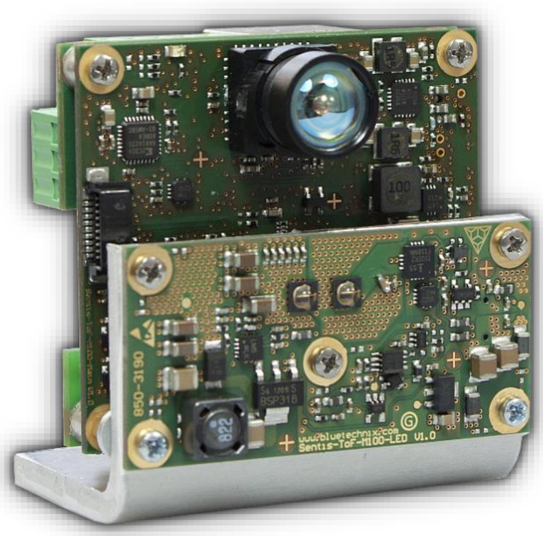
BLUETECHNIX
Embedding Ideas

Sentis-ToF-M100

Software User Manual

Version 10



Bluetechnix

Waidhausenstraße 3/19
A-1140 Vienna
AUSTRIA

office@bluetechnix.com
www.bluetechnix.com

Sentis-ToF-M100 – Software User Manual

Document No.: 900-308 / A

Publication date: February 11, 2016

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

1 General Information ... 6
1.1 Symbols Used ... 6
2 Overview ... 7
3 Interfacing ... 8
3.1 Control Interface ... 8
3.1.1 Ethernet ... 8
3.2 Data Interface ... 17
3.2.1 Ethernet ... 17
3.3 GPIOs ... 20
3.4 Frame trigger signal (TRIG) ... 20
3.5 Modulation Signal Interface ... 20
3.6 Status LED ... 20
4 Camera Features ... 21
4.1 Basic Settings ... 21
4.2 Image Processing Chain ... 21
4.2.1 Image filtering ... 22
4.2.2 Pixel invalidation ... 23
4.3 Camera Data Format ... 24
4.3.1 Distances and Amplitudes ... 24
4.3.2 XYZ Point Cloud ... 25
4.3.3 XYZ Point Cloud and Amplitude ... 26
4.4 Segmentation and Min Max Registers ... 27
4.4.1 Sliding average for Min Max registers. ... 29
4.5 Automatic Exposure Control (AEC) ... 29
4.6 Pixel invalidation ... 29
4.7 Modulation Frequency ... 29
4.8 Frame-rate and Integration Time ... 29
4.9 Manual Frame Trigger ... 30
4.9.1 Hardware Trigger ... 30
4.9.2 Software Trigger ... 30
4.10 Over Temperature Protection ... 30
4.11 Communication Keep Alive (CKA) ... 30
4.12 Alternative illumination interface on the 100pol connector ... 30
4.13 Save Registers ... 32
4.14 Ethernet/IP Settings ... 32

© Bluetechnix 2016

4.14.1 MAC Address ... 32
4.14.2 IP Settings ... 33
4.15 Reset to Factory Default ... 33
4.16 Bootloader and Firmware Update ... 33
4.16.1 Boot Sequence ... 33
4.16.2 Bootloader default settings ... 35
5 Software ... 36
5.1 Demo Application ... 36
5.2 Getting Started Software Development Example ... 36
5.3 Camera Firmware Development KITs ... 36
5.3.1 VDSP++ Development Package ... 36
5.3.2 µCLinux Development Package ... 36
6 Register Description ... 38
6.1.1 General registers ... 38
6.1.2 Registers for Segmentation ... 42
6.1.3 More General Registers ... 43
6.1.4 Registers for Automatic Exposure Control ... 44
6.1.5 Registers for Test Commands ... 44
6.1.6 Registers for Device Update (Only Bootloader) ... 45
6.1.7 Registers for Filter Configuration ... 46
6.1.8 Registers for Ethernet configuration ... 46
6.2 Support ... 50
6.2.1 General Support ... 50
6.3 Software Packages ... 50
6.4 Related Products ... 50
7 Product History ... 51
7.1 Version Information ... 51
7.1.1 Sentis-ToF-M100 ... 51
7.2 Anomalies ... 51
7.3 Document Revision History ... 51
8 Index ... **Error! Bookmark not defined.**

© Bluetechnix GmbH 2016
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.

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.

2 Overview

The document describes the necessary steps and settings to work with the Sentis$^{\text{ToF}}$ -M100 and describes the firmware dependent interfaces.

This document applies to firmware version >=2.1.x.

For a hardware compatibility list please refer to our support site.

Software and documentation

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

3 Interfacing

The Sentis^{ToF} - M100 provides two types of data and control interfaces. Ethernet over the Interface Board or I^{2}C/SSDI available on the 100pol board to board connector of the Sentis Main-Board. This firmware supports the Ethernet interface only. Please contact support@bluetechnix.com for information about I^{2}C/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 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.

3.1 Control Interface

As control interface the Ethernet interface is used.

3.1.1 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 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 boot loader 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 3-1: 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.

3.1.1.1 Register read

Command frame

Addr	Field	Type	Value	Description
0x00	Preamble	Uint16 (high byte 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 (high byte 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 (high byte 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 (high byte first)	0x0	No <Data> present [2]
0x3E	HeaderCrc16	Uint16 (high byte first)	<CRC16 checksum>	Checksum over 60 bytes of Header: 0x02 – 0x3D [1]

Table 3-2: 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
0x00	Preamble	Uint16 (high byte 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 (high byte 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 (high byte 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 (high byte first)	<CRC32 checksum>	Checksum over <Data> 2)
0x3E	HeaderCrc16	Uint16 (high byte first)	<CRC16 checksum>	Checksum over 60 bytes of Header: 0x02 – 0x3D 1)
0x40	Data	byte[] (high byte first)	<result data>	Result: One or more 16 bit values

Table 3-3: 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 3-4: Register read flag description

Result codes

Please refer to Table 3-18.

3.1.1.2 Register write

Command frame

Addr	Field	Type	Value	Description
0x00	Preamble	Uint16 (high byte 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

Addr	Field	Type	Value	Description
0x06	Flags	Uint16	Refer to table	Optional flags
0x08	Length	Uint32 (high byte first)	<# of bytes to write>	The number of bytes to write. Must be a multiple of two and match length of <Data> in bytes. The length divided by two represents the # of registers to write.
0x0C	RegisterAddress	Uint16 (high byte 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 (high byte first)	<CRC32 checksum>	Checksum over <Data> [2]
0x3E	HeaderCrc16	Uint16 (high byte 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 3-5: 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 (high byte 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 (high byte first)	0	No <Data> present
0x0C	RegisterAddress	Uint8 (high byte 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 (high byte first)	0x0	No <Data> present [2]

Addr	Field	Type	Value	Description
0x3E	HeaderCrc16	Uint16 (high byte first)	<CRC16 checksum>	Checksum over 60 bytes of Header: 0x02 – 0x3D [1)]

Table 3-6: 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 3-7: Register write flag description

Result codes

Please refer to Table 3-18.

3.1.1.3 Reset

Command frame

Addr	Field	Type	Value	Description
0x00	Preamble	Uint16 (high byte 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 (high byte 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 (high byte first)	0x0	No <Data> present [2)]
0x3E	HeaderCrc16	Uint16 (high byte first)	<CRC16 checksum>	Checksum over 60 bytes of Header: 0x02 – 0x3D [1)]

Table 3-8: 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 (high byte 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 (high byte first)	0x0	No <Data> present
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 (high byte first)	0x0	No <Data> present [2)]
0x3E	HeaderCrc16	Uint16 (high byte first)	<CRC16 checksum>	Checksum over 60 bytes of Header: 0x02 – 0x3D [1)]

Table 3-9: 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 3-10: Reset flag description

Result codes

Please refer to Table 3-18.

3.1.1.4 Flash Update

Command frame

Addr	Field	Type	Value	Description
0x00	Preamble	Uint16 (high byte 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 boot loader 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

Addr	Field	Type	Value	Description
0x06	Flags	Uint16	Refer to table	Optional flags (remember: DataCrc32 is mandatory, the appropriate flag has been set to 0)
0x08	Length	Uint32 (high byte first)	<# of bytes to write>	The size of the binary file to flash
0x0C	FlashAddress	Uint32 (high byte first)	<Flash Address>	A generic file is flashed to this address. When Flashing a boot loader or application it is ignored.
0x10	Reserved (42 bytes)	Uint8[]		Ignored
0x3A	DataCrc32	Uint32 (high byte first)	<CRC32 checksum>	Checksum over <Data> [2)]
0x3E	HeaderCrc16	Uint16 (high byte 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 3-11: 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.

Response frame

Addr	Field	Type	Value	Description
0x00	Preamble	Uint16 (high byte 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 boot loader 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 (high byte 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 (high byte first)	0x0	No <Data> present [2)]
0x3E	HeaderCrc16	Uint16 (high byte first)	<CRC16 checksum>	Checksum over 60 bytes of Header: 0x02 – 0x3D [1)]

Table 3-12: 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.

Subcommand

SubCommand	Description
0	SPI-flash
1	Parallel flash
2	OTP

Table 3-13: Flash update subcommand description

Flags

Flags	Description
Bit 0	1: Ignore DataCrc32

Table 3-14: Flash update flag description

Result codes

Please refer to Table 3-18.

3.1.1.5 Alive

Command frame

Addr	Field	Type	Value	Description
0x00	Preamble	Uint16 (high byte 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 (high byte first)	0x0	No <Data> present [2)]
0x3E	HeaderCrc16	Uint16 (high byte first)	<CRC16 checksum>	Checksum over 60 bytes of Header: 0x02 – 0x3D [1)]

Table 3-15: 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 (high byte 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 (high byte 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 (high byte first)	0x0	No <Data> present 2)
0x3E	HeaderCrc16	Uint16 (high byte first)	<CRC16 checksum>	Checksum over 60 bytes of Header: 0x02 – 0x3D 1)

Table 3-16: 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 3-17: 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 file size (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 3-18: Result codes

3.2 Data Interface

The data interface is available over internet.

Note

Different data interfaces uses different header formats. The data section (the frame content) is common for all data interfaces. Please refer to 4.3 for information on the available frame data formats.

3.2.1 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.

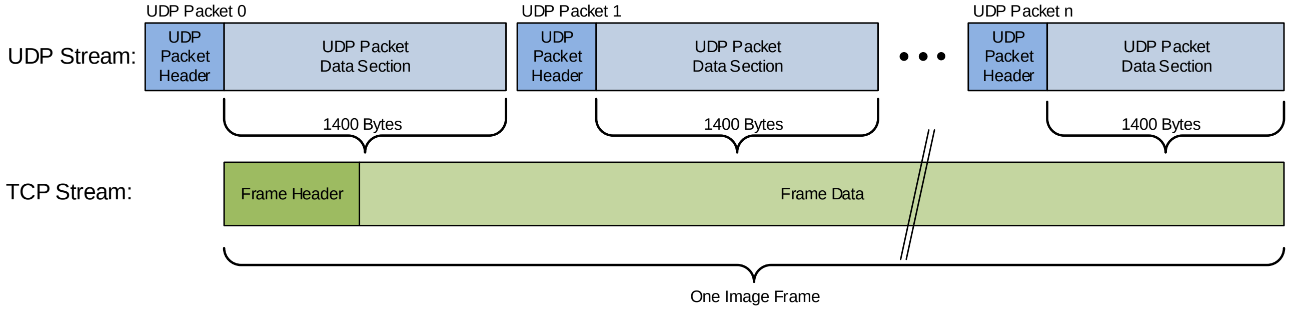


Figure 3-1: UDP and TCP streaming data format

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.

3.2.1.1 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 3-1).

Addr	Field	Type	Value	Description
0x00	Version	Uint16 (high byte first)	0x0001	Protocol version
0x02	FrameCounter	Uint16 (high byte first)		Continuous frame counter. On an overrun it restarts at 0.
0x04	PacketCounter	Uint16 (high byte first)		Actual packet #. The frame data must be recomposed in order of the packet #.
0x06	DataLength	Uint16 (high byte first)		Length of the image data section of the current packet.
0x08	FrameSize	Uint32 (high byte first)		Size of the image data. It may be used to calculate the expected # of packets for a frame.
0x0C	PacketCRC32	Uint16 (high byte 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 3-19: 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.

Flags

Flags	Description
Bit 0	1: Ignore DataCrc32

Table 3-20: UDP packet header flag description

Image data

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 4.3. Below you can find the format of the 64 byte image header.

Addr	Field	Type	Value	Description
0x00	Reserved	Uint16	0xFFFF	
0x02	HeaderVersion	Uint16 (high byte first)	0x0003	Current header version
0x04	ImageWidth	Uint16 (high byte first)		Width of the image in pixels.
0x06	ImageHeight	Uint16 (high byte 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 (high byte first)		The content is the same as in the register *ImageDataFormat*).
0x0C	Timestamp	Uint32 (high byte first)		Timestamp of the actual image in µs
0x10	FrameCounter	Uint16 (high byte 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 (high byte first)		Content of the register *FirmwareInfo*
0x1E	Reserved			
0x3E	CRC16	Uint16 (high byte first)		CRC16 checksum over the header without the first two bytes and the CRC16 checksum itself (addr 0x02 to addr 0x3D) [1)]

Table 3-21: 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.

3.2.1.2 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 3-21 followed by the image content as described in 4.3. 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.

3.3 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 5.3 for more information. Please refer to the Hardware User Manual for information about the GPIO interface connector.

3.4 Frame trigger signal (TRIG)

The frame trigger signal is available on pin #3 of the GPIO connector on the Interface-Board (refer to the Hardware User Manual). 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, the integration time 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*. Please refer to the Hardware User Manual for information about the GPIO interface connector.

3.5 Modulation Signal Interface

The modulation light interface is only available by removing the LED-board and mounting the appropriate Adapter-board (refer to the Hardware User Manual to see how mounting the Adapter-board). 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.

3.6 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 3-22: Status LED meaning

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

4 Camera Features

4.1 Basic Settings

The module comes up according to the reset (default) values as described in the register description section (refer to 6).

4.2 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.

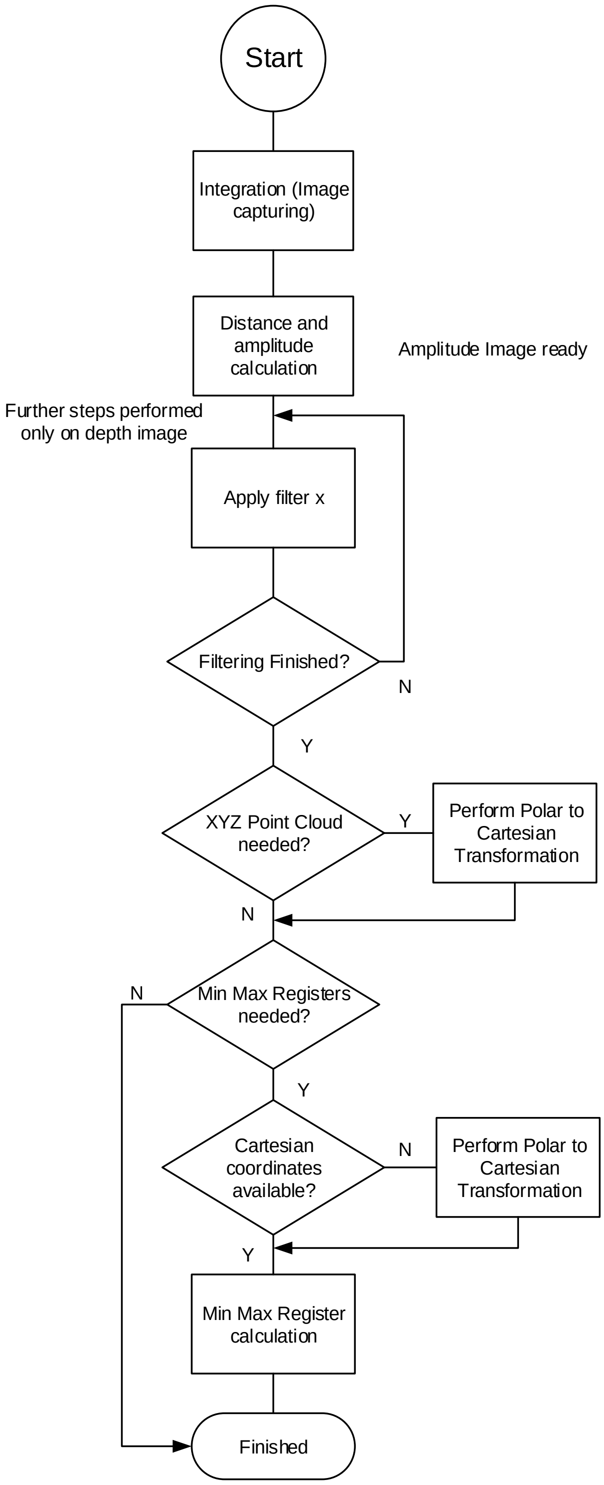

Start
Integration (Image capturing)
Distance and amplitude calculation
Amplitude Image ready
Further steps performed only on depth image
Apply filter x
Filtering Finished?
N
Y
XYZ Point Cloud needed?
Y
Perform Polar to Cartesian Transformation
N
Min Max Registers needed?
N
Y
Cartesian coordinates available?
N
Perform Polar to Cartesian Transformation
Y
Min Max Register calculation
Finished

Figure 4-1: Image processing flow

4.2.1 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.

4.2.1.1 Median Filter

A 3 x 3 Median Filter can be applied.

Register: *FilterMedianConfig*

Only the number of iterations can be configured.

4.2.1.2 Average Filter

Register: *FilterAverageConfig*

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

4.2.1.3 Gauss Filter

Register: *FilterGaussConfig*

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

4.2.1.4 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.

4.2.1.5 Bilateral filter

Register: *FilterBilateralConfig*

The sigma R and sigma D factor (weight for radius and weight for data) can be selected.

! **Note**

The bilateral filter may drastically reduce the frame-rate.

4.2.2 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*.

4.3 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:

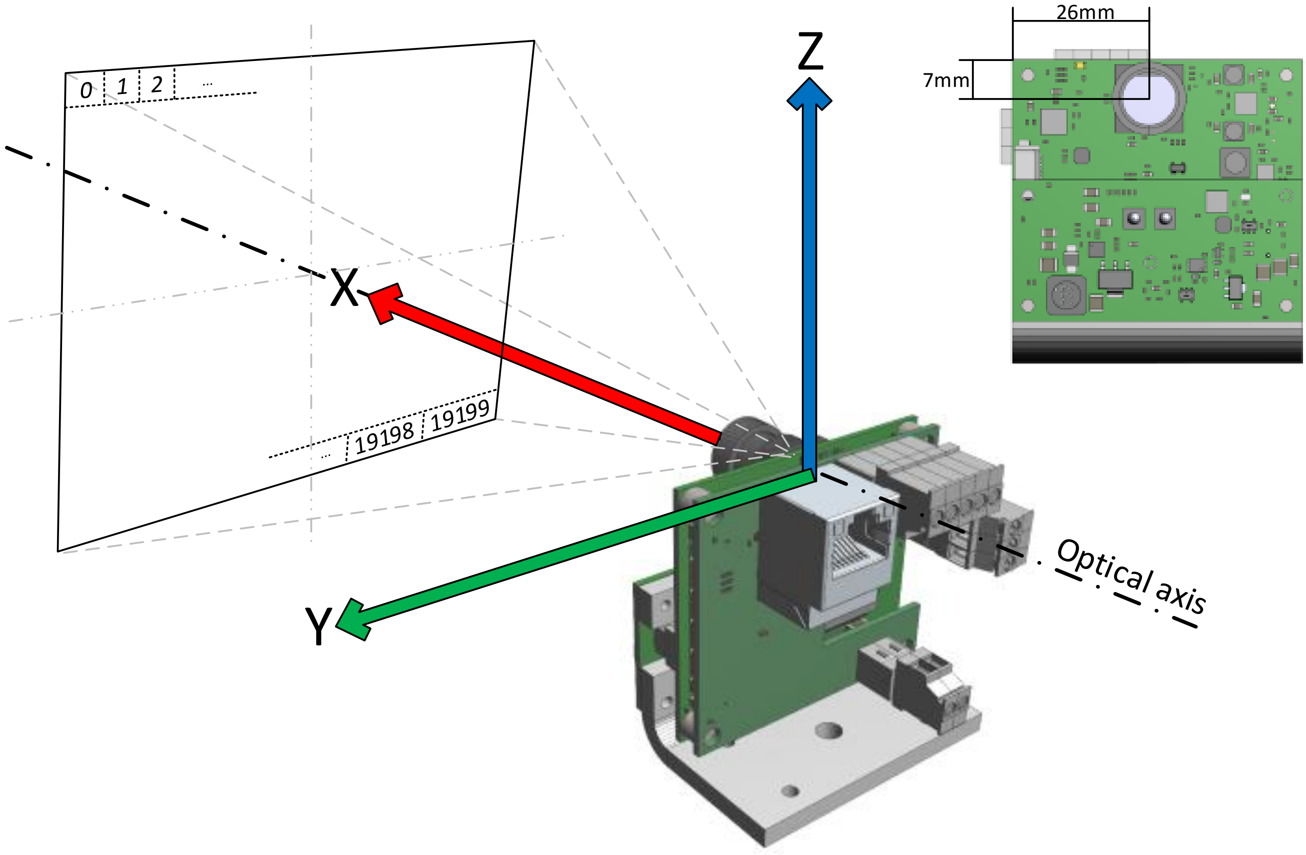


Figure 4-2: Sentis-ToF-M100 Coordinate System

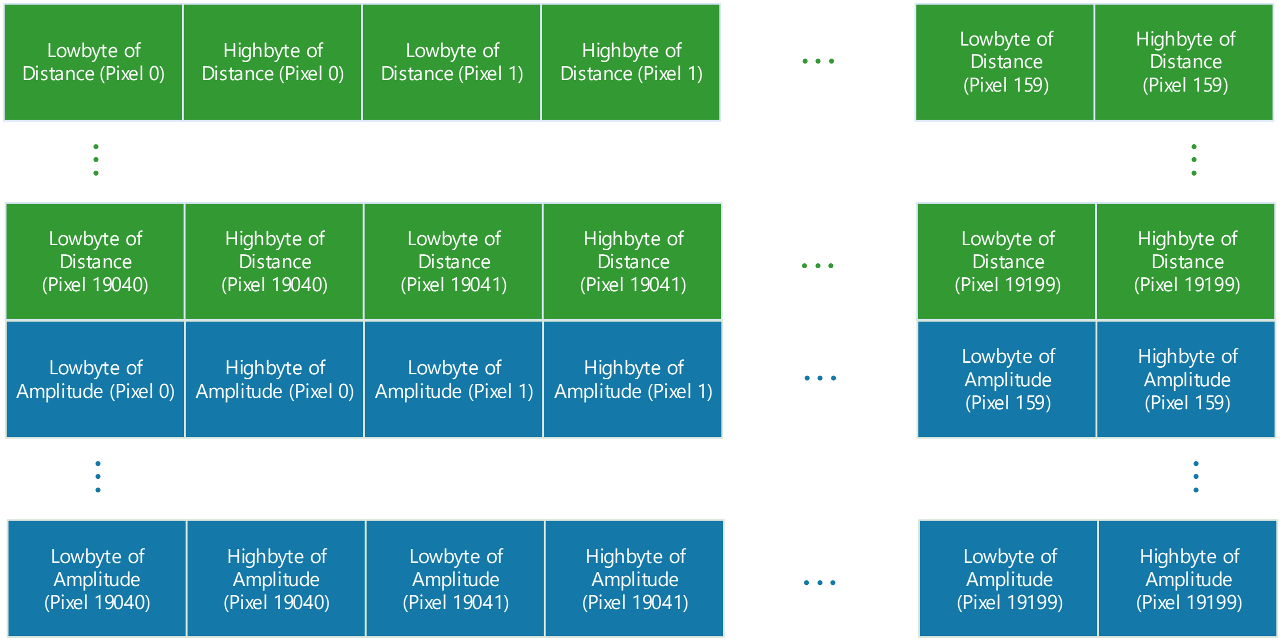
Which image format will be transferred can be selected by the register *ImageDataFormat*. The following section describes each of the supported formats in detail. Only the data section which contains the image data of the transferred frame will be described. For information about the packet format and meta-data please refer to 3.2 according to the interface used.

4.3.1 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



Lowbyte of Distance (Pixel 0)	Highbyte of Distance (Pixel 0)	Lowbyte of Distance (Pixel 1)	Highbyte of Distance (Pixel 1)	…	Lowbyte of Distance (Pixel 159)	Highbyte of Distance (Pixel 159)
⋮						⋮
Lowbyte of Distance (Pixel 19040)	Highbyte of Distance (Pixel 19040)	Lowbyte of Distance (Pixel 19041)	Highbyte of Distance (Pixel 19041)	…	Lowbyte of Distance (Pixel 19199)	Highbyte of Distance (Pixel 19199)
Lowbyte of Amplitude (Pixel 0)	Highbyte of Amplitude (Pixel 0)	Lowbyte of Amplitude (Pixel 1)	Highbyte of Amplitude (Pixel 1)	…	Lowbyte of Amplitude (Pixel 159)	Highbyte of Amplitude (Pixel 159)
⋮						⋮
Lowbyte of Amplitude (Pixel 19040)	Highbyte of Amplitude (Pixel 19040)	Lowbyte of Amplitude (Pixel 19041)	Highbyte of Amplitude (Pixel 19041)	…	Lowbyte of Amplitude (Pixel 19199)	Highbyte of Amplitude (Pixel 19199)

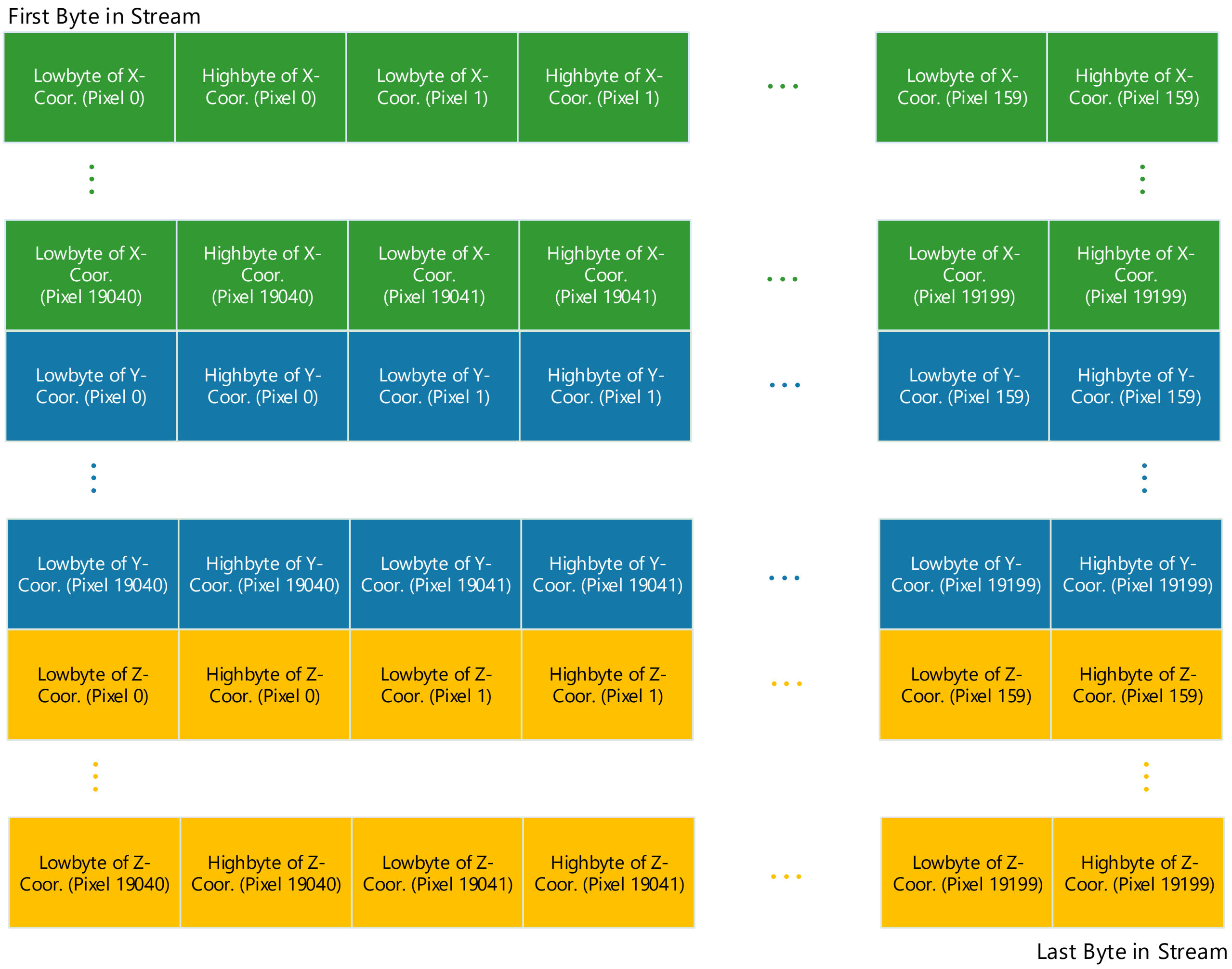
Last Byte in Stream

Figure 4-3: Data-stream of Distance and Amplitude data

4.3.2 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.**



First Byte in Stream

Lowbyte of X-Coor. (Pixel 0)	Highbyte of X-Coor. (Pixel 0)	Lowbyte of X-Coor. (Pixel 1)	Highbyte of X-Coor. (Pixel 1)	…	Lowbyte of X-Coor. (Pixel 159)	Highbyte of X-Coor. (Pixel 159)
⋮					⋮	
Lowbyte of X-Coor. (Pixel 19040)	Highbyte of X-Coor. (Pixel 19040)	Lowbyte of X-Coor. (Pixel 19041)	Highbyte of X-Coor. (Pixel 19041)	…	Lowbyte of X-Coor. (Pixel 19199)	Highbyte of X-Coor. (Pixel 19199)
Lowbyte of Y-Coor. (Pixel 0)	Highbyte of Y-Coor. (Pixel 0)	Lowbyte of Y-Coor. (Pixel 1)	Highbyte of Y-Coor. (Pixel 1)	…	Lowbyte of Y-Coor. (Pixel 159)	Highbyte of Y-Coor. (Pixel 159)
⋮					⋮	
Lowbyte of Y-Coor. (Pixel 19040)	Highbyte of Y-Coor. (Pixel 19040)	Lowbyte of Y-Coor. (Pixel 19041)	Highbyte of Y-Coor. (Pixel 19041)	…	Lowbyte of Y-Coor. (Pixel 19199)	Highbyte of Y-Coor. (Pixel 19199)
Lowbyte of Z-Coor. (Pixel 0)	Highbyte of Z-Coor. (Pixel 0)	Lowbyte of Z-Coor. (Pixel 1)	Highbyte of Z-Coor. (Pixel 1)	…	Lowbyte of Z-Coor. (Pixel 159)	Highbyte of Z-Coor. (Pixel 159)
⋮					⋮	
Lowbyte of Z-Coor. (Pixel 19040)	Highbyte of Z-Coor. (Pixel 19040)	Lowbyte of Z-Coor. (Pixel 19041)	Highbyte of Z-Coor. (Pixel 19041)	…	Lowbyte of Z-Coor. (Pixel 19199)	Highbyte of Z-Coor. (Pixel 19199)

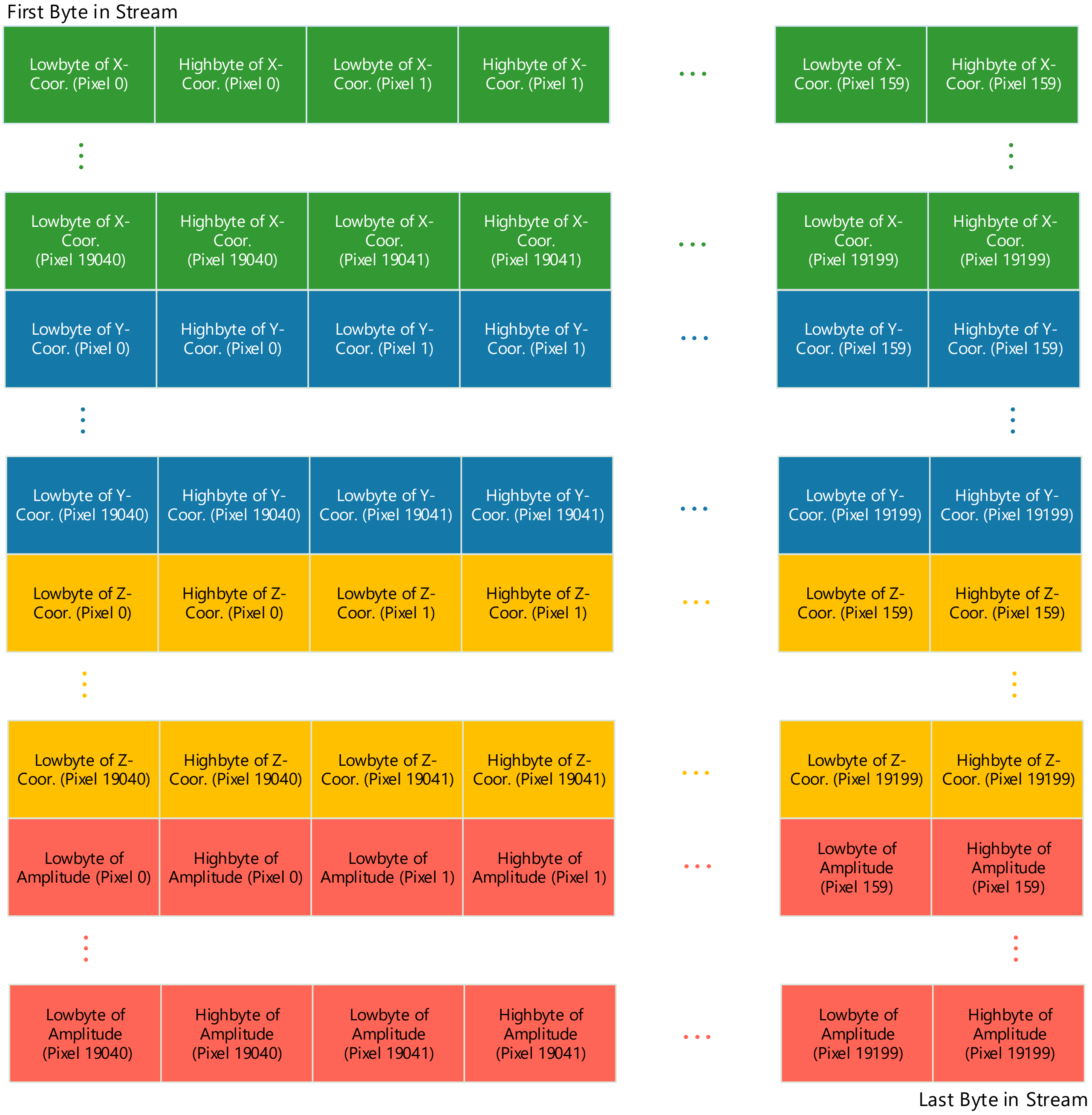
Last Byte in Stream

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

4.3.3 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.**



First Byte in Stream

Lowbyte of X-Coor. (Pixel 0)	Highbyte of X-Coor. (Pixel 0)	Lowbyte of X-Coor. (Pixel 1)	Highbyte of X-Coor. (Pixel 1)	…	Lowbyte of X-Coor. (Pixel 159)	Highbyte of X-Coor. (Pixel 159)
⋮					⋮	
Lowbyte of X-Coor. (Pixel 19040)	Highbyte of X-Coor. (Pixel 19040)	Lowbyte of X-Coor. (Pixel 19041)	Highbyte of X-Coor. (Pixel 19041)	…	Lowbyte of X-Coor. (Pixel 19199)	Highbyte of X-Coor. (Pixel 19199)
Lowbyte of Y-Coor. (Pixel 0)	Highbyte of Y-Coor. (Pixel 0)	Lowbyte of Y-Coor. (Pixel 1)	Highbyte of Y-Coor. (Pixel 1)	…	Lowbyte of Y-Coor. (Pixel 159)	Highbyte of Y-Coor. (Pixel 159)
⋮					⋮	
Lowbyte of Y-Coor. (Pixel 19040)	Highbyte of Y-Coor. (Pixel 19040)	Lowbyte of Y-Coor. (Pixel 19041)	Highbyte of Y-Coor. (Pixel 19041)	…	Lowbyte of Y-Coor. (Pixel 19199)	Highbyte of Y-Coor. (Pixel 19199)
Lowbyte of Z-Coor. (Pixel 0)	Highbyte of Z-Coor. (Pixel 0)	Lowbyte of Z-Coor. (Pixel 1)	Highbyte of Z-Coor. (Pixel 1)	…	Lowbyte of Z-Coor. (Pixel 159)	Highbyte of Z-Coor. (Pixel 159)
⋮					⋮	
Lowbyte of Z-Coor. (Pixel 19040)	Highbyte of Z-Coor. (Pixel 19040)	Lowbyte of Z-Coor. (Pixel 19041)	Highbyte of Z-Coor. (Pixel 19041)	…	Lowbyte of Z-Coor. (Pixel 19199)	Highbyte of Z-Coor. (Pixel 19199)
Lowbyte of Amplitude (Pixel 0)	Highbyte of Amplitude (Pixel 0)	Lowbyte of Amplitude (Pixel 1)	Highbyte of Amplitude (Pixel 1)	…	Lowbyte of Amplitude (Pixel 159)	Highbyte of Amplitude (Pixel 159)
⋮					⋮	
Lowbyte of Amplitude (Pixel 19040)	Highbyte of Amplitude (Pixel 19040)	Lowbyte of Amplitude (Pixel 19041)	Highbyte of Amplitude (Pixel 19041)	…	Lowbyte of Amplitude (Pixel 19199)	Highbyte of Amplitude (Pixel 19199)

Last Byte in Stream

Figure 4-5: Data-stream of XYZ Point Cloud and Amplitude

4.4 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 4-6. The vertical field of view can be configured either by the top and bottom opening angle (*TopOpeningAngle* and *BottomOpeningAngle*) (see Figure 4-7) 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.

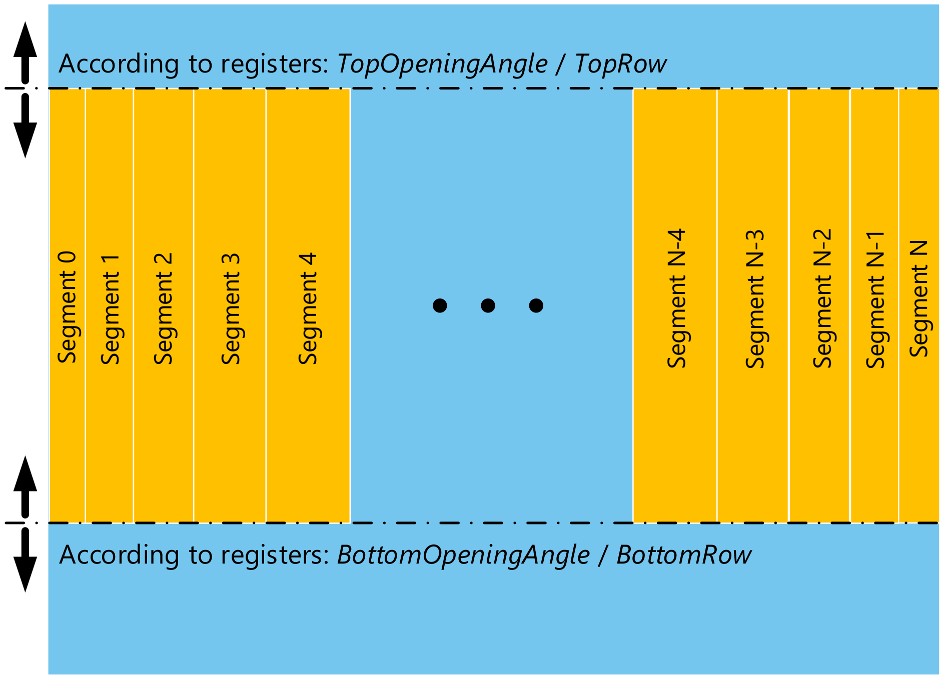

According to registers: *TopOpeningAngle / TopRow*
Segment 0
Segment 1
Segment 2
Segment 3
Segment 4
• • •
Segment N-4
Segment N-3
Segment N-2
Segment N-1
Segment N
According to registers: *BottomOpeningAngle / BottomRow*

Figure 4-6: Segmentation of image

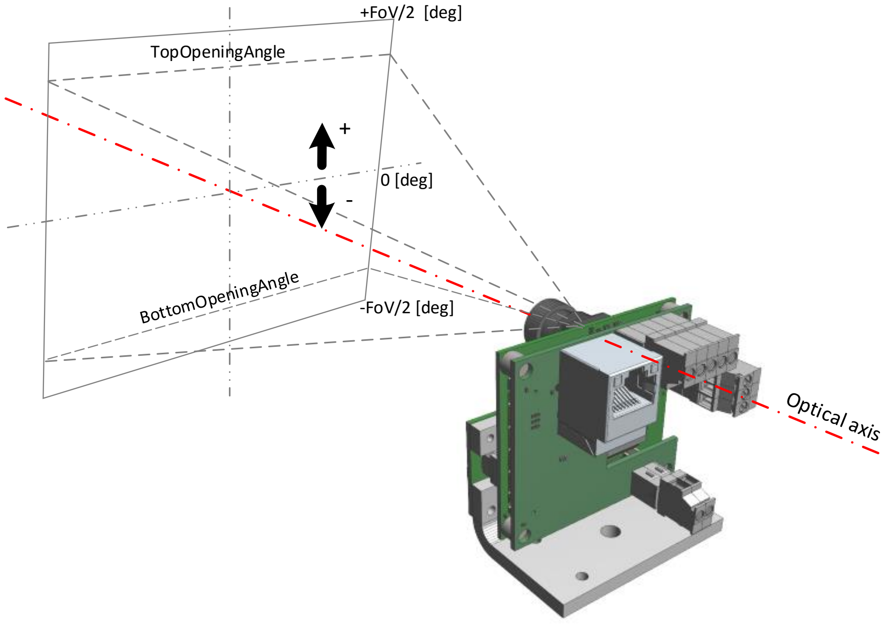


Figure 4-7: Meaning of top and bottom opening angle

4.4.1 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*.

4.5 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*. There are some registers for the configuration of the AEC.

4.6 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*.

4.7 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.

4.8 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! Refer to the Hardware User Manual for more information.

Note

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

4.9 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]. Register *NOF_Sequ* (0x0120) controls the number of frames captured after a hardware or software trigger event (default: 1).

4.9.1 Hardware Trigger

The camera provides an external connector where a hardware trigger can be applied. Please refer to 3.4 for more information.

4.9.2 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*.

4.10 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.

4.11 Communication Keep Alive (CKA)

The communication keep alive feature should improve a stable operation in environments where a high availability of the camera services will be needed.

If the CKA feature has been enabled by writing a value >0 to the register *CommKeepAliveTimeout* (0x004e), the host computer must periodically write the reset value 0xca82 to the register *CommKeepAliveReset* (0x004f). If the reset value will not be written within the programmed timeout the device reboots.

The timeout can be set by the register *CommKeepAliveTimeout* by writing the timeout value in seconds. This value will also be saved, if the register map will be saved in flash. But after a reboot the timeout check starts only after the first write of the reset value in register *CommKeepAliveReset*.

4.12 Alternative illumination interface on the 100pol connector

By setting Bit[12] of the *Mode1* (0x0019) register the alternative illumination interface can be enabled. The internal LED-board interface will be disabled. Changes to this bit takes only effect after a reboot or power cycle.

4.13 Temperature Sensor Manager

Registers 0x290 – 0x02CF are used to configure the Sentis' temperature sensors. Please refer to 6.1.10 for more information about the individual registers.
The default configuration is intended for standard operation with the Sentis Ledboard. Sensor configuration depends on the Sentis' hardware configuration and can be modified to conform to the hardware setup in use.

Example Configuration: Sentis M100 and Sentis Ledboard

Addr (hex)	Register Name	Value (hex)	Description
0298	TempDevBusConfig2	0081	Bit[0-6]: … Address … 0x01 Bit[7-13]: … Device Type 1 … OWIRE LIM <V2.1 (or Sentis Ledboard) Bit[14-15]: … Bus Nr. 0 … Internal I2C/OWIRE Bus
0299	TempDevCompConfig2	3201	Bit[0-7]: … Compensation Group 1 … Illumination
029A	TempDevConfig2		Not supported by Sentis Ledboard
02CC	TempDevBusConfig15	0198	Bit[0-6]: … Address … 0x18 Bit[7-13]: … Device Type 3 … I2C_ADT74 Bit[14-15]: … Bus Nr. 0 … Internal I2C/OWIRE Bus
02CD	TempDevCompConfig15	6402	Bit[0-7]: … Compensation Group 2 … Sensor
02CE	TempDevConfig15	0000	Not used

Example Configuration: Sentis M100 and 2x ToF Flasher V3.0.0

Addr (hex)	Register Name	Value (hex)	Description
029C	TempDevBusConfig3	0101	Bit[0-6]: … Address … 0x01 Bit[7-13]: … Device Type 2 … OWIRE LIM V2.1 Bit[14-15]: … Bus Nr. 0 … Internal I2C/OWIRE Bus
029D	TempDevCompConfig3	3201	Bit[0-7]: … Compensation Group 1 … Illumination Bit[8-15]: … All 4 LIMs are equally averaged for temperature compensation
029E	TempDevConfig3	0007	PEN enabled, both segments enabled
029C	TempDevBusConfig4	0102	Bit[0-6]: … Address … 0x02 Bit[7-13]: … Device Type 2 … OWIRE LIM V2.1 Bit[14-15]: … Bus Nr. 0 … Internal I2C/OWIRE Bus
029D	TempDevCompConfig4	3201	Bit[0-7]: … Compensation Group 1 … Illumination Bit[8-15]: … All 4 LIMs are equally averaged for temperature compensation
029E	TempDevConfig4	0007	PEN enabled, both segments enabled
029C	TempDevBusConfig5	0103	Bit[0-6]: … Address … 0x03 Bit[7-13]: … Device Type 2 … OWIRE LIM V2.1 Bit[14-15]: … Bus Nr. 0 … Internal I2C/OWIRE Bus
029D	TempDevCompConfig5	3201	Bit[0-7]: … Compensation Group 1 … Illumination Bit[8-15]: … All 4 LIMs are equally averaged for temperature compensation
029E	TempDevConfig5	0007	PEN enabled, both segments enabled

Addr (hex)	Register Name	Value (hex)	Description
029C	TempDevBusConfig6	0104	Bit[0-6]: … Address … 0x04 Bit[7-13]: … Device Type 2 … OWIRE LIM V2.1 Bit[14-15]: … Bus Nr. 0 … Internal I2C/OWIRE Bus
029D	TempDevCompConfig6	3201	Bit[0-7]: … Compensation Group 1 … Illumination Bit[8-15]: … All 4 LIMs are equally averaged for temperature compensation
029E	TempDevConfig6	0007	PEN enabled, both segments enabled
02CC	TempDevBusConfig15	0198	Bit[0-6]: … Address … 0x18 Bit[7-13]: … Device Type 3 … I2C_ADT74 Bit[14-15]: … Bus Nr. 0 … Internal I2C/OWIRE Bus
02CD	TempDevCompConfig15	6402	Bit[0-7]: … Compensation Group 2 … Sensor
02CE	TempDevConfig15	0000	Not used

Example Configuration: Sentis M100 and LIM V2.1 on 100pol connector (connected via I2C)

Addr (hex)	Register Name	Value (hex)	Description
0290	TempDevBusConfig0	4219	Bit[0-6]: … Address … 0x19 Bit[7-13]: … Device Type 4 … I2C_LIM Bit[14-15]: … Bus Nr. 1 … I2C/OWIRE over 100pol connector
0291	TempDevCompConfig0	3201	Bit[0-7]: … Compensation Group 1 … Illumination
0292	TempDevConfig0	0007	PEN enabled, both segments enabled
02CC	TempDevBusConfig15	0198	Bit[0-6]: … Address … 0x18 Bit[7-13]: … Device Type 3 … I2C_ADT74 Bit[14-15]: … Bus Nr. 0 … Internal I2C/OWIRE Bus
02CD	TempDevCompConfig15	6402	Bit[0-7]: … Compensation Group 2 … Sensor
02CE	TempDevConfig15	0000	Not used

4.14 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.

4.15 Ethernet/IP Settings

4.15.1 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.

4.15.2 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.

4.16 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*.

4.17 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^{2}C. The I^{2}C 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

4.17.1 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.

© Bluetechnix 2016

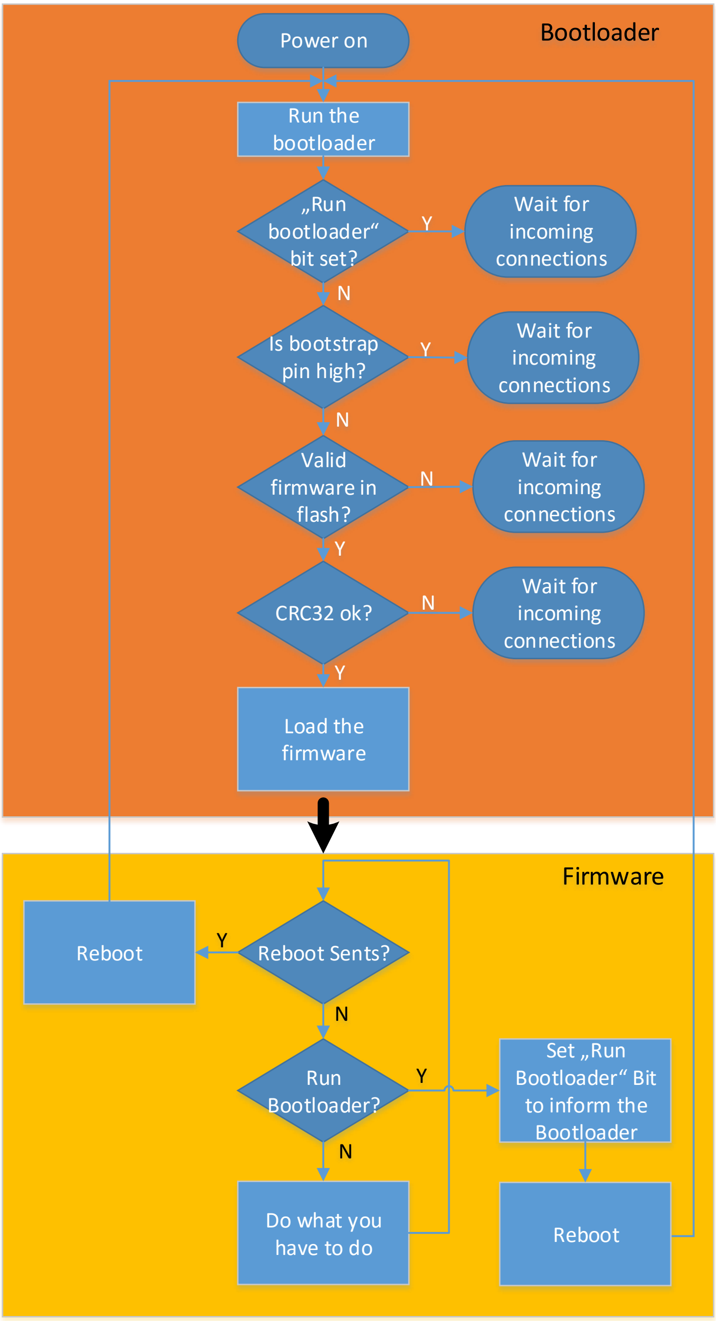

Bootloader
Power on
Run the bootloader
„Run bootloader" bit set?
Y
Wait for incoming connections
N
Is bootstrap pin high?
Y
Wait for incoming connections
N
Valid firmware in flash?
N
Wait for incoming connections
Y
CRC32 ok?
N
Wait for incoming connections
Y
Load the firmware
Firmware
Reboot
Y
Reboot Sents?
N
Run Bootloader?
Y
Set „Run Bootloader" Bit to inform the Bootloader
N
Do what you have to do
Reboot

Figure 4-8: Boot sequence

4.17.1.1 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 to the Hardware User Manual for more information) 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.

4.17.2 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!

5 Software

5.1 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

5.2 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

5.3 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.

5.3.1 VDSP++ Development Package

VDSP++ is an Integrated Development 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

5.3.2 µCLinux Development Package

Bluetechnix provides a µCLinux Board Support Package (BSP) for the Sentis-ToF-M100 which can be used by developers to develop their 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

6 Register Description

!

Note

Some critical registers are password protected. To enable the functionality a specific value must be written to the CmdEnablePasswd register prior to enable the functionality. This should prevent from unattended enabling of certain functions.

6.1.1 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[2]: 0..Full On, 1..Sleep Mode Bit[4]: 1..Manual Trigger Bit[6]: 1..Clear status register Bit[8]: 1..Start boot loader (Start boot loader requires writing 0x5e6b into register CmdEnablePasswd (0x0022)
0003	Status	0040	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[7]: 1..Regmap recovered after watchdog reset Bit [12]: Frame Faulty
0004	ImageDataFormat	0000	R/W	Bit[3:6]: 0… 2 bytes depth-data / 2 bytes amp-data 3… X/Y/Z coordinates (2 bytes in signed format for each coordinate) 4… X/Y/Z coordinates and amp-data (2 bytes in signed format for each coordinate,2 bytes unsigned for the amp value) 5… X/Y/Z coordinates and OHD color Information for each pixel (2 bytes in signed format for XYZ, 2 bytes color information in RGB565) 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° 9… depth-data and X/Y/Z coordinates (2 byte unsigned for the depth value , 2 byte in signed format for each coordinate) 10… X coordinate and amp-data (2 bytes in signed format, 2 bytes unsigned for amp-data)

© Bluetechnix 2016

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
				11… 4 channels Test mode: arithmetic functions with coordinates as input (2 bytes ascending index; 2 bytes constant 0xbeef; 2 bytes ascending squared index; 2 bytes constant 0x0000) 12… 2 bytes depth-data
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	07D0	R/W	Modulation frequency index: 0.. 5 MHz 1.. 7,5 MHz 2.. 10 MHz 3.. 15 MHz 4.. 20 MHz 5.. 25 MHz 6.. 30 MHz Or a modulation Frequency in 10 [kHz] (may take a while to apply and no calibration data will be available)
000A	Framerate	0028	R/W	Frame rate [Hz]
000B	HardwareConfiguration	005A	R/W	Lens opening angle identifier
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)
000F	CalibrationCommand	0000	R/W	Bit[0:7]: Cmd code 19.. Calibrate DistOffset of the current frequency
0010	ConfidenceThresLow	012C	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 Bit[12]: 1..Use illumination interface on 100pol connector. Applies only after a reset or power cycle. Be aware to save the register map in flash before reboot/power cycle.
001A	CalculationTime		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		R	Temperature of LED-Board in 0,01[°C] (FFFF: Sensor not available).

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
001C	MainboardTemp		R	Temperature of Main-Board in 0,01[°C] (FFFF: Sensor not available).
001D	LinearizationAmplitude	0190	R/W	Amplitude for Linearization Function [float value x 10000]
001E	LinearizationPhaseShift	1B58	R/W	Amplitude for Linearization Function [float value x 10000]
001F	FrameTime		R	Time between the last two frames. In 0,1[ms]
0020	RealWorldXcoordinate	0000	R/W	Distance to the calibration target [mm].
0021	CalibrationExtentended	0000	R	Bit[0-7]: Status/error 0.. Idle 20.. Erasing flash 161.. Operation done 255.. Generic error 254.. NVM error 253.. Dist calib calc error 252.. Out of memory 251.. Lens calib pattern not recognized 250.. Wrong version of data in NVM 249.. Led board failure 248.. Invalid modulation frequency 247.. Hardware fault Bit[10]: 1..Error occurred Bit[11]: 1..No FPN Calibration data in NVM Bit[12]: 1..No FPPN Calibration data in NVM Bit[13]: 1..No Dist Calibration data in NVM Bit[14]: 1..No Lens Calibration data in NVM Bit[15]: 1..Temperature compensation error
0022	CmdEnablePasswd	0000	R/W	Set a password for critical operations: 0x4877: Register map flash operations (register CmdExec 0x0033) 0x5e6b: Test commands (register TestConfig 0x01c0)
0024	MaxLedTemp	2328	R/W	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	4000	R	Bit[14-15]: Firmware Load Counter. This counter is reset by the firmware. It counts the boot trials. In boot loader mode it is used to detect a firmware load problem
002D	TempCompGradient		R/W	Factor 'c' of the temperature compensation function: y [mm] = a/1000 * x^3 + b/10000 * x^2 + c/10000 * x + u
002E	ApplicationVersion		R	See "FirmwareInfo (0x0008)" for bit description, in boot loader mode this register contains the firmware info of the flashed application
002F	DistCalibGradient	4000	R/W	Gradient of dist value, interpreted as fixed comma shifted by 14 binary digits

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
0030	TempCompGradient2		R/W	Factor 'b' of the temperature compensation function: y [mm] = a/1000 * x³ + b/10000 * x² + c/10000 * x + u
0033	CmdExec	0000	R/W	Initiate an operation (after setting the password in register 0x0022): 0xC2AE.. Clear RegMap in flash 0x9E20.. Read RegMap from flash 0x909A.. Read factory RegMap 0xDD9E.. Save RegMap in flash
0034	CmdExecResult	0000	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 low byte 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 time zone (stored in OTP flash)
003C	TempCompGradient3		R/W	Factor 'a' of the temperature compensation function: y [mm] = a/1000 * x³ + b/10000 * x² + c/10000 * x + u
003D	BuildYearMonth		R	Build date/time Bit[14-4]: Year Bit[3-0]: Month
003E	BuildDayHour		R	Build date/time Bit[9-5]: Day Bit[4-0]: Hour
003F	BuildMinuteSecond		R	Build date/time Bit[11-6]: Minute Bit[5-0]: Second
0040	UpTimeLow		R	Lower 16 bit of uptime in [s]
0041	UpTimeHigh		R	Higher 16 bit of uptime in [s]
0042	AkfPlausabilityCheckAmpLimit	0032	R/W	Limit for the akf plausibility check
004e	CommKeepAliveTimeout	0000	R/W	Timeout in seconds for communication keep alive
004f	CommKeepAliveReset	0000	R/W	Write 0xca82 to reset the communication keep alive watchdog

Table 6-1: 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.

6.1.2 Registers for Segmentation

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
0050	SegmentCount	0020	R/W	Nr. Of Segments/Macro cells
0051	TopOpeningAngle	1)	R/W	Top Opening Angle in 0,01[°] [4]
0052	TopRow	1)	R/W	Bit[0-7]: Top Row (0-119)
0053	BottomOpeningAngle	1)	R/W	Top Opening Angle in 0,01[°] [4]
0054	BottomRow	1)	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
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]

Table 6-2: 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 segment 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.

Note 5): This value varies from unit to unit.

6.1.3 More General Registers

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
00C1	DistOffset0	1)	R/W	An offset for distance values when operating at modulation frequency with index 0
00C2	DistOffset1	1)	R/W	An offset for distance values when operating at modulation frequency with index 1
00C3	DistOffset2	1)	R/W	An offset for distance values when operating at modulation frequency with index 2
00C4	DistOffset3	1)	R/W	An offset for distance values when operating at modulation frequency with index 3
00C5	DistOffset4	1)	R/W	An offset for distance values when operating at modulation frequency with index 4
00C6	DistOffset5	1)	R/W	An offset for distance values when operating at modulation frequency with index 5
00C7	DistOffset6	1)	R/W	An offset for distance values when operating at modulation frequency with index 6
00C8	DistOffset7	1)	R/W	An offset for distance values when operating at modulation frequency with index 7
00C9	DistOffset8	1)	R/W	An offset for distance values when operating at modulation frequency with index 8
00CA	DistOffset9	1)	R/W	An offset for distance values when operating at modulation frequency with index 9
0100	UserDefined0	0	R/W	For any purpose
0101	UserDefined1	0	R/W	For any purpose
0102	UserDefined2	0	R/W	For any purpose
0103	UserDefined3	0	R/W	For any purpose
0104	UserDefined4	0	R/W	For any purpose
0105	UserDefined5	0	R/W	For any purpose
0106	UserDefined6	0	R/W	For any purpose
0107	UserDefined7	0	R/W	For any purpose
0108	UserDefined8	0	R/W	For any purpose
0109	UserDefined9	0	R/W	For any purpose

Table 6-3: General registers

Note 5): This value varies from unit to unit.

6.1.4 Registers for Sequencing

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
0120	NofSequ	1	R/W	Number of frames that are recorded after manual trigger event

6.1.5 Registers for Automatic Exposure Control

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
01A9	AecAvgWeight0	4444	R/W	Bit[15-12]: Weight for average, area 1 Bit[11-8]: Weight for average, area 2 Bit[7-4]: Weight for average, area 3 Bit[3-0]: Weight for average, area 4
01AA	AecAvgWeight1	44CC	R/W	Bit[15-12]: Weight for average, area 5 Bit[11-8]: Weight for average, area 6 Bit[7-4]: Weight for average, area 7 Bit[3-0]: Weight for average, area 8
01AB	AecAvgWeight2	C44C	R/W	Bit[15-12]: Weight for average, area 9 Bit[11-8]: Weight for average, area 10 Bit[7-4]: Weight for average, area 11 Bit[3-0]: Weight for average, area 12
01AC	AecAvgWeight3	FC44	R/W	Bit[15-12]: Weight for average, area 13 Bit[11-8]: Weight for average, area 14 Bit[7-4]: Weight for average, area 15 Bit[3-0]: Weight for average, area 16
01AD	AecAvgWeight4	CCC4	R/W	Bit[15-12]: Weight for average, area 17 Bit[11-8]: Weight for average, area 18 Bit[7-4]: Weight for average, area 19 Bit[3-0]: Weight for average, area 20
01AE	AecAvgWeight5	4444	R/W	Bit[15-12]: Weight for average, area 21 Bit[11-8]: Weight for average, area 22 Bit[7-4]: Weight for average, area 23 Bit[3-0]: Weight for average, area 24
01AF	AecAvgWeight6	4000	R/W	Bit[15-12]: Weight for average, area 25
01B0	AecAmpTarget	02BC	R/W	Auto exposure target amplitude value to which the controller is controlling to
01B1	AecTintStepMax	0021	R/W	Auto exposure maximum change of integration time percentage. The relative change of the integration time will be lower than this percentage
01B2	AecTintMax	2710	R/W	Auto exposure maximum integration time the controller calculates
01B3	AecKp	0028	R/W	Proportional part of the auto exposure controller in percent
01B4	AecKi	000F	R/W	Integral part of the auto exposure controller in percent
01B5	AecKd	0000	R/W	Differential part of the auto exposure controller in percent

Table 6-4: Registers for automatic exposure control

6.1.6 Registers for Test Commands

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
01C0	TestConfig	0000	R/W	Bit[0]: 1…CPLD LED Security Feature

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
				Bit[1]: 1… Watchdog Test

Table 6-5: Registers for test commands

6.1.7 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 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 6-6: Registers for device update

6.1.8 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[3]: 1… enable Bilateral Filter Bit[4]: 1… enable Sliding Average Bit[5]: 1… enable fast lookup table mode 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[11]: 1… enable temperature compensation Bit[12]: 1… enable scaling via register DistCalibGradient (0x002F) Bit[13]: 1… enable offsets via registers DistCalibOffsetX (0x00C1 onwards) Bit[14]: 1… enable akf plausibility check (affected pixel have a distance of 1)
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
01E4	FilterBilateralConfig	1082	R/W	Bit[0-5]: Sigma R (factor for weighing for radius, max: 6) Bit[6-11]: Sigma D (factor for weighting for data, max: 6) Bit[12-15]: Nr of iterations
01E5	FilterSlafConfig	0005	R/W	Bit[0-7]: …Window size

Table 6-7: Register for filter configuration

Note 1): [Bit0 – 9]: Distance to the Object, Bit [10-15]: Refer to 20120709_pmd_vertical_lookup_table.csv" for more information about the coding of the height of the object.

6.1.9 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

Addr (hex)	**Register Name**	**Default Value (hex)**	**R/W**	**Description**
				Bit[2]: 1.. Ignore CRC for UDP streaming
0241	Eth0Mac2	ACDE	R/W	Low byte and byte 1 of MAC address (default value differs in factory config)
0242	Eth0Mac1	4801	R/W	Byte 2 and byte 3 of MAC address (default value differs in factory config)
0243	Eth0Mac0	0203	R/W	Byte 4 and high byte of MAC address (default value differs in factory config)
0244	Eth0Ip0	000A	R/W	Low word of IP address
0245	Eth0Ip1	C0A8	R/W	High word of IP address
0246	Eth0Snm0	FF00	R/W	Low word of subnet mask
0247	Eth0Snm1	FFFF	R/W	High word of subnet mask
0248	Eth0Gateway0	0000	R/W	Low word of gateway
0249	Eth0Gateway1	0000	R/W	High word of gateway
024A	Eth0TcpStreamPort	2710	R/W	Port for TCP streaming
024B	Eth0TcpConfigPort	2711	R/W	Port for TCP control interface
024C	Eth0UdpStreamIp0	0001	R/W	Low word of IP address for UDP stream
024D	Eth0UdpStreamIp1	E000	R/W	High word of IP address for UDP stream
024E	Eth0UdpStreamPort	2712	R/W	Port for UDP streaming
024F	Eth0UdpStreamStride	1	R/W	Stride for UDP stream 0: Don't stream any frame 1: Stream every frame x: Stream every xth frame

Table 6-8: Registers for Ethernet configuration

6.1.10 Registers for Temperature Management

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
028F	TempDevMaxIllTempOffset	000f	R/W	Bit[0-15]: … Temperature offset in °C for illumination temperature devices. If supported by the device the temperature threshold for security shutdown is set by using register MaxLedTemp (0x0024) added by this offset, in 0,01[°C]
0290	TempDevBusConfig0	0000	R/W	Bit[0-6]: … Address Bit[7-13]: … Device Type 0 … unconfigured 1 … OWIRE LIM <V2.1 (or Sentis Ledboard) 2 … OWIRE LIM V2.1 3 … I2C_ADT74 4 … I2C_LIM Bit[14-15]: … Bus Nr. 0 … Internal I2C/OWIRE Bus 1 ... I2C/OWIRE over 100pol connector
0291	TempDevCompConfig0	0000	R/W	Bit[0-7]: … Compensation Group 0 … unconfigured 1 … Illumination 2 … Sensor 4 … None Bit[8-15]: … this factor (register value divided by 100) is applied to the temperature of this

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
				sensor. The sum of factors of one compensation group has to be 100.
0292	TempDevConfig0	0000	R/W	Device specific configuration of temperature device 0 LIM: Bit[0]: 1 … enable PEN Bit[1]: 1 … enable LED Segment 1 Bit[2]: 1 … enable LED Segment 2 Bit[5]: 1 … Fan manually on, 0 … Fan auto mode Bit[15]: is set by the firmware at start of device configuration and cleared as soon as the configuration was successful.
0293	TempDevSysStatus0	0000	R/W	System Status of temperature device 0 Bit[0]: … Device specific status has error bits set Bit[1-11]: … unused Bit[13]: … Temperature Plausibility Error Bit[14]: … Sensor IO Error Bit[15]: … Initialization Error
0294	TempDevBusConfig1	0000	R/W	See register TempDevBusConfig0
0295	TempDevCompConfig1	0000	R/W	See register TempDevCompConfig0
0296	TempDevConfig1	0000	R/W	See register TempDevConfig0
0297	TempDevSysStatus1	0000	R/W	See register TempDevSysStatus0
0298	TempDevBusConfig2	0000	R/W	See register TempDevBusConfig0
0299	TempDevCompConfig2	0000	R/W	See register TempDevCompConfig0
029A	TempDevConfig2	0000	R/W	See register TempDevConfig0
029B	TempDevSysStatus2	0000	R/W	See register TempDevSysStatus0
029C	TempDevBusConfig3	0000	R/W	See register TempDevBusConfig0
029D	TempDevCompConfig3	0000	R/W	See register TempDevCompConfig0
029E	TempDevConfig3	0000	R/W	See register TempDevConfig0
029F	TempDevSysStatus3	0000	R/W	See register TempDevSysStatus0
02A0	TempDevBusConfig4	0000	R/W	See register TempDevBusConfig0
02A1	TempDevCompConfig4	0000	R/W	See register TempDevCompConfig0
02A2	TempDevConfig4	0000	R/W	See register TempDevConfig0
02A3	TempDevSysStatus4	0000	R/W	See register TempDevSysStatus0
02A4	TempDevBusConfig5	0000	R/W	See register TempDevBusConfig0
02A5	TempDevCompConfig5	0000	R/W	See register TempDevCompConfig0
02A6	TempDevConfig5	0000	R/W	See register TempDevConfig0
02A7	TempDevSysStatus5	0000	R/W	See register TempDevSysStatus0
02A8	TempDevBusConfig6	0000	R/W	See register TempDevBusConfig0
02A9	TempDevCompConfig6	0000	R/W	See register TempDevCompConfig0
02AA	TempDevConfig6	0000	R/W	See register TempDevConfig0
02AB	TempDevSysStatus6	0000	R/W	See register TempDevSysStatus0
02AC	TempDevBusConfig7	0000	R/W	See register TempDevBusConfig0
02AD	TempDevCompConfig7	0000	R/W	See register TempDevCompConfig0
02AE	TempDevConfig7	0000	R/W	See register TempDevConfig0
02AF	TempDevSysStatus7	0000	R/W	See register TempDevSysStatus0
02B0	TempDevBusConfig8	0000	R/W	See register TempDevBusConfig0
02B1	TempDevCompConfig8	0000	R/W	See register TempDevCompConfig0
02B2	TempDevConfig8	0000	R/W	See register TempDevConfig0
02B3	TempDevSysStatus8	0000	R/W	See register TempDevSysStatus0

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
02B4	TempDevBusConfig9	0000	R/W	See register TempDevBusConfig0
02B5	TempDevCompConfig9	0000	R/W	See register TempDevCompConfig0
02B6	TempDevConfig9	0000	R/W	See register TempDevConfig0
02B7	TempDevSysStatus9	0000	R/W	See register TempDevSysStatus0
02B8	TempDevBusConfig10	0000	R/W	See register TempDevBusConfig0
02B9	TempDevCompConfig10	0000	R/W	See register TempDevCompConfig0
02BA	TempDevConfig10	0000	R/W	See register TempDevConfig0
02BB	TempDevSysStatus10	0000	R/W	See register TempDevSysStatus0
02BC	TempDevBusConfig11	0000	R/W	See register TempDevBusConfig0
02BD	TempDevCompConfig11	0000	R/W	See register TempDevCompConfig0
02BE	TempDevConfig11	0000	R/W	See register TempDevConfig0
02BF	TempDevSysStatus11	0000	R/W	See register TempDevSysStatus0
02C0	TempDevBusConfig12	0000	R/W	See register TempDevBusConfig0
02C1	TempDevCompConfig12	0000	R/W	See register TempDevCompConfig0
02C2	TempDevConfig12	0000	R/W	See register TempDevConfig0
02C3	TempDevSysStatus12	0000	R/W	See register TempDevSysStatus0
02C4	TempDevBusConfig13	0000	R/W	See register TempDevBusConfig0
02C5	TempDevCompConfig13	0000	R/W	See register TempDevCompConfig0
02C6	TempDevConfig13	0000	R/W	See register TempDevConfig0
02C7	TempDevSysStatus13	0000	R/W	See register TempDevSysStatus0
02C8	TempDevBusConfig14	0000	R/W	See register TempDevBusConfig0
02C9	TempDevCompConfig14	0000	R/W	See register TempDevCompConfig0
02CA	TempDevConfig14	0000	R/W	See register TempDevConfig0
02CB	TempDevSysStatus14	0000	R/W	See register TempDevSysStatus0
02CC	TempDevBusConfig15	0000	R/W	See register TempDevBusConfig0
02CD	TempDevCompConfig15	0000	R/W	See register TempDevCompConfig0
02CE	TempDevConfig15	0000	R/W	See register TempDevConfig0
02CF	TempDevSysStatus15	0000	R/W	See register TempDevSysStatus0
02D0	TempDevTemperature0	0000	R	Temperature of temperature device 0, see LedboardTemp (0x001B)
02D1	TempDevTemperature1	0000	R	See TempDevTemperature0
02D2	TempDevTemperature2	0000	R	See TempDevTemperature0
02D3	TempDevTemperature3	0000	R	See TempDevTemperature0
02D4	TempDevTemperature4	0000	R	See TempDevTemperature0
02D5	TempDevTemperature5	0000	R	See TempDevTemperature0
02D6	TempDevTemperature6	0000	R	See TempDevTemperature0
02D7	TempDevTemperature7	0000	R	See TempDevTemperature0
02D8	TempDevTemperature8	0000	R	See TempDevTemperature0
02D9	TempDevTemperature9	0000	R	See TempDevTemperature0
02DA	TempDevTemperature10	0000	R	See TempDevTemperature0
02DB	TempDevTemperature11	0000	R	See TempDevTemperature0
02DC	TempDevTemperature12	0000	R	See TempDevTemperature0
02DD	TempDevTemperature13	0000	R	See TempDevTemperature0
02DE	TempDevTemperature14	0000	R	See TempDevTemperature0
02DF	TempDevTemperature15	0000	R	See TempDevTemperature0
02E0	TempDevStatus0	0000	R	Device specific configuration
02E1	TempDevStatus1	0000	R	See TempDevStatus0
02E2	TempDevStatus2	0000	R	See TempDevStatus0
02E3	TempDevStatus3	0000	R	See TempDevStatus0
02E4	TempDevStatus4	0000	R	See TempDevStatus0
02E5	TempDevStatus5	0000	R	See TempDevStatus0

© Bluetechnix 2016

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
02E6	TempDevStatus6	0000	R	See TempDevStatus0
02E7	TempDevStatus7	0000	R	See TempDevStatus0
02E8	TempDevStatus8	0000	R	See TempDevStatus0
02E9	TempDevStatus9	0000	R	See TempDevStatus0
02EA	TempDevStatus10	0000	R	See TempDevStatus0
02EB	TempDevStatus11	0000	R	See TempDevStatus0
02EC	TempDevStatus12	0000	R	See TempDevStatus0
02ED	TempDevStatus13	0000	R	See TempDevStatus0
02EE	TempDevStatus14	0000	R	See TempDevStatus0
02EF	TempDevStatus15	0000	R	See TempDevStatus0

Table 6-9:6.8 Registers for Temperature Management

6.2 Support

6.2.1 General Support

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

Support Link

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

6.3 Software Packages

Software packages and software downloads are for registered customers only

Software Package

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

6.4 Related Products

- ToF-Flash
- ToF-Flash Adapter
- Debug Adapter

7 Product History

7.1 Version Information

7.1.1 Sentis-ToF-M100

Version	Release date	Firmware Version
X-Grade	May 2013	V0.1.0
Initial Release	July 2013	V1.0.0
Update	January 2014	V2.0.1
Update	July 2014	V2.1.0
Update	July 2015	V2.5.0

Table 7-1: Overview Sentis-ToF-M100 product changes

! **Note**

Please refer to our support site for additional information about product changes.

7.2 Anomalies

Applies to	Date	Description
V2.5.0	July 2015	Sentis M100 LED Board temperature readout might fail at rare intervals, indicated by the LED-Board temperature sensor error flag (Status Register 0x03) and the transmitted temperature value (0xFF).

Table 7-2 – Product anomalies

7.3 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. 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. CmdEnablePasswd register added. Description for CRC calculation added.
5	2014 01 01	Document updated to firmware version 2.x.x. Hardware and Software related documentation split in two documents.
6	2014 02 20	Missing reference added.
7	2014 03 06	Register Mode0 corrected. Register CmdEnablePasswd corrected.
8	2014 03 07	Formatted
9	2014 07 20	Update to firmware version V2.1.x Alternative illumination interface description and registers added. Communication keep alive feature description and registers added.
10	2015 06 23	Update to firmware version V2.5.x

Version	Date	Document Revision
		Temperature Sensor Manager description and registers added.

Table 7-3: Revision history