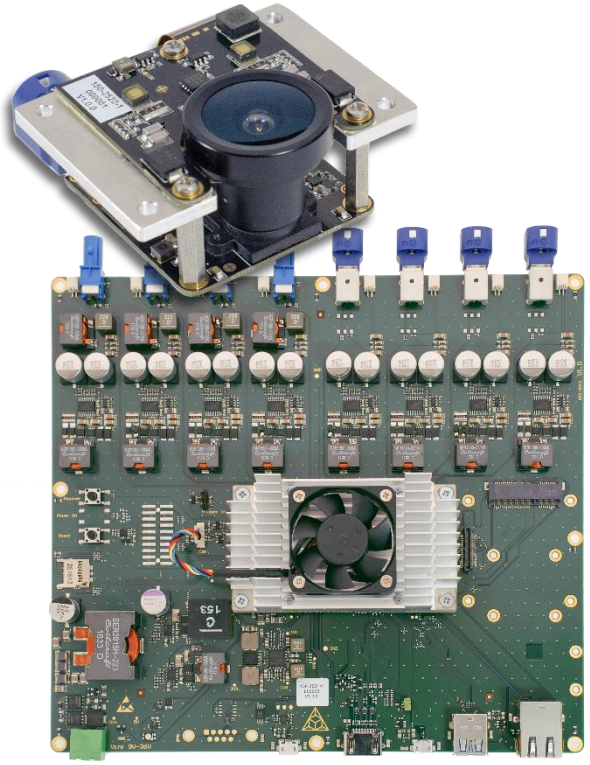
BLUETECHNIX
Embedding Ideas

multi-ToF platform



Software User Manual

Version 1

BECOM BLUETECHNIX GmbH

Gutheil-Schoder-Gasse 17
1230 Wien
AUSTRIA

office@bluetechnix.com
www.bluetechnix.com

multi-ToF platform – Software User Manual

Document No.: 900-308 / A

Publication date: February 22, 2018

Subject to change without notice. Errors excepted.

Table of Contents

1 General Information 6
1.1 Symbols Used 6
2 Overview 7
3 Interfacing 8
3.1 Control Interface 8
3.2 Data Interface 8
3.3 Camera Discovery 9
3.4 Secure Shell (SSH) Login 9
3.5 Debug UART 9
4 BltTofApp 11
5 Camera Features 12
5.1 Basic Settings 12
5.2 ToF Image Processing Chain 12
5.3 ToF Image Filters 13
5.3.1 Median Filter 14
5.3.2 Bilateral filter 14
5.3.3 Average filter 14
5.3.4 Sliding Average Filter 14
5.3.5 Frame Average Filter 14
5.4 ToF Image Enhancements 15
5.4.1 Pixel invalidation 15
5.4.2 Temperature compensation 15
5.5 Camera Coordinate System 15
5.6 Camera Data Format 16
5.6.1 Distances and Amplitudes 17
5.6.2 XYZ Point Cloud 18
5.6.3 XYZ Point Cloud and Amplitudes 18
5.6.4 Distances and XYZ Point Cloud 18
5.6.5 X Coordinate and Amplitudes 18
5.6.6 Distances 18
5.6.7 4 phases without image processing 18
5.6.8 Raw Distances and Amplitudes 18
5.6.9 Distance, Amplitudes, and Confidences 19
5.7 ToF Modulation Frequency 19
5.8 Frame Rate and Integration Time 19

5.9 Distance Offset Calibration 20
5.10 Temperature Sensors 20
5.10.1 Over Temperature Protection 20
5.11 Save Registers 21
5.12 Reset to Factory Default 21
5.13 Error Indication 21
6 Register Description 22
6.1 General 22
6.2 Distance Offset 25
6.3 User Defined 25
6.4 General 26
6.5 Sequencing 26
6.6 Device Update 27
6.7 Filter Configuration 27
6.8 Advanced Image processing 28
6.9 Ethernet configuration 28
7 Firmware History 29
7.1 Version Information 29
7.2 Anomalies 29
8 Software 30
8.1 BltTofApi 30
8.2 MATLAB SDK 30
8.3 BltTofSuite 30
9 Support 31
9.1 General Support 31
9.2 Software Downloads 31
9.3 Camera Development Package 31
10 Document Revision History 32
A List of Figures and Tables 33

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.

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

BECOM BLUETECHNIX takes no liability for any damages and errors causing of the usage of this product. The user of this product is responsible by himself for the functionality of his application. He is allowed to use the product 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 BECOM BLUETECHNIX (http://www.bluetechnix.com).

Warning

Due to technical requirements components may contain dangerous substances.

1 General Information

This guide applies to the multi-ToF platform camera platform from BECOM BLUETECHNIX. 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 multi-ToF platform and describes the firmware dependent interfaces.

This document applies to firmware version 0.3.0.

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

Software and documentation

https://support.bluetechnix.at/index.html

3 Interfacing

The multi-ToF platform provides control and data interfaces via Gigabit-Ethernet.

The control interface is used to set and read the configuration of the multi-ToF platform via a set of registers. Refer to chapter 6 for a detailed register description.

The data interface provides a continuous stream of ToF data depending on the configuration.

Bluetechnix provides an abstraction of control and data interfaces by means of Refer to chapter 8.1 for the *BltTofApi*.

3.1 Control Interface

The multi-ToF platform can be configured using a TCP/IP connection. For the control interface the multi-ToF platform 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

3.2 Data Interface

A UDP stream delivers ToF and/or color data from the multi-ToF platform. Each UDP packet contains a UDP streaming header and up to 1400 bytes of frame data (Ethernet, IP, and UDP headers are not shown in Figure 3-1).

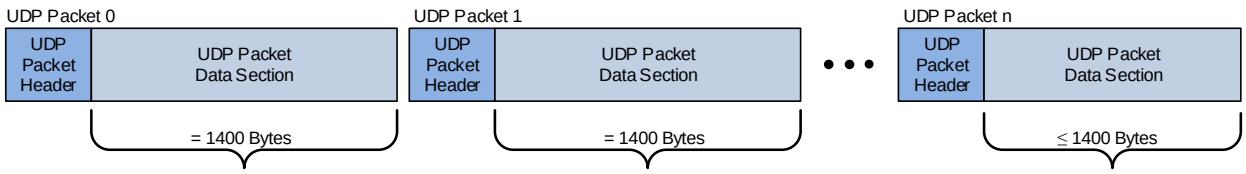


Figure 3-1: UDP streaming data format

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.

The UDP streaming is enabled by factory default. The multi-ToF platform streams to the following IP settings:

- **IP-Address**: 192.168.0.1
- **UDP port**: 10002

Note

The UDP stream settings can be configured using the ***Eth0_*** registers.

3.3 Camera Discovery

The multi-ToF platform supports a discovery protocol via UDP/IP. It allows the discovery of the camera within the Ethernet network and the retrieval of camera properties.

The discovery service listens on UDP address 255.255.255.255 (broadcast) and port 11003.

Note

Discovery is supported by the BltTofApi and the BltTofSuite
Chapter 8

3.4 Secure Shell (SSH) Login

The multi-ToF platform camera features an OpenSSH server listening to TCP port 22.

	Root account
Username	nvidia
Default password	nvidia

Table 3-1: Default login credentials

3.5 Debug UART

Note

Using the Debug UART is optional.

The multi-ToF platform features a debug UART, which is the primary debug interface for the boot loader as well as the Linux kernel.

The Debug UART is available via a micro-USB-connector, with a UART-to-USB converter behind. To be able to access the serial terminal via the Debug UART, you need an appropriate driver installed in your OS.

Most Linux distributions come with an appropriate driver and create a device node `/dev/ttyUSB...` dynamically.

Additionally, one needs a serial terminal emulator, e.g., Minicom or C-Kermit for Linux, or TeraTerm for Windows OS. The emulator has to be configured with the following settings:

Baud rate	115200
Data bits	8
Parity	none
Stop bits	1
Flow control	none

Table 3-2: Debug UART settings

The Debug UART also allows to log in to the camera's Linux OS. Please see chapter 3.4 for the default login and how to change it.

4 BltTofApp

The BltTofApp configures the connected frontend, captures the image data stream, processes the image processing chain (refer to 5.2) and streams the calculated data over the ethernet.

The application is located at '/home/nvidia/BltTofApp' and gets started automatically from '/etc/rc.local' at boot time. The first argument of the BltTofApp is used to select the slot on which the frontend is connected. This argument can range from 0 to 3.

Start the application manually:

sudo ./BltTofApp [slot] &

Stop the application:

sudo killall BltTofApp

5 Camera Features

5.1 Basic Settings

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

Each camera has been pre-configured with a factory-default register map.

5.2 ToF Image Processing Chain

The following flow diagram shows the image processing chain of the camera for the ToF sensor data. Filters can be applied individually to distance data. XYZ point cloud data is calculated from distance data on demand.

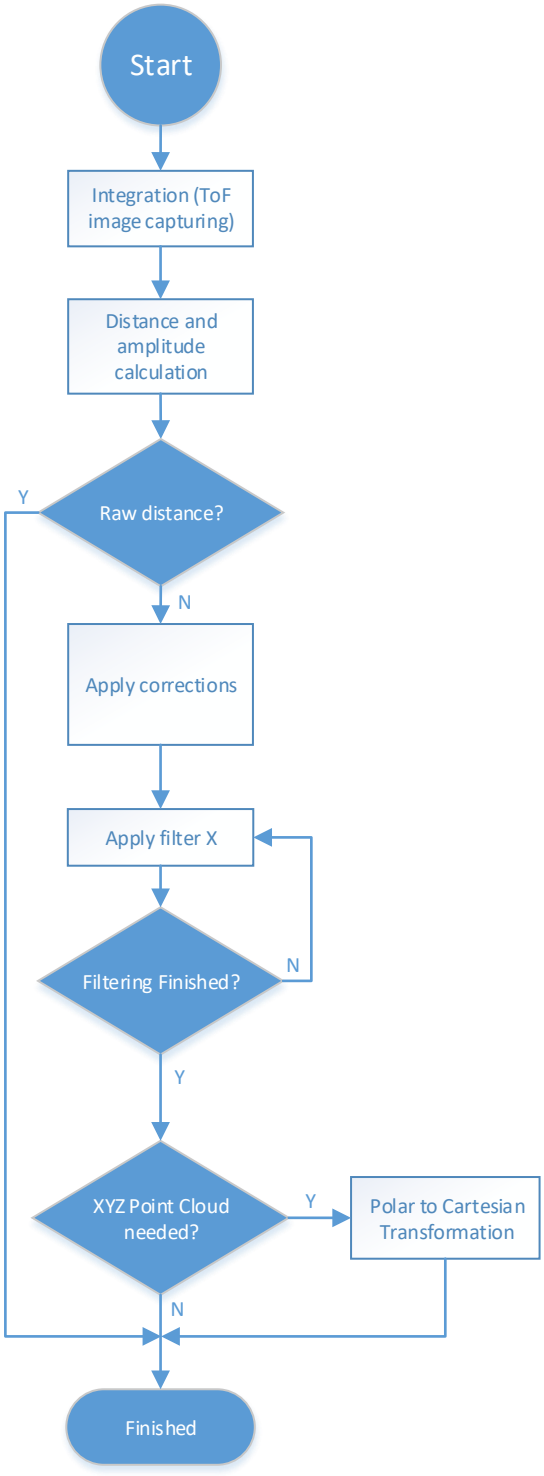


Figure 5-1: Image processing flow

5.3 ToF Image Filters

After the distance and amplitude calculation, filters can be applied to the distance data. Each of the filters provides one or more configuration parameters. The iteration count for each filter can also be configured. The

filters can be enabled or disabled by writing the ***ImgProcConfig*** (distance data). Enabling more than one filter is possible but each added filter reduces the maximum achievable frame rate (as does the number of iterations).

The filters are applied in the following order:

1. Frame Average filter
2. Sliding Average filter
3. Average filter
4. Median filter
5. Bilateral filter

5.3.1 Median Filter

A 3x3 median filter can be applied.

Register: ***FilterMedianConfig***

The number of iterations is configurable.

5.3.2 Bilateral filter

Registers: ***FilterBilateralConfig, FilterBilateralConfig2***

Configuration options are σ_R (Width of range kernel), σ_S (Width of spatial kernel), number of iterations, and window size.

5.3.3 Average filter

Registers: **FilterAverageConfig**

Configuration option is the filter size.

5.3.4 Sliding Average Filter

Register: ***FilterSLAFconfig***

A sliding average filter over up to 255 frames can be applied. The number of frames is configurable. An increasing number of frames will not decrease the frame rate but may add blurring effects.

5.3.5 Frame Average Filter

Register: ***FilterFrameAverageConfig***

A frame average filter over up to 255 frames can be applied. The number of frames is configurable.

The frame rate of the data interface will be divided by the number of configured frames to be averaged, e.g., if the camera is configured to 40 frames per second, and the frame average filter with number 4 is used, the resulting output frame rate will be 10.

5.4 ToF Image Enhancements

5.4.1 Pixel invalidation

The multi-ToF platform provides an on-board check for invalid pixels:

- Underexposed pixels: The amplitude is too low for the distance value to be trustworthy. The multi-ToF platform sets the pixel distance to the maximum value. The threshold is set via register ***ConfidenceThresLow***.
- Overexposed pixels: The amplitude is too high for the distance value to be trustworthy. The multi-ToF platform sets the pixel distance to the minimum value. The threshold is set via register ***ConfidenceThresHigh***.
- Invalid pixels: The multi-ToF platform features an additional amplitude check called ACF (auto correlation function) Plausibility Check. It detects inconsistent pixels e.g. in case of fast movement in the scene.

5.4.1.1 Distance values

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.

For invalid pixels, the distance value is set to 0x0001.

5.4.1.2 XYZ values

If the amplitude of the reflected signal is below a threshold (underexposure) the X value of the appropriate pixel is set to 32767 (0x7FFF), i.e., the largest positive Int16 value. Y and Z values are set to 0.

If the amplitude of the reflected signal is above a threshold (overexposure) the X of the appropriate pixel is set to 0. Y and Z values are set to 0 as well.

If the ACF plausibility check classified the pixel's distance as invalid, the X value of the appropriate pixel is set to 1. Y and Z values are set to 0.

5.4.2 Temperature compensation

The camera firmware continuously monitors temperatures of the frontend and corrects the measured distance with cubic polynomials.

5.5 Camera Coordinate System

The camera coordinate system is depicted in Figure 5-2.

Pixel numbering starts in the upper left corner of the pixel array, seen from the camera's point of view.

Distance data always contains the measured distance from the ToF sensor to the viewed scene.

Point cloud data contains X, Y and Z coordinates for each ToF pixel. Point cloud data is calculated via intrinsic lens parameters for the ToF sensor and optical system, where X=Y=Z=0, is the front-top-right edge of the camera, seen from the camera's point of view.

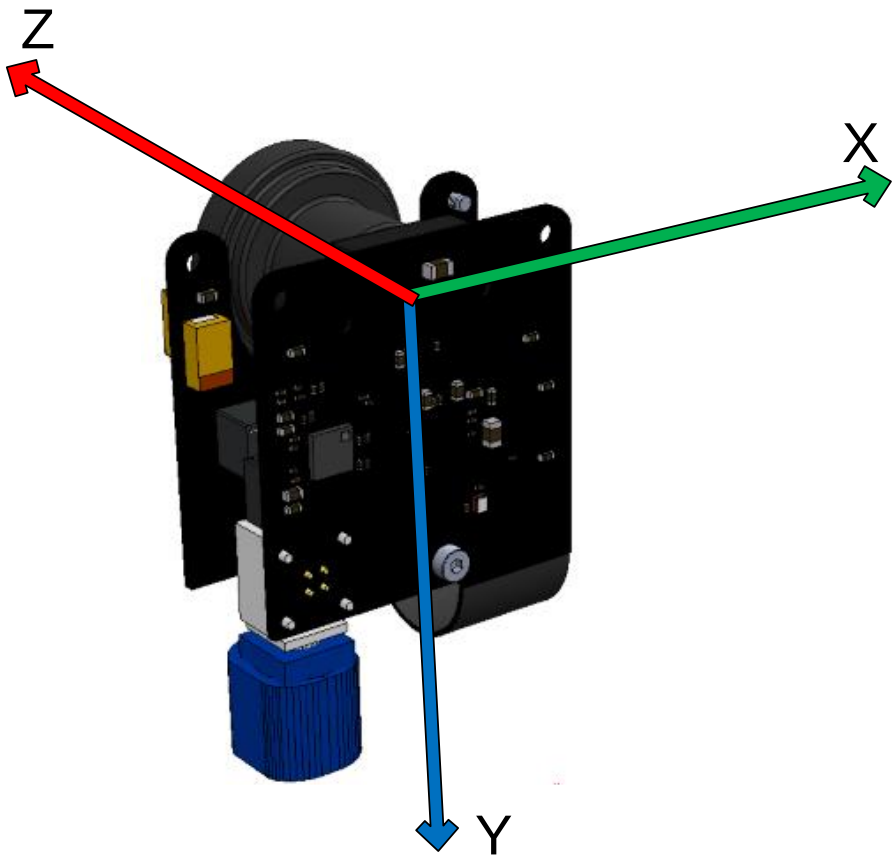


Figure 5-2 multi-ToF platform Default Coordinate System (1)

Note

On protocol level the coordinate system may differ. Please contact Bluetechnix support for more information.

For point cloud calculation, the camera requires proper calibration file (Lens calibration). This calibrations are identified by the ***HardwareConfiguration*** identifier (register). To check for proper calibration, registers ***CalibStatus*** and ***CalibStatus2*** may be read.

5.6 Camera Data Format

The camera provides up to four data channels via its data interface. The meaning of each data channel depends on the selected data format. The factory default setting provides an array of distance data and an array of amplitude data.

Alternatively, a 3D XYZ point cloud can be provided. Refer to chapter 5.4 for a description of the coordinate system of the camera.

Note

Data formats with XYZ point clouds cannot be enabled if there is no proper lens calibration data stored on the camera. Please use register ***HardwareConfig*** to configure the ToF lens calibration hardware identifier and register ***CalibStatus*** to read the status of lens calibration availability.

Color data from the CMOS image sensor can also be selected.

A channel can carry one of the following data types:

- Distance data from the ToF sensor, in millimeters, as 16-bit unsigned (Uint16) values. Resolution is always 304x240 pixels.
- Amplitude data from the ToF sensor, as 16-bit unsigned (Uint16) values. Resolution is always 304x240 pixels.
- Confidence data, as 8-bit unsigned values. Resolution is always 304x240 pixels.
- Z coordinate values, in millimeters, as 16-bit signed (Int16) values. No negative values allowed. Resolution is always 304x240.
- X coordinate values, in millimeters, as 16-bit signed (Int16) values. Resolution is always 304x240.
- Y coordinate values: Same as X.
- Raw depth data (without scaling, corrections and filters), as 16-bit unsigned (Uint16) values. Resolution is always 304x240 pixels.
- Raw phase data (0°, 90°, 180°, 270°), as 16-bit unsigned (Uint16) values. Resolution is always 304x240 pixels.

Which image format will be transferred can be selected by the register ***ImageDataFormat***. The following sections describe 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 chapter 3.2.

5.6.1 Distances and Amplitudes

In this mode the distances and amplitudes will be transferred in progressive mode, first the distance array (channel 0), then the amplitude array (channel 1). The stream starts always with pixel #0.

The **distances** are coded in **millimeters** as **Uint16**, the **amplitudes** also as **Uint16**.

5.6.2 XYZ Point Cloud

In this mode the XYZ point cloud will be transferred in progressive mode, first the X coordinate array (channel 0) then the Y (channel 1) and Z (channel 2) coordinate array. The stream starts always with pixel #0.

The **coordinates** are coded in **millimeters** as **Int16.**

5.6.3 XYZ Point Cloud and Amplitudes

In this mode the XYZ point cloud and the amplitude will be transferred in progressive mode, first the X coordinate array (channel 0), then the Y (channel 1) and Z (channel 2) coordinate array, then the amplitude array (channel 3). The stream starts always with pixel #0.

The **coordinates** are coded in **millimeters** as **Int16** the **amplitudes** as **Uint16.**

5.6.4 Distances and XYZ Point Cloud

In this mode the distances and the XYZ point cloud will be transferred in progressive mode, first the distances array (channel 0), then X (channel 1), Y (channel 2), and Z (channel 3) coordinate arrays. The stream starts always with pixel #0.

The **distances** are coded in millimeters as Uint16. The **coordinates** are coded in **millimeters** as **Int16**.

5.6.5 X Coordinate and Amplitudes

In this mode a single coordinate array, more specifically, the one belonging to the optical axis of the camera (X), is transferred in channel 0, as well as the amplitudes (channel 1).

Coordinate values are coded in **millimeters** as **Int16**. The **amplitudes** are coded as **Uint16**.

5.6.6 Distances

In this mode a single array with distances is transferred in channel 0. The stream starts always with pixel #0.

The **distances** are coded in **millimeters** as **Uint16**.

5.6.7 4 phases without image processing

In this mode the 4 phases (0°, 90°, 180° and 270°) will be transferred in progressive mode, as 16-bit unsigned (Uint16) values

5.6.8 Raw Distances and Amplitudes

In this mode, raw distance values are transferred in channel 0, and amplitude values are transferred in channel 1.

Raw distance values are forwarded as retrieved from the TIM. No conversion to millimeters and no corrections are performed on the values. Filter settings for raw distance values are ignored as well.

5.6.9 Distance, Amplitudes, and Confidences

In this mode, the distances, amplitudes, and confidences will be transferred in progressive mode, first the distance array (channel 0), then the amplitude array (channel 1) and confidence array (channel 2). The stream starts always with pixel #0.

The **distances** are coded in **millimeters** as **Uint16**. The **amplitudes** pixels are in **Uint16** format. The **confidence** pixels are in ***Uint8*** format.

The confidence is a measure of the confidence of the corresponding depth value. It is an exponential function of a target amplitude to the current amplitude. A value of 255 (maximum value) means 100% confidence, whereas a value of 0 (minimum value) means a confidence of 0%.

5.7 ToF Modulation Frequency

The modulation frequency of the illumination is set to 40 MHz per default. Other modulation frequencies can be set using register ***ModulationFrequency***. Be aware that this also changes the ambiguity range of the camera. On writing these registers, if inexact values are supplied, the camera searches for the next possible modulation frequency automatically.

The following modulation frequencies can be selected:

Frequency	Unambiguity Range
12 MHz	12,5m
16 MHz	9,375m
20 MHz	7,5m
24 MHz	6,25m
28 MHz	5,357m
32 MHz	4,688m
36 MHz	4,167m
40 MHz	3,75m

Table 5-1: Pre-defined modulation frequencies

The register content is the frequency in 10-kHz-steps (frequency in Hz/10000). On a register read, the currently selected modulation frequency (again, in 10-kHz-steps), is returned.

Note

The BltTofSuite demo application displays and takes modulation frequencies in MHz and calculates the register content transparently!

5.8 Frame Rate and Integration Time

The frame rate and the integration time of the ToF sensor can be set by using the registers ***Framerate*** and ***IntegrationTime***/***IntTimeSeq1***/***IntTimeSeq2***/***IntTimeSeq3*** (for sequence 0/1/2/3). The ToF sensor's integration time is limited to 1,5ms.

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 follows:

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

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.

5.9 Distance Offset Calibration

Each multi-ToF platform is offset-calibrated for the preconfigured modulation frequency (40MHz) out of factory.

For each pre-defined modulation frequency, there is an absolute offset in millimeters that all distance values are corrected with. The absolute offsets are stored in registers ***DistOffset0*** (for 12 MHz modulation frequency) to ***DistOffset7*** (for 40 MHz). Offsets can be modified by direct register writes.

Distance Offset correction is enabled or disabled in register ***ImgProcConfig***.

The camera has also a built-in offset calibration function, which is described in the following procedure. You let the camera know the real distance and the camera will calculate the correct absolute offset. It uses a square of 4x4 pixels in the center of the distance image.

Procedure

Offset Calibration

1. Place your camera co-planar in front of a uniform white target.
2. Avoid extreme environmental light conditions or avoid light completely.
3. Set the desired modulation frequency using register ***ModulationFrequency***.
4. Check the amplitude in the center of the image and increase/decrease the ***IntegrationTime*** until the amplitude is about 15000 in the center.
5. Measure the real distance in millimeters from the camera to the white target. Write this value into register ***RealWorldXCoordinate***.
6. Write decimal value 19 into register ***CalibrationCommand*** and wait until the CalibrationExtended register, Bits 0..7, read decimal value 161 (= finished).

➤ The appropriate ***DistOffsetX*** register is updated.

7. If you want to keep the setting, don't forget to save registers to flash (Chapter 5.11).

5.10 Temperature Sensors

5.10.1 Over Temperature Protection

The multi-ToF platform firmware has a built-in monitoring for over-temperature condition of the laser diodes. If the temperature exceeds 70°C, the camera will automatically stop illumination, until temperature is below 68°C.

During over-temperature condition, bit 9 of the ***Status*** register is set.

The maximum allowed temperature for the LEDs can be changed in register ***MaxLedTemp***. However, we do not recommend to set values larger than 70°C, because it drastically reduces the lifetime of the LEDs.

5.11 Save Registers

The entire register map can be saved into non-volatile 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.

After camera boot, one may query a ready bit which indicates that all the settings have been applied within the camera: Bit 1 of register ***Ready***.

5.12 Reset to Factory Default

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

5.13 Error Indication

The multi-ToF platform indicates detected errors mainly in the ***Status*** register:

- Bit 3: Indicates a temperature measurement error on at least one LIM. The bit is automatically cleared if the error disappears.
- Bit 4: Indicates a temperature measurement error on the TIM. The bit is automatically cleared if the error disappears.
- Bit 5: Indicates that calibration data are missing, refer to ***CalibStatus*** and ***CalibStatus2***.
- Bit 8: There were errors during firmware boot, the previous firmware was discovered. This bit is not set again after a reboot.
- Bit 9: Indicates the LIM temperature has exceeded the maximum tolerable value
- Bit 11: The status of at least one LIM could not be retrieved, or there is an error condition in one of the LIM status registers ***Lim1Status*** and ***Lim2Status***.
- Bit 15: Indicates a communication error with the TIM, or an error with triggering the TIM. This bit is automatically cleared if the error disappears.

6 Register Description

Note

Some critical registers are password protected. To enable the functionality a specific value must be written to the ***CmdEnablePasswd*** register in advance to enable the functionality. This should prevent from accidentally executing certain functions.

6.1 General

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
0001	Mode0	0001	R/W	Bit[0]: 0…Manual Mode, 1…Video Mode Bit[4]: 1…Manual Trigger (self-clearing bit) Bit[6]: 1…Clear status register (self-clearing bit) Bit[11]: 1…3D Snapshot Trigger (self-clearing bit
0003	Status	0040	R	Bit[2]: 1…Ongoing Calibration Bit[3]: 1…LIM temperature sensor error (Temperature sensors not found, or temporary temperature reading error) Bit[4]: 1…TIM temperature sensor error Bit[5]: 1…Calibration data missing Bit[6]: 1…Factory Regmap was loaded Bit[8]: 1…Previous firmware version was restored Bit[9]: 1…LIM over-temperature Bit[10]: 1…Frame rate or integration time was limited due to PoE constraints Bit[11]: 1…Error condition reported by one of the LIMs, see registers ***Lim1Status*** and ***Lim2Status*** for details Bit[13]: 1…Color sensor stream error Bit[14]: 1…Base board temperature sensor error Bit[15]: 1…TIM error (I2C communication, triggering)
0004	ImageDataFormat	0000	R/W	Bit[3:10]: 0 … Distances and Amplitudes 1 … Distances, Amplitudes, and Confidences 3 … XYZ Point Cloud 4 … XYZ Point Cloud and Amplitudes 9 … Distances and XYZ Point Cloud 10 … Z Coordinate and Amplitudes 12 … Distances 13 … Raw Distances and Amplitudes
0005	IntegrationTime	05DC	R/W	Integration Time [µs] (min: 1, max: 1500)
0006	DeviceType	03FC	R	Hardware specific identification
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 in multiples of 10kHz
000A	Framerate	0028	R/W	ToF frame rate [Hz]
000B	HardwareConfiguration		R/W	Lens opening angle identifier
000C	SerialNumberLowWord		R	Lower 16bit of the 32bit Serial Number

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
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 13…FPPN calibration for the current modulation frequency. Exactly one sequence must be configured in register ***NofSequ***. 16…Clear FPPN calibration data for the current modulation frequency. Always takes modulation frequency of first sequence! 19…Calibrate DistOffset for the current modulation frequency. Exactly one sequence must be configured in register ***NofSequ***.
0010	ConfidenceThresLow	03E8	R/W	Amplitude threshold for valid distance data
0011	ConfidenceThresHigh	EA60	R/W	Amplitude threshold for valid distance data
001B	LedboardTemp		R	Temperature of LIMs in 0.01[°C] (FFFF: Sensor not available).
001C	MainboardTemp		R	Temperature of ToF sensor in 0.01[°C] (FFFF: Sensor not available).
0020	RealWorldXcoordinate	0000	R/W	Distance to the calibration target [mm].
0021	CalibrationExtended	0000	R	Bit[0-7]: Status/error 0…Idle 19…FPPN calibration 20…Erasing flash 21…DistOffset calibration 161…Operation done 246…Wrong image mode (Need distance) or Mode0 setting (Need video mode) 248.. Invalid modulation frequency 255…Generic error Bit[10]: 1…Error occurred Bit[12]: 1…No FPPN Calibration data in NVM Bit[14]: 1…No Lens Calibration data in NVM
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	1B58	R/W	Maximum tolerable LIM 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[°]
002D	TempCompGradientLim		R/W	Factor ‘c’ of the illumination temperature compensation function: y [mm] = a/100000 * x^3 + b/10000 * x^2 + c/1000 * x
0030	TempCompGradient2Lim		R/W	Factor ‘b’ of the illumination temperature compensation function: y [mm] = a/100000 * x^3 + b/10000 * x^2 + c/1000 * x
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

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
				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
0035	FactoryMacAddr2		R	Hi byte and byte 4 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	TempCompGradient3Lim		R/W	Factor ‘a’ of the temperature compensation function: y [mm] = a/100000 * x^3 + b/10000 * x^2 + c/1000 * x
003D	BuildYearMonth		R	Firmware Build date/time Bit[14-4]: Year Bit[3-0]: Month
003E	BuildDayHour		R	Firmware Build day/hour Bit[9-5]: Day Bit[4-0]: Hour
003F	BuildMinuteSecond		R	Firmware 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	AcfPlausCheckAmpLimit	0064	R/W	Limit for the ACF plausibility check
0046	ProcessorStatus		R	Bit[0:7]…Temperature of the processor in °C (0xFF: Sensor not available) Bit[8:15]…Processor speed in 10-MHz-steps
0048	Lim1Status	0000	R	Status word of LIM #1 or external illumination. Bits[0..4]: Overcurrent on LED segments 0..4 Bits[8..12]: Open load on LED segments 0..4 Bit[14]: Temperature sensor error Bit[15]: Could not read status word (communication failure)
004A	TempCompGradientTim		R/W	Factor ‘c’ of the ToF sensor temperature compensation function: y [mm] = a/100000 * x^3 + b/10000 * x^2 + c/1000 * x
004B	TempCompGradient2Tim		R/W	Factor ‘b’ of the ToF sensor temperature compensation function: y [mm] = a/100000 * x^3 + b/10000 * x^2 + c/1000 * x

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
004C	TempCompGradient3Tim		R/W	Factor 'a' of the ToF sensor temperature compensation function: y [mm] = a/100000 * x^3 + b/10000 * x^2 + c/1000 * x

Table 6-1: General registers

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

6.2 Distance Offset

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
00C1	DistOffset0		R/W	An offset for distance values when operating at modulation frequency 5 MHz
00C2	DistOffset1		R/W	An offset for distance values when operating at modulation frequency 7.5 MHz
00C3	DistOffset2		R/W	An offset for distance values when operating at modulation frequency 10 MHz
00C4	DistOffset3		R/W	An offset for distance values when operating at modulation frequency 15 MHz
00C5	DistOffset4		R/W	An offset for distance values when operating at modulation frequency 20 MHz
00C6	DistOffset5		R/W	An offset for distance values when operating at modulation frequency 25 MHz
00C7	DistOffset6		R/W	An offset for distance values when operating at modulation frequency 30 MHz
00C8	DistOffset7		R/W	An offset for distance values when operating at modulation frequency 40 MHz

Table 6-2: Distance Offset registers

6.3 User Defined

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
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: User Defined registers

6.4 General

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
010A	TempCompGradientBaseboard		R/W	Factor 'c' of the ToF base board temperature compensation function: y [mm] = a/100000 * x^3 + b/10000 * x^2 + c/1000 * x + u
010B	TempCompGradient2Baseboard		R/W	Factor 'b' of the ToF base board temperature compensation function: y [mm] = a/100000 * x^3 + b/10000 * x^2 + c/1000 * x + u
010C	TempCompGradient3Baseboard		R/W	Factor 'a' of the ToF base board temperature compensation function: y [mm] = a/100000 * x^3 + b/10000 * x^2 + c/1000 * x + u
010D	BaseboardTemp		R	Temperature of baseboard in 0,01[°C] (FFFF: Sensor not available).
0118	CalibStatus2		R	Bit[0]: … No wiggling calibration data in NVM Bit[1]: 1 … No geometric model parameters for 3D sensor in NVM Bit[2]: 1 … No overlay calibration data in NVM Bit[3]: 1 … No geometric model parameters for 2D sensor in NVM

Table 6-4: General registers

6.5 Sequencing

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
0120	NofSequ	1	R/W	Number of sequences recorded by the ToF sensor without wait time in between, 1…4
0121	IntTimeSeq1	05DC	R/W	Integration time to be used for capturing sequence 1 NOTE: Sequence 0 integration time is set via register ***IntegrationTime***
0122	IntTimeSeq2	05DC	R/W	Integration time to be used for capturing sequence 2
0123	IntTimeSeq3	05DC	R/W	Integration time to be used for capturing sequence 3
0128	ModFreqSeq1	07D0	R/W	Modulation frequency to be used for capturing sequence 1 Register description: See ***ModulationFrequency***
0129	ModFreqSeq2	07D0	R/W	Modulation frequency to be used for capturing sequence 2 Register description: See ***ModulationFrequency***
012A	ModFreqSeq3	07D0	R/W	Modulation frequency to be used for capturing sequence 3 Register description: See ***ModulationFrequency***

Table 6-5: Registers for Sequencing

6.6 Device Update

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
01D1	FileUpdateStatus	0000	R	0…idle 3…out_of_memory 6…file crc error 8…erasing flash 9…flashing 11…erasing failed 12…flashing failed 14…update success 16…header version conflict 18…wrong fw identifier 20…data inconsistent 21…in progress 255…protocol violation

Table 6-6: Registers for device update

6.7 Filter Configuration

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
01E0	ImgProcConfig	28C0	R/W	Bit[0]: 1…enable Median Filter for Distance Image Bit[1]: 1… enable Average Filter Bit[3]: 1…enable Bilateral Filter for Distance Image Bit[4]: 1…enable Sliding Average for Distance Image Bit[6]: 1…enable wiggling compensation for Distance Image Bit[7]: 1…enable FPPN compensation for Distance Image Bit[10]: 1…enable FrameAverage Filter for Distance Image Bit[11]: 1…enable temperature compensation for Distance Image Bit[13]: 1…enable offsets via registers ***DistOffsetX*** (0x00C1 onwards) for Distance Image Bit[14]: 1… enable ACF plausibility check (affected pixels 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 2… 2x2 Pixel Bit[8-15]: Nr of iterations
01E4	FilterBilateralConfig	13DE	R/W	Bit[0-5]: Sigma R (Width of range kernel) Bit[6-11]: Sigma S (Width of spatial kernel) Bit[12-15]: Nr. Of iterations
01E5	FilterSlafConfig	0005	R/W	Bit[0-7]: Window size
01E6	FilterBilateralConfig2	0003	R/W	Bit[0-5]: Square size (=> Window size = square size x square size)

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
01E7	FilterFrameAverageConfig	0002	R/W	Bit[0-7]: Number of Frames
01E9	ImgProcConfig2		R/W	Bit[0]: 1… enable Median Filter for Amplitude Image Bit[1]: 1… enable Bilateral Filter for Amplitude Image Bit[2]: 1… enable Sliding Average Filter for Amplitude Image Bit[3]: 1… enable FrameAverage Filter for Amplitude Image

Table 6-7: Register for filter configuration

6.8 Advanced Image processing

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
01F0	ImgProcAdvanced	0	R/W	Bit[0]: 1… enable Combine/HDR Mode (combine several sequences to one) Bit[3]: 1… enable path length correction Bit[7]: 1 … enable Vernier

Table 6-8: Registers for Advanced Image Processing

6.9 Ethernet configuration

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
024B	Eth0TcpCtrlPort	2711	R/W	Port for TCP control interface
024C	Eth0UdpStreamIp0	0001	R/W	Low word of IP address for UDP stream Writing this register has no immediate effect.
024D	Eth0UdpStreamIp1	E000	R/W	High word of IP address for UDP stream **Writing this register will update the network configuration with the new UDP stream address.**
024E	Eth0UdpStreamPort	2712	R/W	Port for UDP streaming
0256	Eth0UdpColorStreamIp0	0001	R/W	Low word of destination IP address for Color sensor UDP data stream
0257	Eth0UdpColorStreamIp1	E000	R/W	High word of destination IP address for Color sensor UDP data stream **Writing this register will update the IP address.**
0258	Eth0UdpColorStreamPort	2716	R/W	UDP port for Color sensor UDP data stream
0259	Eth0UdpPacketSize	0578	R/W	Packet size for UDP data interface
025A	Eth0LinkSpeed	03E8	R	Link speed [Mbps]

Table 6-9: Registers for Ethernet configuration

7 Firmware History

7.1 Version Information

Firmware Version	Status	Release date	Changes
0.3.1	Beta	Feb 2018	

Table 7-1: Overview multi-ToF platform firmware changes

Note

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

7.2 Anomalies

Applies to	Date	Description
0.3.1	Feb 2018	The produced frame rate doesn't match exactly the value set.
0.3.1	Feb 2018	Single frame trigger is not working

Table 7-2: Firmware anomalies

8 Software

8.1 BltTofApi

SDK for ToF products: Bluetechnix 'Time of Flight' API

In order to create a common interface for our products we define the interfaces between a ToF device and an application. The main part of this model is the *BltTofApi* which is written in C for platform independency.

Please visit our support Wiki to get information and to download the SDK.

Bluetechnix 'Time of Flight' API

https://support.bluetechnix.at/wiki/ (Section Software)

8.2 MATLAB SDK

MATLAB SDK for ToF products: BltTofApi Matlab SDK

The MATLAB SDK is able to access the *BltTofApi* interface and will therefore be compatible with any device with an existing library implementing the *BltTofApi*.

Bluetechnix 'Time of Flight' API Matlab SDK

https://support.bluetechnix.at/wiki/ (Section Software)

8.3 BltTofSuite

For the first evaluation of the camera and to evaluate different settings and configurations a .NET demo application for Microsoft Windows is provided: *BltTofSuite*. The demo application can be downloaded from our support web site.

Software and documentation

https://support.bluetechnix.at/index.html

9 Support

9.1 General Support

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

Support Link

https://support.bluetechnix.at/index.html

9.2 Software Downloads

Camera support packages are available for registered customers only. Please contact Bluetechnix support if you do not yet have an account.

Software Download Portal

https://support.bluetechnix.at/software/

9.3 Camera Development Package

The camera offers the possibility to bring your own application onto the multi-ToF platform.

The multi-ToF platform is based on an embedded ARM Linux system based on the i.MX6 Quad-core processor from Freescale Inc.

Please contact Bluetechnix support for more information.

10 Document Revision History

Version	Date	Document Revision
1	2018 02 19	Initial version of the document

Table 10-1: Revision history

A List of Figures and Tables

Figures

Figure 3-1: UDP streaming data format 8
Figure 5-1: Image processing flow 13
Figure 5-2 multi-ToF platform Default Coordinate System (1) 16

Tables

Table 3-1: Default login credentials 9
Table 3-2: Debug UART settings 10
Table 5-1: Pre-defined modulation frequencies 19
Table 6-1: General registers 25
Table 6-2: Distance Offset registers 25
Table 6-3: User Defined registers 25
Table 6-4: General registers 26
Table 6-5: Registers for Sequencing 26
Table 6-6: Registers for device update 27
Table 6-7: Register for filter configuration 28
Table 6-8: Registers for Advanced Image Processing 28
Table 6-9: Registers for Ethernet configuration 28
Table 7-1: Overview multi-ToF platform firmware changes 29
Table 7-2: Firmware anomalies 29
Table 10-1: Revision history 32