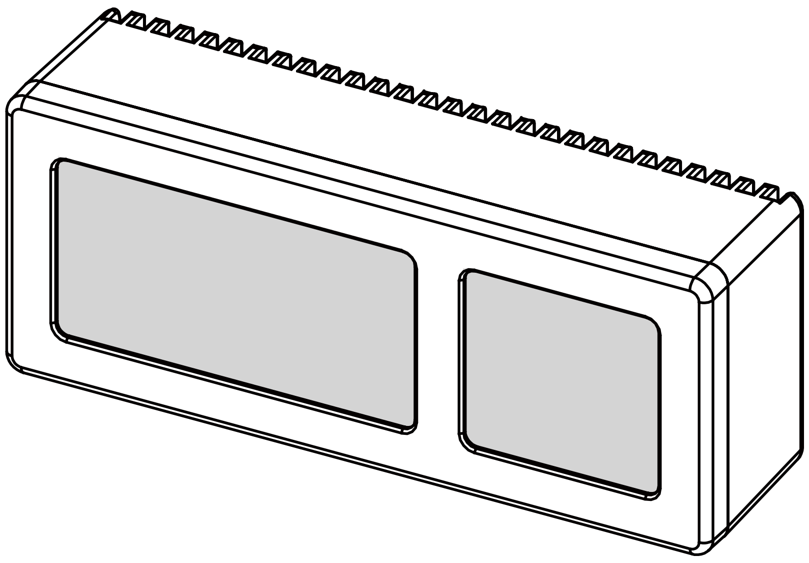
HYBO | iLidar ToF

Complete Solid-State 3D LiDAR

iTFS Series User Manual (EN)



Important Information

Notification

- Use of the product constitutes your agreement that you understand and agree to comply with the instructions and precautions listed in this document.
- Before using the product, be sure to familiarize yourself with the safety regulations and operating instructions. Improper use of the product may cause problems with the product and may result in malfunctions that can cause personal injury or property damage. Therefore, before using the product for the first time, be sure to familiarize yourself with the product-related materials, such as the Quick Start Guide and User Manual.
- The user assumes all risk of damage that may result from the use of or access to the product, product accessories, and all materials. HYBO Inc. shall not be liable for any personal or material damage or legal disputes arising directly or indirectly from the use of this product. Therefore, you understand and agree that you use this product at your own will and are responsible for any personal or material damage or third party personal or material damage arising from the use or inability to use the product. In order to avoid any unforeseen accidents, the user must observe safe and legal usage methods, including product-related materials.

Warnings

- Before using the product, be sure to familiarize yourself with the safety regulations and operating instructions. Improper use may cause problems with the equipment and may result in malfunctions that may cause personal injury or property damage. Therefore, be sure to read the product literature before using the product for the first time.
- Do not disassemble, use, modify, or repair the product in any way. In particular, illegal modification or alteration of the optical part through which the laser is emitted is prohibited as it may cause serious eye damage. If you need to reconfigure the product's performance, please do not modify the product. Contact HYBO Inc. to discuss product customization.
- Looking at the product at close range with a broken optical window can cause eye damage. Therefore, before using the product, check the condition of the optical window on the front of the product. If the optical window is broken, do not use the product and immediately turn off the power if it is operating. If you are unavoidably close to a product with a broken optical window, wear eye protection for the LASER CLASS 3R.
- If the optical window on the front of the product is broken, do not operate the product and contact HYBO Inc. after-sales service center for assistance.
- Do not press the optical window on the front of the product with sharp objects such as fingernails, screwdrivers, etc. Damage to the optical window can cause product malfunction, such as incorrect measurement data. Also, if the optical window gets water droplets, dust, or other debris, wipe it off with a soft cloth to keep it clean. Objects on the optical window can cause poor measurement data.
- Do not cover the optical window on the front of the product with covers, stickers, etc. as this may cause poor measurement data. If the product is to be used in an installed format, install

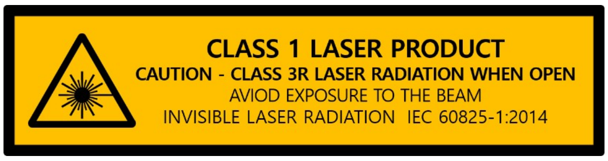
and use the product in accordance with the keep-out zones specified in the installation guide.

- The surface of the optical window on the front of the product may have minor bends and scratches, which is normal.
- Be careful not to drop the product on the floor, as damage may result in failure.
- When using 12 VDC through the I/O connector, check the pinmap properly. Incorrect power connection may cause the product to malfunction. Failures caused by incorrect connection methods are not covered by warranty service.
- Use cable specifications approved by HYBO Inc. Failure to do so may result in product damage.
- The product may be damaged if the cables and connectors used are connected while wet.
- When the product is in operation, heat is generated by the light source. The temperature at the back of the product may rise to about 60℃, which is normal. If the temperature on the back of the product rises excessively, please use the rear mount holes to attach it to a metal plate capable of conducting heat, etc. to dissipate the heat.

Legal Notices

Laser Safety

- The iTFS Series uses an 940 nm infrared invisible laser light source. This product is classified as CLASS 1 in the safety classification of laser products as defined by KS C IEC 60825-1:2014, a Korean standard equivalent to the international standard IEC 60825-1.


CLASS 1 LASER PRODUCT
CAUTION - CLASS 3R LASER RADIATION WHEN OPEN
AVIOD EXPOSURE TO THE BEAM
INVISIBLE LASER RADIATION IEC 60825-1:2014

- Although the CLASS 1 rating is classified as eye-safe, avoid direct viewing of the front optical window (transmitter) for a long time and do not use other optical equipment to view the front optical window (transmitter).
- If the product's front optical window is removed, it may result in a level of light exposure equivalent to CLASS 3R. Therefore, please keep the cautionary signs shown below in mind and never disassemble the optical window and front cover.

Hot Surface Warning

- The iTFS series generates heat during product operation due to the light source. The temperature on the back of the product may rise to about 60℃ (@ Room temperature), which is normal. If the temperature on the back of the product rises excessively, there is a risk of low temperature burns if you touch the product directly with your skin. Therefore, please measure the temperature of the product to keep it in a proper state and, if necessary, attach it to a metal plate capable of heat conduction using the rear mounting hole to dissipate the heat.

Certifications

KC: Registration of Broadcasting and Communication Equipment

- The iTFS Series is KC certified under the Korea Communications Equipment Act (Enforcement Decree of the Radio Act).

Company name	**HYBO Inc.**
Product name	iLidar-ToF
Model name	iTFS-110/iTFS-80
Certification number	R-R-h2b-iTFS-110
Manufacturer	HYBO Inc.
Country of manufacture	Republic of Korea
Date of manufacture	March 0000(see hard copy)
After-sales service center	02)597-4905

Sensor Overview

ilidar-ToF:iTFS

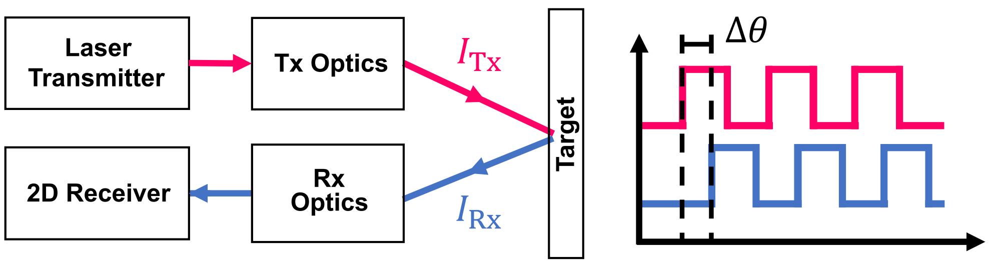
- iLidar ToF: iTFS is a 3D solid-state lidar that offers a cost-effective solution compared to traditional lidar sensors on the market. It features a solid-state design with no moving parts and a highly efficient optical system to achieve a measurement range of up to 20 meters.
- The iTFS series can be used for obstacle avoidance by mobile robots and for monitoring the presence of people in hazardous areas in industrial environments. It can also be used for a variety of other applications, such as detecting vehicle entry and exit in parking lots.
- One of the key advantages of the iTFS Series is the modularity of the optical components. By swapping out the optical modules, different specifications can be offered while maintaining the sensor's form factor. This allows users to customize the sensor's performance for different environments and applications. If you need such kind of customization, please contact HYBO Inc.

Fundamentals of Indirect Time-of-Flight Sensors

- iLidar ToF: iTFS uses the indirect ToF (iToF) method to measure distance. The transmitter generates amplitude modulated continuous wave (AMCW) light with a laser, and the light reflected from the measured object returns to the sensor. The ToF image sensor in the receiver decodes the light at the same frequency as the transmitter. Finally, the modulation phase difference ($\Delta\theta$) between the transmitter and receiver is calculated, and the distance is calculated as follows

$$d = \frac{c}{2} \cdot \frac{\Delta\theta}{2\pi f}$$

- Where f is the AMCW frequency, c is the speed of light, and d is the distance from the object.

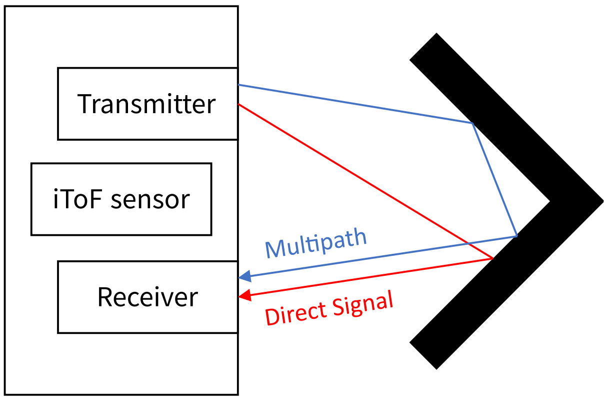


Known Limitations

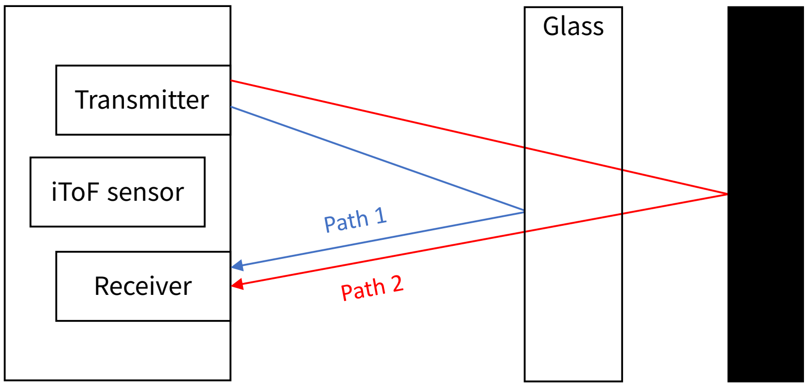
- The iToF method has known limitations. When using iTFS sensors, please be aware of the following factors that contribute to ranging errors.

Multi-Path Error

- One of the known limitations of the iToF distance measurement is the multi-path error. As shown in the figure below, if the light reflected from the transmitter to the object has two light paths (Direct Signal and Multipath), the distance measured by the ToF image sensor may be greater than the actual distance in the direction.

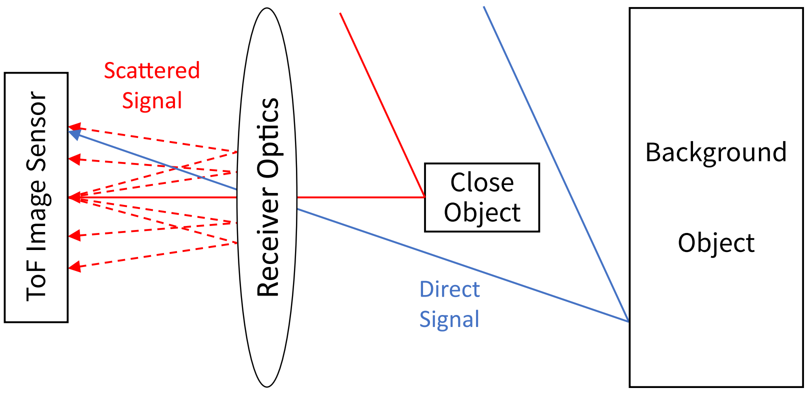


- Also, when viewing an object through a transparent object such as glass in front of the sensor, the effects of the light path through the glass (Path 1) and the light path through the object (Path 2) can be combined, causing the depth value in that direction to be measured between the glass and the object.



Scattering

- Scattering is a phenomenon caused by the reflection of light inside the light receiver of the iToF sensor. Typically, scattering occurs when objects are very close to the iToF sensor, or when highly reflective objects such as mirrors are present, causing errors in the depth value. Specifically, the low signal strength of the background object and the signal from scattering from the close object can interfere, causing the distance of the background object to appear closer than it actually is.



Specifications

[Specifications of iTFS Series]

Features	Specifications					
	iTFS-110			iTFS-80		
	mode 1 [1]	mode 2	mode 3	mode 1	mode 2	mode 3
Range[2]	0.3-8 m	0.05-12 m	0.05-16 m	0.3-10 m	0.05-15 m	0.05-20 m
Resolution	0.4° ×0.4°	0.4° ×0.8°	0.8° ×0.8°	0.3° ×0.3°	0.3° ×0.6°	0.6° ×0.6°
FoV[3]	110° × 60°			80° × 45°		
Accuracy	Error level: ± (3~5 cm + 2% of distance measurement)					
Framerate	Typ. 12.5 Hz (Up to 20 Hz with heatsink andreduced RoI)					
Dimensions	115 mm × 46 mm × 31.5 mm					
Weight	200 g					
Power	Avg. 6 W / Max. 12 W (12VDC or PoE)					
Interface	Ethernet (RJ-45) / UART (Molex)					
Output	Depth and Intensity Images					
Certification	KC					
Sunlight Immunity	~ 33 klux (80%ranging performance @ mode2 with 80% diffuse-reflective target)					
Illumination	940-nm IR Laser					
Eye safety	CLASS 1 (based on IEC 60825-1:2014)					

[1] On-the-fly configuration available (mode, framerate, and output data)

[2] Measured at centered ROI by using 80% diffuse-reflective target

[3] Range-guaranteed scope. Working horizontal FoVs are 120° and 90° for iTFS-110 and iTS-80 respectively.

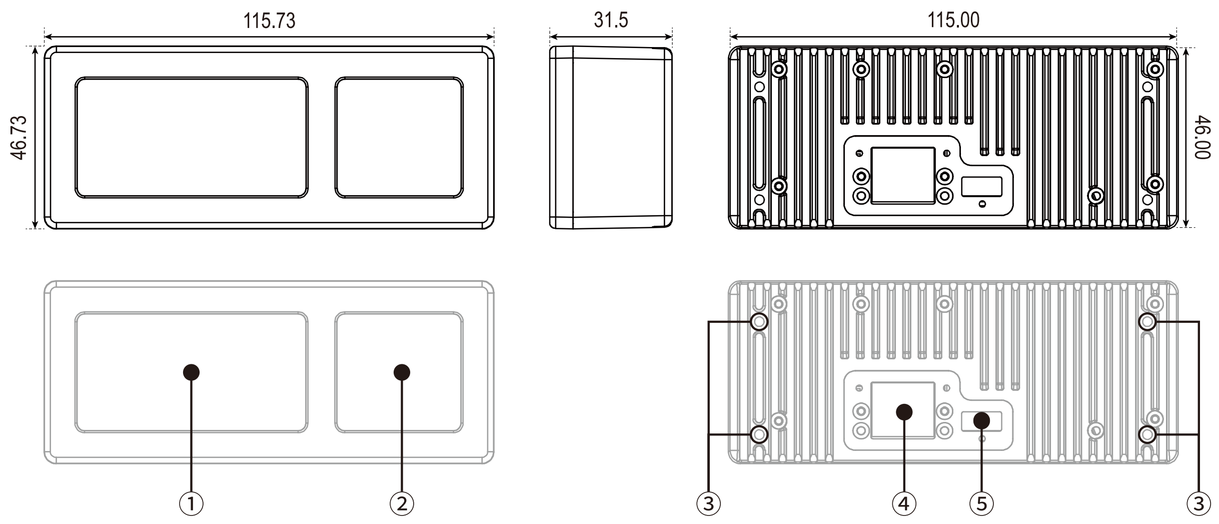
Mechanical Parts

What's in the box?

iTFS-110 × 1 (or iTFS-80 × 1)	LAN Cable 1m × 1	Molex Connector 30cm × 1

- iTFS-110 x1 (or iTFS-80 x1): This is the lidar. Be sure to remove the protective tape covering the product.
- LAN cable (1 m) x1: This is the LAN cable for receiving lidar data. iTFS Series requires a communication speed of 100 Mbps or higher, so be sure to use a CAT.5 or higher UTP cable. The included LAN cable is a CAT.5e UTP cable. When using PoE to power the lidar, no additional line connection is required.
- Molex connector (30 cm) x1: This is an add-on cable for use with terminals that can apply 12 VDC directly without using PoE. It also provides features such as TRIGGER and STROBE. The included connector has a part number of Molex 51021-0600 and includes a half-cut cable with 22 AWG or larger flame retardant wire. If you make your own cable, please check this information to determine compatibility.

Mechanical Drawing



- The product sizes and designations for the iTFS series are as follows (unit: mm)

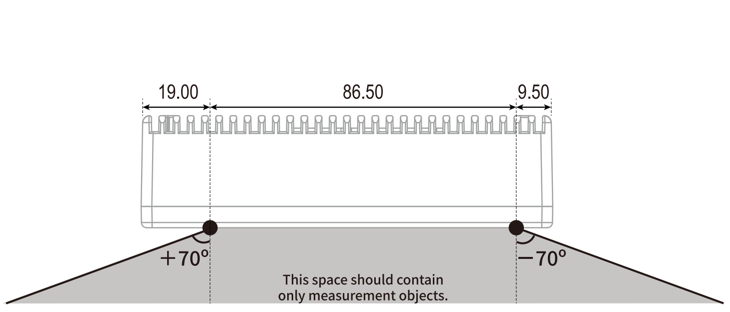
   1. Optical window (transmitter)
   2. Optical window (receiver)
   3. Screw holes for mounting, M3 4-mm depth
   4. Data RJ45 connector, PoE
   5. I/O connector, Molex pico-blade 6-pos [1 – 2(RED): 12VDC, 3: TRIGGER, 4: STROBE, 5 – 6(BLACK): GND]

- The 3D CAD model (STL) is available.

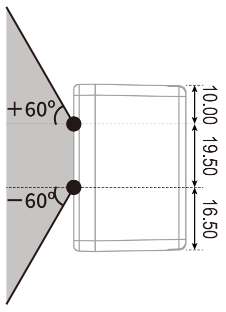
Installation Guidelines

- To ensure maximum performance when installing iTFS Series, please keep in mind the keep-out zone according to the model name as follows (Unit: mm)

iTFS-110

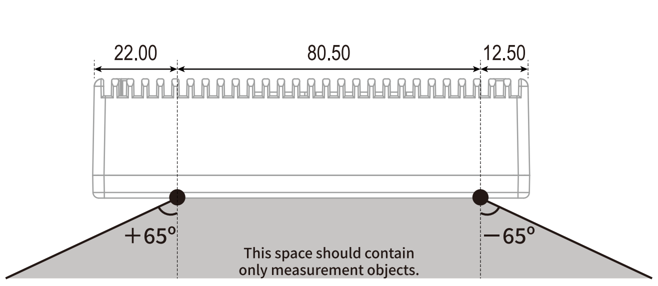


<Top View>

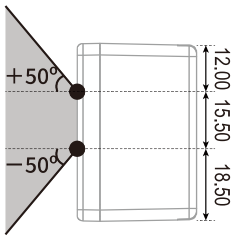


<Side View>

iTFS-80

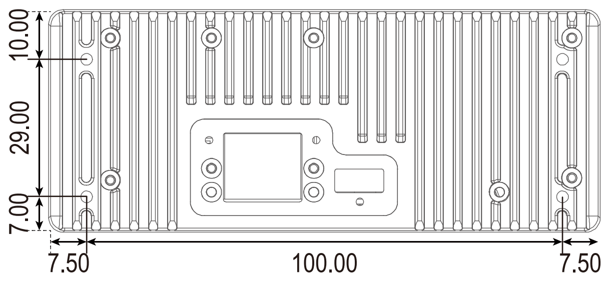


<Top View>



<Side View>

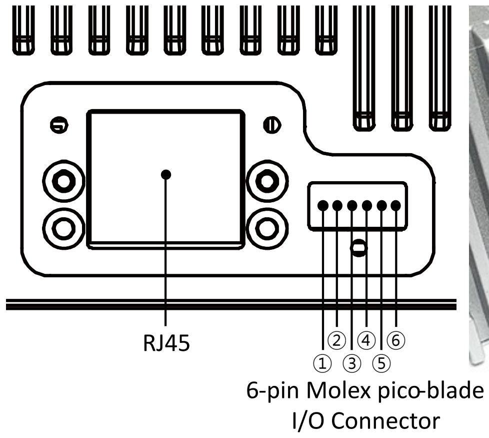
- The rear of the product provides screw holes (M3, 4-mm depth) for mounting the product. Please refer to the screw hole locations below.



Electrical Connection

Pinout

- The iTFS series has an RJ45 connector and a 6-pin Molex pico-blade. The RJ45 port is used for power supply (PoE) and data transfer. The 6-pin Molex pico-blade is used for power supply (12V DC) and TRIGGER/STROBE function.



- The RJ45 port on the iTFS supports PoE and PoE+ specifications. When supplying power via the PoE feature, please ensure that the power supply meets the following standards

[PoE Power Source Requirements]

Property	PoE	PoE+	Unit
Maximum Power	15.4	30.0	W
Voltage Range	44.0-57.0	50.0-57.0	V
Maximum current	350	600	mA

- The connection information for the Molex I/O pins is shown below.

[Molex I/O Pin Map]

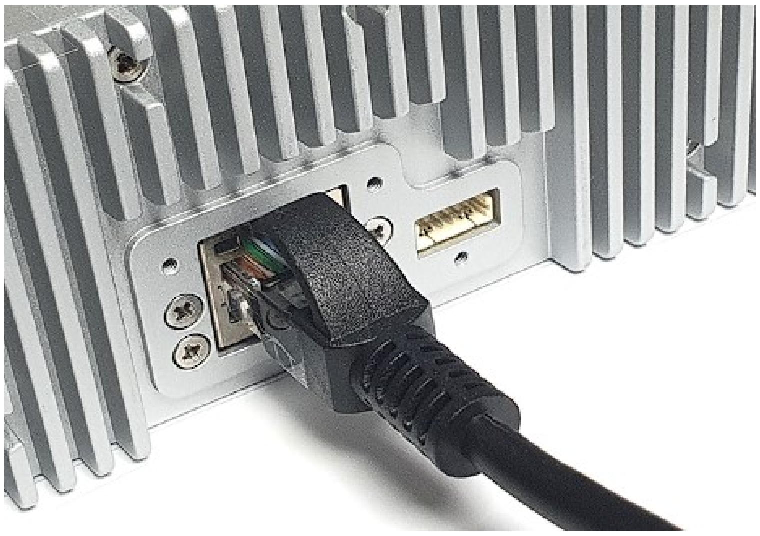
Pin	Name	Rating/Description
1	12VDC	12V (min: 11.6V, Max: 12.4V)
2	12VDC	12V (min: 11.6V, Max: 12.4V)
3	TRIGGER	TTL (Typ: 5V, min: 3.3V)
4	STROBE	TTL
5	GND	GND
6	GND	GND

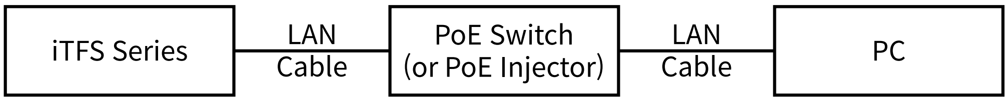
Power

- The iTFS Series consumes an average of 6 W and a maximum of 12 W. For your convenience, the iTFS Series offers both PoE power supply and 12VDC power supply. Please connect the product's power and LAN cables to operate the product in the preferred method. (Note that the overall temperature of the sensor may be approximately 4 degrees higher when using the PoE function compared to 12VDC power).
  - PoE: RJ45 port, supports PoE and PoE+ standards
  - 12VDC: Supplied via 6-pin Molex pico-blade

Connection Example with PoE

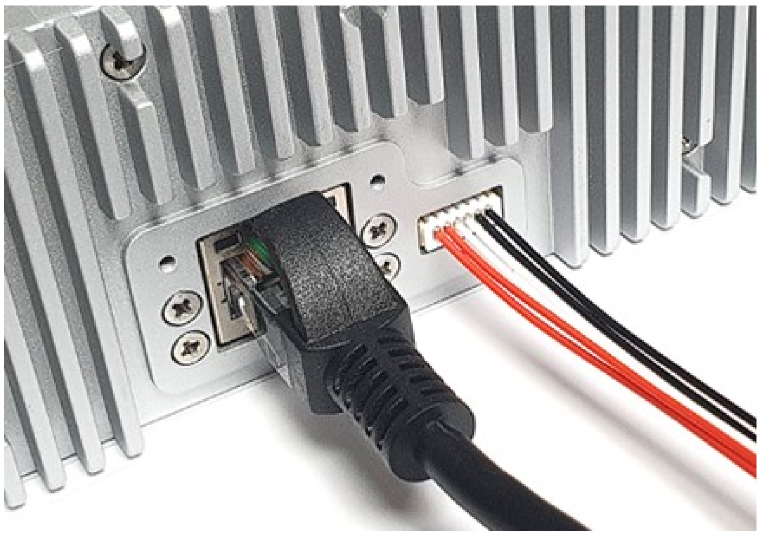
- Power supply via PoE is possible by using a LAN switch or PoE injector that supports PoE function. Please refer to the connection diagram below to configure the product. (No additional line connections are required to power the lidar using PoE)

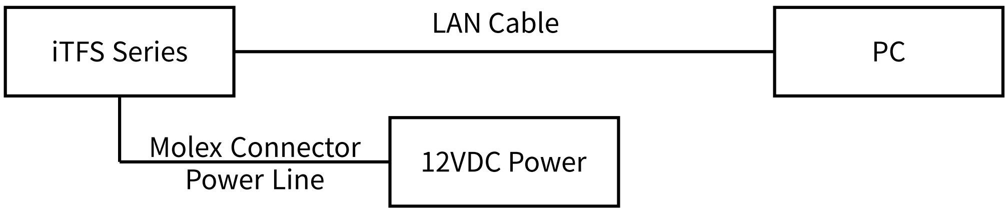




Connection Example with 12VDC

- You can connect 12VDC through the Molex connector to supply power. Please check the connection diagram below to configure the product.




iTFS Series
LAN Cable
PC
Molex Connector
Power Line
12VDC Power

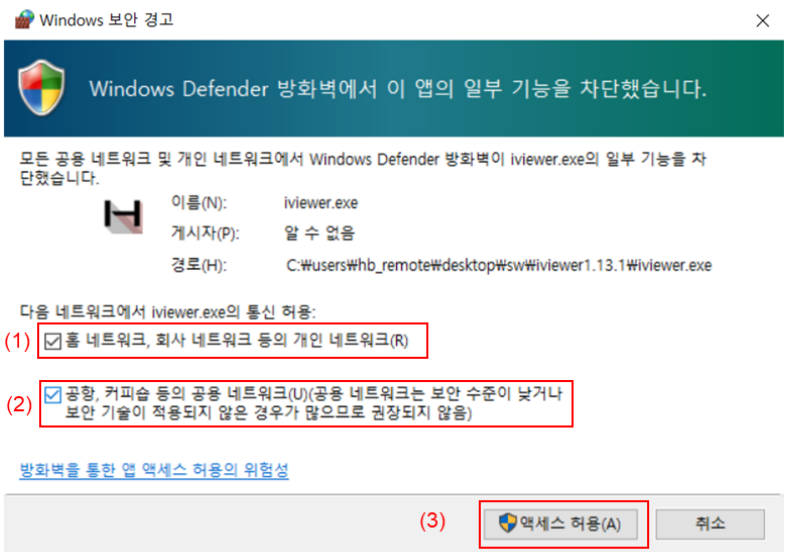
Sensor Operation

Network Configuration

- The iTFS sensor sends and receives data to and from the connected device via UDP communication, and the address of the connected PC or device must be set correctly for normal data reception. To do this, please change the IP address of the LAN port connected to the iTFS sensor to the following, depending on the OS you are using. The IP address listed below is the initial settings of the sensor. If you change the IP address setting of the sensor, you should enter the corresponding IP address value.

Windows OS

- Open the network properties settings of the Ethernet device to which the iTFS sensor is connected. Select Internet Protocol version 4 (TCP/IPv4), and set the IP address and subnet mask. The default values are; IP address: 192.168.5.2/subnet mask: 255.255.255.0.
- If this is the first time you are running iTFS-related software in a Windows environment, you will see the following Windows security warning prompt. This window is displayed because the iTFS sensor sends and receives data through UDP communication, and you must allow all communication as shown below for normal execution (the appearance of the settings window may differ depending on the Windows version).



Windows 보안 경고

Windows Defender 방화벽에서 이 앱의 일부 기능을 차단했습니다.

모든 공용 네트워크 및 개인 네트워크에서 Windows Defender 방화벽이 iviewer.exe의 일부 기능을 차단했습니다.

이름(N):	iviewer.exe
게시자(P):	알 수 없음
경로(H):	C:₩users₩hb_remote₩desktop₩sw₩iviewer1.13.1₩iviewer.exe

다음 네트워크에서 iviewer.exe의 통신 허용:

(1) ☑ 홈 네트워크, 회사 네트워크 등의 개인 네트워크(R)

(2) ☑ 공항, 커피숍 등의 공용 네트워크(U)(공용 네트워크는 보안 수준이 낮거나 보안 기술이 적용되지 않은 경우가 많으므로 권장되지 않음)

방화벽을 통한 앱 액세스 허용의 위험성

Linux(Ubuntu)

- Enter the settings for the LAN port that the iTFS sensor is connected to. On the IPv4 tab, select Set manually, and set the IP address and subnet mask. The default values are; IP address: 192.168.5.2/subnet mask: 255.255.255.0.

iViewer:Simple iLidar Data Viewer

- iViewer is a real-time data viewer for iLidar-ToF sensors. iViewer allows you to connect sensors, view measurement data, and set sensor parameters. The latest version of iViewer can be downloaded from the iLidar-ToF Github.

Packets and Parameters

Common Packet Structure

- Devices connected to an iTFS sensor send and receive data over UDP communication. The packets used have the following structure Index and Size are in bytes.

[Common Packet Structure Details]

Name	Index	Size	Type	Value
STX0	0	1	uint8	0xA5
STX1	1	1	uint8	0x5A
ID	2	2	uint16	-
LEN	4	2	uint16	N
PAYLOAD	6	N	-	-
ETX0	N+6	1	uint8	0xA5
ETX1	N+7	1	uint8	0x5A

STX

- The beginning of the packet.

ID

- An ID that indicates the type of packet.

[List of ID]

Name	ID	LEN	Direction	Description
IMG	0x0000	1282	Sensor → User	Image Data
STATUS	0x0010	28	Sensor → User	Current state of the sensor (summary)
STATUS_FULL	0x0011	312	Sensor → User	Current state of the sensor (detailed)
INFO	0x0020	110	Sensor ↔ User	Check or set sensor parameters
INFO_V2	0x0021	166	Sensor ↔ User	Check or set sensor parameters
CMD	0x0030	4	Sensor ← User	Commands for sensor behavior

LEN

- The length (number of bytes) of the PAYLOAD. It has a fixed value based on the ID.

PAYLOAD

- This is the part of the packet that contains the actual information, which varies depending on the packet's ID. For information about each ID, refer to the section that corresponds to each ID.

ETX

- The end of the packet.

IMG Packet

- IMG packets are packets that carry depth or intensity image data taken by the sensor, and are passed from the sensor to the user. The image from the sensor is delivered in the form of a few rows. The IMG packet has the following PAYLOAD.

[IMG Packet Payload]

Name	Index	Size	Type
row_index	0	1	uint8
mframe	1	1	uint8
data[0]	2	2	uint16
data[1]	4	2	uint16
...	...	...	...
data[639]	1280	2	uint16

- See How to Handle Image and Point Cloud to learn how to get the full image using IMG packets.

row_index

- The image taken by the sensor is divided into several rows and transfered to the user. The row_index indicates the order of the packet in the image.

mframe

- Contains the capturing mode and frame information for the current packet. It has bitwise information as shown in the table below.

[mframe Bit Information]

Bit	Name
0	capture_frame_short[0]
1	capture_frame_short[1]

Bit	Name
2	capture_frame_short[2]
3	capture_frame_short[3]
4	capture_frame_short[4]
5	capture_frame_short[5]
6	capture_mode[0]
7	capture_mode[1]

- capture_mode: The capture mode of the sensor. See the [List of capture_mode] table.
- capture_frame_short: Number of captured frames. Increments from 0 to 63 and then returns to 0.

data

- The depth or intensity data at each pixel. The unit of the depth value is [mm].

STATUS Packet

- The STATUS packet indicates the current state of the sensor. The STATUS packet is passed from the sensor to the user. The PAYLOAD of a STATUS packet is as follows;

[STATUS Packet Payload]

Name	Index	Size	Type
capture_mode	0	1	uint8
capture_frame	1	1	uint8
sensor_sn	2	2	uint16
sensor_time_th	4	8	uint64
sensor_time_tl	12	2	uint16
sensor_frame_status	14	2	uint16
sensor_temp_rx	16	2	int16
sensor_temp_core	18	2	int16
sensor_vcsel_level	20	2	int16
sensor_power_level	22	2	int16
sensor_warning	24	4	uint32

capture_mode

- Indicates the capture mode of the sensor. See Chapter How to Handle Depth Image and Point Cloud for how to organize images based on capture_mode.

[List of capture_mode]

Mode	capture_mode	Max. Resolution	Description
Gray	0	320x240	940 nm gray camera without illumination
NB	1	320x160	No binning
VB	2	320x80	Vertical binning
HV	3	160x80	Horizontal and Vertical binning

capture_frame

- Number of captured frames. Increases from 0 to 255 and then returns to 0.

sensor_sn

- Serial number of the sensor.

sensor_time_th & sensor_time_tl

- Indicates the sensor running time. sensor_time_th is in [ms], sensor_time_tl is in [us]. You can get the exact action time in [us] with the following formula;

$$\text{time} = \text{sensor\_time\_th} \times 1000 + \text{sensor\_time\_tl} \quad [\text{us}]$$

sensor_temp_rx & sensor_temp_core

- Contains the temperature information from the sensor. You can divide this value by 100 to get the actual Celsius temperature.
- rx: temperature of the imager / core: temperature of the sensor core

sensor_vcsel_level & sensor_power_level

- Contains the voltage information from the sensor. You can divide this value by 100 to get the actual voltage [V].
- vcsel: voltage of the transmitter / power: voltage of the transmitter source

sensor_warning

- Warning flag information that indicates a sensor anomaly.

STATUS_FULL Packet

- The STATUS_FULL packet details the current state of the sensor. The STATUS_FULL packet has the following PAYLOAD

[STATUS_FULL Packet Payload]

Name	Index	size	Type
capture_mode	0	1	uint8_t
capture_frame	1	1	uint8_t
sensor_sn	2	2	uint16_t
sensor_time_th	4	8	uint64_t
sensor_time_tl	12	2	uint16_t
sensor_frame_status	14	2	uint16_t
sensor_temp_rx	16	2	int16_t
sensor_temp_core	18	2	int16_t
sensor_temp	20	2x4	int16_t
sensor_vcsel_level	28	2	int16_t
sensor_vcsel_on	30	2x4x16	int16_t
sensor_power_level	158	2	int16_t
sensor_power_on	160	2x4x16	int16_t
sensor_level	288	2x10	int16_t
sensor_warning	308	4	uint32_t

capture_mode

- Indicates capturing mode of the sensor. Please check table [List of capture_mode].

capture_frame

- Number of captured frames. Increases from 0 to 255 and then returns to 0.

sensor_sn

- Serial number of the sensor.

sensor_time_th & sensor_time_tl

- Indicates the sensor running time. sensor_time_th is in [ms], sensor_time_tl is in [us]. You can get the exact action time in [us] with the following formula;

$$\text{time} = \text{sensor\_time\_th} \times 1000 + \text{sensor\_time\_tl} \quad [\text{us}]$$

sensor_temp_rx & sensor_temp_core

- Contains the temperature information from the sensor. You can divide this value by 100 to get the actual Celsius temperature.
- rx: temperature of the imager / core: temperature of the sensor core

sensor_temp

- It contains temperature information for four points inside the case. You can divide this value by 100 to get the actual Celsius temperature.

sensor_vcsel_level

- Contains the transmitter voltage information from the sensor. You can divide this value by 100 to get the actual voltage [V].

sensor_vcsel_on

- Information about the dynamic voltages series of the emitter used for debugging purposes. [4]:DCS0,1,2,3 / [16]:time information

sensor_power_level

- Contains the transmitter source voltage information from the sensor. You can divide this value by 100 to get the actual voltage [V].

sensor_power_on

- Contains the transmitter voltage information from the sensor. [4]:DCS0,1,2,3 / [16]:time information

sensor_level

- Voltage information at each point inside the sensor, used for debugging purposes. [10]: the voltage at each point

sensor_warning

- Warning flag information that indicates a sensor anomaly.

INFO Packet

- INFO packets provide information about the current operation parameter of the sensor. INFO packets can be sent from the sensor to the user, or from the user to the sensor.
- For more information about changing sensor settings via INFO packets, see the Configuration Process.

[INFO Packet Payload]

Name	Index	Size	Type	Authority

Name	Index	Size	Type	Authority
sensor_sn	0	2	uint16	R-
sensor_hw_id	2	1x30	uint8	R-
sensor_fw_ver	32	1x3	uint8	R-
sensor_fw_date	35	1x12	char	R-
sensor_fw_time	47	1x9	char	R-
sensor_calib_id	56	4	uint32	R-
capture_mode	60	1	uint8	RW
capture_row	61	1	uint8	RW
capture_period	62	2	uint16	RW
capture_shutter	64	2x5	uint16	RW
capture_limit	74	2x2	uint16	RW
data_output	78	1	uint8	RW
arb	79	1	uint8	RW
data_baud	80	4	uint32	RW
data_sensor_ip	84	1x4	uint8	RW
data_dest_ip	88	1x4	uint8	RW
data_subnet	92	1x4	uint8	RW
data_gateway	96	1x4	uint8	RW
data_port	100	2	uint16	RW
sync	102	1	uint8	RW
lock	103	1	uint8	R-
sync_delay	104	2	uint16	RW
arb_timeout	106	4	uint32	RW

sensor_sn

- Serial number of the sensor.

sensor_hw_id

- Hardware ID of the sensor.

sensor_fw_ver

- Firmware version of the sensor.

sensor_fw_date

- Build date of the sensor.

sensor_fw_time

- Build time of the sensor.

sensor_calib_id

- Calibration data id of the sensor.

capture_mode

- Indicates capturing mode of the sensor. Please check table [List of capture_mode].

capture_row

- The number of pixel rows to use for capturing depth and intensity values on the image sensor.
- The number of capturing rows is counted from the central row of the image.
- Range: Multiples of 4 between 4 (default: 160). We recommend keeping the default value (160).

capture_period

- The capturing period in unit [ms].
- Range: 50~1000 (default: 80)

capture_shutter

- Shutter lengths to use when acquiring 1 frame of data. Set the five values [SH1, SH2, SH3, SH4, G] in units of [us].
- The iTFSs have High-Dynamic Range (HDR) function for more reliable depth image acquisition. SH1~SH4 are the shutter lengths used to capture HDR depth and intensity images.
- SH1~SH4: The shutter lengths for the depth and intensity capturing mode (mode: 1~3). Range: 0, 2~600 (default: [400,40,4,2])
- G: gray camera shutter length (mode: 0). Range: 2~10000 (default: 800)

capture_limit

- Set to output a depth value 0 for pixels whose the intensity value is lower than the capture_limit at each HDR level.
- Range: 0~500 (default: 200)

data_output

- Set the kind of output data from the sensor.

[data_output Bit Information]

Bit	Name
0	data_output_depth[0]
1	data_output_intensity[0]
2	data_output_status[0]
3~7	reserved

- data_output_depth: depth output flag
    - 0: do not output depth
    - 1: do output depth
- data_output_intensity: intensity output flag
    - 0: do not output intensity
    - 1: do output intensity
- data_output_status: STATUS output flag
    - 0: output STATUS packet
    - 1: output STATUS_FULL packet

data_sensor_ip

- IP address of the sensor.

data_dest_ip

- Destination IP of the sensor. The sensor will send data to the destination IP.

data_subnet, data_gateway

- Subnet and gateway.

data_port

- Program port information.

sync

- sync has information about the synchronization method and the behavior of the STROBE pin. The sync byte contains the following parameters;

[sync Bit Information]

Bit	Name
0	sync_mode[0]
1	sync_mode[1]
2	strobe[0]

Bit	Name
3	strobe[1]
4~7	reserved

- sync_mode: synchronization method
  - 0: do not use synchronization
  - 1: UDP-based synchronization
  - 2: TRIGGER-based synchronization
- strobe can be set to indicate exact illumination time of the sensor. When the strobe function is on, the STROBE pin is set to 5V while the sensor is illuminating and has a value of 0V the rest of the time.
  - 0: Turn off strobe
  - 1: Turn on strobe
- For more information about the synchronization feature, see theSynchronization.

sync_delay

- Valid when synchronization is enabled. This parameter determines how much to delay capturing at the time of synchronization.
- Range: 0 ~ capture_period
- For more information about the synchronization feature, see theSynchronization.

arb

- Sets the method of the Auto-Reboot (arb). If the Auto-Reboot is enabled, the sensor will automatically reboot if it does not receive a synchronization signal from the user for arb_timeout [ms]. The parameters contained in the arb byte are as follows;

[arb Bit Information]

Bit	Name
0	arb_mode[0]
1	arb_mode[1]
2~7	reserved

- arb_mode: the synchronization signal detection criteria
  - 0: disable Auto-Reboot
  - 1: Detect SYNC packets via UDP
  - 2: Detect SYNC signal via TRIGGER pin

arb_timeout

- Sets the time when an automatic reboot will occur. The unit is [ms].

lock

- lock indicates the status of the configuration lock. If set to 1, this prevents the sensor configuration changes via INFO packets. The lock status cannot be changed by sending INFO packets, and only can be changed by COMMAND packets.

CMD Packet

- CMD packets are packets that issue commands to the sensor. cmd_id is the command ID and cmd_msg is an additional message used as needed.

[CMD Packet Information]

Name	cmd_id	cmd_msg
Index	0	2
Size	2	2
Type	uint16	uint16

- The list of possible cmd_id and cmd_msg is as follows;

[cmd_id and cmd_msg Information]

Name	cmd_id	cmd_msg	Description
Sync	0x0000	0x0000	Synchronization for multiple sensors
Measure	0x0100	0x0000	Start capturing
Pause	0x0101	0x0000	Pause capturing
Reboot	0x0102	0x0000	Reboot sensor
Store	0x0103	0x0000	Store current setting
Factory Reset	0x0200	serial_number	Factory reset
Read Info	0x0300	0x0000	Query info
Redirect	0x0400	0x0000	Redirect destination IP
Lock	0x0500	serial_number	Set configuration lock bit
Unlock	0x0501	serial_number	Clear configuration lock bit

Sync

- This command broadcasts when the sensors are synchronized when using multiple sensors. For more information about the synchronization feature, see theSynchronization.

Measure

- Start capturing.

Pause

- Pause capturing.

Reboot

- Reboot sensor.

Store

- Saves the current sensor's settings (INFO) to the internal FLASH so that they are retained after the sensor reboots.

Factory Reset

- When the sensor receives this packet, the sensor compares its serial number and the serial number in the packet. If it matches, the sensor will factory reset and set all parameters to default.

Read Info

- Used when a user queries INFO from a sensor. Upon receiving this packet, the sensor sends an INFO packet with the current settings to data_dest_ip.

Redirect

- Set the sensor's data_dest_ip value to the IP address of the user who sent the Redirect command.

Lock/Unlock

- The received sensor turns the lock byte of the INFO ON or OFF. A sensor with the lock byte ON will not update the sensor settings when it receives an INFO packet from the user.

INFO V2 Packet

- INFO V2 packets provide information about the current operation parameter of the sensor and is supported by firmware from F/W V1.5.0 onwards. INFO V2 packets can be sent from the sensor to the user, or from the user to the sensor.
- For more information about changing sensor settings via INFO V2 packets, see the Configuration Process.

[INFO Packet Payload]

Name	Index	Size	Type	Authority

Name	Index	Size	Type	Authority
sensor_sn	0	2	uint16	R-
sensor_hw_id	2	1x30	uint8	R-
sensor_fw_ver	32	1x3	uint8	R-
sensor_fw_date	35	1x12	char	R-
sensor_fw_time	47	1x9	char	R-
sensor_calib_id	56	4	uint32	R-
sensor_fw0_ver	60	1x3	uint8	R-
sensor_fw1_ver	63	1x3	uint8	R-
sensor_fw2_ver	66	1x3	uint8	R-
sensor_model_id	69	1	uint8	R-
sensor_boot_mode	70	1	uint8	R-
capture_mode	71	1	uint8	RW
capture_row	72	1	uint8	RW
capture_shutter	73	2x5	uint16	RW
capture_limit	83	2x2	uint16	RW
capture_period_us	87	4	uint32	RW
capture_seq	91	1	uint8	RW
data_output	92	1	uint8	RW
data_baud	93	4	uint32	RW
data_sensor_ip	97	1x4	uint8	RW
data_dest_ip	101	1x4	uint8	RW
data_subnet	105	1x4	uint8	RW
data_gateway	109	1x4	uint8	RW
data_port	113	2	uint16	RW
data_mac_addr	115	1x6	uint8	RW
sync	121	1	uint8	RW
sync_trig_delay_us	122	4	uint32	RW
sync_ill_delay_us	126	2x15	uint16	RW
sync_trig_trim_us	156	1	uint8	RW
sync_ill_trim_us	157	1	uint8	RW

Name	Index	Size	Type	Authority
sync_output_delay_us	158	2	uint16	RW
arb	160	1	uint8	RW
arb_timeout	161	4	uint32	RW
lock	165	1	uint8	RW

sensor_sn

- Serial number of the sensor.

sensor_hw_id

- Hardware ID of the sensor.

sensor_fw_ver

- Firmware version of the sensor.

sensor_fw_date

- Build date of the sensor.

sensor_fw_time

- Build time of the sensor.

sensor_calib_id

- Calibration data id of the sensor.

sensor_fw0_ver

- The backup firmware version of the sensor.

sensor_fw1_ver

- The firmware version on sector 1 of the flash memory.

sensor_fw2_ver

- The firmware version on sector 2 of the flash memory.

sensor_model_id

- Model ID of the sensor.

sensor_boot_mode

- Boot mode register of the sensor.

capture_mode

- Indicates capturing mode of the sensor. Please check table [List of capture_mode].

capture_row

- The number of pixel rows to use for capturing depth and intensity values on the image sensor.
- The number of capturing rows is counted from the central row of the image.
- Range: Multiples of 4 between 4 (default: 160). We recommend keeping the default value (160).

capture_shutter

- Shutter lengths to use when acquiring 1 frame of data. Set the five values [SH1, SH2, SH3, SH4, G] in units of [us].
- The iTFSs have High-Dynamic Range (HDR) function for more reliable depth image acquisition. SH1~SH4 are the shutter lengths used to capture HDR depth and intensity images.
- SH1~SH4: The shutter lengths for the depth and intensity capturing mode (mode: 1~3). Range: 0, 2~600 (default: [400,40,4,2])
- G: gray camera shutter length (mode: 0). Range: 2~10000 (default: 800)

capture_limit

- Set to output a depth value 0 for pixels whose the intensity value is lower than the capture_limit at each HDR level.
- Range: 0~500 (default: 200)

capture_period_us

- Set the capture period of the sensor in [us] units. Setting range: 80000~1000000

capture_seq

- Sets the shooting sequence of the shutter defined in capture_shutter.
- 0 (default): Forward, shooting in SH1-SH2-SH3 order.
- 1: Backward, shoots SH3-SH2-SH1

data_output

- Set the kind of output data from the sensor.

[data_output Bit Information]

Bit	Name
0	data_output_depth[0]
1	data_output_intensity[0]
2	data_output_status[0]
3~7	reserved

- data_output_depth: depth output flag
  - 0: do not output depth
  - 1: do output depth
- data_output_intensity: intensity output flag
  - 0: do not output intensity
  - 1: do output intensity
- data_output_status: STATUS output flag
  - 0: output STATUS packet
  - 1: output STATUS_FULL packet

data_baud

- Set the baud rate for UART communication. Setting range: 9600~60000000 (default: 115200)

data_sensor_ip

- IP address of the sensor.

data_dest_ip

- Destination IP of the sensor. The sensor will send data to the destination IP.

data_subnet, data_gateway

- Subnet and gateway.

data_port

- Program port information.

data_mac_addr

- MAC address of the sensor.

sync

- sync has information about the synchronization method and the behavior of the STROBE pin. The sync byte contains the following parameters;

[sync Bit Information]

Bit	Name
0	sync_mode[0]
1	sync_mode[1]
2	strobe[0]
3	strobe[1]
4~7	reserved

- sync_mode: synchronization method
    - 0: do not use synchronization
    - 1: UDP-based synchronization
    - 2: TRIGGER-based synchronization
- strobe can be set to indicate exact illumination time of the sensor. When the strobe function is on, the STROBE pin is set to 5V while the sensor is illuminating and has a value of 0V the rest of the time.
    - 0: Turn off strobe
    - 1: Turn on strobe
- For more information about the synchronization feature, see theSynchronization.

sync_trig_delay_us

- The delay value from the synchronization point when using the sync feature.
- Setting range: 0 to capture_period_us (default: 0)
- For more information about the synchronization feature, see theSynchronization.

sync_ill_delay_us

- Value that sets an additional time delay between raw images being taken when using the sync feature.
- Default: 0
- For more information about the synchronization feature, see theSynchronization.

sync_trig_trim_us

- This is a value that is trimmed from the sync_trig_delay_us value to account for communication delays and internal clock errors when using the sync feature.
- The value of (sync_trig_delay_us - sync_trig_trim_us) is used as the delay value from the synchronization point.
- Default value: 4
- For more information about the synchronization feature, see theSynchronization.

sync_ill_trim_us

- This is a value that is trimmed from the sync_ill_delay_us value to account for communication delays and internal clock errors when using the sync function.
- The value of (sync_ill_delay_us[0,1,2,...] - sync_ill_trim_us) is used as the actual time delay value between raw images.
- Default value: 2
- For more information about the synchronization feature, see theSynchronization.

sync_output_delay_us

- This value sets an additional time delay between capturing data and sending it to the user after the sensor finishes internal calculations.
- This can be used to prevent traffic overload when using multiple sensors.
- Default: 0

arb

- Sets the method of the Auto-Reboot (arb). If the Auto-Reboot is enabled, the sensor will automatically reboot if it does not receive a synchronization signal from the user for arb_timeout [ms]. The parameters contained in the arb byte are as follows;

[arb Bit Information]

Bit	Name
0	arb_mode[0]
1	arb_mode[1]
2~7	reserved

- arb_mode: the synchronization signal detection criteria
  - 0: disable Auto-Reboot
  - 1: Detect SYNC packets via UDP
  - 2: Detect SYNC signal via TRIGGER pin

arb_timeout

- Sets the time when an automatic reboot will occur. The unit is [ms].

lock

- lock indicates the status of the configuration lock. If set to 1, this prevents the sensor configuration changes via INFO packets. The lock status cannot be changed by sending INFO packets, and only can be changed by COMMAND packets.

CMD Packet

- CMD packets are packets that issue commands to the sensor. cmd_id is the command ID and cmd_msg is an additional message used as needed.

[CMD Packet Information]

Name	cmd_id	cmd_msg
Index	0	2
Size	2	2
Type	uint16	uint16

- The list of possible cmd_id and cmd_msg is as follows;

[cmd_id and cmd_msg Information]

Name	cmd_id	cmd_msg	Description
Sync	0x0000	0x0000	Synchronization for multiple sensors
Measure	0x0100	0x0000	Start capturing
Pause	0x0101	0x0000	Pause capturing
Reboot	0x0102	0x0000	Reboot sensor
Store	0x0103	0x0000	Store current setting
Factory Reset	0x0200	serial_number	Factory reset
Read Info	0x0300	0x0000	Query info
Redirect	0x0400	0x0000	Redirect destination IP
Lock	0x0500	serial_number	Set configuration lock bit
Unlock	0x0501	serial_number	Clear configuration lock bit

Sync

- This command broadcasts when the sensors are synchronized when using multiple sensors. For more information about the synchronization feature, see theSynchronization.

Measure

- Start capturing.

Pause

- Pause capturing.

Reboot

- Reboot sensor.

Store

- Saves the current sensor's settings (INFO) to the internal FLASH so that they are retained after the sensor reboots.

Factory Reset

- When the sensor receives this packet, the sensor compares its serial number and the serial number in the packet. If it matches, the sensor will factory reset and set all parameters to default.

Read Info

- Used when a user queries INFO from a sensor. Upon receiving this packet, the sensor sends an INFO packet with the current settings to data_dest_ip.

Redirect

- Set the sensor's data_dest_ip value to the IP address of the user who sent the Redirect command.

Lock/Unlock

- The received sensor turns the lock byte of the INFO ON or OFF. A sensor with the lock byte ON will not update the sensor settings when it receives an INFO packet from the user.

How to Handle Image and Point Cloud

- This chapter describes how to obtain images and point clouds from output data from sensors.

Image Data

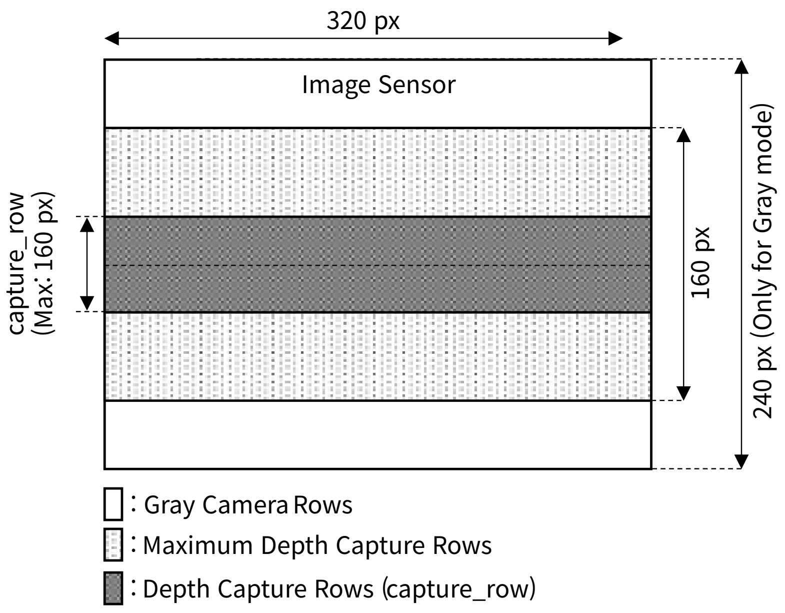
- This chapter covers how the data is acquired depending on the sensor's mode and how the user receives the image data. For the actual implementation, please refer to the OpenCV Example in C and Python in Software Examples.

Capture Row

- capture_row, which is defined in the INFO packet, is a parameter that selects the capturing rows on the image sensor. It can be set to any multiple of 4 in the range of 4 to 160. The total number of pixels in the iTFS sensor image element is 320x240. Depending on the operating mode (capture_mode), the following rows are captured;

[Number of capturing pixel row based on capture_row]

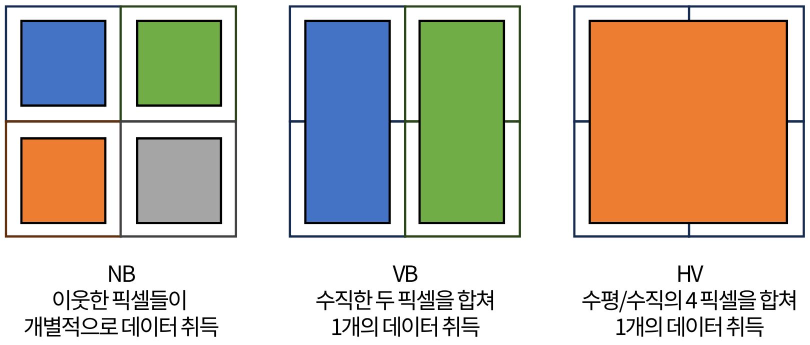
Mode	capture_mode	Capturing Rows
Gray	0	320x240 (fixed)
NB/VB/HV	1,2,3	320 x [capture_row]


320 px
Image Sensor
capture_row (Max: 160 px)
160 px
240 px (Only for Gray mode)
: Gray Camera Rows
: Maximum Depth Capture Rows
: Depth Capture Rows (capture_row)

Binning

- The iTFS sensor provides a built-in binning function that combines signals between neighboring pixels, increasing the strength of the signal at the expense of reducing resolution. There are three binning modes available depending on capture_mode. ([List of capture_mode])

- NB(No Binning): Do not use binning. Full resolution.
- VB(Vertical Binning): Measured by summing the signals of two vertically neighboring pixels. Half vertical resolution.
- HV(Horizontal and Vertical binning): Measured by summing the signals of four horizontally and vertically neighboring pixels. Half horizontal and vertical resolution.

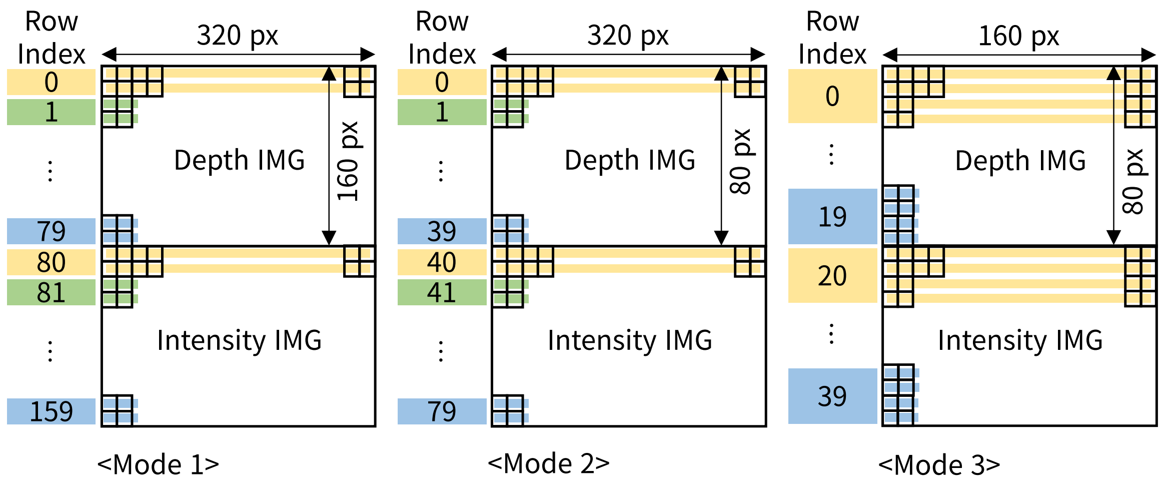


Accumulating Image Packet

- The total bytes of the image taken by the sensor is larger than 1500 bytes, which is the Maximum Transmission Unit (MTU) of Ethernet. Therefore, to ensure reliable transmission of the image data, the depth image and intensity image are divided into several lines and sent to the user via IMG packets. The user can extract the image information from the IMG packet and stack them according to the row_index to get the raw depth image and raw intensity image.

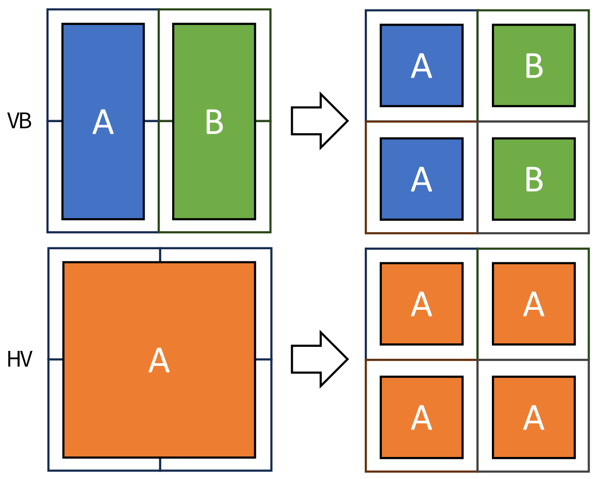
[Mode and Image Format]

Mode	Total Pixels	Rows per IMG Packet	Depth row_index	Intensity row_index
NB	320 x 160 x 2	2	0~79	80~159
VB	320 x 80 x 2	2	0~39	40~79
HV	160 x 80 x 2	4	0~19	20~39



- For VB and HV, the resolution of the output image is different from NB. To get images with the same resolution regardless of the mode, restore the resolution in the following way.
  - VB: Restores the resolution by copying each pixel value once in the vertical direction.

  - HV: Restores the resolution by copying each pixel value once in both the horizontal and vertical directions.



Point Cloud

- This chapter covers how to reconstruct point clouds from image data. For an actual implementation, please refer to the PCL Example of C in Software Examples.
- To obtain a point cloud from an iTFS sensor, you need to convert from the image coordinate system to the lidar local coordinate system and then to the world coordinate system, as shown below.

Image Coordinates
LiDAR Local Coordinates
World Coordinates

Image Coordiantes to LiDAR Local Coordinates

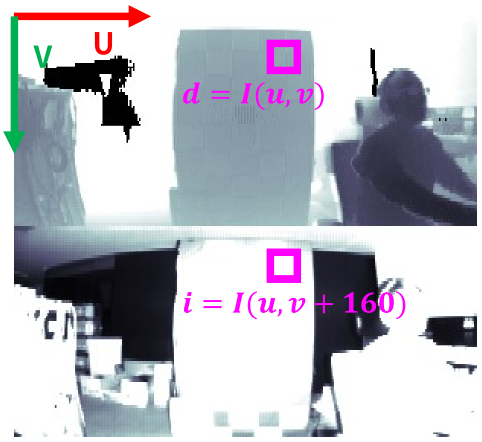
- You need to perform a conversion from the image coordinate system to the lidar's three-dimensional coordinate system. This requires depth and intensity image data configured using the procedures in the Image Data chapter, as well as a camera intrinsic data file (provided in *.dat format for iTFS sensors).
  - In the image coordinate system (U, V), the depth value is $d = I(u, v)$ and the intensity value is $i = I(u, v + 160)$.
  - The pixel-by-pixel three-dimensional orientation vector $V(u, v)$ defined in the intrinsic data file:

$$V(u, v) = (V_x, V_y, V_z)[v][u] = \mathrm{vec}[v][u][3]$$

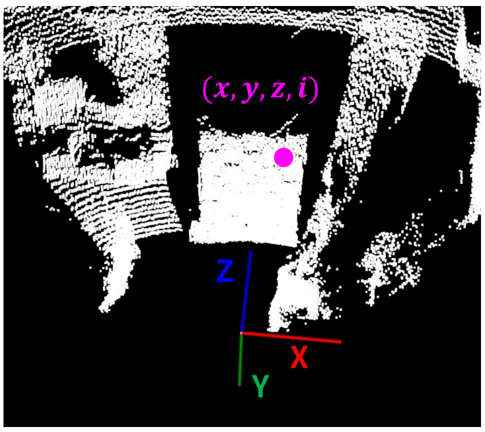
$$V_{\{x,y,z\}} = \mathrm{vec}[v + (240 - 160)/2][u][\{0, 1, 2\}]$$

- Using the values above, the coordinates $p(x, y, z)$ of the point corresponding to each pixel are calculated as follows;

$$p(x, y, z) = d(u, v) \times V(u, v) = (d \times V_x, d \times V_y, d \times V_z)$$



<Image Coordinates>

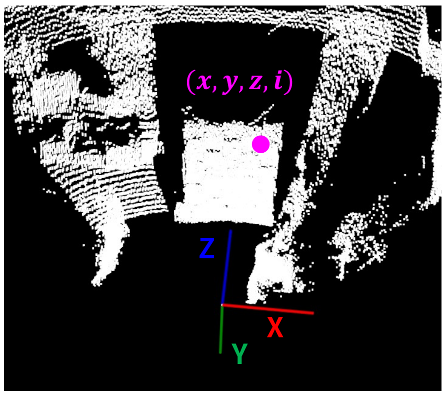


<LiDAR Local Coordinates>

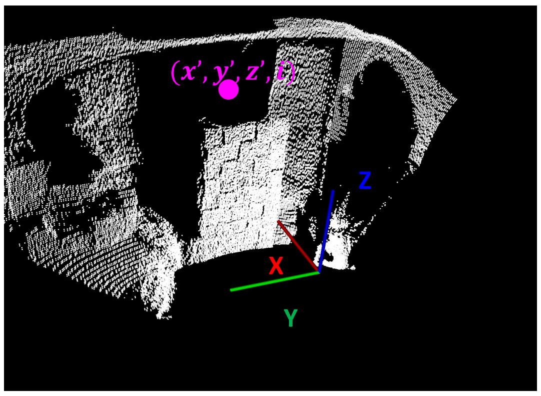
LiDAR Local Coordinates to World Coordinates

- You will need to perform a conversion from the lidar's local coordinate system to the world coordinate system. To do this, you will need;
  - Coordinates $p(x, y, z)$ of each point in the point cloud in the local coordinate system
  - Rotation matrix R_o that aligns the X, Y axis of the local coordinate system to the world coordinate system
  - 6 dof position and orientation information of the lidar in the world coordinate system $R|T$
- Using the above information, we can determine the position p_{world} of the point cloud in the world coordinate system as follows;

$$p_{\text{world}} = R \times R_o \times p + T$$



<LiDAR Local Coordinates>



<World Coordinates>

Software Examples

ilidar-api-cpp (C/C++)

- The ilidar-api-cpp example explains how to control or acquire data from an iTFS sensor using the C/C++ language. The example consists of three detailed programs.
  - Helloworld: describes the APIs that provide connectivity, control, and parameterization of sensors.
  - OpenCV Example: describes how to use the OpenCV library to display data acquired from an iTFS sensor in the form of depth and intensity images.
  - PCL Example: Describes how to use the Point Cloud Library (PCL) to display data acquired from iTFS sensors in the form of a three-dimensional point cloud.
- For more information about this example, please refer the following page: https://github.com/ilidar-tof/ilidar-api-cpp

ilidar-api-py (Python)

- The ilidar-api-py example teaches you how to use the Python language to acquire data coming from an iTFS sensor. The example consists of two small codes;
  - Helloworld: describes the APIs that provide connectivity, control, and parameterization of sensors.
  - OpenCV Example: describes how to use the OpenCV library to display data acquired from an iTFS sensor in the form of depth and intensity images.
- For more information about this example, please refer the following page: https://github.com/ilidar-tof/ilidar-api-py

ilidar-api-ros (ROS1)

- ilidar-api-ros is a ROS package that acquires and topics data from iTFS sensors in the ROS environment. The package also provides a three-dimensional point cloud representation of the acquired data from iTFS sensors using RViz. The following are the topics provided by the package;
  - /ilidar/depth: depth image
  - /ilidar/intensity: intensity image
  - /ilidar/points: 3D point cloud
  - /ilidar/gray: 940 nm gray view image due to external light source, enabled only when capture_mode is 0
- For more information about this example, please refer the following page: https://github.com/ilidar-tof/ilidar-api-ros

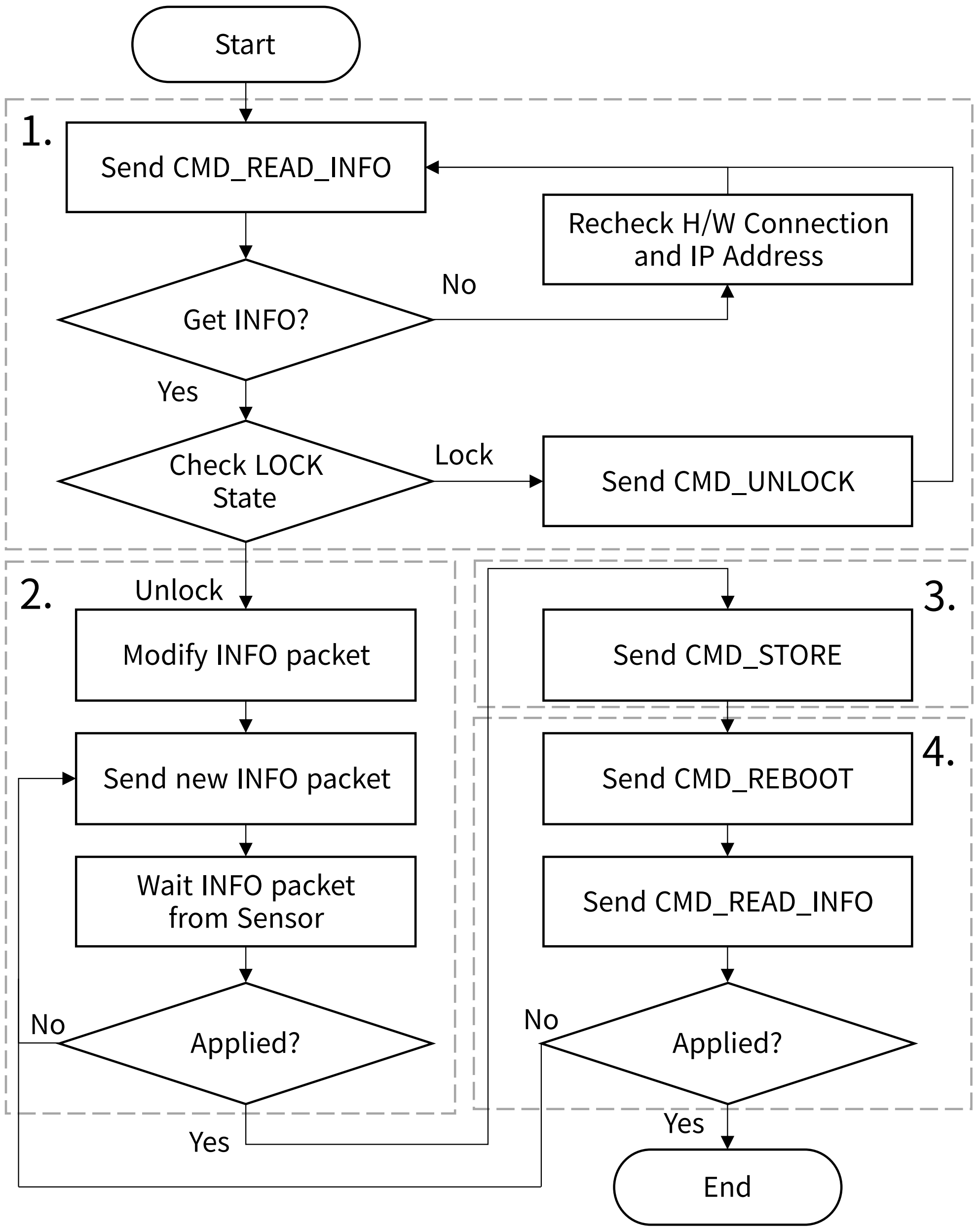
ilidar-api-ros2 (ROS2)

- Support for ROS2 is currently undergoing.

Configuration Process

- Checking and changing sensor operation parameters is done via CMD and INFO (or INFO_V2) packets. For a detailed description of each packet, see the previous chapter.
- The following flowchart shows how to change the sensor settings.

1. Send a CMD_READ_INFO packet to the sensor. If the iTFS sensor successfully receives CMD_READ_INFO, it sends the sensor's configuration information in the INFO packet. If the user does not receive the INFO packet, please check the hardware and IP related settings. Check the value of lock in the INFO packet and send CMD_UNLOCK if the configuration is locked to make the sensor's settings changeable.
2. Prepare and send the INFO packet for the new configuration. The sensor changes its settings to the values in the INFO packet it receives from the user. To prevent unintentional changes to the settings, **prepare a new INFO packet by copying the received INFO packet and changing the values only where the settings need to be changed**. The sensor receiving the INFO packet behaves as follows;
    1. Check each value in the INFO packet and change it to the default value if it is outside the valid range.
    2. Apply the changed settings.
    3. Send the new INFO packet to the user → The user receives the packet and can verify that the settings were changed successfully.
3. The sensor that normally receives the INFO packet will behave according to the changed settings (except for IP address-related values, which will take effect after a reboot). However, **user must send CMD_STORE to preserve the settings after a reboot**. The sensor that receives CMD_STORE will save their current settings to FLASH memory and read them back after reboot. Tip) If you make a mistake with the sensor settings, you can revert to the previous settings by performing a reboot without sending CMD_STORE.
4. Reboot the sensor by sending CMD_REBOOT, followed by CMD_READ_INFO, and check the return INFO packet to see if the settings were saved.

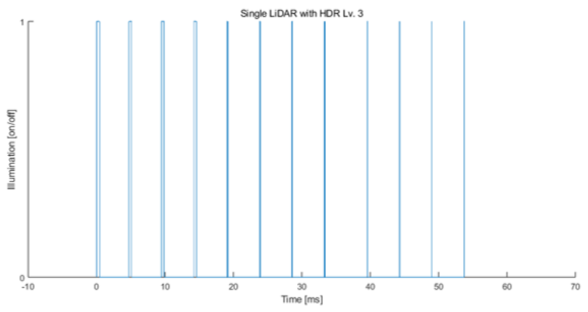

Start
1.
Send CMD_READ_INFO
Recheck H/W Connection and IP Address
No
Get INFO?
Yes
Check LOCK State
Lock
Send CMD_UNLOCK
2.
Unlock
Modify INFO packet
Send new INFO packet
Wait INFO packet from Sensor
No
Applied?
Yes
3.
Send CMD_STORE
4.
Send CMD_REBOOT
Send CMD_READ_INFO
No
Applied?
Yes
End

Synchronization

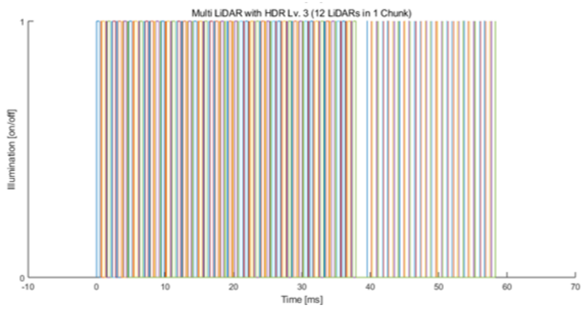
- The iTFS sensor is a Flash-type lidar, which means that it emits light to the entire RoI, and light reflected from objects is also received by the entire RoI. As a result, interference may occur with sensors that use the same wavelength and modulation frequency. (iLidar-ToF: iTFS series uses a 940 nm light source)
- To avoid interference when using multiple iTFS sensors, Time Division Multiple Access (TDMA) function is included. This chapter describes the synchronization method for this.

Illumination Timing Diagrams

- The iTFS sensor captures images at different times depending on capture_mode and HDR level. For example, when a sensor set to Mode 1, HDR 3 acquires data, it takes 12 measurements, four for each HDR step, and the measurement time window (the time between shooting and receiving light) is shown in the figure below.



- Therefore, when operating multiple sensors with the same Mode 1, HDR 3 in the same space, interference can be avoided by staggering the times at which individual lidars capture data so that they do not overlap in the time domain, as shown in the figure below (each color represents a different iTFS sensor).



- This requires that all sensors share the same reference time (synchronization), and that you set which time window each sensor chooses from the reference time. The rest of the chapter describes this.

Synchronization Methods

- The synchronization method changes depending on the sync_mode value set in the sync byte in INFO.

[sync_mode Information]

Type	sync_mode	Description
None	0	No synchronization
UDP	1	Synchronizatino with UDP
Trigger	2	Synchronization with Trigger pin

UDP Synchronization

- UDP synchronization is a synchronization method based on the time when the sensor receives a CMD_SYNC packet over UDP. This usually consists of a user broadcasting CMD_SYNC packets.
- For effects due to network latency, a delay of 3 to 4 [us] or less in the receipt of SYNC packets by individual sensors is operational.
- For UDP synchronization, the sensor has the following settings;
  - sync_mode: 1
  - sync_trig_delay_us: integer between 0~ capture_period_us, unit: [us]
- Sequence of actions:
  - (1) When a sensor receives a SYNC packet, it sets received time as its reference time.
  - (2) Start capturing data after (sync_trig_delay_us - sync_trig_trim_us) [us] from the reference time, and capture and send data every capture_period thereafter.
  - (3) Over time, the reference time between sensors will drift out of sync, so the user periodically broadcasts SYNC packets to keep the reference time in sync.

Trigger Synchronization

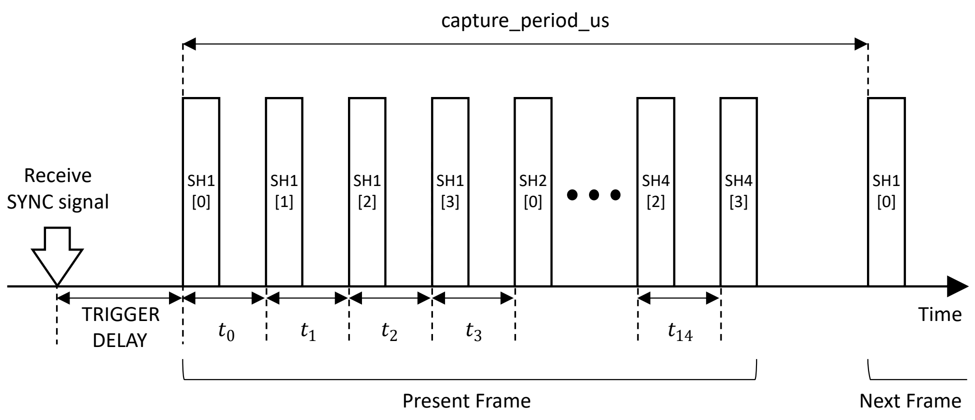
- Trigger synchronization is a synchronization method that uses the signal from the TRIGGER pin of the sensor as the reference time. The user must manually input the synchronization signal to the simultaneously used sensors.
- For Trigger synchronization, the sensor has the following settings;
  - sync_mode: 2
  - sync_trig_delay_us: integer between 0~ capture_period_us, unit: [us]
- Sequence of actions:
  - (1) When a sensor receives a TTL high signal in its TRIGGER pin, it sets received time as its reference time.
  - (2) Start capturing data after (sync_trig_delay_us - sync_trig_trim_us) [us] from the reference time, and capture and send data every capture_period thereafter.
  - (3) Over time, the reference time between sensors will drift out of sync, so the user periodically set the TTL high in TRIGGER pins to keep the reference time in sync.

Optical Synchronization (in development)

- A way for individual sensors to synchronize by analyzing the light emitted by other sensors without wired synchronization. Currently under development.

Fine-Tuning Time Window

- To avoid interference between sensors, you can fine-tune the measurement time windows of individual sensors.



- When the Synchronization feature is enabled, the sensor starts acquiring data after a TRIGGER DELAY based on when the SYNC signal is received. The TRIGGER DELAY can be set as follows
    - sync_trig_delay_us: Delay settings value
    - sync_ill_trim_us: Compensation value for communication delays

$$\text{TRIGGER DELAY} = (\text{sync\_trig\_delay\_us}) - (\text{sync\_trig\_trim\_us})$$

- Four raw images are taken for each HDR step. 16 raw images are taken for the maximum 4-step HDR setting to calculate the depth image and intensity image. When the time interval between raw images in the same frame is $t_i(i = 0 \sim 14)$, you can adjust the interval by setting a non-zero value for sync_ill_delay_us[i]. If the value set is shorter than Minimum delay, it will behave as Minimum delay.
    - Minimum delay
        - Mode 1: capture_shutter + 3900 [us]
        - Mode 2: capture_shutter + 1950 [us]
        - Mode 3: capture_shutter + 975 [us]
    - sync_ill_delay_us [i]: Delay settings value for each [i]
    - sync_ill_trim_us: Compensation value for internal clock error, apply the same value to all [i]

$$t_i = \text{MAX}(\text{Minimum delay}, \text{sync\_ill\_delay\_us[i]} - \text{sync\_ill\_trim\_us})$$

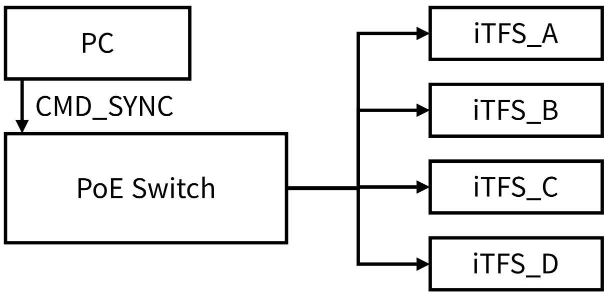
Multi Sensor Examples

- The following is how to use four iTFS sensors simultaneously via UDP synchronization. For reliable synchronization, please check the following;
    - Make sure all sensors have the same capture_mode value.
    - Make sure all sensors have the same capture_period value.
    - Make sure all sensors have the same sync_mode value.
    - Make sure that the sensor-specific sync_delay value is set so that the illumination profiles do not overlap.
- An example of a sensor setup that satisfies the above conditions is shown below.

[Configuration for 4 iTFS Synchronization]

Parameters	iTFS_A	iTFS_B	iTFS_C	iTFS_D
capture_mode	1			
capture_period	80			
capture_shutter	[400,80,16,0,800]			
sync	1			
sync_delay	0	20	40	60

- Connect the PC and iTFS sensors as shown below, and set the PC to broadcast CMD_SYNC packets periodically (approximately once every 2-3 minutes).



Sensor Maintenance Guide

Hardware Maintenance and Safety Guide

Electrical Requirements and Safety Conditions

- Always ensure that the electrical junctions of the wires connecting to the product are well-connected. Loose contacts can cause malfunction, poor signal quality, or fire.
- This product is not waterproof. Do not allow water, conductive liquids, flammable powders and liquids, or other liquids to come into contact with the sensor. In particular, do not allow liquids to come into contact with the electrical junctions of the sensor.
- High humidity can cause condensation inside the product and cause electrical damage.
- Avoid direct contact of the human body or conductive objects with electrical junctions of the product, even when the product is not energized.
- Use devices that have been certified for performance and safety to power and signal the product.
- Check the allowable voltage range of the product's power supply before connecting. Unstable power supply or voltage outside the permissible voltage range may cause permanent damage to the product.
- To ensure stable operation of the product, monitor the following values in the STATUS packet, and if they are out of the allowed range, stop using the product immediately and check the power supply method.

[Electrical Monitoring List]

Parameters	Description	Minimum	Maximum
sensor_vcsel_level	Transmitter Voltage	11.00	11.65
sensor_power_level	Transmitter Power Voltage	17.5	21.5

Temperature and Heat Dissipation

- Do not allow the product and the power device to become damaged or overheated.
- When the product is in operation, the sensor surface temperature may rise above the temperature of a typical environment (25 degrees Celsius). Direct skin contact with the surface of the product or lingering around the product may cause burns, so be sure to check the surface temperature of the product before touching it.
- To ensure the normal operation of the product, please allow enough space around the heatsink on the back of the product to allow air to flow freely.
- To maintain the normal operation of the product, the user may need to introduce additional heat dissipation methods such as ventilators and fans.
- To ensure the safe use of the product, please monitor the following parameters of the STATUS packet and STATUS_FULL packet, and stop using the product immediately if they are out of the allowable range.

[Thermal Monitoring List]

Parameters	Description	Minimum	Maximum
sensor_temp_rx	Image sensor temperature (Celsius)	-25	90
sensor_temp_core	PCB core temperature (Celsius)	-25	100
sensor_temp	Case temperature (Celsius)	-25	70

Optical Windows

- If the optical window is broken, stop using immediately in order to avoid personal injury.
- In general, the maintenance of the product requires the removal of dust and smudges from the optical window. Dust and smudges can negatively affect the behavior of the product, especially the measurement accuracy. In environments where dust and smudges are likely to be present, check the optical window periodically and clean it according to the methods below.
- Remove dust: If there is dust on the optical window, wiping it off directly can cause damage. In this case, use compressed air to remove dust from the optical window.
- Removing smudges: Use an optical lens tissue to remove smudges from the optical window. If there is also dust on the optical widow, please use compressed air to remove the dust before removing the smudge.

Warning Code

- The last part of the STATUS packet PAYLOAD consists of a 4-byte sensor_warning. Each bit value contains warning code information about an anomaly occurring during sensor operation. The user should monitor this value and react accordingly when a warning occurs.
- Voltage issues: Make sure you have a good power supply.
- Temperature issues: Ensure good airflow around the sensor. If necessary, provide additional heat dissipation, such as a fan.
- If the problem persists, please contact the manufacturer of the product.

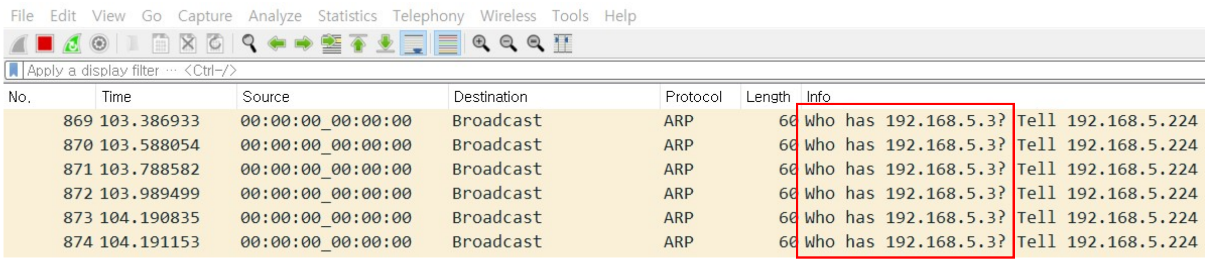
[List of Warning Code]

Bit	Description	Bit	Description
2	Reserver Overvoltage	14	5V Overvoltage
3	Reserver Undervoltage	15	5V Undervoltage
4	Transmitter Overvoltage	16	10V Overvoltage
5	Transmitter Undervoltage	17	10V Undervoltage
6	REF Overvoltage	18	-10V Overvoltage
7	REF Undervoltage	19	-10V Undervoltage
8	BAT Overvoltage	20	Receiver Overheat

Bit	Description	Bit	Description
9	BAT Undervoltage	21	Receiver Freezing
10	Input Overvoltage	22	MCU Overheat
11	Input Undervoltage	23	MCU Freezing
12	1.8V Overvoltage	24	Case Overheat
13	1.8V Undervoltage	25	Case Freezing

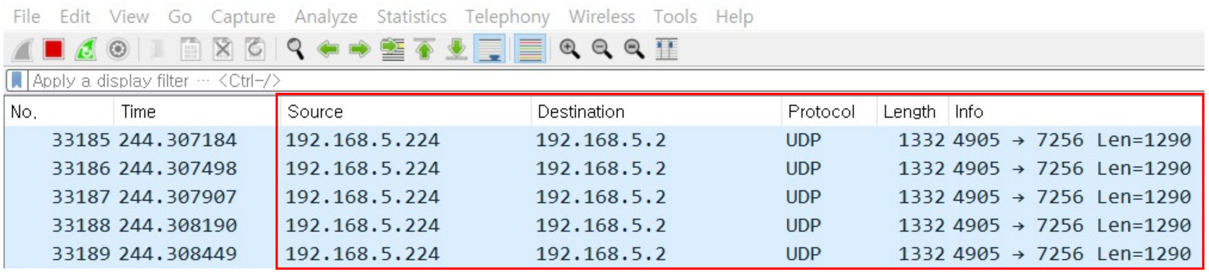
I need to verify that data from the sensor is reaching my PC / I forgot the sensor's destination IP address

- You can use an external program, Wireshark, to analyze all socket communication packets that reach your PC. Install Wireshark, and connect your sensor to your PC. Then run Wireshark, and select the socket device that the sensor is connected to.
- If the destination IP address of the sensor is different from the IP address of the PC, Info displays a statement that the iTFS sensor is looking for the destination IP.



```
No.   Time        Source             Destination  Protocol  Length  Info
  869 103.386933  00:00:00_00:00:00  Broadcast    ARP           60  Who has 192.168.5.3? Tell 192.168.5.224
  870 103.588054  00:00:00_00:00:00  Broadcast    ARP           60  Who has 192.168.5.3? Tell 192.168.5.224
  871 103.788582  00:00:00_00:00:00  Broadcast    ARP           60  Who has 192.168.5.3? Tell 192.168.5.224
  872 103.989499  00:00:00_00:00:00  Broadcast    ARP           60  Who has 192.168.5.3? Tell 192.168.5.224
  873 104.190835  00:00:00_00:00:00  Broadcast    ARP           60  Who has 192.168.5.3? Tell 192.168.5.224
  874 104.191153  00:00:00_00:00:00  Broadcast    ARP           60  Who has 192.168.5.3? Tell 192.168.5.224
```

- If the destination IP address of the sensor and the IP address of your PC match, you'll see data being sent from the iTFS sensor to your PC.



```
No.     Time        Source         Destination  Protocol  Length  Info
  33185 244.307184  192.168.5.224  192.168.5.2  UDP         1332  4905 → 7256 Len=1290
  33186 244.307498  192.168.5.224  192.168.5.2  UDP         1332  4905 → 7256 Len=1290
  33187 244.307907  192.168.5.224  192.168.5.2  UDP         1332  4905 → 7256 Len=1290
  33188 244.308190  192.168.5.224  192.168.5.2  UDP         1332  4905 → 7256 Len=1290
  33189 244.308449  192.168.5.224  192.168.5.2  UDP         1332  4905 → 7256 Len=1290
```