(highbyte first)		
0x06	ImageHeight	Uint16 (highbyte first)		Height of the image in pixels.
0x08	NofChannels	Uint8		Nof data channels. Depends on the image format
0x09	BytesPerPixel	Uint8		Bytes per pixel of the image data.
0x0A	ImageFormat	Uint16 (highbyte first)		The content is the same as in the register *ImageDataFormat* (please refer to 8.6.1)
0x0C	Timestamp	Uint32 (highbyte first)		Timestamp of the actual image in µs
0x10	FrameCounter	Uint16 (highbyte first)		Continuous frame counter. On an overrun it restarts at 0.
0x12	Reserved			
0x1A	MainTemp	Uint8		Main-Board temperature in °C + 50. Decrement this field by 50 to get the current Main-Board temperature.
0x1B	LEDtemp	Uint8		LED-Board temperature in °C + 50. Decrement this field by 50 to get the current LED-Board temperature.
0x1C	FirmwareVersion	Uint16 (highbyte first)		Content of the register *FirmwareInfo*
0x1E	Reserved			
0x3E	CRC16	Uint16 (highbyte first)		CRC16 checksum over the header without the first two bytes and the CRC16 checksum itself (addr 0x02 to addr 0x3D) [1)]

Table 5-22: Image data header

Note 1): For the CRC16 calculation the CRC-CCITT is used (Polynom: 0x1021, start value: 0). Please ask the Bluetechnix support for an implementation example of the CRC-CCITT.

5.2.1.2 TCP

To use the TCP connection you must open a TCP connection to the Sentis module. On factory default the module is listening for incoming data connections on the following IP-Address and TCP port:

- **IP-Address**: 192.168.0.10
- **TCP port**: 10000

Once the connection is established the Sentis immediately streams the image data starting with the image header as described in Table 5-22 followed by the image content as described in 6.3. The image content depends on the selected image format.

! **Note**

Disable the UDP streaming using the register *Eth0Config* if you want to use TCP otherwise the performance may decrease.

5.2.2 SSDI

The SSDI interface is based on a standard SPI interface. It is used to transfer the image data. Find below a description of the associated signals.

Note

Be aware that the SSDI interface is only available on the 100pol board to board connector of the Sentis Main-Board. Please refer to 4.3 for information about the connector.

#	Name	Description	Type (Sensor view)	Status
1	DOUT0	Primary data output channel	O, 3.3V TTL	
2	DOUT1	Secondary data output channel	O, 3.3V TTL	n.a.
3	DIN	Data input	I, 3.3V TTL	
4	CLK	Clock	I, 3.3V TTL	
5	nSS	Slave select	I, 3.3V TTL	
6	nINT	Frame interrupt signal	O, 3.3V TTL	Meaning of the pin may be firmware dependent.
6	MnS	Master/Slave select	O, 3.3V TTL	Currently only slave is supported. Pullup to GND required. Usage of the pin may be firmware dependent.

Table 5-23: SSDI timing characteristics

This interface is used as data interface to stream out the image data and corresponding meta-data (frame header).

5.2.2.1 Timing characteristics

The following figures describe the timing characteristics of the SSDI interface:

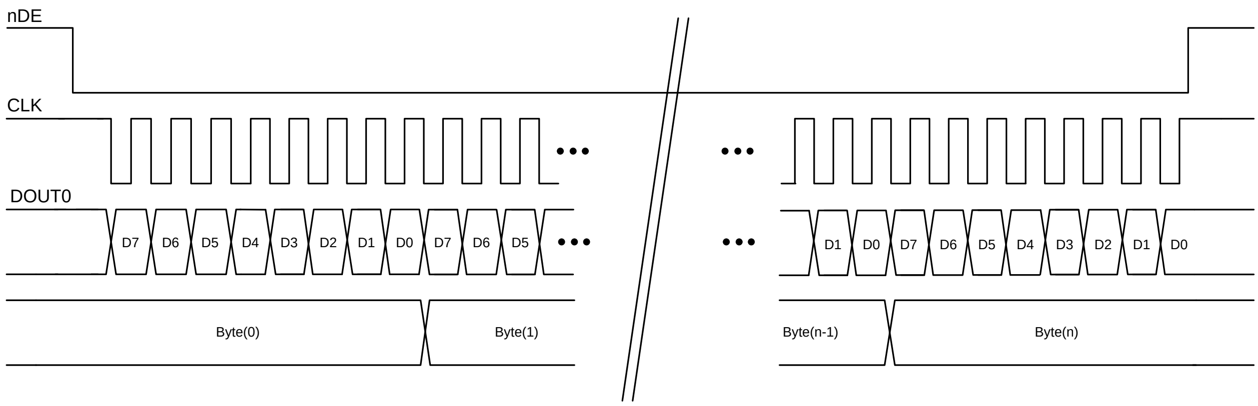


Figure 5-9: SSDI timing characteristics

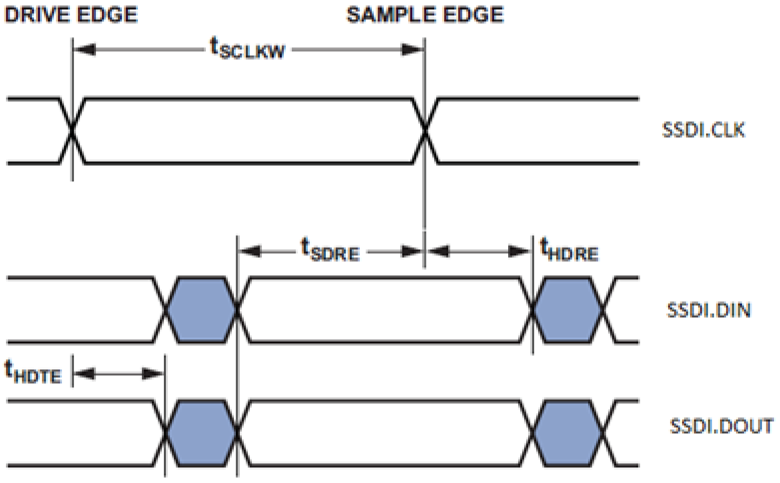


Figure 5-10: Clock to data in signal timing characteristics

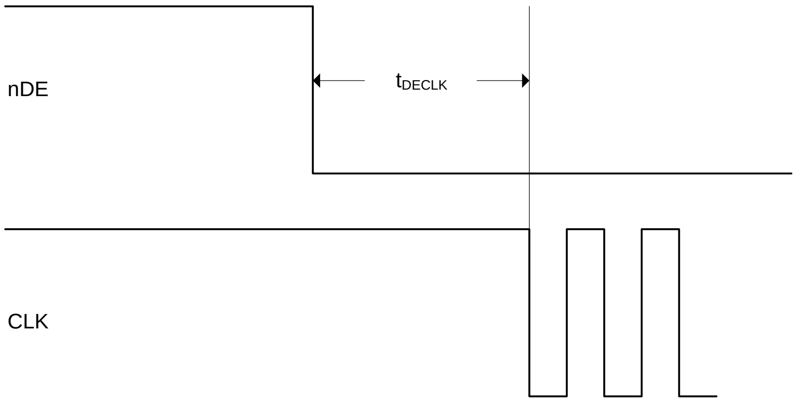


Figure 5-11: nDE assertion to CLK signal timing

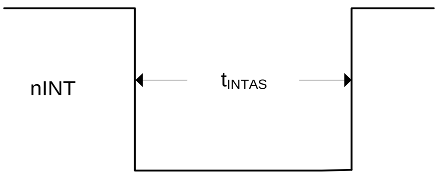


Figure 5-12: Interrupt assertion timing

Name	Description	Min	Typical	Max	Unit
t_{SCLKW}	Clock period	33	40		ns
t_{SDRE}	Data setup time	5			ns
t_{HDRE}	Data hold time	4			ns
t_{SCLKW} -t_{HDTE}	Data stable before sample edge	7			ns

Name	Description	Min	Typical	Max	Unit
t_{DECLK}	nDE signal asserted to first negative clock edge	2.5	3		µs
t_{INTAS}	Interrupt assertion time	0.2	0.35		µs

Table 5-24: SSDI timing characteristics

5.2.2.2 Data format

The streamed data is split into a 64 byte image header and the image data section. The format of the image data depends on the selected image format and is described in chapter 6.3.

The image data header format is the same as for TCP/IP Ethernet streaming (please refer to Table 5-22).

5.3 GPIOs

Three free programmable GPIOs are available. The GPIOs are not used by the default software and can be freely programmed with the *"Camera Firmware Development KITs"*. Please refer to chapter 7.3 for more information.

5.4 Frame trigger signal (TRIG)

The frame trigger signal is available on the GPIO connector of the Interface-Board (refer to 4.5.4). The signal is high active and edge sensitive and can be used to trigger a frame capturing. Once triggered the frame is available on the data interface with a latency of approx. 20ms depending on the configuration of the video processing chain and the filter configuration. To use the hardware trigger you have to disable the video mode and to enable the manual mode in register *Mode0*.

5.5 Modulation Signal Interface

The modulation light interface is only available by removing the LED-board and mounting the appropriate Adapter-board (refer to 0). The interface provides the modulation signal according to the selected modulation frequency as well as a one wire communication line between the external illumination module and the Sentis.

5.6 Status LED

The Status LED is used to give some basic information about the status of the Sentis. The following table shows the meaning depending on the mode.

Mode	LED signaling
Bootloader mode	Toggles every second
Video mode	Toggling with each frame (signals the frame-rate)
Manual mode	Toggles with each frame (signals frame capturing)

Table 5-25: Status LED meaning

The Status LED can be disabled using the register *Mode1*.

6 Camera Features

6.1 Basic Settings

The module comes up according to the reset (default) values as described in the register description section (refer to 8.6).

6.2 Image Processing Chain

The following flow diagram shows the image processing chain of the module for the depth data. For the amplitude data currently no post processing will be performed.

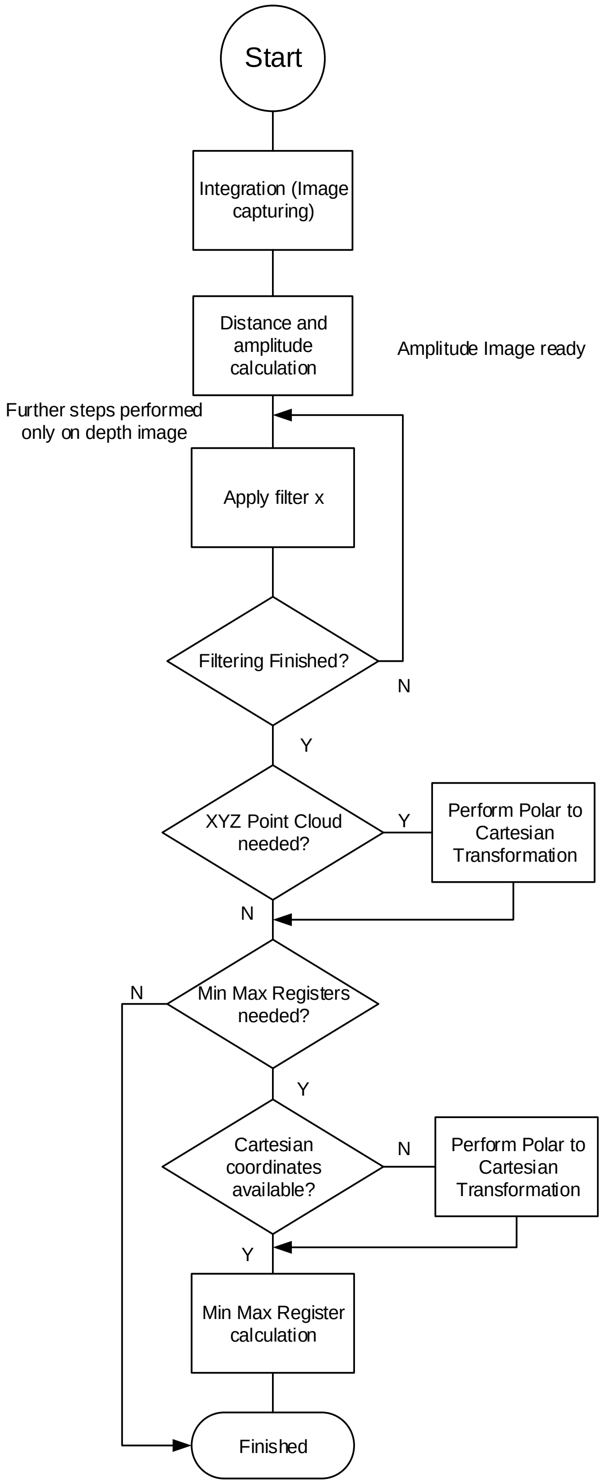


Figure 6-1: Image processing flow

6.2.1 Image filtering

After the distance and amplitude calculation some filters can be applied to the depth data. The amplitude data will be left unchanged. Each of the filter provides a configuration parameter. The iteration count for each filter can also be configured. The filters can be enabled or disabled by using the *ImgProcConfig*. Enabling more than one filter is possible but this may reduce the frame rate. Also the number of iterations influences the achievable maximum frame rate.

6.2.1.1 Median Filter

A 3 x 3 Median Filter can be applied.

Register: *FilterMedianConfig*

Only the number of iterations can be configured.

6.2.1.2 Average Filter

Register: *FilterAverageConfig*

Either a 3x3 or a 5x5 Average filter can be applied.

6.2.1.3 Gauss Filter

Register: *FilterGaussConfig*

Either a 3x3 or a 5x5 Gaussian filter can be applied.

6.2.1.4 Sliding Average Filter

Register: *FilterSLAFconfig*

A sliding average Filter over up to 20 frames can be applied. The number of frames can be configured. Only the distance data will be averaged. The amplitude data will be left unchanged. An increasing number of frames will not decrease the frame-rate but may add blurring effects.

6.2.2 Pixel invalidation

The Sentis provides an onboard check for invalid pixels. If the amplitude of the reflected signal is below a threshold (underexposure) the distance value of the appropriate pixel will be set to 0xFFFF. If the amplitude is too high (overexposure) the distance value will be set to 0x0000. The lower and upper bound for invalidating pixels can be set by using the registers *ConfidenceThresLow* and *ConfidenceThresHigh*.

6.3 Camera Data Format

The camera provides up to four data channels. The meaning of each data channel depends on the selected data format. The factory default setting provides an array of depth data in millimeters as 16 bit unsigned (Uint16) and an array of grayscale values (Amplitudes) also as 16bit unsigned for each pixel. In addition also a 3D XYZ point-cloud is provided. The following image shows the coordinate system from a cameras point of view:

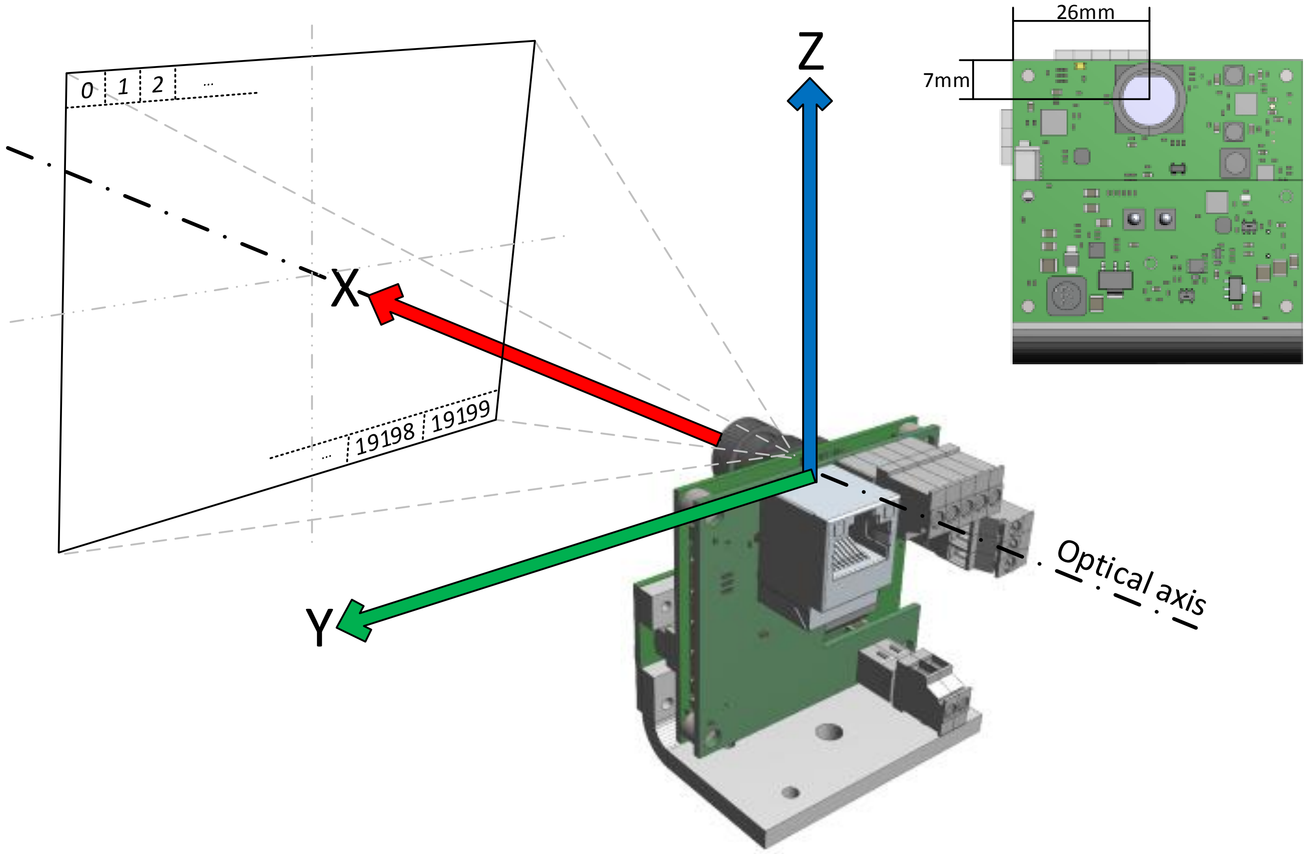


Figure 6-2: Sentis-ToF-M100 Coordinate System

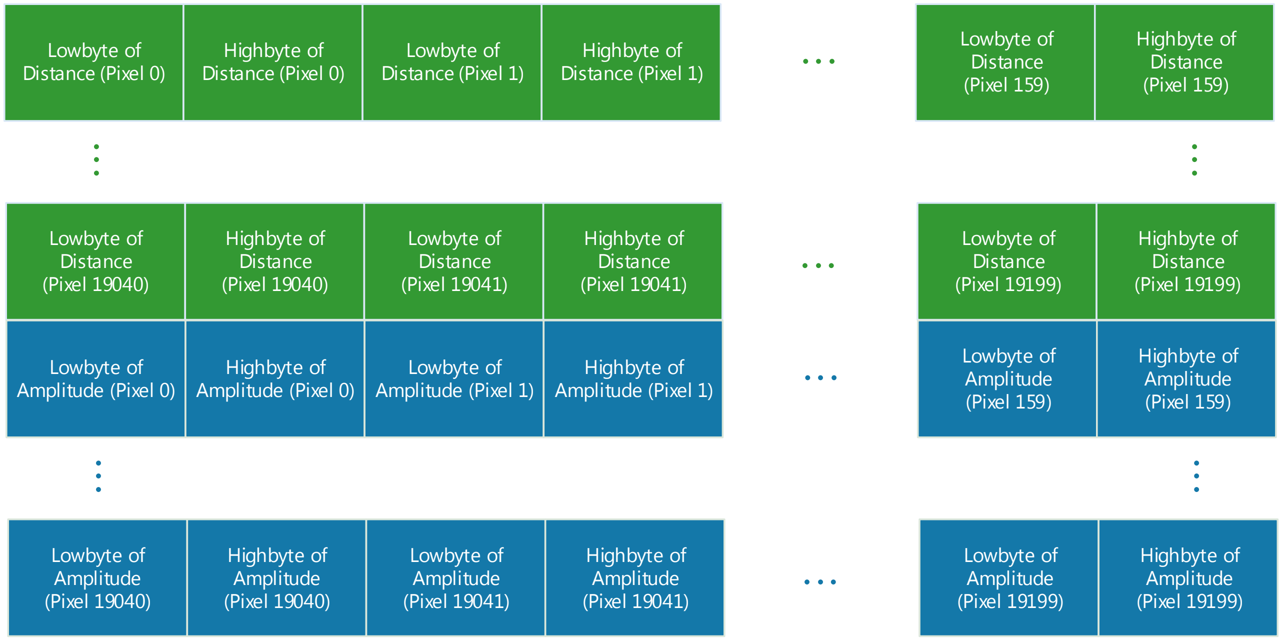
Which image format will be transferred can be selected by the register *ImageDataFormat*. Please refer to 8.3. The following section describes each of the supported formats in detail. Only the data section which contains the image data of the transferred frame will be described. For information about the packet format and meta-data please refer to 5.2 according to the interface used.

6.3.1 Distances and Amplitudes

In this mode the distances and amplitudes will be transferred in progressive mode, first the distance array then the amplitude array. The stream starts always with pixel #0.

The **distances** are coded in **[mm]** as **Uint16**, the **amplitudes** also as **Uint16**.

First Byte in Stream



Last Byte in Stream

Figure 6-3: Data-stream of Distance and Amplitude data

6.3.2 XYZ Point Cloud

In this mode the XYZ point cloud will be transferred in progressive mode, first the X coordinate array then the Y and Z coordinate array. The stream starts always with pixel #0.

The **coordinates** are coded in **[mm]** as **Int16.**

Figure 6-4: Data-stream of XYZ Point Cloud

6.3.3 XYZ Point Cloud and Amplitude

In this mode the XYZ point cloud and the amplitude will be transferred in progressive mode, first the X coordinate array then the Y and Z coordinate array. The stream starts always with pixel #0.

The **coordinates** are coded in **[mm]** as **Int16** the **amplitudes** as **Uint16.**

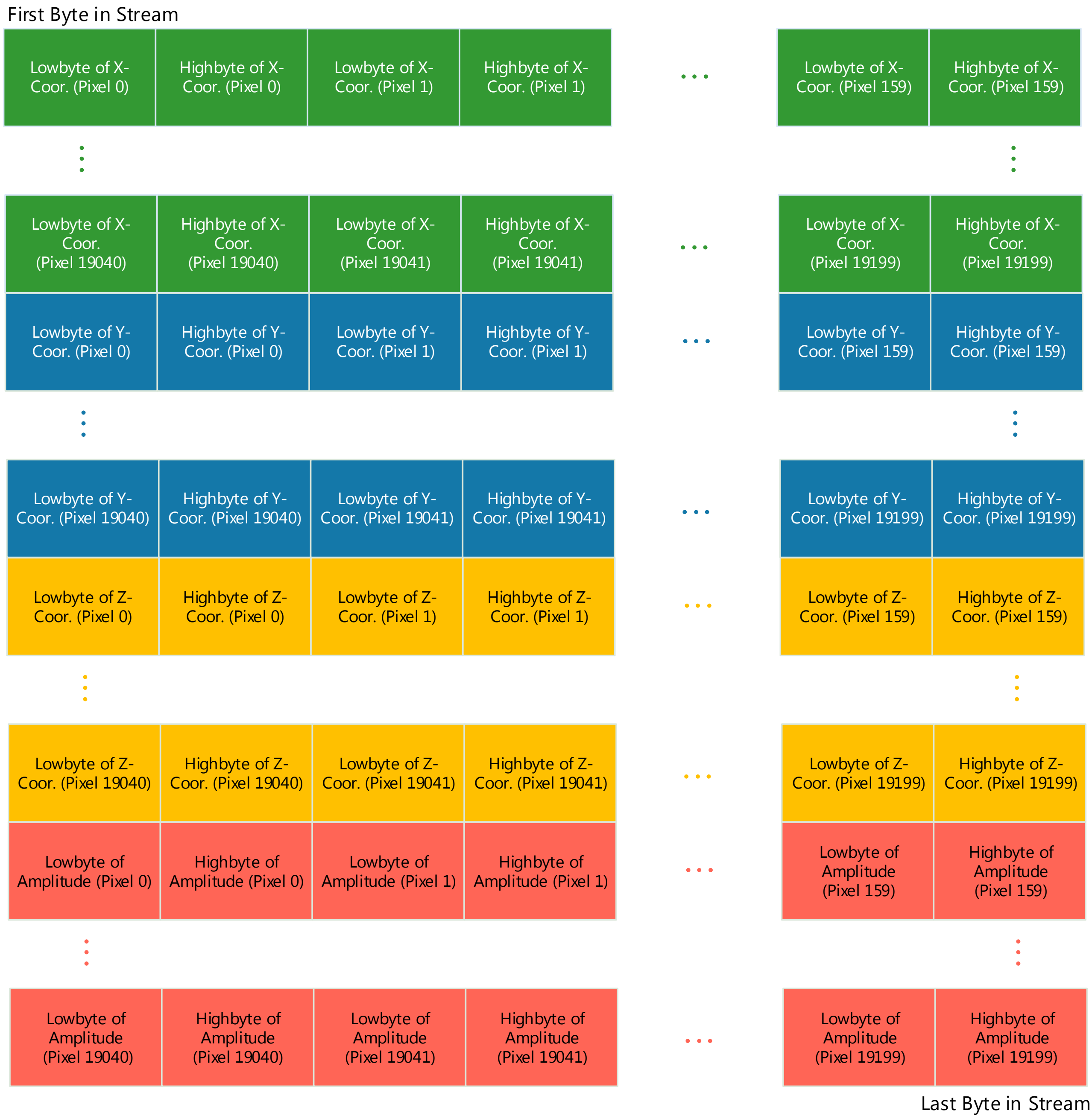


Figure 6-5: Data-stream of XYZ Point Cloud and Amplitude

6.4 Segmentation and Min Max Registers

The Sentis sensor module is capable to deliver a segmented interpretation of the image. For this feature the horizontal field of view is divided into a configurable set of slices (up to 32) using the *SegmentCount* register and for each slice (segment) the minimum and maximum distance value is calculated and readable by the registers *SegMinDistanceX* and *SegMaxDistanceX* where X represents the segment number (from 0 up to 31). This results in the division of the image in columns as shown in Figure 6-6. The vertical field of view can be configured either by the top and bottom opening angle (*TopOpeningAngle* and *BottomOpeningAngle*) (see

Figure 6-7) or by the top and bottom row (*TopRow* and *BottomRow*). Be aware that if the angle register is set, the appropriate row register will be updated automatically and vice versa.

The content of the min/max registers represents the X-coordinate and therefore the minimum and maximum distance projected to the optical axis.

This feature can be used to implement a simple “radar” like obstacle detection.

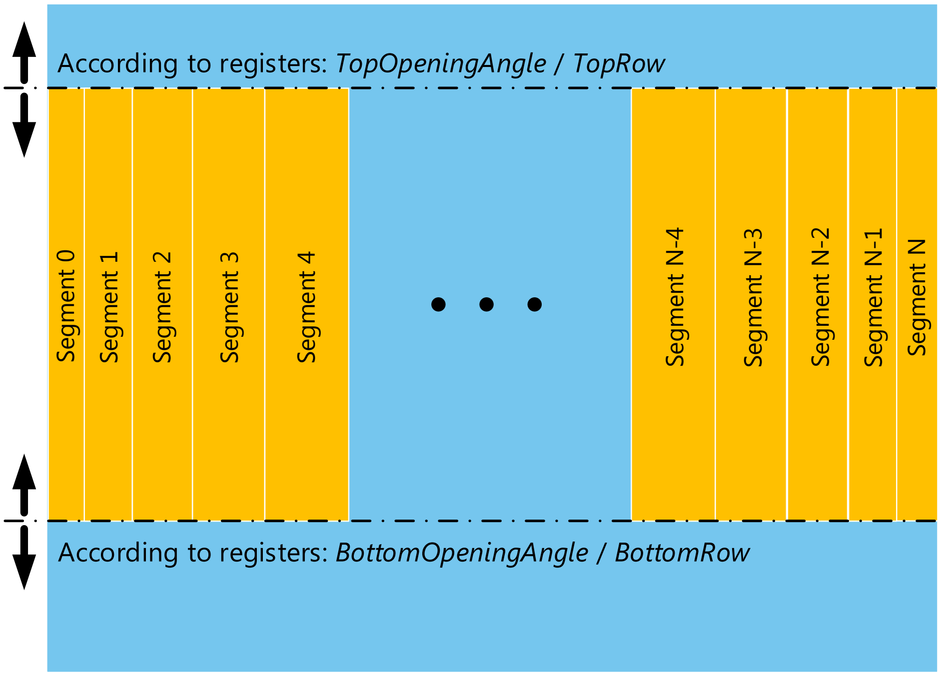


Figure 6-6: Segmentation of image

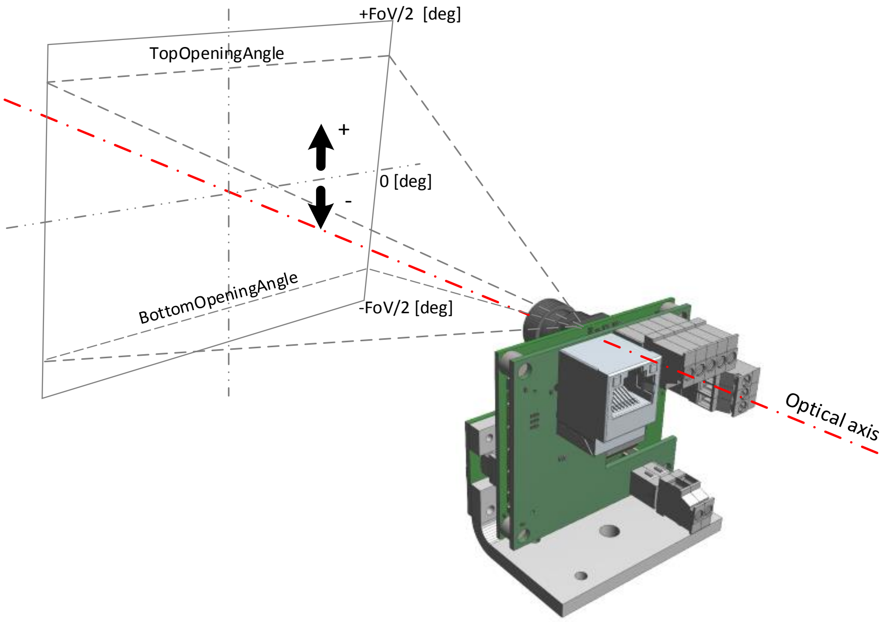


Figure 6-7: Meaning of top and bottom opening angle

6.4.1 Sliding average for Min Max registers.

For each Min and Max value an average over several frames can be calculated. To enable the sliding average calculation enable the appropriate Bit in register *Mode1.* The window size can be configured using *SegMinSlidingMedianWindowSize*.

6.5 Automatic Exposure Control (AEC)

The Sentis provides an automatic exposure control feature which controls the integration time according to the current observed scene. The AEC is disabled by default and must be enabled in the register *Mode1*.

6.6 Pixel invalidation

The Sentis has an integrated invalidation for pixels where the calculated amplitude is too high or too low. If the amplitude is to low the distance in the depth map of this pixel will be set to 0xFFFF if the amplitude is too high it will be set to 0x0000. The thresholds for this invalidation can be set using the registers *ConfidenceThresLow* and *ConfidenceThresHigh*.

6.7 Modulation Frequency

The modulation frequency is set to 20 MHz per default. Other modulation frequencies can be set using the register *ModulationFrequency*. Be aware that this also changes the ambiguity range of the camera.

The following modulation frequencies can be selected: 5MHz *(Index 0)*, 7.5MHz *(Index 1)*, 10MHz *(Index 2)*, 15MHz *(Index 3)*, 20MHz *(Index 4)*, 25MHz *(Index 5)*, 30MHz *(Index 6)*. In the register you can either write the frequency in kHz or the index. On a read of the register you get the current selected modulation frequency.

Other frequencies between 2,5Mhz and 30MHz can be set as well, but as there is no calibration data available for other frequencies than the above it may result in unpredictable distance values.

6.8 Frame-rate and Integration Time

The frame-rate and the integration time can be set by using the registers *Framerate* and *IntegrationTime*.

The camera integration time is limited by hardware to 27ms.

The maximum frame rate is 45fps but may be reduced by increasing the integration time. The combination of frame-rate and integration time influences the input current as well as the dissipated heat and will be characterized by the *"Frame-rate Integration Time Product"* (FITP) which has been defined as follow:

$$FITP = t_{INT}\,[ms] \cdot fps\left[\frac{1}{s}\right] \cdot 4$$

Caution

Be careful in setting different integration times and frame-rate combinations. Not all combinations are possible! Without appropriate cooling the device may be damaged! Please refer to 8.4.2 for valid values of the FITP.

!

Note

If the Auto Exposure Control is enabled the integration time will be set automatically and the register *IntegrationTime* should not be written!

6.9 Manual Frame Trigger

There are two types of manual trigger. To enable the manual trigger you have to disable the video mode in register *Mode0,* Bit[0].

6.9.1 Hardware Trigger

The camera provides an external connector where a hardware trigger can be applied. Please refer to 5.4 for more information.

6.9.2 Software Trigger

In addition to the hardware trigger a software trigger is available. To start a frame capturing by software, set the appropriate bit (Bit[4]) in register *Mode0*.

6.10 Over Temperature Protection

The Sentis provides an onboard over temperature protection for the LED-Board. If the LED-Board temperature reaches 90°C the frame capturing will be paused. It will be resumed if the temperature returns below 90° degree.

6.11 Save Registers

The entire register map can be saved in the flash using the register *CmdExec*. It will be restored from flash after a reboot or power cycle. Use this feature to save a user specific configuration.

6.12 Ethernet/IP Settings

6.12.1 MAC Address

A dedicated Ethernet MAC address from Bluetechnix MAC address pool is assigned to each Sentis module by factory default. This MAC address is saved in the OTP and cannot be changed by the user. The user is able to give the module his own MAC address using the register *Eth0Mac0* to *Eth0Mac2*. Be aware to make the changes persistent you have to safe the register map in flash using register *CmdExec* otherwise the changes will be lost on a reboot or power cycle. When the register map in the flash will be cleared the factory default MAC address from OTP will be loaded.

6.12.2 IP Settings

The IP Settings of the Sentis can be changes by the user using the *Eth0_* registers. A change of the IP settings (TCP/UDP port or IP-Address) will take affect after the next reconnect. To make the changes persistent you have to save the register map by writing a dedicated value to the *CmdExec* register.

6.13 Reset to Factory Default

The Module can be reset to the factory default register settings by deleting the saved register map. This can be done by writing a dedicated value to the register *CmdExec*.

6.14 Bootloader and Firmware Update

The Sentis will be delivered with a bootloader which is capable to update the onboard firmware. The communication with the bootloader will be done using dedicated TCP/IP command frames over the control interface connection or a set of registers over I^2C. The I^2C interface is only available on the 100 pol. Main-Board connector.

Bluetechnix provides a .NET based tool for updating the Sentis firmware over Ethernet. Please refer to our support site.

Sentis firmware update tool

https://support.bluetechnix.at/wiki/Sentis-ToF-M100_Camera

6.14.1 Boot Sequence

After a power on or reboot the bootloader will be started. The bootloader checks if a valid firmware is installed and tries to start the firmware. If no application can be found or the bootstrap pin is high at startup the bootloader stays in bootloader mode waiting for incoming TCP connection requests.

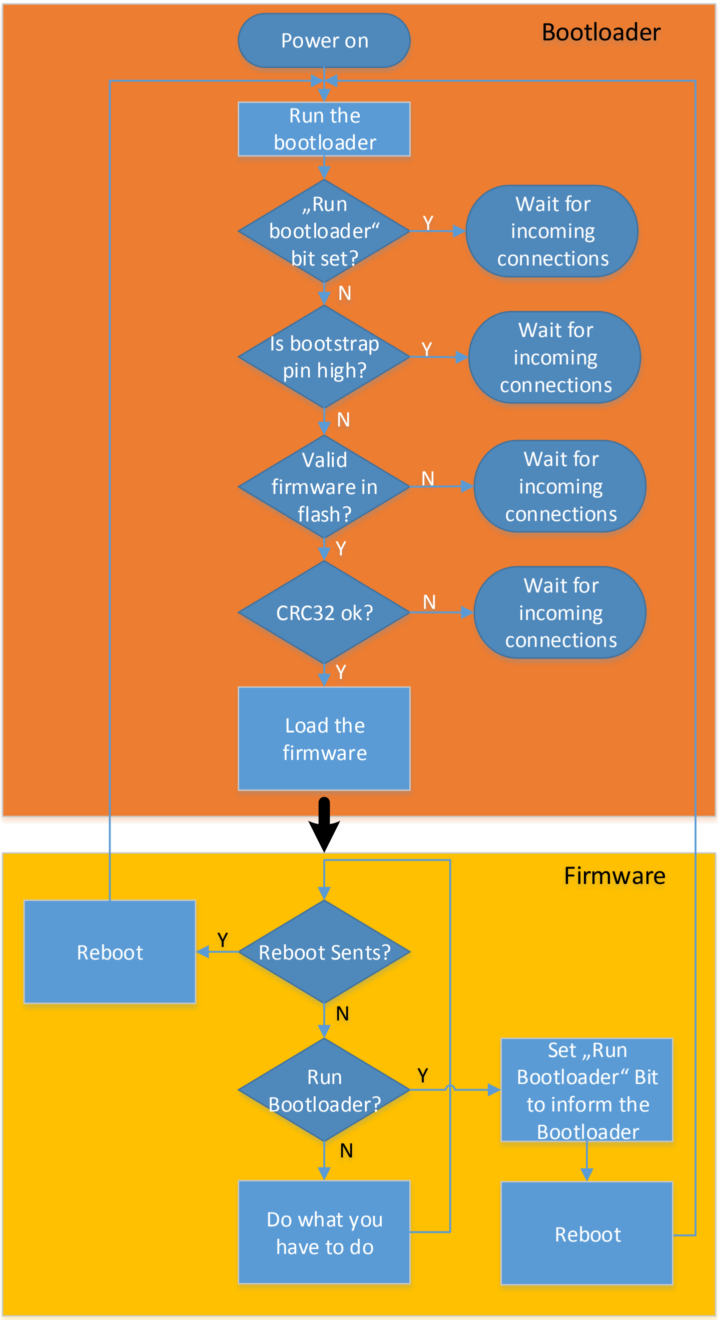


Figure 6-8: Boot sequence

6.14.1.1 Bootloader Bootstrap Pin

To force the module to stay in bootloader mode without starting the firmware a bootstrap pin can be used. To stay in bootloader mode connect pin #2 of the GPIO connector (refer to 4.5.4) to the positive input voltage before applying the supply voltage to the Sentis. The bootloader samples this pin immediately after startup and if it is high the bootloader doesn't start the firmware but remains in bootloader mode waiting for incoming commands or TCP connections on the control interface port.

6.14.2 Bootloader default settings

- **IP-Address:** 192.168.0.10
- **TCP port for the control interface:** 10001
- **MAC Address:** Factory default MAC address

!

Note

The bootloader doesn't use any saved register map but always factory default register settings. That means that any changes in the IP-Settings made for the firmware are not valid for the bootloader!

7 Software

7.1 Demo Application

For the first evaluation of the camera and to evaluate different settings and configurations a demo application is provided. The demo application can be downloaded from our support web site. Refer to the 'Quick Start Guide' for more information and visit our support site.

Software and documentation

https://support.bluetechnix.at/wiki/Sentis-ToF-M100_Camera

7.2 Getting Started Software Development Example

To facilitate the integration of the Sentis module in your own application a getting started example will be available on our download site. Please refer to our support site.

Software and documentation

https://support.bluetechnix.at/wiki/Sentis-ToF-M100_Camera

7.3 Camera Firmware Development KITs

The camera offers the possibility to develop your own firmware or to bring your application on the Sentis-ToF-M100. Using the dual-core processor ADSP-BF561 from Analog Devices Inc., one core is reserved for the calculation of the depth data, the other one can be used by the customers for their own applications.

Two different types of packages will be available for developing applications for the Sentis-ToF-M100.

7.3.1 VDSP++ Development Package

VDSP++ is an Integrated Devlopment Environment (IDE) provided by Analog Devices Inc. for the Blackfin processors. Bluetechnix provides a VDSP++ project where the user can put his own application code.

Refer to our support site for more information.

Software and documentation

https://support.bluetechnix.at/wiki/Sentis-ToF-M100_Camera

7.3.2 µCLinux Development Package

Bluetechnix provides a µCLinux Board Support Package (BSP) for the Sentis-ToF-M100 which can be used by delevopers to delevop there own Linux based applications which directly runs on the Sentis-ToF-M100.

Refer to our support site for more information.

Software and documentation

https://support.bluetechnix.at/wiki/Sentis-ToF-M100_Camera

8 Appendix

8.1 Operating Conditions

Symbol	Parameter	Min	Typical	Max	Unit
V_{IN}	Input supply voltage	18	24	30	V
I_{IN}	Input current [1)]	TBD	TBD [1)]	TBD	mA
	Input current without Interface-Board [1)]	TBD	TBD	TBD	mA
	Input current (Mainboard only) [3)]	TBD	TBD	TBD	mA
T	Operating Temperature [2)]	TBD		TBD[2)]	°C
T	Storage Temperature	-40		+125	°C
FITP[4)]	Frame-rate Integration Time Product			TBD	

Table 8-1: Operating Conditions

!

1) Note

Valid for a frame-rate of 40fps and an integration time of 1500µs. The input current depends on the applied frame-rate and integration time. Please refer to 8.1.1.

2) Note

The maximum operating temperature depends on the frame-rate and integration time. Refer to Figure 2-1 for recommended integration time to frame-rate combinations.

3) Note

The typical value is measured with both BF561 cores active and medium CPU load.

4) Note

Refer to 8.4.2 for valid frame-rate to integration time combinations.

8.1.1 Input current

The input current depends on the selected frame-rate (fps) and the integration time (t_{INT}). The following figure shows typical values. The values for the x axis shows the FITP which has been calculated with the following equation:

$$FITP = t_{INT}\,[ms] \cdot fps\left[\frac{1}{s}\right] \cdot 4$$

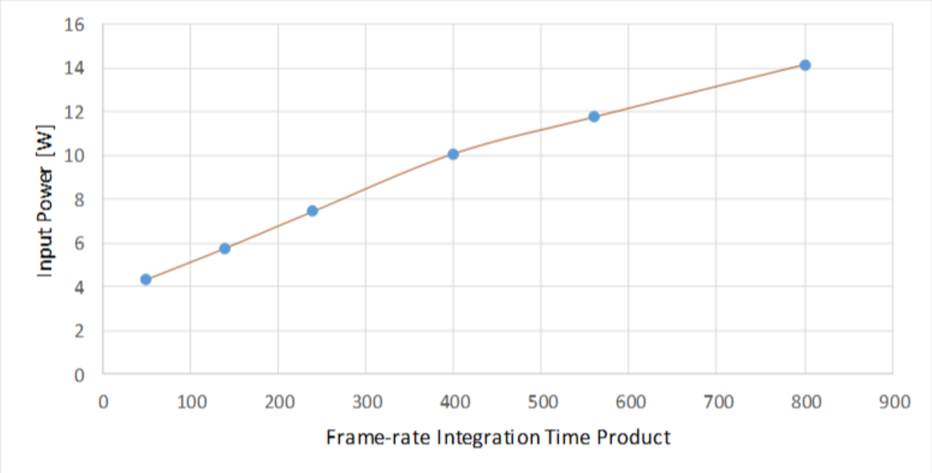


Figure 8-1: Input power depending on frame-rate integration time product

8.2 Optical Characteristics

Symbol	Parameter	Min	Typical	Max	Unit
#LEDs	Nr. of LEDs		2		
$\Lambda_{CENTROID}$	Centroid-Wavelength of Illumination		850		nm
$\Delta\lambda$	Spectral Bandwidth		30		nm
I_e	Radiant intensity				W/sr
FoV_H	Horizontal Field of View		90		Deg
FoV_V	Vertical Field of View		67,5		Deg

8.3 Measurement Specifications

8.3.1 Measurement Environmental Conditions

All the following measurements have been acquired at the following constant environmental conditions.

Parameter	Value
Temperature	23 °C
Humidity	35 %
Ambient light	2 kLux
Modulation Frequency	20 MHz
Frame-rate	25 fps

Table 8-2: Environmental Specification

8.3.2 Typical Reproducibility

The following table shows the standard deviation over 100 samples.

Measuring range [mm]	White target (90%) [mm]	Integration time [ms]	Gray target (18%) [mm]	Integration time [ms]
100			TBD	0,5
300	TBD	0,5	TBD	0,5
500	TBD	0,5	TBD	0,5

Measuring range [mm]	White target (90%) [mm]	Integration time [ms]	Gray target (18%) [mm]	Integration time [ms]
700	TBD	0,5	TBD	1,0
900	TBD	0,5	TBD	1,0
1100	TBD	1,0	TBD	1,5
1300	TBD	1,0	TBD	1,5
1500	TBD	1,0	TBD	2,0
1700	TBD	1,5	TBD	5,0
1900	TBD	1,5	TBD	5,0
2100	TBD	1,5	TBD	10,0
2300	TBD	2,0	TBD	10,0
2500	TBD	2,0	TBD	10,0
2700	TBD	2,0	TBD	10,0
2900	TBD	2,0	TBD	10,0

Table 8-3: Typical Reproducibility

8.3.3 Typical Integration Time

Measuring range [mm]	Integration time for white target (90%) [ms]	Integration time for gray target (18%) [ms]
500	1,0	1,0
1000	1,0	1,5
1500	1,5	2,0
2000	1,5	3,0
2500	2,0	5,0
3000	2,5	10,0

Table 8-4: Typical Integration Time

8.3.4 Typical Range

Integration time [ms]	Minimum distance for white target (90%) [mm]	Maximum distance for white target (90%) [mm]	Minimum distance for gray target (18%) [mm]	Maximum distance for gray target (18%) [mm]
0,5	300	1500	100	500
1,0	300	2000	100	1000
1,5	300	2000	300	1500
2,0	500	3000 1)	300	2000
2,5	500	3000 1)	300	2000
3,0	500	3000 1)	300	2000
5,0	500	4000 1)	300	3000 1)
10,0	500	6000 1)	500	3000 1)

Table 8-5: Typical Range

8.3.5 Accuracy of Distances

The following table has been determined by calibrating the device at a distance of 1500mm and an integration time of 1,5ms. For applications with specific environment optimized calibration may improve the error results.

Measuring range [mm]	White target (90%) [mm]	Integration time [ms]	Gray target (18%) [mm]	Integration time [ms]
500	TBD	1,0	TBD	1,0
1000	TBD	1,0	TBD	1,5

Measuring range [mm]	White target (90%) [mm]	Integration time [ms]	Gray target (18%) [mm]	Integration time [ms]
1500	TBD	1,5	TBD	2,0
2000	TBD	1,5	TBD	3,0
2500	TBD	2,0	TBD	5,0
3000	TBD	2,5	TBD	10,0

Table 8-6: Accuracy of Distances

8.4 Environmental considerations

8.4.1 Temperature on the Cooling Plate

The following figure shows the expected case temperature depending on the frame-rate integration time product (FITP) and the ambient temperature. The FITP has been calculated as follow:

$$FITP = t_{INT}\,[ms] \cdot fps\left[\frac{1}{s}\right] \cdot 4$$

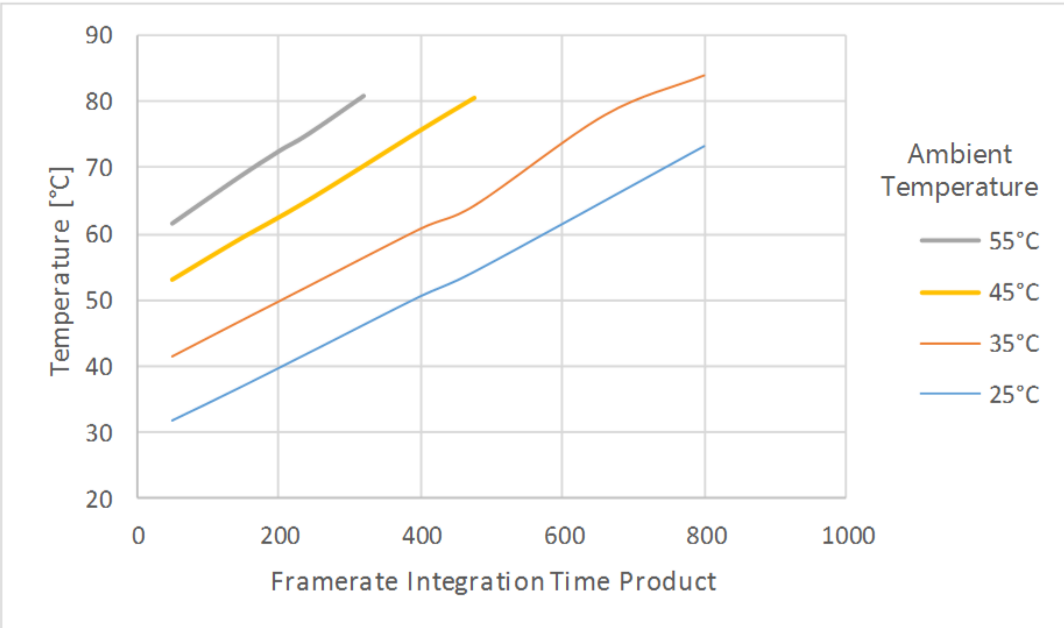


Figure 8-2: Expected cooling plate temperature depending on frame-rate integration time product

The temperature on the cooling plate can be reduced by mounting an additional heat sink on the cooling plate.

8.4.2 Integration Time vs. Frame-rate

The following table shows recommended frame-rate integration time combinations depending on the ambient temperature.

Caution

Be careful to not stress the device beyond the limits, otherwise you may damage the device.

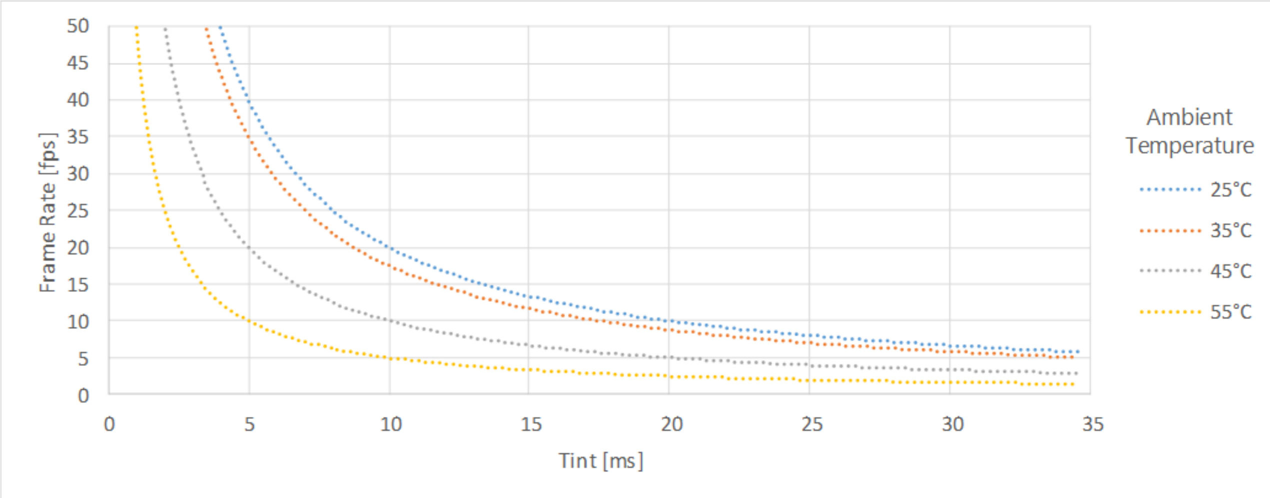


Figure 8-3: Integration time vs. frame-rate

The diagram takes care to limit the FITP in a way that the temperature on the cooling plate doesn't exceed 70°C. Using an appropriate heat sink higher values of the FITP may be applied.

Caution

The user is responsible to take care for an appropriate cooling if the Sentis is mounted into a case.

8.5 Mechanical Outline

All dimensions are given in mm.

Mechanical outline of the 'Bounding Box':

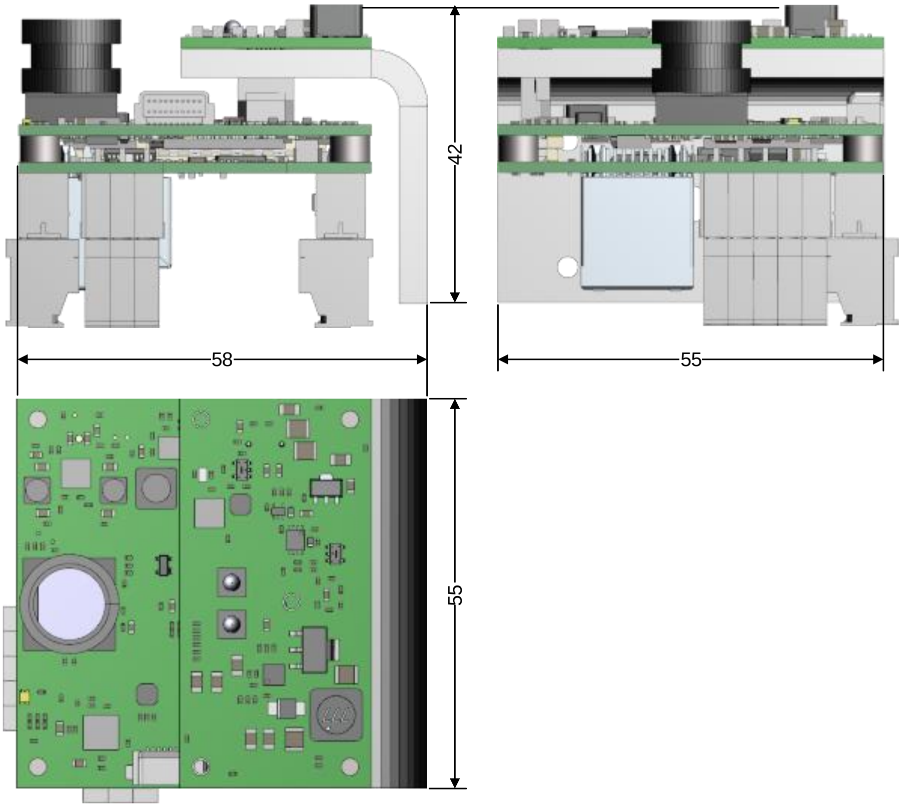


Figure 8-4: Mechanical Outline of the Bounding Box

8.6 Register Description

8.6.1 General registers

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
0001	Mode0	0001	R/W	Bit[0]: 0..Manual Mode, 1.. Video Mode Bit[4]: 1..Manual Trigger Bit[5]: Reserved Bit[6]: 1..Clear status register Bit[7]: Reserved Bit[8]: 1..Start Bootloader
0003	Status	0000	R	Bit[0]: 0..Application Mode, 1..Bootloader Mode Bit [1]: Frame Ready Bit[2]: 1..Ongoing Calibration Bit[3]: 1..LED-Board temperature sensor error Bit[4]: 1..Main-Board temperature sensor error Bit[5]: 1..Calibration data missing Bit[6]: 1..Factory Regmap was loaded Bit [12]: Frame Faulty
0004	ImageDataFormat	0000	R/W	Bit[3:6]: 0… 2 Byte Depth-Data / 2 Byte Amp-Data 3… X/Y/Z Coordinates (2 Byte in signed format for each coordinate) 4… X/Y/Z Coordinates and Amp-Data (2

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
				byte in signed format for each coordinate,2 byte unsigned for the amp value) 7… 4 Phases without image processing 4 times 2 bytes for 0°, 90°, 180° and 270° 8… 4 Phases without image processing 4 times 2 bytes for 270°, 180°, 90° and 0°
0005	IntegrationTime	05DC	R/W	Integration Time [µs] (min: 50, max: 25000)
0006	DeviceType	A9C1	R	Hardware specific identification
0007	DeviceInfo		R	Bit[0-3]: PCB Revision 3) Bit[4-7]: BOM Revision
0008	FirmwareInfo		R	Bit[0-5]: Non Functional Revision Bit[6-10]: Minor Revision Bit[11-15]: Major Revision
0009	ModulationFrequency	4E20	R/W	Modulation Frequency [10000Hz]
000A	Framerate	0028	R/W	Framerate [Hz]
000B	HardwareConfiguration	005A	R/W	Bit[0-7]: Lens Angle [°] (90)
000C	SerialNumberLowWord		R	Lower 16bit of the 32bit Serial Number
000D	SerialNumberHighWord		R	Higher 16bit of the 32bit Serial Number
000E	FrameCounter		R	Frame Counter (increments on every captured frame)
0010	ConfidenceThresLow	0064	R/W	Amplitude threshold for valid distance data
0011	ConfidenceThresHigh	3A98	R/W	Amplitude threshold for valid distance data
0019	Mode1	0800	R/W	Bit[3]: 0..AEC Off, 1..AEC On Bit[7]: 1..MinMax calculation On Bit[8]: 1..Disable status LED Bit[11]: 1..Enable Sliding Median for Min Registers
001A	CalculationTime	0000	R	Calculation time for the last frame in 100[µs]. The inverse of this value shows the maximum achievable frame rate based on the CPU load.
001B	LedboardTemp	0000	R	Temperature of LED-Board in 0,01[°C] (FFFF: Sensor not available).
001C	MainboardTemp	0000	R	Temperature of Main-Board in 0,01[°C] (FFFF: Sensor not available).
001D	LinearizationAmplitude	01AE	R/W	Amplitude for Linearization Function [float value x 10000]
001E	LinearizationPhaseShift	1B58	R/W	Amplitude for Linearization Function [float value x 10000]
0022	CmdExecPassword		R/W	Password for enabling command execution of commands written to CmdExec 3) Execution password: 0x4877
0024	MaxLedTemp	2328		Maximum tolerable LED-Board temperature 0,01[°C]
0026	HorizontalFov	2)	R	Horizontal field of view in 0,01[°]
0027	VerticalFov	2)	R	Vertical field of view in 0,01[°]
002B	TriggerDelay	0000	R/W	Delay between trigger assertion (either software or hardware) and image capturing [ms]
002C	BootloaderStatus		R	Bit[13-15]: Firmware Load Counter. This counter is reset by the firmware. It counts the boot trials. In Bootloader mode it is used to detect a firmware load problem
002D	TemperatureCompensationGradient	1004	R/W	Gradient of the temperature compensation

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
				0,001[mm/°C]
002E	ApplicationVersion		R	See "FirmwareInfo (0x0008)" for bit description, in Bootloader mode this register contains the firmware info of the flashed application
002F	DistCalibGradient	0000	R/W	Gradient of dist value, interpreted as fixed comma shifted by 14 binary digits
0030	DistCalibOffset	0000	R/W	Offset of dist value, interpreted as signed short
0033	CmdExec		R/W	Initiate an operation: 0xC2AE.. Clear RegMap in flash 0x9E20.. Read RegMap from flash 0x909A.. Read factory RegMap 0xDD9E.. Save RegMap in flash 3)
0034	CmdExecResult		R	Result code of the operation initiated using CmdExec 1.. Success Other.. Error
0035	FactoryMacAddr2		R	Highest and second highest byte of the MAC address stored in OTP flash
0036	FactoryMacAddr1		R	Byte 3 and 2 of the MAC address stored in OTP flash
0037	FactoryMacAddr0		R	Byte 1 and lowbyte of the MAC address stored in OTP flash
0038	FactoryYear		R	Production year (stored in OTP flash)
0039	FactoryMonthDay		R	Bit[0-7]: Production day (stored in OTP flash) Bit[8-15]: Production month (stored in OTP flash)
003A	FactoryHourMinute		R	Bit[0-7]: Production hour (stored in OTP flash) Bit[8-15]: Production minute (stored in OTP flash)
003B	FactoryTimezone		R	Production timezone (stored in OTP flash)

Table 8-7: General register

Note 1): The number of median iterations may have an impact on the achievable frame rate. The frame rate may decrease on incrementing this register.

Note 2): The content depends on the mounted lens and the calibration data and represents the real viewing angles.

Note 3): To enable command execution of commands written to CmdExec register it is necessary to write 0x4877 to the CmdExecPassword register before each command execution.

8.6.2 Registers for Segmentation

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
0050	SegmentCount	0020	R/W	Nr. Of Segments/Macrocells
0051	TopOpeningAngle	1)	R/W	Top Opening Angle in 0,01[°] 4)
0052	TopRow	0000	R/W	Bit[0-7]: Top Row (0-119)
0053	BottomOpeningAngle	1)	R/W	Top Opening Angle in 0,01[°] 4)

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
0054	BottomRow	0077	R/W	Bit[0-7]: Bottom Row (0-119)
0055	SegmentMinCountThreshold	1388	R/W	Bit[0-15] Percentage of pixel per segment where the distance has to be 0 to set the minimum distance to 0, [Percent * 100] 4)
0056	SegmentMaxCountThreshold	1388	R/W	Bit[0-15] Percentage of pixel per segment where the distance has to be 0xFFFF to set the maximum distance to 0xFFFF, [Percent * 100]
0057	SegMinSlidingMedianWindowSize	0005	R/W	Windows size for the Sliding Median Filter, has to be odd. (Maximum: 15)
005F	SegMinMaxError		R	Error for the min max calculation (cleared on read): 0..No error (currently no error available)
0060	SegMinDistance0		R	Minimum Distance of Segment 0, 0.. Amplitude to high 3), 0xFFFF..Amplitude to low 2).
0061	SegMinDistance1		R	Minimum Distance of Segment 1, 0.. Amplitude to high 3), 0xFFFF..Amplitude to low 2)
	…		…	…
007E	SegMinDistance30		R	Minimum Distance of Segment 30, 0.. Amplitude to high 3), 0xFFFF..Amplitude to low 2)
007F	SegMinDistance31		R	Minimum Distance of Segment 31, 0.. Amplitude to high 3), 0xFFFF..Amplitude to low 2)
0080	SegMaxDistance0		R	Maximum Distance of Segment 0, 0.. Amplitude to high 3), 0xFFFF..Amplitude to low 2)
0081	SegMaxDistance1		R	Maximum Distance of Segment 1, 0.. Amplitude to high 3), 0xFFFF..Amplitude to low 2)
	…		…	…
009E	SegMaxDistance30		R	Maximum Distance of Segment 30, 0.. Amplitude to high 3), 0xFFFF..Amplitude to low 2)
009F	SegMaxDistance31		R	Maximum Distance of Segment 31, 0.. Amplitude to high 3), 0xFFFF..Amplitude to low 2)
00C1	DistOffset0		R/W	An offset for distance values when operating at modulation frequency with index 0
00C2	DistOffset1		R/W	An offset for distance values when operating at modulation frequency with index 1
00C3	DistOffset2		R/W	An offset for distance values when operating at modulation frequency with index 2
00C4	DistOffset3		R/W	An offset for distance values when operating at modulation frequency with index 3
00C5	DistOffset4		R/W	An offset for distance values when operating at modulation frequency with index 4
00C6	DistOffset5		R/W	An offset for distance values when operating at modulation frequency with index 5
00C7	DistOffset6		R/W	An offset for distance values when operating at modulation frequency with index 6
00C8	DistOffset7		R/W	An offset for distance values when operating at modulation frequency with index 7
00C9	DistOffset8		R/W	An offset for distance values when operating at modulation frequency with index 8
00CA	DistOffset9		R/W	An offset for distance values when operating at modulation frequency with index 9

Table 8-8: Register for segmentation

Note 1): The content depends on the lens and the calibration data and will be set per default to the field of view. Dedicated registers will be provided for each segment. This causes a bit more overhead if all segments must be read. But it is also possible to read only the segments of interest

Note 2): This means that the amplitude is too low for correct distance measurement. The object might be underexposed.

Note 3): This means the amplitude is too high for correct measurement. The object might be overexposed.

Note 4): The top angle must be greater than or equal to the bottom angle. The top angle cannot be higher than the Field of View angle and the bottom angle cannot be below the Field of View angle.

8.6.3 Registers for Automatic Exposure Control

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
01B0	AecAmpTarget	0320	R/W	Auto exposure target amplitude value to which the controller is controlling to
01B1	AecTintStepMax	07D0	R/W	Auto exposure maximum change of integration time per second
01B2	AecTintMax	1B58	R/W	Auto exposure maximum integration time the controller calculates
01B3	AecKp	0046	R/W	Proportional part of the auto exposure controller in percent
01B4	AecKi	0002	R/W	Integral part of the auto exposure controller in percent
01B5	AecKd	0003	R/W	Differential part of the auto exposure controller in percent

Table 8-9: Register for automatic exposure control

8.6.4 Registers for Device Update (Only Bootloader)

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
01D1	FileUpdateStatus	0000	R	0.. idle 1.. ok 2.. max_filesize_exceeded 3.. out_of_memory 4.. buffer_overrun 5.. packet crc error 6.. file crc error 7.. file ok 8.. erasing flash 9.. flashing 10.. verifying 11.. erasing failed 12.. flashing failed

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
				13.. verifying failed 14.. update success 15.. wrong packet nr 16.. header version conflict 17.. missing fw identifier 18.. wrong fw identifier 19.. flash boundary exceeded 20.. data inconsistent 21.. in progress 255.. protocol violation
01D2	FileSizeLow	0000	R/W	Size of the file to be transmitted
01D3	FileSizeHigh	0000	R/W	Size of the file to be transmitted
01D4	FlashStartAddLow	0000	R/W	Position in flash to write to
01D5	FlashStartAddrHigh	0000	R/W	Position in flash to write to
01D6	FileCrcLow	0000	R/W	Crc sum of the file to be transmitted
01D7	FileCrcHigh	0000	R/W	Crc sum of the file to be transmitted

Table 8-10: Register for device update

8.6.5 Registers for Filter Configuration

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
01E0	ImgProcConfig	2FC1	R/W	Bit[0]: 1… enable Median Filter Bit[1]: 1… enable Average Filter Bit[2]: 1… enable Gauss Filter Bit[4]: 1… enable Sliding Average Bit[6]: 1… enable wiggling compensation Bit[7]: 1… enable FPPN compensation Bit[8]: 1… enable ModFreq scaling Bit[9]: 1… enable scaling to [mm] Bit[10]: 1… enable additive offset Bit[11]: 1… enable temperature compensation Bit[12]: 1… enable scaling via register DistCalibGradient (0x002F) Bit[13]: 1… enable scaling via register DistCalibOffset (0x0030)
01E1	FilterMedianConfig	0001	R/W	Bit[0-7]: … Nr of Median Iterations
01E2	FilterAverageConfig	0100	R/W	Bit[0-7]: 0… 3x3 Pixel 1… 5x5 Pixel Bit[8-15]:… Nr of Iterations
01E3	FilterGaussConfig	0100	R/W	Bit[0-7]: 0… 3x3 Pixel 1… 5x5 Pixel Bit[8-15]:… Nr of Iterations
01E5	FilterSLAFconfig	0005	R/W	Bit[0-7]: …Window size

Table 8-11: Register for filter configuration

8.6.6 Registers for Ethernet Configuration

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
0240	Eth0Config	0006	R/W	Bit[0]: 1.. Enable DHCP Bit[1]: 1.. Enable UDP streaming Bit[2]: 1.. Ignore CRC for UDP streaming
0241	Eth0Mac2		R/W	Highbyte and byte 4 of MAC address
0242	Eth0Mac1		R/W	Byte 3 and byte 2 of MAC address
0243	Eth0Mac0		R/W	Byte 1 and lowbyte of MAC address
0244	Eth0Ip0		R/W	Lowword of IP address
0245	Eth0Ip1		R/W	Highword of IP address
0246	Eth0Snm0		R/W	Lowword of subnetnask
0247	Eth0Snm1		R/W	Highword of subnetmask
0248	Eth0Gateway0		R/W	Lowword of gateway
0249	Eth0Gateway1		R/W	Highword of gateway
024A	Eth0TcpStreamPort		R/W	Port for TCP streaming
024B	Eth0TcpConfigPort		R/W	Port for TCP control interface
024C	Eth0UdpStreamIp0		R/W	Lowword of IP address for UDP stream
024D	Eth0UdpStreamIp1		R/W	Highword of IP address for UDP stream
024E	Eth0UdpStreamPort		R/W	Port for UDP streaming

Table 8-12: Registers for Ethernet configuration

8.7 Support

8.7.1 General Support

General support for products can be found at Bluetechnix' support site

Support Link

https://support.bluetechnix.at/wiki/Sentis-ToF-M100_Camera

8.8 Software Packages

Software packages and software downloads are for registered customers only

Software Package

https://support.bluetechnix.at/wiki/Sentis-ToF-M100_Camera

8.9 Related Products

- ToF-Flash

© Bluetechnix 2013

- ToF-Flash Adapter
- Debug Adapter

9 Product History

9.1 Version Information

9.1.1 Sentis-ToF-M100

Version	Release date	Firmware Version
X-Grade	May 2013	V0.1.0

Table 9-1: Overview Sentis-ToF-M100 product changes

9.2 Anomalies

Applies to	Date	Description

Table 9-2 – Product anomalies

9.3 Document Revision History

Version	Date	Document Revision
1	2013 06 05	First preliminary of the document
2	2013 07 08	ImageDataFormat field in image header corrected. Some typos corrected. Figure for TopOpeningAngle and BottomOpeningAngle added. Incorrect over- and underexposure data values corrected. Figure 8-2 updated. Offset registers added. Some corrections to register description table.
3	2013 09 03	Wrong UDP protocol header version corrected. High- and low-word mismatch in “Registers for Ethernet” corrected.
4	2013 09 12	Frame header table entries corrected. CmdExecPassword register added. Description for CRC calculation added.

Table 9-3: Revision history

10 Index