BLUETECHNIX
Embedding Ideas

EPC610-ToF-Module

Software User Manual

Version 1

Bluetechnix

Waidhausenstraße 3/19
A-1140 Vienna
AUSTRIA

office@bluetechnix.com
www.bluetechnix.com

EPC610-ToF-Module – Software User Manual

Document No.: 900-308 / A

Publication date: June 18, 2014

Subject to change without notice. Errors excepted.

Table of Contents

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 EPC610-ToF-Module 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 EPC610-ToF-Module and describes the firmware dependent interfaces.

This document applies to firmware version 1.x.x.

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

Software and documentation

https://support.bluetechnix.at/wiki/TinyToF

3 Interfacing

The EPC610-ToF-Module has one UART Interface which can be used as control interface as well as data interface in case of a full duplex connection (for example RS232). A half-duplex connection type (for example RS485 with one differential pair) needs special configuration of the module.

The board-to-board connector supports also SPI and I2C but those interfaces are not supported by the firmware.

3.1 UART Interface

3.1.1 Register Read

3.1.1.1 Register Read Command

Addr	Len	Field	Value	Description
00	2	Preamble	0xa1ef	Unique identifier, start of header
02	1	ProtocolVersion	1	This document refers to version V1.0
03	1	Receiver	<id>	ID of communication target
04	1	Transmitter	<id>	ID of communication source
05	1	Command	1	Command code for register read
06	2	Length	2	Length of data
08	1	HeaderCrc	<CRC8 sum>	Checksum over 6 bytes of header: 0x2 – 0x7
09	2	RegisterAddress	<addr>	Address of the register to read
0B	2	DataCrc16	<CRC16 sum>	Checksum over data
0D				

3.1.1.2 Register Read Response

Addr	Len	Field	Value	Description
00	2	Preamble	0xa1ef	Unique identifier, start of header
02	1	ProtocolVersion	1	This document refers to version V1.0
03	1	Receiver	<id>	ID of communication target
04	1	Transmitter	<id>	ID of communication source
05	1	Command	4	Command code for response
06	2	Length	6	Length of data
08	1	HeaderCrc	<CRC8 sum>	Checksum over 6 bytes of header: 0x2 – 0x7
09	1	CommandOfResponse	1	Command code for register read
0A	2	RegisterAddress	<addr>	Address of the register to read
0C	2	RegisterData	<regData>	Content of the register to read
0E	1	Status	<status>	Refer to table “Status Codes”
0B	2	DataCrc16	<CRC16 sum>	Checksum over data
0D				

3.1.2 Register Write

3.1.2.1 Register Write Command

Addr	Len	Field	Value	Description
00	2	Preamble	0xa1ef	Unique identifier, start of header
02	1	ProtocolVersion	1	This document refers to version V1.0
03	1	Receiver	<id>	ID of communication target

Addr	Len	Field	Value	Description
04	1	Transmitter	<id>	ID of communication source
05	1	Command	2	Command code for register write
06	2	Length	4	Length of data
08	1	HeaderCrc	<CRC8 sum>	Checksum over 6 bytes of header: 0x2 – 0x7
09	2	RegisterAddress	<addr>	Address of the register to write
0B	2	RegisterContent	<regData>	Content to write to register
0D	2	DataCrc16	<CRC16 sum>	Checksum over data
0F				

3.1.2.2 Register Write Response

Addr	Len	Field	Value	Description
00	2	Preamble	0xa1ef	Unique identifier, start of header
02	1	ProtocolVersion	1	This document refers to version V1.0
03	1	Receiver	<id>	ID of communication target
04	1	Transmitter	<id>	ID of communication source
05	1	Command	4	Command code for response
06	2	Length	4	Length of data
08	1	HeaderCrc	<CRC8 sum>	Checksum over 6 bytes of header: 0x2 – 0x7
09	1	CommandOfResponse	2	Command code for register write
0A	2	RegisterAddress	<addr>	Address of the register to read
0C	1	Status	<status>	Refer to table “Status Codes”
0D	2	DataCrc16	<CRC16 sum>	Checksum over data
0F				

3.1.3 Data Stream

Addr	Len	Field	Value	Description
00	2	Preamble	0xa1ef	Unique identifier, start of header
02	1	ProtocolVersion	1	This document refers to version V1.0
03	1	Receiver	<id>	ID of communication target
04	1	Transmitter	<id>	ID of communication source
05	1	Command	3	Command code for data stream
06	2	Length	<length>	Length of Image data (header + pixel data)
08	1	HeaderCrc	<CRC8 sum>	Checksum over 6 bytes of header: 0x2 – 0x7
09	1	Version	1	This document refers to version V1.0
0A	1	ImageWidth		Number of pixels per row
0B	1	ImageHeight		Number of rows
0C	2	ImageFormat		Describes number of channels, their meaning and number of bytes per pixel
0E	4	Timesatamp		Timestamp in[]
12	2	FrameCounter		Consecutive numbers
14	1	MainboardTemp		
15	1	LedboardTemp		
17	2	FirmwareVersion		
19	n	PixelData		The actual pixel data in above described format
19 + n	2	DataCrc16	<CRC16 sum>	Checksum over data
1B + n				

3.1.4 Flash Update

3.1.4.1 Flash Update - Init

Addr	Len	Field	Value	Description
00	2	Preamble	0xa1ef	Unique identifier, start of header
02	1	ProtocolVersion	1	This document refers to version V1.0
03	1	Receiver	<id>	ID of communication target
04	1	Transmitter	<id>	ID of communication source
05	1	Command	5	Command code for flash update
06	2	Length	11	Length of flash init + header
08	1	HeaderCrc	<CRC8 sum>	Checksum over 6 bytes of header: 0x2 – 0x7
09	1	Version	1	This document refers to version V1.0
0A	1	FlashCommand	0	Command code for flash update init
0B	4	XX	XX	Ignored
0F	1	Target	<target>	0: Pixel list 0xff: Specific flash data
10	4	SpecificFlashAddress	<addr>	Address in flash memory
14	2	DataCrc16	<CRC16 sum>	Checksum over data
16				

3.1.4.2 Flash Update – Init Response

Addr	Len	Field	Value	Description
00	2	Preamble	0xa1ef	Unique identifier, start of header
02	1	ProtocolVersion	1	This document refers to version V1.0
03	1	Receiver	<id>	ID of communication target
04	1	Transmitter	<id>	ID of communication source
05	1	Command	4	Command code for response
06	2	Length	6	Length of response + header
08	1	HeaderCrc	<CRC8 sum>	Checksum over 6 bytes of header: 0x2 – 0x7
09	1	CommandOfResponse	5	Command code for flash update
0A	1	Version	1	This document refers to version V1.0
0B	1	FlashCommand	0	Command code for flash update init
0C	2	XX	XX	Ignored
0E	1	Status	<status>	Refer to table “Flash Update Status Codes”
0D	2	DataCrc16	<CRC16 sum>	Checksum over data
0F				

3.1.4.3 Flash Update – Set CRC32

Addr	Len	Field	Value	Description
00	2	Preamble	0xa1ef	Unique identifier, start of header
02	1	ProtocolVersion	1	This document refers to version V1.0
03	1	Receiver	<id>	ID of communication target
04	1	Transmitter	<id>	ID of communication source
05	1	Command	5	Command code for flash update
06	2	Length	10	Length of crc + header
08	1	HeaderCrc	<CRC8 sum>	Checksum over 6 bytes of header: 0x2 – 0x7
09	1	Version	1	This document refers to version V1.0
0A	1	FlashCommand	1	Command code for flash update set crc32
0B	4	XX	XX	Ignored
0F	4	CRC32	<CRC32 sum>	CRC sum over the whole file to be transmitted

Addr	Len	Field	Value	Description
13	2	DataCrc16	<CRC16 sum>	Checksum over data
15				

3.1.4.4 Flash Update – Set CRC32Response

Addr	Len	Field	Value	Description
00	2	Preamble	0xa1ef	Unique identifier, start of header
02	1	ProtocolVersion	1	This document refers to version V1.0
03	1	Receiver	<id>	ID of communication target
04	1	Transmitter	<id>	ID of communication source
05	1	Command	4	Command code for response
06	2	Length	6	Length of response + header
08	1	HeaderCrc	<CRC8 sum>	Checksum over 6 bytes of header: 0x2 – 0x7
09	1	CommandOfResponse	5	Command code for flash update
0A	1	Version	1	This document refers to version V1.0
0B	1	FlashCommand	1	Command code for flash update set crc32
0C	2	XX	XX	Ignored
0E	1	Status	<status>	Refer to table “Flash Update Status Codes”
0D	2	DataCrc16	<CRC16 sum>	Checksum over data
0F				

3.1.4.5 Flash Update – Set Packet

Addr	Len	Field	Value	Description
00	2	Preamble	0xa1ef	Unique identifier, start of header
02	1	ProtocolVersion	1	This document refers to version V1.0
03	1	Receiver	<id>	ID of communication target
04	1	Transmitter	<id>	ID of communication source
05	1	Command	5	Command code for flash update
06	2	Length	6 + <packetLength>	Length of packet data + header
08	1	HeaderCrc	<CRC8 sum>	Checksum over 6 bytes of header: 0x2 – 0x7
09	1	Version	1	This document refers to version V1.0
0A	1	FlashCommand	2	Command code for flash update set packet
0B	2	PacketNumber	<packetNr>	Index of packet starting at 0
0D	2	PacketSize	<packetSize>	Size of packet without header
0F	n	PacketData	<data>	
0F + n	2	DataCrc16	<CRC16 sum>	Checksum over data
11 + n				

3.1.4.6 Flash Update – Set Packet Response

Addr	Len	Field	Value	Description
00	2	Preamble	0xa1ef	Unique identifier, start of header
02	1	ProtocolVersion	1	This document refers to version V1.0
03	1	Receiver	<id>	ID of communication target
04	1	Transmitter	<id>	ID of communication source
05	1	Command	4	Command code for response
06	2	Length	6	Length of response + header
08	1	HeaderCrc	<CRC8 sum>	Checksum over 6 bytes of header: 0x2 – 0x7
09	1	CommandOfResponse	5	Command code for flash update
0A	1	Version	1	This document refers to version V1.0
0B	1	FlashCommand	2	Command code for flash update set packet

Addr	Len	Field	Value	Description
0C	2	PacketNumber	<packetNr>	
0E	1	Status	<status>	Refer to table “Flash Update Status Codes”
0D	2	DataCrc16	<CRC16 sum>	Checksum over data
0F				

3.1.4.7 Flash Update – Get Max Packet Size

Addr	Len	Field	Value	Description
00	2	Preamble	0xa1ef	Unique identifier, start of header
02	1	ProtocolVersion	1	This document refers to version V1.0
03	1	Receiver	<id>	ID of communication target
04	1	Transmitter	<id>	ID of communication source
05	1	Command	5	Command code for flash update
06	2	Length	6	Length of payload
08	1	HeaderCrc	<CRC8 sum>	Checksum over 6 bytes of header: 0x2 – 0x7
09	1	Version	1	This document refers to version V1.0
0A	1	FlashCommand	3	Command code for flash update get max packet size
0B	4	XX	XX	Ignored
0F	2	DataCrc16	<CRC16 sum>	Checksum over data
11				

3.1.4.8 Flash Update – Get Max Packet Size Response

Addr	Len	Field	Value	Description
00	2	Preamble	0xa1ef	Unique identifier, start of header
02	1	ProtocolVersion	1	This document refers to version V1.0
03	1	Receiver	<id>	ID of communication target
04	1	Transmitter	<id>	ID of communication source
05	1	Command	4	Command code for response
06	2	Length	8	Length of response + header
08	1	HeaderCrc	<CRC8 sum>	Checksum over 6 bytes of header: 0x2 – 0x7
09	1	CommandOfResponse	5	Command code for flash update
0A	1	Version	1	This document refers to version V1.0
0B	1	FlashCommand	3	Command code for flash update get max packet size
0C	2	XX	XX	Ignored
0E	1	Status	<status>	Refer to table “Flash Update Status Codes”
0F	2	MaxPacketSize	<maxPacketSize>	
11	2	DataCrc16	<CRC16 sum>	Checksum over data
13				

3.1.4.9 Flash Update – Finalize

Addr	Len	Field	Value	Description
00	2	Preamble	0xa1ef	Unique identifier, start of header
02	1	ProtocolVersion	1	This document refers to version V1.0
03	1	Receiver	<id>	ID of communication target
04	1	Transmitter	<id>	ID of communication source
05	1	Command	5	Command code for flash update
06	2	Length	6	Length of payload

Addr	Len	Field	Value	Description
08	1	HeaderCrc	<CRC8 sum>	Checksum over 6 bytes of header: 0x2 – 0x7
09	1	Version	1	This document refers to version V1.0
0A	1	FlashCommand	4	Command code for flash update finalize
0B	4	XX	XX	Ignored
0F	2	DataCrc16	<CRC16 sum>	Checksum over data
11				

3.1.4.10 Flash Update – Finalize Response

Addr	Len	Field	Value	Description
00	2	Preamble	0xa1ef	Unique identifier, start of header
02	1	ProtocolVersion	1	This document refers to version V1.0
03	1	Receiver	<id>	ID of communication target
04	1	Transmitter	<id>	ID of communication source
05	1	Command	4	Command code for response
06	2	Length	6	Length of response + header
08	1	HeaderCrc	<CRC8 sum>	Checksum over 6 bytes of header: 0x2 – 0x7
09	1	CommandOfResponse	5	Command code for flash update
0A	1	Version	1	This document refers to version V1.0
0B	1	FlashCommand	4	Command code for flash update finalize
0C	2	XX	XX	Ignored
0E	1	Status	<status>	Refer to table “Flash Update Status Codes”
0F	2	DataCrc16	<CRC16 sum>	Checksum over data
11				

3.1.5 Error

Addr	Len	Field	Value	Description
00	2	Preamble	0xa1ef	Unique identifier, start of header
02	1	ProtocolVersion	1	This document refers to version V1.0
03	1	Receiver	<id>	ID of communication target
04	1	Transmitter	<id>	ID of communication source
05	1	Command	0xFF	Command code for error
06	2	Length	1	Length of data
08	1	HeaderCrc	<CRC8 sum>	Checksum over 6 bytes of header: 0x2 – 0x7
09	1	Error	<error>	0: Version not supported
0D	2	DataCrc16	<CRC16 sum>	Checksum over data
0F				

3.1.6 Status Codes

Status	Description
00	Ok
0F	Illegal write: The Address is not valid or the register is not write-enabled
11	Register end reached
FC	DataCrc16 mismatch
FF	Unknown command

3.1.7 Flash Update Status Codes

Status	Description
00	Ok
01	Unknown target
02	Invalid flash address
03	CRC32 error
04	Wrong packet number
05	Wrong packet size
06	Unknown command
07	Unsupported version
08	Not initialized
09	Protocol violation
0A	In progress error
0B	Flashing failed
0C	Max size exceeded

3.2 GPIOs

The EPC610-ToF-Module provides two Inputs and two Outputs. The current state of these IOs can be checked via the IO State Register. Alternatively the Alive LED feature can be configured to either of the two Outputs.

Please refer to the Hardware User Manual for information about the GPIO interface.

3.3 Frame trigger signal (TRIG)

The frame trigger signal is available on IN_1, the first of the two Inputs (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 the video mode has to be disabled in register Mode0.

3.4 Status LED

The Status LED is used to give some basic information about the status of the module. The output of the status LED can be configured to OUT_1, OUT_2 or none. This behavior can be configured in the Mode0 register. The following table shows the meaning depending on the mode.

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

Table 3-1: Status LED meaning

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.

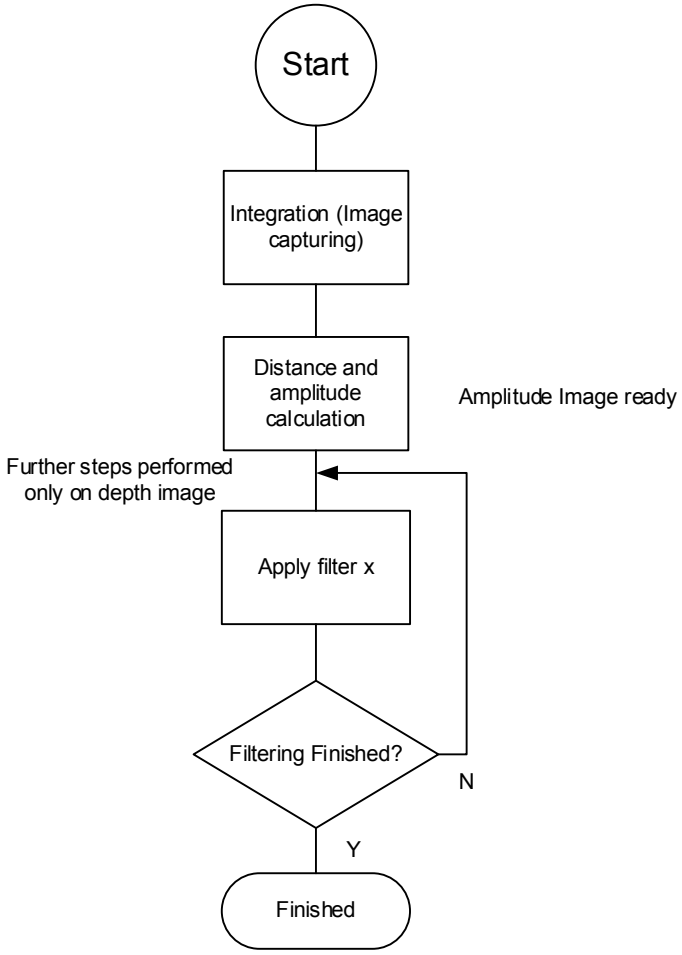

Figure 4-1: Image processing flow

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. 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 Local Average Filter

Register: *FilterLAVconfig*

Only a 2x2 local average filter can be applied.

4.2.1.2 Sliding Average Filter

Register: *FilterSLAFconfig*

A sliding average filter over up to 5 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.

4.2.1.3 Frame Average Filter

Register: *FilterFAVconfig*

A frame average filter over up to 5 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 decrease the frame rate because the configured number of frames are gathered and the averaged frame is transmitted over the data interface.

4.3 Camera Data Format

The camera provides two 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. The following image shows the coordinate system from a cameras point of view:

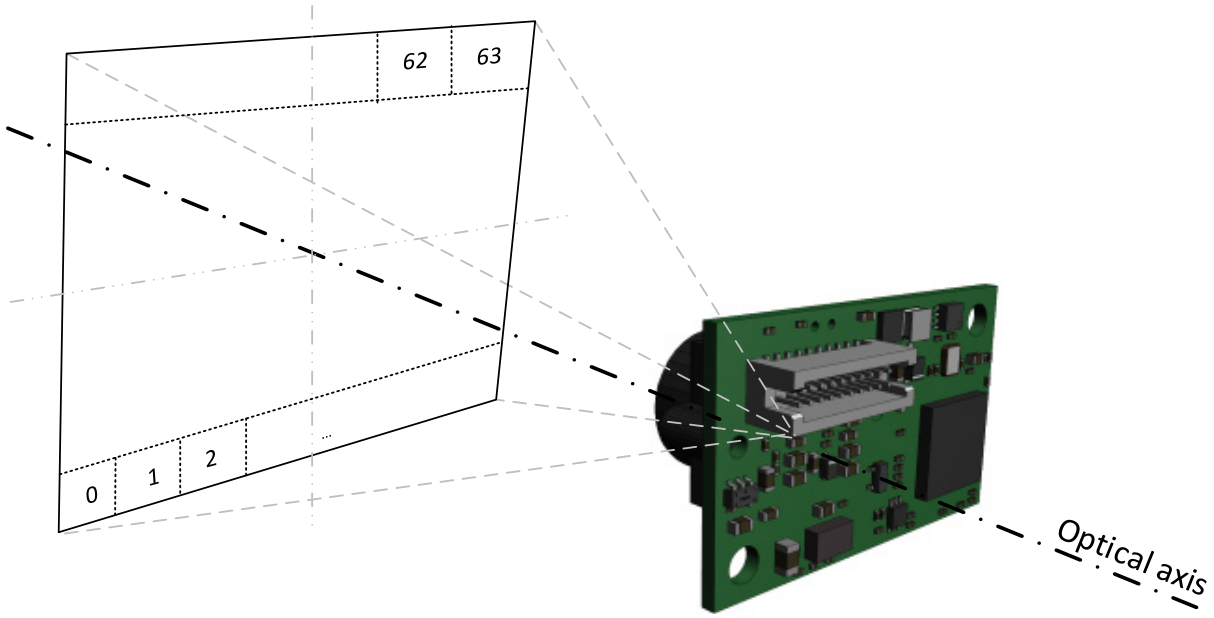

Figure 4-2: EPC610-ToF-Module optical arrangement

The image format to be transferred can be configured in 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.

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**.

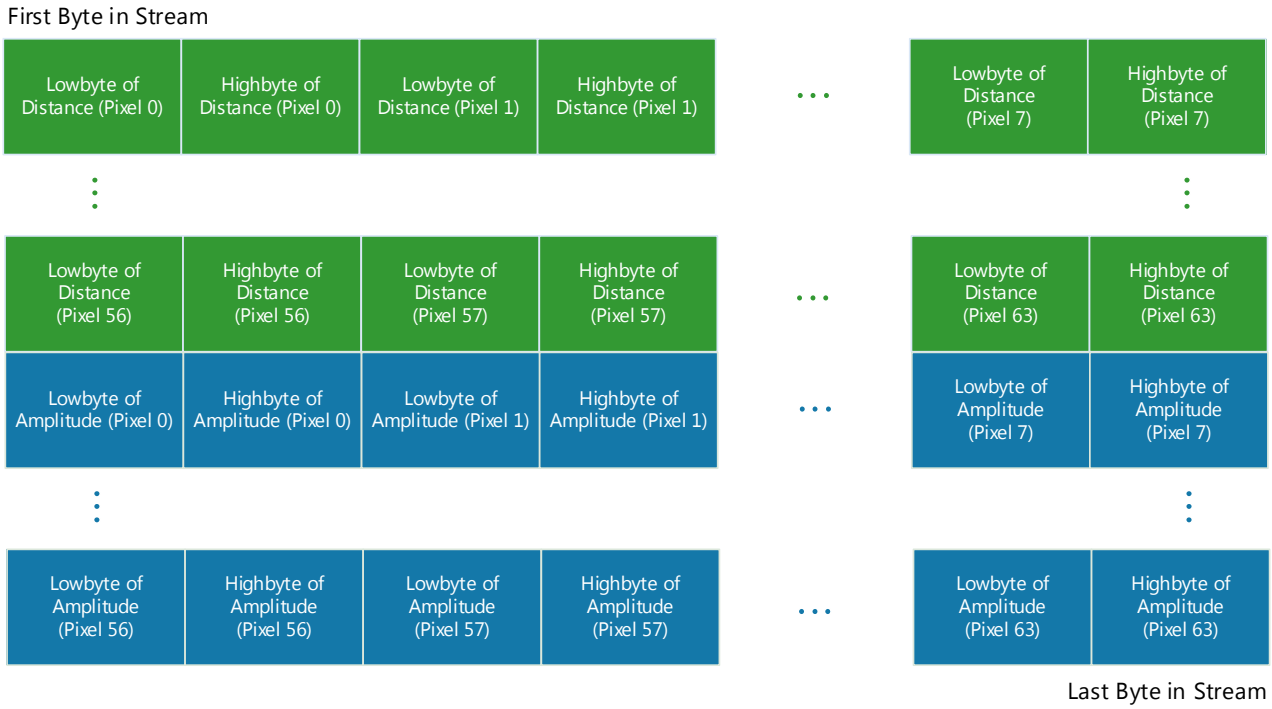

Figure 4-3: Data-stream of distance and amplitude data

4.4 High Dynamic Range (HDR)

The module provides an automatic sweep through different integration times to get the optimal illumination for each pixel. The target amplitude can be configured in the *HDRamplitudeThreshold* register. This feature can be enabled in the *Mode0* register.

There are two modes for HDR imaging. The first mode will take images starting with the integration time configured at register *HDRintegrationTimeStart* and sweeps until every pixel has a valid amplitude value or the integration time configured at register *HDRintegrationTimeStop* is reached. This mode is slow for most of the scenarios. This approach delivers good results for fast changing conditions.

The second mode starts with the lowest integration time that delivered a valid pixel from the last image and reduces it by one step. So this will scores good frame rates but can be slow on fast changing situations because it may deliver some over illuminated images until the integration time is low enough.

These two modes can be switched in the *Mode0* register.

4.5 Pixel invalidation

The EPC610-ToF-Module 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) or the EPC610 sensor signals a saturated pixel 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*.

4.6 Modulation Frequency

The modulation frequency is set to 10 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.

In the register you can either write the frequency in 10 kHz steps or the index. On a read of the register you get the current selected modulation frequency.

Other frequencies between 1,25Mhz and 20 MHz 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.

The modulation frequency has also impact on the maximum integration time. At 10 MHz the maximum integration time is 52,6ms. At 20 MHz this maximum is reduced to the half of 26,3ms.

Modulation Frequency [MHz]	Max. Integration time [ms]	Integration time multiplier
20	26,3	0,5
10	52,6	1
6,67	78,9	1,5
5	105,2	2
4	131,5	2,5
3,33	157,8	3
2,86	184,1	3,5
2,5	210,4	4
2,22	236,7	4,5
2	263	5
1,82	289,3	5,5
1,67	315,6	6
1,54	341,9	6,5
1,43	368,2	7
1,33	394,5	7,5
1,25	420,8	8

Table 4-1: dependency between modulation frequency and maximum integration time

4.7 Integration time

The integration time can be set only in predefined steps. The resulting integration time depends on the used modulation frequency. The following table shows the steps for 10MHz integration time. For other modulation frequencies the integration time multiplier (see Table 4-1) has to be applied to get the resulting integration time.

Integration time steps [µs]
1,6
3,2
6,4
12,8
25,6
51,2
103
205
408
818
1640
3280
6560
13200
26300
52500

Table 4-2: Integration time steps at 10 MHz

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 module integration time is limited by hardware to 52,6ms at 10MHz modulation frequency.

The maximum frame rate is 18fps 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 High Dynamic Range is enabled the integration time will be set automatically and the register *IntegrationTime* has no influence!

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].

4.9.1 Hardware Trigger

The camera provides an external connector where a hardware trigger can be applied. Please refer to 3.3 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 module supports no automatic shutdown to prevent overheating.

4.11 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. This feature can be used to save a user specific configuration.

4.12 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*.

4.13 Firmware Update

A firmware update can be performed over the UART boot protocol. See https://support.bluetechnix.at/wiki/TinyToF for more details.

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/TinyToF

5.2 Getting Started Software Development Example

To facilitate the integration of the EPC610 ToF 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/TinyToF

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 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[1]: 1..HDR On Bit[4]: 1..Manual Trigger Bit[5]: 1..HDR Auto Limits On Bit[9]: 1..Alive LED on Out1 Bit[10]: 1..Alive LED on Out2
0002	Madr	0000	R/W	Bit[0-7]: module address for serial bus communication
0003	Status	0040	R	Bit[0]: 0..Application Mode, 1..Bootloader Mode Bit[4]: 1..Main-Board temperature sensor error Bit[6]: 1..Factory Regmap was loaded Bit[7]: 1..Regmap recovered after watchdog reset
0004	ImageDataFormat	0000	R/W	Bit[3:10]: 0… 2 bytes depth-data / 2 bytes amp-data
0005	IntegrationTime	CD78	R/W	Integration Time [µs]
0006	DeviceType	7A3D	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	03E8	R/W	Modulation frequency index: 0.. 20 MHz 1.. 10 MHz 2.. 6,67 MHz 3.. 5 MHz 4.. 4 MHz 5.. 3,33 MHz 6.. 2,86 Mhz
000A	Framerate	0028	R/W	Framerate [Hz]
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	0032	R/W	Amplitude threshold for valid distance data
0011	ConfidenceThresHigh	FFFF	R/W	Amplitude threshold for valid distance data
001A	CalculationTime		R	Calculation time for the last frame in 10[µs]. The inverse of this value shows the maximum achievable frame rate based on the CPU load.
001C	MainboardTemp		R	Temperature of Main-Board in 0,01[°C] (FFFF: Sensor not available).

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
001F	FrameTime		R	Time between the last two frames. In 0,1[ms]
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)
002D	TempCompGradient		R/W	Factor 'c' of the temperature compensation function: y [mm] = a/10000 * x^3 + b/10000 * x^2 + c/1000 * x + u
0030	TempCompGradient2		R/W	Factor 'b' of the temperature compensation function: y [mm] = a/10000 * x^3 + b/10000 * x^2 + c/1000 * x + u
0033	CmdExec	0000	R/W	Initiate an operation: 0xC2AE.. Clear RegMap in flash 0x9E20.. Read RegMap from flash 0x909A.. Read factory RegMap 0xDD9E.. Save RegMap in flash Writing this register must be preceded by writing 0x4877 into register CmdEnablePasswd (0x0022)
0034	CmdExecResult	0000	R	Result code of the operation initiated using CmdExec 1.. Success Other.. Error
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)
003C	TempCompGradient3		R/W	Factor 'a' of the temperature compensation function: y [mm] = a/10000 * x^3 + b/10000 * x^2 + c/1000 * 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]

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.

6.2 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

Table 6-2: General registers

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

6.3 Registers for GPIOs

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
00d0	IOstate0	0000	R/W	Bit[0]: … state of IN_1 (only R) Bit[1]: … state of IN_2 (only R) Bit[8]: … state of OUT_1 (R/W) Bit[9]: … state of OUT_2 (R/W)

Table 6-3: Registers for GPIOs

6.4 Registers for Wiggling correction

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
00e0	WigglingAmp0	0000	R/W	Wiggling correction amplitude when operating at modulation frequency with index 0
00e1	WigglingAmp1	0050	R/W	Wiggling correction amplitude when operating at modulation frequency with index 1
00e2	WigglingAmp2	0000	R/W	Wiggling correction amplitude when operating at modulation frequency with index 2

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
00e3	WigglingAmp3	0000	R/W	Wiggling correction amplitude when operating at modulation frequency with index 3
00e4	WigglingAmp4	0000	R/W	Wiggling correction amplitude when operating at modulation frequency with index 4
00e5	WigglingAmp5	0000	R/W	Wiggling correction amplitude when operating at modulation frequency with index 5
00e6	WigglingAmp6	0000	R/W	Wiggling correction amplitude when operating at modulation frequency with index 6
00e7	WigglingAmp7	0000	R/W	Wiggling correction amplitude when operating at modulation frequency with index 7
00e8	WigglingAmp8	0000	R/W	Wiggling correction amplitude when operating at modulation frequency with index 8
00e9	WigglingAmp9	0000	R/W	Wiggling correction amplitude when operating at modulation frequency with index 9
00f0	WigglingPhase0	0000	R/W	Wiggling correction phase when operating at modulation frequency with index 0
00f1	WigglingPhase1	128E	R/W	Wiggling correction phase when operating at modulation frequency with index 1
00f2	WigglingPhase2	0000	R/W	Wiggling correction phase when operating at modulation frequency with index 2
00f3	WigglingPhase3	0000	R/W	Wiggling correction phase when operating at modulation frequency with index 3
00f4	WigglingPhase4	0000	R/W	Wiggling correction phase when operating at modulation frequency with index 4
00f5	WigglingPhase5	0000	R/W	Wiggling correction phase when operating at modulation frequency with index 5
00f6	WigglingPhase6	0000	R/W	Wiggling correction phase when operating at modulation frequency with index 6
00f7	WigglingPhase7	0000	R/W	Wiggling correction phase when operating at modulation frequency with index 7
00f8	WigglingPhase8	0000	R/W	Wiggling correction phase when operating at modulation frequency with index 8
00f9	WigglingPhase9	0000	R/W	Wiggling correction phase when operating at modulation frequency with index 9

Table 6-4: Registers for Wiggling correction

6.5 Registers for Filter Configuration

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
01E5	FilterSlafConfig	0001	R/W	Bit[0-7]: …Window size
01E7	FilterFrameAverageConfig	0001	R/W	Bit[0-3]: ... Number of frames
01E8	FilterLocalAverageConfig	0001	R/W	Bit[0-3]: … Number of square pixel

Table 6-5: Register for filter configuration

6.6 Registers for Switch application configuration

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
0260	SwitchConfig	0000	R/W	Bit[0]: … display switch state on outputs
0261	SwitchState	0000	R	Bit[0-3]: … state of the switch application
0262	SwitchThreshold1	01C2	R/W	Lower threshold of the switch application in [mm]
0263	SwitchThreshold2	0320	R/W	Upper threshold of the switch application in [mm]

Table 6-6: Registers for Switch application configuration

6.7 Registers for HDR configuration

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
0270	HDRintegrationTimeStart	0002	R/W	If bit[5] of Mode0 register is not set this is the start integration time [µs] for HDR imaging
0271	HDRintegrationTimeStop	CB20	R	If bit[5] of Mode0 register is not set this is the end integration time [µs] for HDR imaging
0272	HDRamplitudeThreshold	012C	R/W	Target amplitude for HDR imaging

Table 6-7: Registers for HDR configuration

6.8 Registers for distance correction

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
0280	DistCorrConfig	0007	R/W	Bit[0]: … enable output in [mm] Bit[1]: … enable temperature correction Bit[2]: … enable wiggling correction

Table 6-8: Registers for distance correction configuration

6.9 Support

6.9.1 General Support

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

Support Link

https://support.bluetechnix.at/wiki/EPC610-ToF-Module_Camera

6.10 Software Packages

Software packages and software downloads are for registered customers only

Software Package

https://support.bluetechnix.at/wiki/EPC610-ToF-Module_Camera

6.11 Related Products

- DEV-EPC6XX

7 Product History

7.1 Version Information

7.1.1 EPC610-ToF-Module

Version	Note	Release date	Firmware Version
0	X-Grade	June 2014	V0.1.0

Table 7-1: Overview EPC610-ToF-Module product changes

Note

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

7.2 Anomalies

Applies to	Date	Description

Table 7-2 – Product anomalies

7.3 Document Revision History

Version	Date	Document Revision
1	2014 06 17	First preliminary of the document

Table 7-3: Revision history

8 List of Abbreviations

Abbreviation	Description
ADI	Analog Devices Inc.
AI	Analog Input
AMS	Asynchronous Memory Select
AO	Analog Output
CM	Core Module
DC	Direct Current
DSP	Digital Signal Processor
eCM	Enhanced Core Module
EBI	External Bus Interface
ESD	Electrostatic Discharge
GPIO	General Purpose Input Output
I	Input
I²C	Inter-Integrated Circuit
I/O	Input/Output
ISM	Image Sensor Module
LDO	Low Drop-Out regulator
MTBF	Mean Time Between Failure
NC	Not Connected
NFC	NAND Flash Controller
O	Output
OS	Operating System
PPI	Parallel Peripheral Interface
PWR	Power
RTOS	Real-Time Operating System
SADA	Stand Alone Debug Agent
SD	Secure Digital
SoC	System on Chip
SPI	Serial Peripheral Interface
SPM	Speech Processing Module
SPORT	Serial Port
TFT	Thin-Film Transistor
TISM	Tiny Image Sensor Module
TSC	Touch Screen Controller
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
USBOTG	USB On The Go
ZIF	Zero Insertion Force

Table 8.1: List of abbreviations

A List of Figures and Tables

Figures

Tables