Argos3D - P33x

Software User Manual

BECOM Systems GmbH

Gutheil-Schoder-Gasse 17

1230 Wien

AUSTRIA

office.systems@becom-group.com

systems.becom-group.com

Argos3D - P33x – Software User Manual

Document No.: 900-308 / A

Publication date: June 29, 2020

Subject to change without notice. Errors excepted.

Windows is a registered trademark of Microsoft.

Table of Contents

© BECOM Systems GmbH 2020

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.

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

BECOM Systems 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 BECOM Systems www.becom-group.com

Warning

Due to technical requirements components may contain dangerous substances.

1   General Information

This guide applies to the Argos3D - P33x camera platform from BECOM Systems. 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 Argos3D - P33x and describes the firmware dependent interfaces.

This document applies to firmware version 1.0.x.

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

Software and documentation

support.systems.becom-group.com

3   Interfacing

The Argos3D - P33x provides control and data interfaces via Gigabit-Ethernet.

The control interface is used to set and read the configuration of the Argos3D - P33x via a set of registers. Refer to chapter 6 for a detailed register description.

The data interface provides a continuous stream of ToF (and color) data depending on the configuration.

Note

BECOM Systems provides an abstraction of control and data interfaces by means of the *BltTofApi*. If you use this API, you need not be familiar with control and data interface in detail. Refer to chapter 7.1 for the *BltTofApi*.

3.1   Control Interface

The Argos3D - P33x can be configured using a TCP/IP connection. For the control interface the Argos3D - P33x 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 writing register **Eth0Gateway1**.

Once a TCP connection has been established the Argos3D - P33x can be configured using a dedicated set of command frames. The camera 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 Argos3D - P33x
Flash Update	Used to either update the firmware or the boot loader
Alive	Used to keep the TCP control connection alive. If no command is sent for 10 seconds, the camera closes the control interface connection and waits for a new incoming connection request.

Command frame	Description
Up to 5 concurrent control connections are supported.	

Table 3-1: Supported command frames

Note

The control interface is covered by the BltTofApi. For information please refer to
support.systems.becom-group.com

3.2 Data Interface

A UDP stream delivers ToF and/or color data from the Argos3D - P33x. 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).
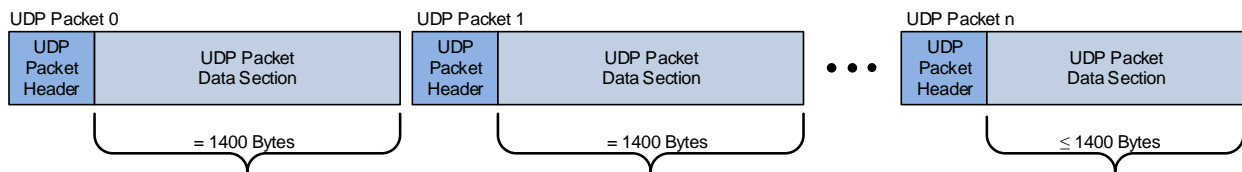
Figure 3-1: UDP streaming data format

The UDP streaming is enabled by factory default. The Argos3D - P33x streams to the following IP settings:

- **IP-Address**: Multicast address 224.0.0.1

- **UDP port**: 10002

Note

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

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 camera 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 your Ethernet network as the stream must be spread to all active links of switches/hubs and routers.

Note

The data interface is covered by the BltTofApi. For information please refer to
support.systems.becom-group.com

3.3   Camera Discovery

The Argos3D - P33x 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 7

3.4   Manual frame triggers

The default mode of the camera is video mode, where it streams continuously with configured frame rate. To use manual frame triggering, the video mode must be disabled in register **Mode0**.

A frame can be triggered by either

- Hardware trigger

- Software trigger: See register **Mode0**.

Both will trigger the capturing of as many sequences on the ToF sensor, as is configured in register **NofSequ**, as well as a transition to low on the trigger output.

If color data is transferred via the data interface, the color sensor operates at either 15 or 30 frames per second in the background (depending on register settings in **ColorStreamParams**). A color image is assigned to each captured ToF image by means of both internal capture timestamps. If no color image is available (if the ToF sensor frame rate is higher than the color sensor frame rate), an empty color image is transferred.

3.5   External Illumination Interface

Please contact BECOM Systems support for information about the external illumination interface.

3.6   GPIOs

The camera features 2 general-purpose inputs and 2 or 4 general-purpose outputs. Please see the register description in chapter 5.3 for more information.

3.7   Status Indicator LED

The Argos3D - P33x camera features an RGB LED to indicate its internal status via different colors:

- Green: No error condition detected

- Yellow: Calibration data is missing

- Red: The input voltage is outside the required range (18-29V)

- Red blinking (ca. 1 Hz): Error condition detected, please consult *Status* register.

3.8 PoE power control

If the camera is powered via a PoE connection (and not via VAUX), the firmware is able to detect the PoE power class it is currently connected to.

It will automatically adjust its current consumption to the maximum allowed power.

The following classes can be detected (see register *PoEStatus*):

Power class	Max. allowed power	Action
PoE	13 W	Power limitation to 13 W [1]
PoE+	25.5 W	Power limitation to 25.5 W
PoE++	not specified	No power limitation
VAUX supply	not specified	No power limitation

Table 3-2: Supported power supplies

Note 1): PoE mode does not allow for any illumination, so there is no reasonable usage of the camera with this power supply.

The power is limited by limiting the max. allowed FITP (Frame rate-Integration Time-Product), i.e., by limiting either the frame rate or the integration time (or both). Please see chapter 4.9 for details.

The detected PoE class can be overridden by means of register **PoEOverride**.

3.9 Secure Shell (SSH) Login

The Argos3D - P33x camera features an OpenSSH server listening to TCP port 22.

	Root account	User account
Username	root	user
Default password	root	user

Table 3-3: Default login credentials

3.9.1 Change default password

Change default password

1. Log in via SSH, e.g., `ssh root@192.168.0.10`
2. Type `passwd root` or `passwd user`
3. Supply the new password for two times
4. Copy the file `/etc/shadow` (containing the encrypted passwords) to the non-volatile settings partition, to be restored again on next camera reboot: `cp /etc/shadow /mnt/settings`

3.10 Debug UART

Note

Using the Debug UART is optional.

The Argos3D - P33x 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 for the FTDI FT234 device.

Windows OS Device Driver Download

http://www.ftdichip.com/Drivers/VCP.htm

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 3-4: Debug UART settings

The Debug UART also allows to log in to the camera's Linux OS. Please see chapter 3.9 for the default login and how to change it.

4 Camera Features

4.1 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.

4.2 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.
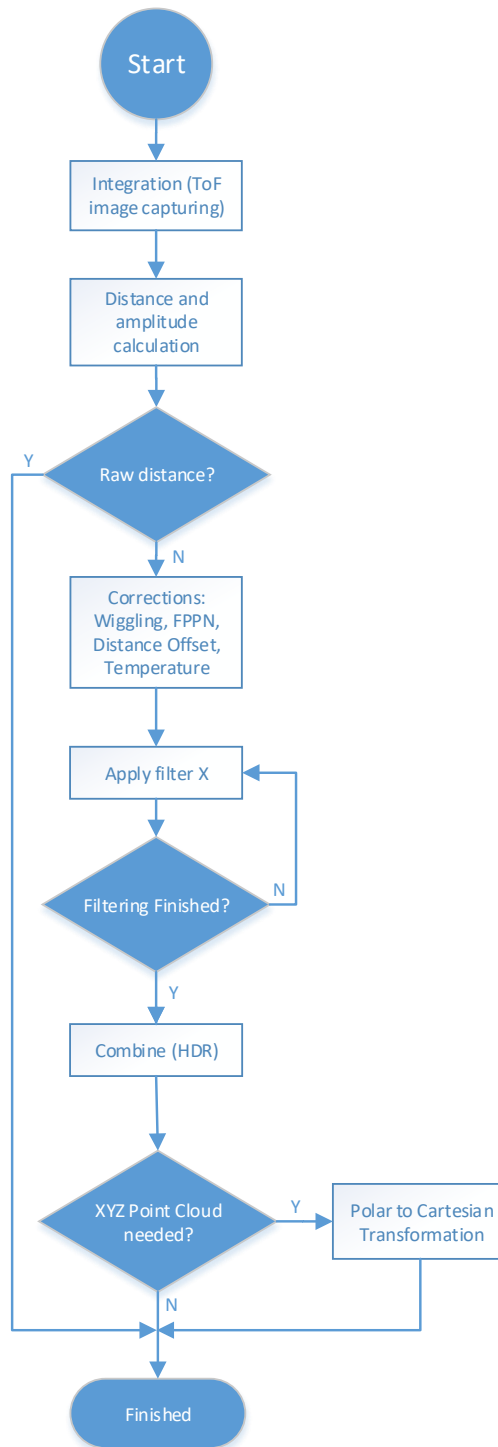
Figure 4-1: Image processing flow

4.3 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) and **_ImgProcConfig2_** (amplitude data) registers.

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

4.3.1   Median Filter

A 3x3 median filter can be applied.

Register: *FilterMedianConfig*

The number of iterations is configurable.

4.3.2   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.

4.3.3   Average filter

Registers: **FilterAverageConfig**

Configuration option is the filter size.

4.3.4   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.

4.3.5   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.

4.4 ToF Image Enhancements

4.4.1 Pixel invalidation

The Argos3D - P33x provides an on-board check for invalid pixels:

- Underexposed pixels: The amplitude is too low for the distance value to be trustworthy. The Argos3D - P33x 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 Argos3D - P33x sets the pixel distance to the minimum value. The threshold is set via register **ConfidenceThresHigh**.

- Invalid pixels: The Argos3D - P33x 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.

4.4.1.1 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.

4.4.1.2 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.

4.4.2 Temperature compensation

The camera firmware continuously monitors temperatures of LIMs, TIM, and base board, and corrects the measured distance with cubic polynomials.

4.4.3 Combine sequences (HDR)

In order to improve the accuracy of distance data, the camera features the ability to combine up to 4 sequences from the ToF sensor into one frame with improved accuracy. It uses the confidence values of the input frames to calculate a new distance value for each pixel of the output frame.

The following registers are used to configure Combine mode:

- **NofSequ**: To configure up to 4 sequences. Minimum of 2 required for Combine mode.

- **IntegrationTime**, **IntTimeSeq1**, **IntTimeSeq2**, **IntTimeSeq3**: To configure integration times of up to 4 sequences

- **ModulationFrequency**, **ModFreqSeq1**, **ModFreqSeq2**, **ModFreqSeq3**: To configure modulation frequencies of the sequences

- **ImgProcAdvanced**, bit 0: To enable combine mode

It is recommended that all sequences are recorded with identical modulation frequency when using Combine mode.

4.4.4 Vernier

In order to combine the benefits of a smaller modulation frequency (higher unambiguity range) and a higher modulation frequency (higher precision and accuracy), the camera features the ability to combine two sequences with different modulation frequency into one frame.

The following registers are used to configure Combine mode:

- **NofSequ**: To configure 2 or 4 sequences.

- **IntegrationTime**, **IntTimeSeq1**, **IntTimeSeq2**, **IntTimeSeq3**: To configure integration times of up to 4 sequences

- **ModulationFrequency**, **ModFreqSeq1**, **ModFreqSeq2**, **ModFreqSeq3**: To configure modulation frequencies of the sequences

- **ImgProcAdvanced**, bit 7: To enable combine mode

If **NofSequ** gets configured to 4, the first and the second sequence and the third and the fourth sequence get calculated to one frame.

	Note
	The modulation frequency of sequence 0 must be lower than the modulation frequency of sequence 1 and the modulation frequency of sequence 2 must be lower than the modulation frequency of sequence 3

4.4.5 Vernier + Combine sequences (HDR)

Vernier and Combine sequences can be enabled together to get a high precision frame.

Example register settings:

- **NofSequ**: 4.

- **IntegrationTime**: 600

- **IntTimeSeq1:** 600

- **IntTimeSeq2:** 5200

- **IntTimeSeq3:** 5200

- **ModulationFrequency:** 20MHz

- **ModFreqSeq1:** 80MHz

- **ModFreqSeq2:** 20MHz

- **ModFreqSeq3:** 80MHz

- **ImgProcAdvanced**, set bit 0 and bit 7

4.5 Camera Coordinate System

The camera coordinate system is depicted in Figure 4-3.

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, and via extrinsic parameters. The reference point of the camera

coordinate system, where X=Y=Z=0, is the front-top-right edge of the camera, seen from the camera's point of view.
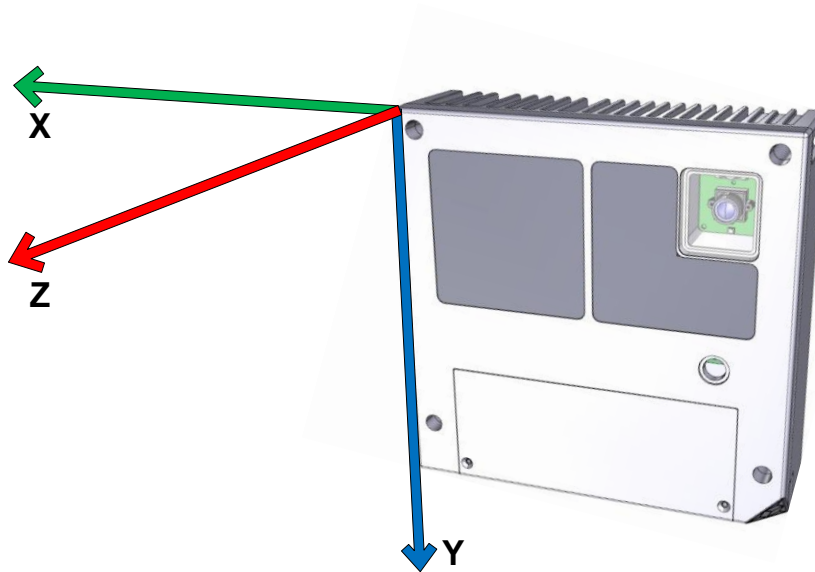
Figure 4-2 Argos3D - P33x Default Coordinate System (1)
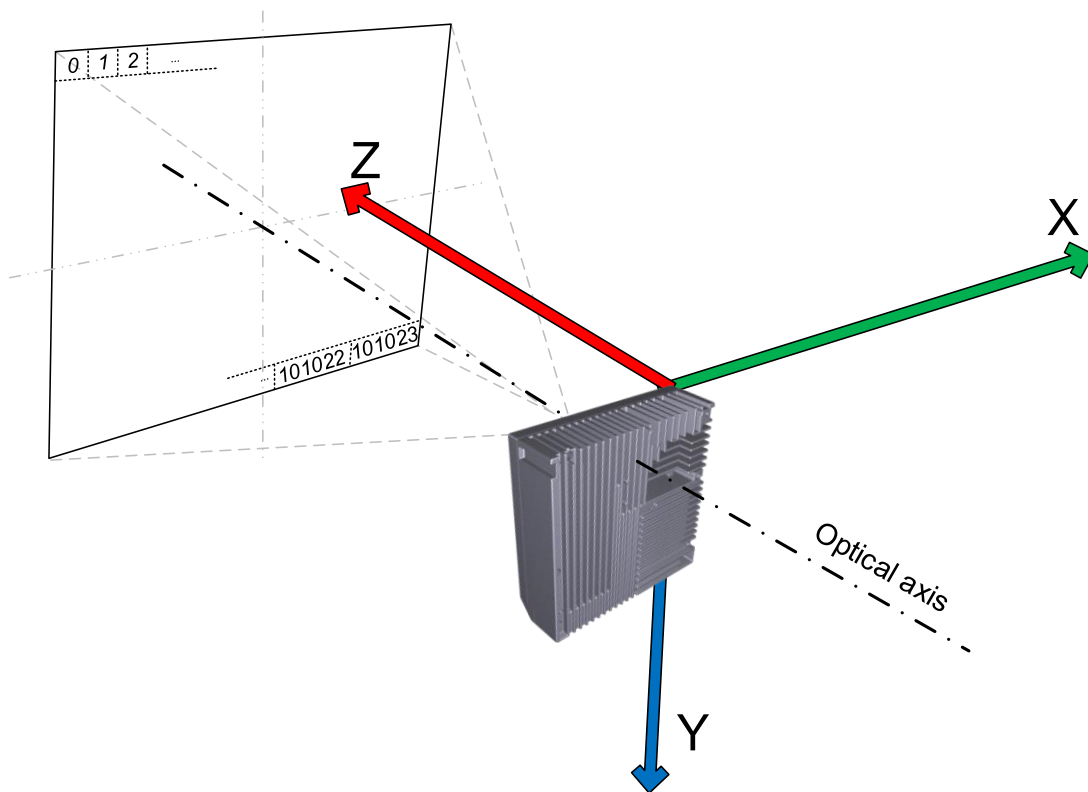
Figure 4-3: Argos3D - P33x Default Coordinate System (2)

Note

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

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

4.6   Color Overlay

The Argos3D - P33x features a color sensor additionally to the ToF sensor. It is able to generate an RGB565 color map that assigns each ToF pixel a corresponding color value from the color sensor. In order to accomplish the mapping, the camera calculates the XYZ point cloud first, and then projects each of these points back to the color sensor array to determine the correct color. Hidden areas are detected by means of ray-tracing, in which case no color or a special color can be assigned (configurable).

Note

The Argos3D - P33x is not delivered fully calibrated for Overlay mode by default. If you wish to upgrade a camera, please consult BECOM Systems Support for an offer.

4.6.1   Camera Settings

To select the overlay image data format, please consult chapter 4.7.8. If you wish to upgrade a camera, please consult BECOM Systems Support for an offer.

For overlay mode, the camera needs certain calibration data which is stored in non-volatile RAM. You can check for these by means of status register bits:

- Lens calibration: Register *CalibStatus*

- Geometric model parameters: Register *CalibStatus2*

- Overlay calibration: Register *CalibStatus2*

The calibration data is identified by means of two registers:

- *HardwareConfiguration*: Identifier for ToF optical system (We often use the vertical opening angle as

  identifier).

- *HardwareConfigColor*: Identifier for color sensor optical system (We often use the vertical opening

  angle as identifier).

Color sensor parameters must be configured using register **ColorStreamParams**. Note that the data format has to be RGB565 and the resolution has to be 640x480 (VGA). The frame rate may be chosen between 15 or 30 fps.

Overlay mode is enabled by selecting mode 5 in register **ImageDataFormat**.

Note

If the camera continues to send distance and amplitude data, and you read the **ImageDataFormat** back as 0, then any of the color sensor configuration is not suitable for this mode, or calibration data is missing (**HardwareConfig** and **HardwareConfigColor** values inappropriate).

Also, if you change any setting while running in overlay mode, that makes the overlay calculation impossible, the camera switches to image data format 0 by itself.

4.6.2 Configuration

There are certain configuration options for the overlay mode:

- Invalidate point cloud points where color assignment is not possible: Register **OverlayConfig**

  A usual scene always contains points that e.g. have invalid depth data, are hidden from the color sensor, etc. In the default setting, these points are marked "invalid" in the X coordinate array, i.e., with value 0x1 (confer to chapter **Error! Reference source not found.**).

- Ray-tracing threshold: Register **OverlayVectorLengthThreshold**

  To determine if an XYZ point is visible from the color sensor, the camera uses a ray-tracing algorithm. This threshold in millimeters allows to adjust the strictness of the algorithm.

- Default color values if assignment from color sensor is not possible: Registers

  OverlayColorInvalidDepth, **OverlayColorNotCalibrated**, **OverlayColorOutside**,

  OverlayColorNoMapping, **OverlayColorHidden**

  RGB565 color values that are filled into the color array if a color assignment is not possible for a point.

4.7 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 4.4 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 352x287 pixels.

- Amplitude data from the ToF sensor, as 16-bit unsigned (Uint16) values. Resolution is always 352x287 pixels.

- Confidence data, as 8-bit unsigned values. Resolution is always 352x287 pixels.

- Z coordinate values, in millimeters, as 16-bit signed (Int16) values. No negative values allowed. Resolution is always 352x287.

- X coordinate values, in millimeters, as 16-bit signed (Int16) values. Resolution is always 352x287.

- Y coordinate values: Same as X.

- Raw depth data (without scaling, corrections and filters), as 16-bit unsigned (Uint16) values. Resolution is always 352x287 pixels.

- Raw phase data (0°, 90°, 180°, 270°), as 16-bit unsigned (Uint16) values. Resolution is always 352x287 pixels.

- Color sensor data in RGB565 format (16 bits per pixel). The resolution is set via register **ColorStreamParams**. The format is depicted in Table 4-1. Note that from one color value, B0, G0, and R0 are the least significant bits. B4, G5, and R4 are the most significant bits.

- Color sensor data in JPEG format. The resolution is set via register **ColorStreamParams**, as is the selection of JPEG instead of RGB565 format.

- Overlay color data: Each point of the XYZ point cloud, measured by the ToF sensor, gets assigned a color value from the color sensor by means of back-projection and ray-tracing to the color sensor. The color format is always RGB565.

High byte								Low byte							
R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
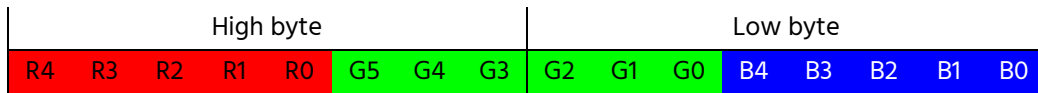
Table 4-1: RGB565 format

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.

4.7.1   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**.
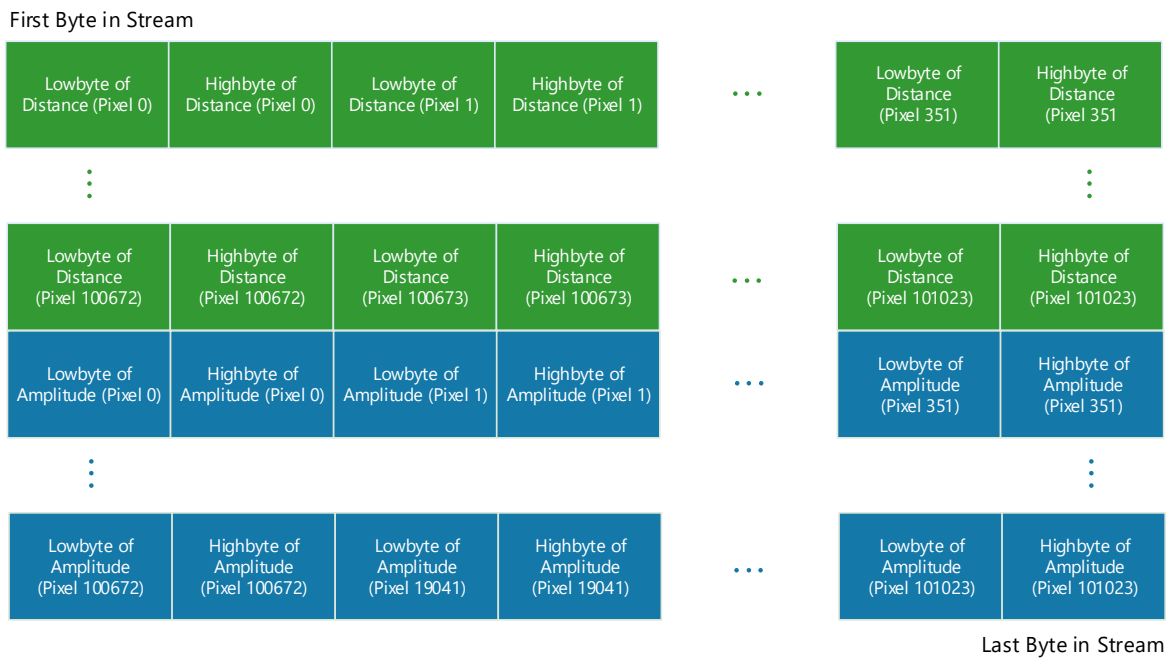
Figure 4-4: Data stream of Distance and Amplitude data

4.7.2   Distances, Amplitudes and Color

In this mode distances and amplitudes from the ToF sensor, as well as the color sensor image will be transferred in progressive mode, first the distance array (channel 0), then the amplitude array (channel 1), then the color sensor data (channel 2). The stream starts always with pixel #0.

The **distances** are coded in **millimeters** as **Uint16**, the **amplitudes** also as **Uint16**. The **color** sensor pixels are coded in **RGB565** (raw) or **JPEG** compressed format.

The resolution, format, and frame rate of the color sensor are set via register **ColorStreamParams**.

Channel 2 (which carries the color data) may have zero length, if no color data is available for the current frame. In this case, fields *ColorSensorWidth*, *ColorSensorHeight*, and *ColorSensorLength* of the frame header are set to zero (refer to section **Error! Reference source not found.**).

In the case the color data is transferred in JPEG format, frame header field *ColorSensorLength* contains the length of the JPEG-compressed image in bytes.
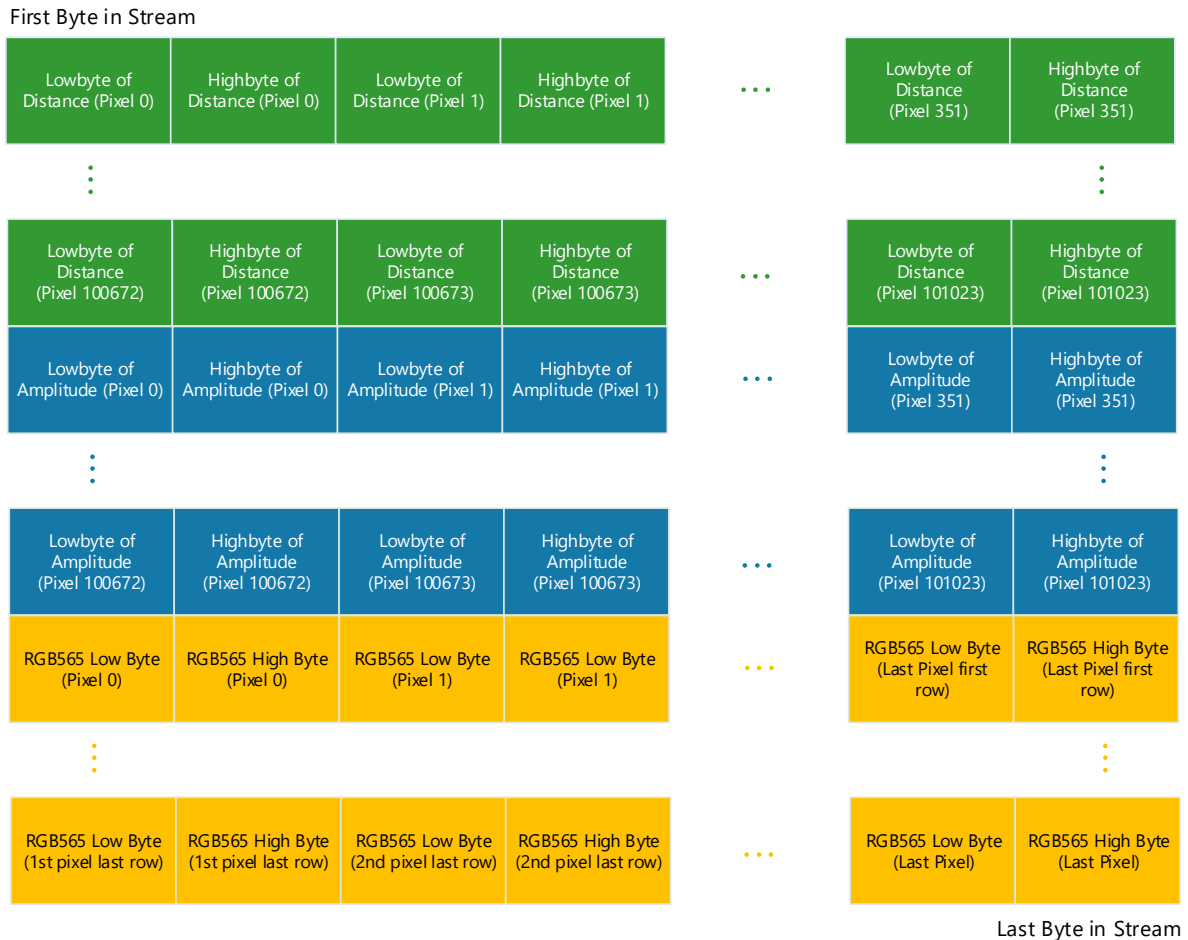
Figure 4-5: Data stream of Distance, Amplitude, RGB565 Color data

4.7.3   Distances and Color

In this mode distances from the ToF sensor and the color sensor image will be transferred in progressive mode, first the distance array (channel 0), then the color sensor data (channel 1). The stream starts always with pixel #0.

The **distances** are coded in **millimeters** as **Uint16**, the **color** sensor pixels are coded in **RGB565** (raw) or **JPEG** compressed format.

The resolution, format, and frame rate of the color sensor are set via register **ColorStreamParams**.

Channel 1 (which carries the color data) may have zero length, if no color data is available for the current frame. In this case, fields *ColorSensorWidth*, *ColorSensorHeight*, and *ColorSensorLength* of the frame header are set to zero (refer to section **Error! Reference source not found.**).

In the case the color data is transferred in JPEG format, frame header field *ColorSensorLength* contains the length of the JPEG-compressed image in bytes.

First Byte in Stream
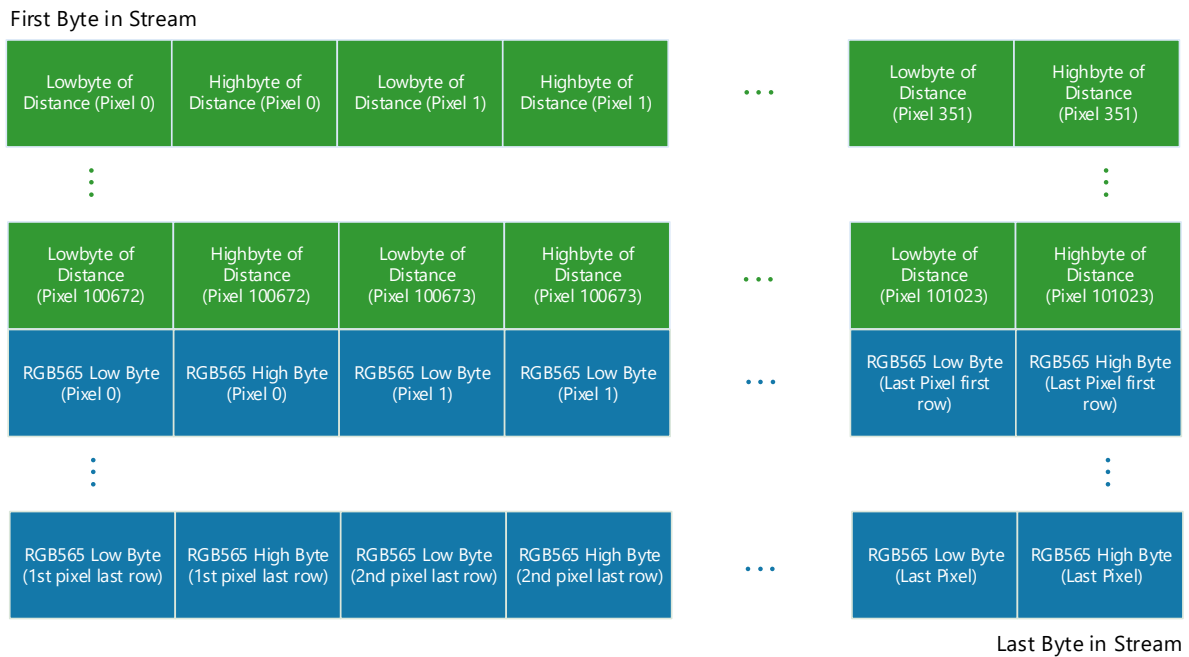
Figure 4-6: Data stream of distance and RGB565 data

4.7.4 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.**

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 351)	Highbyte of X- Coor. (Pixel 351)
Lowbyte of X- Coor. (Pixel 100672)	Highbyte of X- Coor. (Pixel 100672)	Lowbyte of X- Coor. (Pixel 100673)	Highbyte of X- Coor. (Pixel 100673)	...	Lowbyte of X- Coor. (Pixel 101023)	Highbyte of X- Coor. (Pixel 101023)
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 351)	Highbyte of Y- Coor. (Pixel 351)
Lowbyte of Y- Coor. (Pixel 100672)	Highbyte of Y- Coor. (Pixel 100672)	Lowbyte of Y- Coor. (Pixel 100673)	Highbyte of Y- Coor. (Pixel 100673)	...	Lowbyte of Y- Coor. (Pixel 101023)	Highbyte of Y- Coor. (Pixel 101023)
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 351)	Highbyte of Z- Coor. (Pixel 351)
Lowbyte of Z- Coor. (Pixel 100672)	Highbyte of Z- Coor. (Pixel 100672)	Lowbyte of Z- Coor. (Pixel 100673)	Highbyte of Z- Coor. (Pixel 100673)	...	Lowbyte of Z- Coor. (Pixel 101023)	Highbyte of Z- Coor. (Pixel 101023)

Last Byte in Stream

Figure 4-7: Data stream of XYZ Point Cloud

4.7.5 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.**

First Byte in Stream
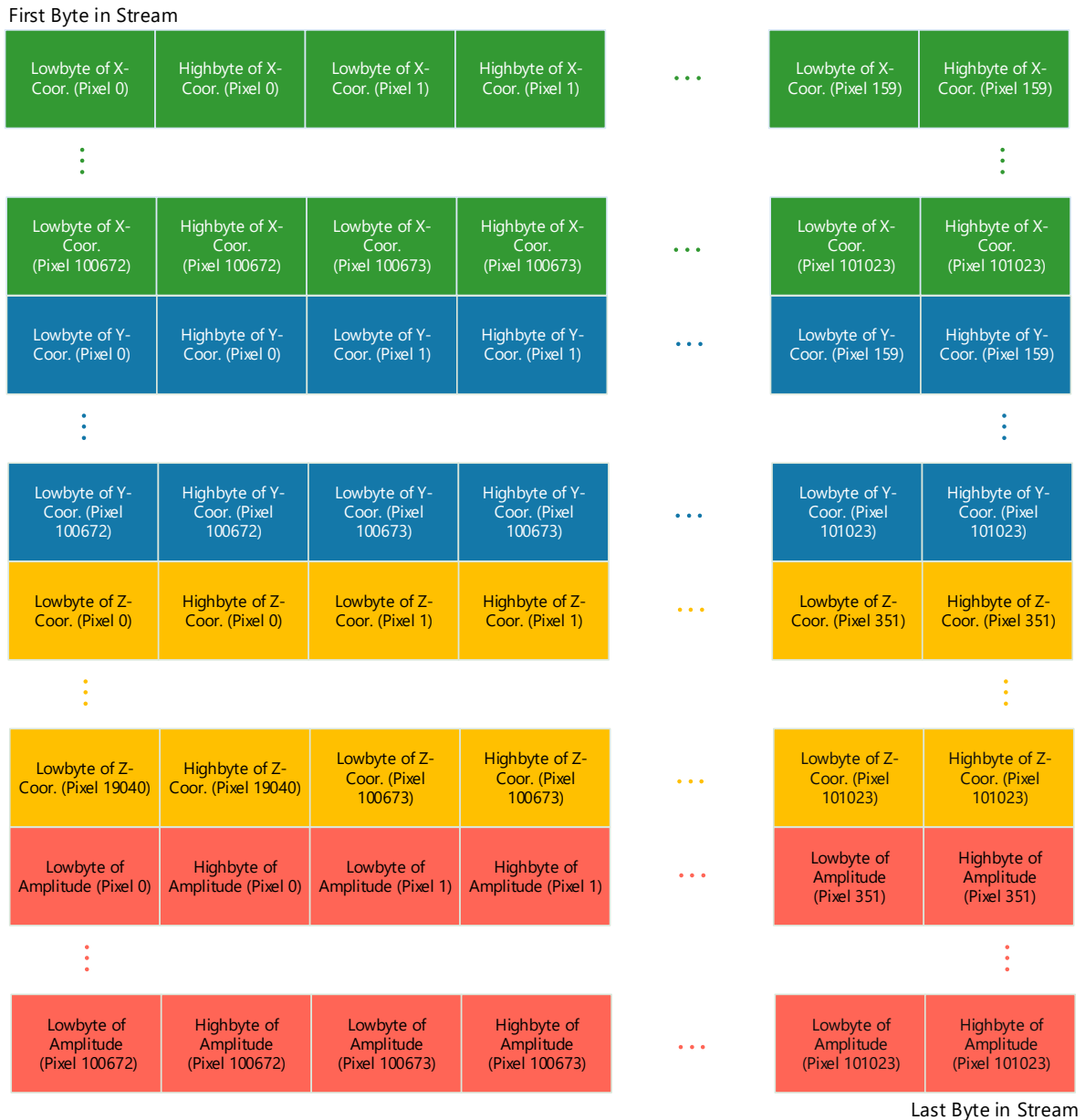
Figure 4-8: Data stream of XYZ Point Cloud and Amplitude

4.7.6   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**.

4.7.7   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**.

4.7.8   XYZ Point Cloud and Color (Overlay)

In this mode XYZ point cloud data and matching color values will be transferred in progressive mode, first the X array (channel 0), then Y (channel 1), Z (channel 2), and RGB565 color array (channel 3). The stream starts always with pixel #0.

The **coordinates** are coded in **millimeters** as **Int16**. The color array contains RGB565 values only.

Please consult chapter 4.6 for proper configuration of this mode.

4.7.9   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**.

4.7.10   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

4.7.11   Test mode

In this mode four arrays with test data are transferred in progressive order.

- Channel 0:   Uint16 value = Pixel Index

- Channel 1:   Uint16 value always constant '0xbeef'

- Channel 2:   Uint16 value = (Pixel Index)^2

- Channel 3:   Uint16 value always constant '0x0000'

4.7.12 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.

4.7.13 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%.

First Byte in Stream

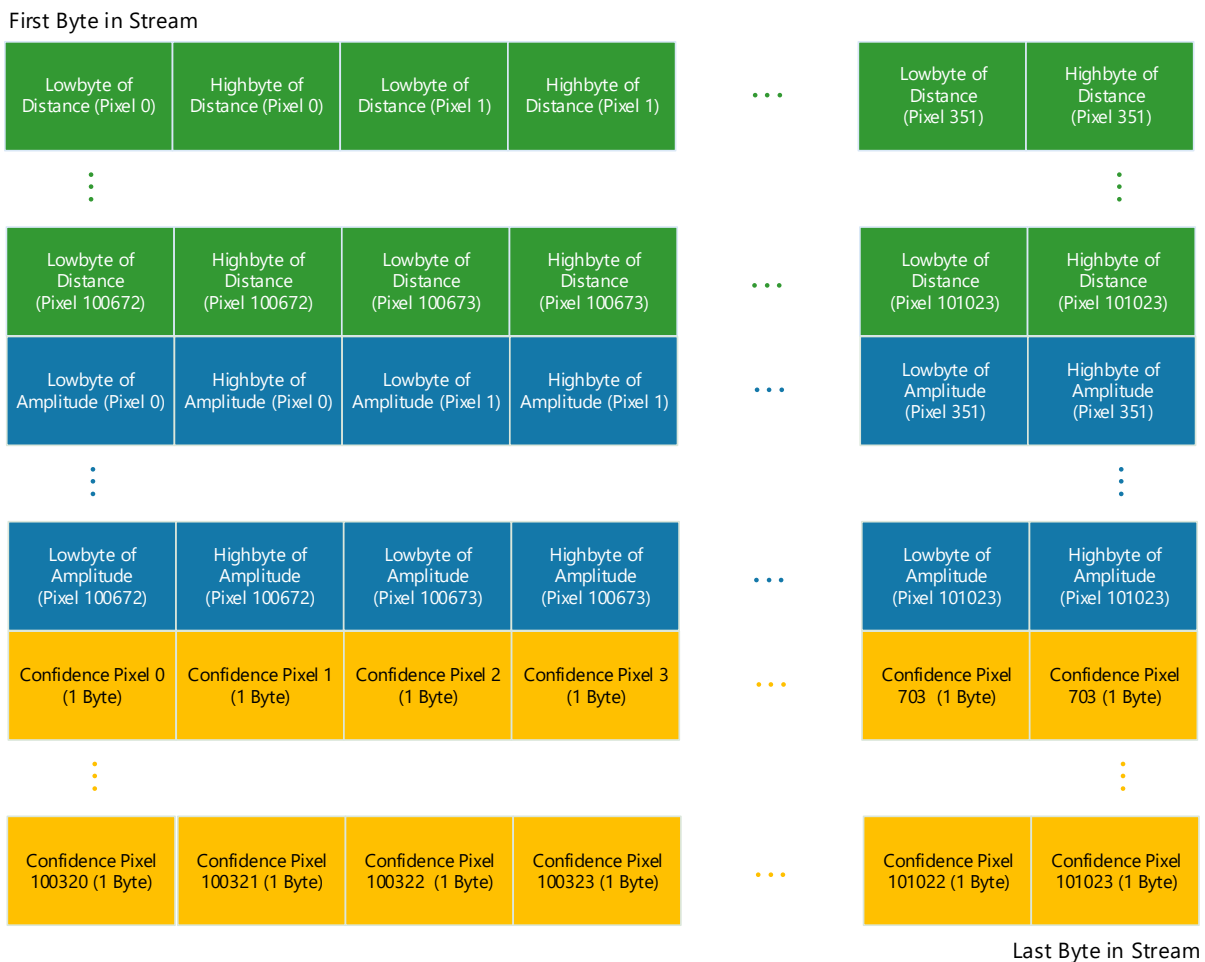| Lowbyte of Distance (Pixel 0)	Highbyte of Distance (Pixel 0)	Lowbyte of Distance (Pixel 1)	Highbyte of Distance (Pixel 1)	...	Lowbyte of Distance (Pixel 351)	Highbyte of Distance (Pixel 351)
Lowbyte of Distance (Pixel 100672)	Highbyte of Distance (Pixel 100672)	Lowbyte of Distance (Pixel 100673)	Highbyte of Distance (Pixel 100673)	...	Lowbyte of Distance (Pixel 101023)	Highbyte of Distance (Pixel 101023)
Lowbyte of Amplitude (Pixel 0)	Highbyte of Amplitude (Pixel 0)	Lowbyte of Amplitude (Pixel 1)	Highbyte of Amplitude (Pixel 1)	...	Lowbyte of Amplitude (Pixel 351)	Highbyte of Amplitude (Pixel 351)
Lowbyte of Amplitude (Pixel 100672)	Highbyte of Amplitude (Pixel 100672)	Lowbyte of Amplitude (Pixel 100673)	Highbyte of Amplitude (Pixel 100673)	...	Lowbyte of Amplitude (Pixel 101023)	Highbyte of Amplitude (Pixel 101023)
Confidence Pixel 0 (1 Byte)	Confidence Pixel 1 (1 Byte)	Confidence Pixel 2 (1 Byte)	Confidence Pixel 3 (1 Byte)	...	Confidence Pixel 703 (1 Byte)	Confidence Pixel 703 (1 Byte)
Confidence Pixel 100320 (1 Byte)	Confidence Pixel 100321 (1 Byte)	Confidence Pixel 100322 (1 Byte)	Confidence Pixel 100323 (1 Byte)	...	Confidence Pixel 101022 (1 Byte)	Confidence Pixel 101023 (1 Byte)

Last Byte in Stream

Figure 4-9: Data stream of Distances, Amplitudes, and Confidences

4.7.14 Distances, Amplitudes, Confidences, and Color

In this mode distances and amplitudes from the ToF sensor, confidence values, as well as the color sensor image will be transferred in progressive mode, first the distance array (channel 0), then the amplitude array (channel 1),

then the confidence array (channel 2), then the color sensor data (channel 3). The stream starts always with pixel #0.

The **distances** are coded in **millimeters** as **Uint16**, the **amplitudes** also as **Uint16**. The confidence values are in **Uint8** format. The **color** sensor pixels are coded in **RGB565** (raw) or **JPEG** compressed format.

The resolution, format, and frame rate of the color sensor are set via register **_ColorStreamParams_**.

Channel 3 (which carries the color data) may have zero length, if no color data is available for the current frame. In this case, fields _ColorSensorWidth_, _ColorSensorHeight_, and _ColorSensorLength_ of the frame header are set to zero (refer to section **Error! Reference source not found.**).

In the case the color data is transferred in JPEG format, frame header field _ColorSensorLength_ contains the length of the JPEG-compressed image in bytes.

4.7.15 Color

This data format is only used for the UDP Color Sensor Stream (see chapter 4.15.1 for more information). This data format cannot be set via register _ImageDataFormat_. When this data format is transferred, field _ImageFormat_ of the frame header (→ chapter **Error! Reference source not found.**) is set to value 22 (0x16).

In this mode the color sensor image is transferred in channel #0. The **color** sensor pixels are coded in **RGB565** (raw) or **JPEG** compressed format. In the case the color data is transferred in JPEG format, frame header field _ColorSensorLength_ contains the length of the JPEG-compressed image in bytes.

The resolution, format, and frame rate of the color sensor are configured via register **_ColorStreamParams_**.

4.8 ToF Modulation Frequency

The modulation frequency of the illumination is set to 20 MHz per default. Other modulation frequencies can be set using registers **_ModulationFrequency_**, **_ModFreqSeq1_**, **_ModFreqSeq2_**, **_ModFreqSeq3_**. 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
5.004 MHz	~30m
7.5 MHz	~20m
10.007 MHz	~15m
15 MHz	~10m
20.013 MHz	~7,5m
25.016 MHz	~6m

Frequency	Unambiguity Range
30 MHz	~5m
40.026 MHz	~3,75m
50.032 MHz	~3m
60 MHz	~2,5m
70 MHz	~2,1m
80.051 MHz	~1,8m

Table 4-2: 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!

4.9 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 24ms.

The color sensor is operating at ca. 15 or ca. 30 frames per second as defined in register **ColorStreamParams**. The color sensor always works with auto exposure control (AEC) and auto gain control (AGC) on.

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.

The FITP is also used for the camera's power control, if it is powered by PoE (see chapter 0). If the camera register **IntegrationTime** (or its pendants for sequences 1, 2, 3) or **Framerate** is modified, the entered value is used to calculate the FITP. Should the FITP violate the maximum allowed power, then the value is automatically corrected before it is applied. The corrected value can be read back via the same register. Also, if any correction of integration time or frame rate took place because of PoE constraints, Bit[10] of the **Status** register will be set.

4.10 Sequencing

The Argos3D - P33x allows the configuration of up to 4 sequences on the TIM. These sequences are captured immediately after each other by the TIM. Each sequence can be configured with individual integration time and modulation frequency.

If the camera's video mode is disabled (see register **Mode0**), and the camera is manually triggered, each trigger will generate as many sequences as currently configured.

The number of sequences can be configured in register **NofSequ**.
For sequence 0, use registers **IntegrationTime** and **ModulationFrequency** to set integration time and modulation frequency, respectively.
For sequence 1, use registers **IntTimeSeq1** and **ModFreqSeq1**.
For sequence 2, use registers **IntTimeSeq2** and **ModFreqSeq2**.
For sequence 3, use registers **IntTimeSeq3** and **ModFreqSeq3**.

Each sequence is processed and streamed by the camera as one individual frame. Each frame header contains the sequence number in field *SequenceNumber*.

4.11 Illumination Pre-Heating

The camera supports pre-heating of the LEDs on the LIMs immediately before capturing. Pre-heating is accomplished by applying the modulation signal to the LEDs prior to starting exposure of the ToF sensor.

Separate pre-heating times can be set for the first phase after a trigger and all consecutive phases.

For setting the pre-heating time for the first phase after a trigger, use register **IllPreheatingTime** (in microseconds).

For setting the pre-heating time for consecutive phases, use register **IllPreheatingTimeConsecutive** (in microseconds).

4.12 Distance Offset Calibration

Each Argos3D - P33x is offset-calibrated 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 5 MHz modulation frequency) to **DistOffset11** (for 80 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 4.16).

4.13 Manual Frame Trigger

There are two types of manual trigger. To enable the manual trigger, the video mode must be disabled in register ***Mode0***, Bit[0].

A manual trigger will start capturing of as many sequences as currently configured in register ***NofSequ***.

4.13.1 Hardware Trigger

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

4.13.2 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***.

4.13.3 ToF Snapshot Function

The camera features a snapshot function which produces exactly one frame with averaged distance (and optionally, amplitude) data per snapshot request. The camera's video mode must be disabled (no continuous streaming). On a snapshot request, the camera is collecting frames from the ToF sensor until the Frame Average filter is able to produce an averaged frame (which depends on filter settings), after which the ToF sensor is stopped again.

In case more than 1 sequences are configured (register ***NofSequ***), one frame is produced for each sequence per snapshot request.

Follow these instructions to use the snapshot function:

- Disable video mode in register **Mode0**.

- Enable the Frame Average filter for distance data in register **ImgProcConfig**.

- Optionally, enable also the Frame Average filter for amplitude data in register **ImgProcConfig2**.

- Configure the number of frames to be averaged in register **FilterFrameAverageConfig**.

- Request a new snapshot by setting the corresponding bit in register **Mode0**. (The bit is self-clearing.)

4.14 Temperature Sensors

4.14.1 Over Temperature Protection

The Argos3D - P33x firmware has a built-in monitoring for over-temperature condition of the LIMs. If the LIM temperature exceeds 70°C, the camera will automatically stop illumination and streaming, 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.

4.14.2 ToF Sensor Temperature

The TIM-U-IRS1020C supports two temperature sensors: The one incorporated into the ToF sensor chip, and a separate temperature sensor near the ToF sensor chip.

Register **TempSensorConfig** allows selecting a temperature sensor. Note that for very small integration times, the temperature cannot be read out from the ToF sensor, and the temperature may be set to "invalid".

4.15 Color Sensor

Data from the color sensor can be transmitted in two different ways (The numbers give the priority):

1. Color channel: Send color data as part of the data streaming protocol, along with data from the ToF sensor. Select an appropriate mode in register **ImageDataFormat**. Select resolution, data format (RGB565 or JPEG compressed), and frame rate of the color sensor in register **ColorStreamParams**.

2. UDP Color Sensor Stream: Send color data with the data streaming protocol, but independent from ToF sensor data. A separate destination IP address and destination UDP port can be configured.

Note that modes *Color Channel* and *UDP Color Sensor Stream* may operate concurrently.

See Table 4-3 for a list of available resolutions/frame rates.

Resolution	Frame rates	Horiz. Opening Angle[1]	Vert. Opening Angle[1]
176x144	15, 30	TBD	TBD
320x240	15, 30	103°	77°
640x480	15, 30	103°	77°
720x480	15, 30	103°	69°
720x576	15, 30	95°	76°
1024x768	15, 30	TBD	TBD
1280x720	15, 30	103°	58°
1920x1080	15, 30 [2]	74°	42°

Table 4-3: Color sensor resolution and frame rate options, plus opening angles

Note 1. The specified opening angles are approximate values only. They are valid only with the default lens. Please consult the Hardware User Manual for details.

Note 2. Depending on the camera's configuration, the maximum value may not be reached.

The pixel orientation of the color image sensor is identical to Figure 4-3.

4.15.1 UDP Color Sensor Stream

In use cases where a continuous color sensor stream is required, but only sporadic ToF sensor images, the camera features a second UDP stream that is dedicated to send color sensor data only. It is compliant to the data interface format in chapter 3.2 and hence compatible to tools and libraries provided by BECOM Systems.

The UDP color sensor stream is enabled in register **ColorStreamParams**, bit 2. The resolution, frame rate, and data format (raw RGB565 or JPEG compressed) can also be configured with this register.

The destination for this stream can be configured independently of the ToF data stream, using the following registers:

- Use registers **Eth0UdpColorStreamIp0**, **Eth0UdpColorStreamIp1** to set the destination IP address.

- Use register **Eth0UdpColorStreamPort** to set the destination UDP port.

4.15.2 Color Sensor Control

The color sensor features automatic exposure and gain control (AEC/AGC) and automatic white balance (AWB). By default, these are switched on.

These features can be controlled via register **ColorSensorControl**. Switching off AEC/AGC and/or AWB means that the sensor freezes the current values for exposure, gain, and white balance.

In case the AEC/AGC feature is switched off, the sensor's exposure can be controlled via register **ColorSensorExposure**, and the gain can be set in register **ColorSensorGain**. Reading from these registers return the current values of the sensor (also in AEC/AGC on mode), or the last user-defined registers, if the color sensor is not operating currently.

4.16 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**.

4.17 Ethernet/IP Settings

4.17.1 MAC Address

A dedicated Ethernet MAC address from BECOM Systems MAC address pool is assigned to each Argos3D - P33x by factory default. This MAC address is saved in the OTP and cannot be changed by the user.

The user is allowed to assign the camera another MAC address using the registers **Eth0Mac0** to **Eth0Mac2**. Be aware that in order to make the changes persistent, the register map must be saved to flash using register **CmdExec,** otherwise the changes will be lost on a reboot or power cycle.

If the register map in the flash will be cleared the factory default MAC address from OTP will be loaded.

4.17.2 IP/TCP/UDP Settings

The IP Settings of the Argos3D - P33x can be changes via the **Eth0_*** registers. A change of the IP settings (IP address, subnet mask, default gateway) will take effect on writing the latter one. Port settings will take effect immediately. UDP destination IP addresses will take effect immediately. Please see the register description for details.

To make the changes persistent the register map must be saved to flash by writing a dedicated value to the **CmdExec** register.

4.18 Reset to Factory Default

The Argos3D - P33x 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**.

Alternatively, a factory reset is executed via the camera's reset signal. (Please consult the Hardware User Manual for details.) It must be active until the firmware is completely booted and the data stream is present.

4.19 Firmware Update

The Argos3D - P33x is capable of updating its firmware (as well as the boot loader). The update procedure is executed using dedicated TCP/IP command frames over the control interface connection.

BECOM Systems provides the "Downloader" tool within the BltTofSuite for updating the Argos3D - P33x firmware over Ethernet.

Note

In order to complete a firmware update, a complete reboot of the camera is required. The camera will NOT reboot automatically.

4.19.1 Firmware Recovery

If a new firmware fails to load for 3 times, the Argos3D - P33x boot loader will recover the old firmware automatically.

After a firmware recovery, Bit[8] of the **Status** register is set.

The Argos3D - P33x also features a firmware load attempt counter, in register **BootStatus**. It is usually 1 (first boot attempt successful). It will lose its value if power is completely removed from the camera.

4.20 Logging

The camera automatically saves log messages to a dedicated partition in the internal flash.

Log data may be retrieved using the Secure Shell login (see chapter 3.9) and can be found at `/mnt/logs/messages*` files. Newest log data is contained in file `messages`.

4.21 Error Indication

The Argos3D - P33x 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**.

- <u>Bit 8</u>: There were errors during firmware boot, the previous firmware was discovered. This bit is not set again after a reboot.

- <u>Bit 9</u>: Indicates the LIM temperature has exceeded the maximum tolerable value

- <u>Bit 11</u>: 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*.

- <u>Bit 13</u>: Error starting the color sensor stream, or error on retrieving color sensor.

- <u>Bit 14</u>: Indicates a temperature measurement error on the base board. The bit is automatically cleared if the error disappears.

- <u>Bit 15</u>: Indicates a communication error with the TIM, or an error with triggering the TIM. This bit is automatically cleared if the error disappears.

5 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.

5.1 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 2 … Distances, Amplitudes and Color 3 … XYZ Point Cloud 4 … XYZ Point Cloud and Amplitudes 5 … X/Y/Z coordinates and color Information for each pixel (Overlay)

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
				6 ... Distances and Color
				9 ... Distances and XYZ Point Cloud
				10 ... Z Coordinate and Amplitudes
				11 ... Test mode
				12 ... Distances
				13 ... Raw Distances and Amplitudes
				21 ... Distances, Amplitudes, Confidences, and Color
0005	IntegrationTime	05DC	R/W	Integration Time [µs] (min: 1, max: 7500)
0006	DeviceType	03FC	R	Hardware specific identification
0007	DeviceInfo		R	Bit[0-3]: PCB Revision of base board 0...V2.1 1...V2.2 2...V2.3
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
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].

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
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[°]
002B	TriggerDelay	0000	R/W	Delay between trigger assertion (either software or hardware) and output trigger assertion (image capturing, trigger output) [ms]
002C	BootStatus		R	Bit[14-15]: Firmware Load Counter. This counter is reset by the firmware. It counts the boot attempts.
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
0032	TimVersion		R	IRS1020 sensor ID
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

BECOM
it's possible.

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
0043	TimSerialLow		R	IRS1020 sensor ID
0044	TimSerialHigh		R	IRS1020 sensor ID
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
0047	RgbLedColor	0300	R/W	RGB565 color value of RGB LED
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

BECOM
it's possible.

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
				Bit[14]: Temperature sensor error
				Bit[15]: Could not read status word (communication failure)
0049	Lim2Status	0000	R	Status word of LIM #2 (Bit description see register *Lim1Status*)
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
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 5-1: General registers

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

5.2 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

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
00C8	DistOffset7		R/W	An offset for distance values when operating at modulation frequency 40 MHz
00C9	DistOffset8		R/W	An offset for distance values when operating at modulation frequency 50 MHz
00CA	DistOffset9		R/W	An offset for distance values when operating at modulation frequency 60 MHz
00CB	DistOffset10		R/W	An offset for distance values when operating at modulation frequency 70 MHz
00CC	DistOffset11		R/W	An offset for distance values when operating at modulation frequency 80 MHz
00CD	DistOffset12		R/W	An offset for distance values when operating at modulation frequency 90 MHz
00CE	DistOffset13		R/W	An offset for distance values when operating at modulation frequency 100 MHz

Table 5-2: Distance Offset registers

5.3   GPIO Control

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
00D0	IOstate0	00--	R/W	Bit[0]: … state of IN_0 (only R) Bit[1]: … state of IN_1 (only R) Bit[8]: … state of OUT_0 (R/W) Bit[9]: … state of OUT_1 (R/W) Bit[10]: … state of OUT_2 (R/W) Bit[11]: … state of OUT_3 (R/W)

Table 5-3: Registers for GPIO Control

5.4   Color Streams

Addr (hex)	Register Name	Default Value (hex)	R/W	Description

00E0	ColorStreamParams	0022	R/W	This register configures various options to transmit data from the color sensor.
				
				Bit[1]: Color channel format
				    0 … RGB565
				    1 … JPEG compressed
				Bit[2]: 1…Enable autonomous UDP data stream
				Bit[4..6]: Color sensor resolution
				    0 … QCIF (176x144)

BECOM
it's possible.

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
				1 ... QVGA (320x240)
				2 ... VGA (640x480)
				3 ... NTSC (720x480)
				4 ... PAL (720x576)
				1 ... XGA (1024x768)
				1 ... 720P (1280x720)
				1 ... 1080P (1920x1080)
				Bit[8]: Color sensor frame rate
				0 ... 15 fps
				1 ... 30 fps
00E1	ColorSensorControl		R/W	Bit[0]: 1 ... AEC/AGC enabled
				Bit[1]: 1 ... AWB enabled
00E2	ColorSensorExposure		R/W	Exposure register value for color sensor
00E3	ColorSensorGain		R/W	Gain register value for color sensor
00E4	ColorTo3DDelayLowWord		R	Difference between the timestamp of the 3D sensor and the color sensor in μs (int32_t)
00E5	ColorTo3DDelayHighWord		R	Difference between the timestamp of the 3D sensor and the color sensor in μs (int32_t)
00E6	HardwareConfigColor		R/W	Lens opening angle identifier for color sensor

Table 5-4: Color Streams Registers

5.5 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 5-5: User Defined registers

5.6 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).
010F	PWM100Temp	FFFF	R/W	Fan control: Temperature at which to turn on fan in 0,01[°C]. Creates a PWM output with duty cycle of 100%.
0110	IllPreheatingTime	0000	R/W	Illumination preheating time for first phase in microseconds
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
011A	LimTempsInconsistentCounter	0000	R	Counts number of inconsistent temperature measurements on LIMs (difference between 2 LIM temperature sensors exceeds certain limit, e.g., 10°C).
011E	TempSensorConfig	0000	R/W	Bit [0]: Sensor input for *MainboardTemp* 0...Use ToF sensor built-in temperature sensor (not available for integration times smaller than 82us) 1...Use temperature sensor on TIM

Table 5-6: General registers

5.7 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 5-7: Registers for Sequencing

5.8 Illumination Configuration

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
0155	IllPreheatingTimeConsecutive	0000	R/W	Pre heating time in microseconds for all phases except the very first after a trigger

Table 5-8: Registers for Illumination Configuration

5.9 Test Commands

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
01C0	TestConfig	0000	R/W	Bit[1]: 1... Watchdog Test

© BECOM Systems 2020

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
				Bit[2]: 1... Simulate color sensor module failure
				Bit[4]: 1... Simulate under-voltage (UVLO detector test)
				Writing this register must be preceded by writing 0x5E6B into register CmdEnablePasswd (0x0022)

Table 5-9: Registers for Test Commands

5.10 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 5-10: Registers for device update

5.11 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

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
				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)
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 5-11: Register for filter configuration

5.12 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 5-12: Registers for Advanced Image Processing

5.13 Ethernet configuration

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
0240	Eth0Config	0006	R/W	Bit[0]: 1.. Enable DHCP Bit[1]: 1.. Enable UDP streaming Bit[2]: 1.. Ignore CRC for UDP streaming
0241	Eth0Mac2		R/W	Byte 5 (=High byte) and byte 4 of MAC address Writing this register has no immediate effect.
0242	Eth0Mac1		R/W	Byte 3 and byte 2 of MAC address Writing this register has no immediate effect.
0243	Eth0Mac0		R/W	Byte 1 and byte 0 (=Low byte) of MAC address **Writing this register will update the network configuration with the new MAC address.**
0244	Eth0Ip0	000A	R/W	Low word of IP address Writing this register has no immediate effect (see register 0x0249).
0245	Eth0Ip1	C0A8	R/W	High word of IP address Writing this register has no immediate effect (see register 0x0249).
0246	Eth0Snm0	FF00	R/W	Low word of subnet mask Writing this register has no immediate effect (see register 0x0249).
0247	Eth0Snm1	FFFF	R/W	High word of subnet mask Writing this register has no immediate effect (see register 0x0249).
0248	Eth0Gateway0	0001	R/W	Low word of gateway Writing this register has no immediate effect (see register 0x0249).

Addr (hex)	Register Name	Default Value (hex)	R/W	Description
0249	Eth0Gateway1	C0A8	R/W	High word of gateway **Writing this register will update the network configuration with new IP address, subnet mask and gateway.**
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
0250	PoEStatus		R	Bit[0..2]: Detected PoE power class 1: PoE (13W limitation) 2: PoE+ (25.5W limitation) 3: PoE++ (no power limitation) 7: VAUX (no power limitation)
0251	PoEOverride	0000	R/W	Bit[0..1]: Override PoE power class 0: Don't override 1: 13W limitation (PoE) 2: 25.5W limitation (PoE+) 3: No limitation (PoE++) **Note: This limit is only checked against if integration time (0x0005) or frame rate (0x000a) is set. No immediate action is taken when this register is written.**
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 5-13: Registers for Ethernet configuration

BECOM
it's possible.

6 Firmware History

6.1 Version Information

Firmware Version	Status	Release date	Changes
0.2.0	Beta	Jan 2016	
0.4.0	Beta	Apr 2016	• Added saturation check on raw phase data • Add support for camera discovery via UDP ➔ See Chapter 3.3 • Add Snapshot mode for averaged 3D frame • ➔ New bit 11 in register **Mode0**
1.0.0	Initial Release	Aug 2017	• First official release
1.1.0	Release	Sep 2017	• Optimized depth calculation (higher frame rates are now possible)
1.2.0	Release	Sep 2017	• Bugfix: GPIOs are working properly now
1.2.1	Release	Dec 2017	• Fixed anomaly 'After switching modulation frequency the camera may stop streaming.'

Table 6-1: Overview Argos3D - P33x firmware changes

Note

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

6.2 Anomalies

Applies to	Date	Description
<=1.1.0	Sep 2017	GPIOs are not working properly
1.2.0	Sep 2017	After switching "ACF plausibility check" off, pixel which was masked invalid stays invalid. Workaround: Switch "ACF plausibility check" off save registers permanently and reboot the camera.
<=1.2.0	Dec 2017	After switching modulation frequency the camera may stop streaming.

Table 6-2: Firmware anomalies

7 Software

7.1 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.

The library which provides this API for Ethernet-based devices as the Argos3D - P33x is the *BtaEthLib* (*BltTofApi* Ethernet Library).

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

Bluetechnix 'Time of Flight' API

↳ support.systems.becom-group.com (Section Documentation)

7.2 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

↳ support.systems.becom-group.com (Section Documentation)

7.3 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

↳ support.systems.becom-group.com

8 Support

8.1 General Support

General support for products can be found at BECOM Systems´ support site

Support Link

↳ support.systems.becom-group.com

8.2 Software Downloads

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

Software Download Portal

↳ support.systems.becom-group.com

8.3 Camera Development Package

The camera offers the possibility to bring your own application onto the Argos3D - P33x.

The Argos3D - P33x is based on an embedded ARM Linux system based on the i.MX6 Quad-core processor from Freescale Inc.

Please contact BECOM Systems support for more information.

9 Document Revision History

Version	Date	Document Revision
10	2019 10 31	Updated Links Streaming protocol details removed
9	2018 12 20	Company Name Change
8	2018 09 19	CD change
7	2017 06 12	Updated chapter 5.4 Color Streams registers Updated anomalies
6	2017 09 26	Updated document for firmware V1.2.0 Updated firmware history Updates anomalies
5	2017 08 28	Updated document for firmware V1.0.0
4	2017 06 22	Company information changed
3	2017 02 03	Index removed Update to P33X
2	2016 04 26	Updated document for firmware V0.4.0 • Added chapter 3.3 (Device discovery) • Added chapter 4.13.3 "ToF Snapshot Function"
1	2016 01 13	Initial version of the document

Table 9-1: Revision history

A List of Figures and Tables

Figures

Tables