

---

# Ouster Sensor API Guide

*Firmware v2.4.0 for all Ouster sensors*

**Ouster**

**Sep 23, 2022**

# TCP and HTTP APIs

---

<b>1</b>	<b>Sensor Configuration</b>	<b>3</b>
1.1	Configurable Parameters . . . . .	3
1.2	Description- Configurable Parameters . . . . .	4
<b>2</b>	<b>TCP API Guide</b>	<b>12</b>
2.1	Querying Sensor Info and Intrinsic Calibration . . . . .	12
2.2	Sensor Configuration Parameter Interface . . . . .	13
2.3	Sensor Status and Calibration . . . . .	16
<b>3</b>	<b>HTTP API Reference Guide</b>	<b>29</b>
3.1	Sensor Metadata . . . . .	29
3.2	System . . . . .	40
3.3	Time . . . . .	43
3.4	Alerts, Diagnostics and Telemetry . . . . .	51
<b>4</b>	<b>API Changelog</b>	<b>59</b>

---

This on-sensor reference document is a quick guide to the TCP and HTTP APIs. For the most up-to-date and comprehensive information about your sensor, please see the sensor user manuals found at [www.ouster.com](http://www.ouster.com)

# 1 Sensor Configuration

## 1.1 Configurable Parameters

Please find a list of all **config\_param** that can be used to configure the sensor below. For detailed description of each parameter refer to [Description- Configurable Parameters](#).

Table1: Overview

Parameter	Type	Valid Values
<code>udp_dest</code>	String	"" (default)
<code>udp_port_lidar</code>	Integer	7502 (default)
<code>udp_port_imu</code>	Integer	7503 (default)
<code>sync_pulse_in_polarity</code>	Keyword	ACTIVE_HIGH (default) ACTIVE_LOW
<code>sync_pulse_out_polarity</code>	Keyword	ACTIVE_LOW (default) ACTIVE_HIGH
<code>sync_pulse_out_frequency</code>	Integer >= 1	1 (default)
<code>sync_pulse_out_angle</code>	Integer [0 ... 360]	360 (default)
<code>sync_pulse_out_pulse_width</code>	Integer >= 0	10 (default)
<code>nmea_in_polarity</code>	Keyword	ACTIVE_HIGH (default) ACTIVE_LOW
<code>nmea_ignore_valid_char</code>	integer [ 0 ... 1 ]	0 (default) 1
<code>nmea_baud_rate</code>	Keyword	BAUD_9600 (default) BAUD_115200
<code>nmea_leap_seconds</code>	Integer >= 0	0 (default)
<code>azimuth_window</code>	List	[0,360000] (default)
<code>signal_multiplier</code>	number [ 1 ... 3 ]	1 (default) 2 3
<code>udp_profile_lidar</code>	Keyword	LEGACY (default) RNG19_RFL8_SIG16_NIR16 RNG19_RFL8_SIG16_NIR16_DUAL RNG15_RFL8_NIR8
<code>udp_profile_imu</code>	Keyword	LEGACY (default)
<code>phase_lock_enable</code>	Boolean	false (default) true

continues on next page

Table 1 - continued from previous page

Parameter	Type	Valid Values
<code>phase_lock_offset</code>	Integer [ 0 ... 360 ]	0 (default)
<code>lidar_mode</code>	Keyword	512x10 1024x10 (default) 2048x10 512x20 1024x20
<code>timestamp_mode</code>	Keyword	TIME_FROM_INTERNAL_OSC (default) TIME_FROM_PTP_1588 TIME_FROM_SYNC_PULSE_IN
<code>multipurpose_io_mode</code>	Keyword	OFF (default) INPUT_NMEA_UART OUTPUT_FROM_INTERNAL_OSC OUTPUT_FROM_SYNC_PULSE_IN OUTPUT_FROM_PTP_1588 OUTPUT_FROM_ENCODER_ANGLE
<code>operating_mode</code>	Keyword	NORMAL (default) STANDBY

## 1.2 Description- Configurable Parameters

---

`udp_dest`

### Description:

- Type: String
- Default: "169.254.198.184"
- Destination to which the sensor sends UDP traffic.

---

**Note:** As of now, setting the `udp_dest` to "@auto" is not supported through the `set_config_param` TCP command. It is only supported through HTTP endpoint **POST** `/api/v1/sensor/config`, and the `set_udp_dest_auto` using TCP command. `udp_ip` flag has been **deprecated** in **Firmware 2.4**, please use `udp_dest` flag.

---

#### udp\_port\_lidar

##### **Description:**

- Type: Integer [ 0 ... 65535 ]
- Default: 7502
- The <port> on `udp_dest` to which lidar data will be sent (7502, default).

#### udp\_port\_imu

##### **Description:**

- Type: Integer [ 0 ... 65535 ]
- Default: 7503
- The <port> on `udp_dest` to which IMU data will be sent (7503, default).

#### sync\_pulse\_in\_polarity

##### **Description:**

- Type: Keyword
- Default: "ACTIVE\_HIGH"
- **Enum:**
  - "ACTIVE\_HIGH"
  - "ACTIVE\_LOW"
- The polarity of SYNC\_PULSE\_IN input, which controls polarity of SYNC\_PULSE\_IN pin when `timestamp_mode` is set in `TIME_FROM_SYNC_PULSE_IN`.

#### sync\_pulse\_out\_polarity

##### **Description:**

- Type: Keyword
- Default: "ACTIVE\_HIGH"
- The polarity of SYNC\_PULSE\_OUT output, if the sensor is set as the master sensor used for time synchronization.

sync\_pulse\_out\_frequency

**Description:**

- Type: Integer  $\geq 1$
- Default: 1
- The output SYNC\_PULSE\_OUT pulse rate in Hz. Valid inputs are integers  $>0$  Hz, but also limited by the criteria described in the Time Synchronization section of the Software User Manual.

sync\_pulse\_out\_angle

**Description:**

- Type: Integer [ 0 ... 360 ]
- Default: 360
- The angle in terms of degrees that the sensor traverses between each SYNC\_PULSE\_OUT pulse. E.g. a value of 180 means a sync pulse is sent out every 180° for a total of two pulses per revolution and angular frequency of 20 Hz if the sensor is 1024x10 Hz lidar mode. Valid inputs are integers between 0 and 360 inclusive but also limited by the criteria described in the Time Synchronization section of Software User Manual.

sync\_pulse\_out\_pulse\_width

**Description:**

- Type: Integer  $\geq 0$
- Default: 10
- The polarity of SYNC\_PULSE\_OUT output, if the sensor is set as the master sensor used for time synchronization. Output SYNC\_PULSE\_OUT pulse width is in ms, increments in 1 ms. Valid inputs are integers greater than 0 ms, but also limited by the criteria described in the Time Synchronization section of Software User Manual.

nmea\_in\_polarity

**Description:**

- Type: Keyword
- Default: "ACTIVE\_HIGH"
- **Enum:**
  - "ACTIVE\_HIGH"
  - "ACTIVE\_LOW"

- Set the polarity of NMEA UART input \$GPRMC messages. See Time Synchronization section in sensor user manual for NMEA use case. Use `ACTIVE_HIGH` if UART is active high, idle low, and start bit is after a falling edge.

#### `nmea_ignore_valid_char`

##### **Description:**

- Type: Integer [ 0 ... 1 ]
- Default: 0
- Set `0` if NMEA UART input \$GPRMC messages should be ignored if valid character is not set, and `1` if messages should be used for time syncing regardless of the valid character.

#### `nmea_baud_rate`

##### **Description:**

- Type: Keyword
- Default: "BAUD\_9600"
- **Enum:**
  - "BAUD\_9600"
  - "BAUD\_115200"
- `BAUD_9600` (default) or `BAUD_115200` for the expected baud rate the sensor is attempting to decode for NMEA UART input \$GPRMC messages.

#### `nmea_leap_seconds`

##### **Description:**

- Type: Integer  $\geq 0$
- Default: 0
- Set an integer number of leap seconds that will be added to the UDP timestamp when calculating seconds since 00:00:00 Thursday, 1 January 1970. For Unix Epoch time, this should be set to `0`.

### azimuth\_window

#### Description:

- Type: List
- Default: [0,360000]
- Set the visible region of interest of the sensor in millidegrees. Only data from within the specified azimuth window bounds is sent. The value should be provisioned as: [min\_bound\_millideg, max\_bound\_millideg]

### signal\_multiplier

#### Description:

- Type: Number [ 1 ... 3 ]
- Default: 1
- The value that the signal\_multiplier is configured. By default the sensor has a signal multiplier value of 1.

For 2x and 3x multipliers, the `azimuth_window` parameter sets the azimuth window that the lasers will be enabled in.

The higher the signal multiplier value, the smaller the maximum azimuth window can be.

Signal Multiplier Value Max Azimuth Window 1: ( Default) 360°, 2: 180°, 3: 120°.

All sensors have equivalent power draw and thermal output when operating at the max azimuth window for a particular signal multiplier value. Therefore, using an azimuth window that is smaller than the maximum allowable azimuth window with a particular signal multiplier value (excluding 1x) can reduce the power draw and thermal output of the sensor.

However, while this can increase the max operating temp of the sensor, it can also degrade the performance at low temps. This discrepancy will be resolved in a future firmware. The table below outlines some example use cases.

### udp\_profile\_lidar

#### Description:

- Type: Keyword
- Default: "LEGACY"
- **Enum:**
  - "LEGACY"
  - "RNG19\_RFL8\_SIG16\_NIR16"
  - "RNG19\_RFL8\_SIG16\_NIR16\_DUAL"



- "RNG15\_RFL8\_NIR8"
- The configuration of the LIDAR data packets. Valid values are **LEGACY** [Default], **RNG19\_RFL8\_SIG16\_NIR16**, **RNG19\_RFL8\_SIG16\_NIR16\_DUAL**, **RNG15\_RFL8\_NIR8**.

#### udp\_profile\_imu

##### **Description:**

- Type: Keyword
- Default: "LEGACY"
- Value: "LEGACY"
- The configuration of the IMU data packets. Valid value is **LEGACY**.

#### phase\_lock\_enable

**Description:** Whether phase locking is enabled. Refer to Phase Lock Section in the Firmware User Manual for more details on using phase lock.

- Type: Boolean
- Default: False
- Whether phase locking is enabled. Refer to Phase Lock Section in the Firmware User Manual for more details on using phase lock.

#### phase\_lock\_offset

##### **Description:**

- Type: Integer [ 0 ... 360000 ]
- Default: 0
- The angle in the Lidar Coordinate Frame that sensors are locked to in millidegrees if phase locking is enabled. Angle is traversed at the top of the second.

#### lidar\_mode

##### **Description:**

- Type: Keyword
- Default: "1024x10"
- **Enum:**
  - "512x10"

- "1024x10"
  - "2048x10"
  - "512x20"
  - "1024x20"
- The horizontal resolution and rotation rate of the sensor. The effective range of the sensor is increased by 15-20% for every halving of the number of points gathered e.g. 512x10 has 15-20% longer range than 512x20.

timestamp\_mode

**Description:**

The method used to timestamp measurements. Valid modes are `TIME_FROM_INTERNAL_OSC`, `TIME_FROM_SYNC_PULSE_IN`, or `TIME_FROM_PTP_1588`.

multipurpose\_io\_mode

**Description:**

- Type: Keyword
- Default: "OFF"
- **Enum:**
  - "OFF"
  - "INPUT\_NMEA\_UART"
  - "OUTPUT\_FROM\_INTERNAL\_OSC"
  - "OUTPUT\_FROM\_SYNC\_PULSE\_IN"
  - "OUTPUT\_FROM\_PTP\_1588"
  - "OUTPUT\_FROM\_ENCODER\_ANGLE"
- Configure the mode of the MULTIPURPOSE\_IO pin. Refer to Time Synchronization section in Firmware user manual for a detailed description of each option.

`operating_mode`

**Description:**

- Type: Any
- Default: "NORMAL"
- Set `NORMAL` to put the sensor into a normal operating mode or `STANDBY` to put the sensor into a low power (5W) operating mode where the motor does not spin and lasers do not fire.

---

**Note:** `auto_start_flag` is deprecated parameter in Firmware 2.4 and later. `auto_start_flag 0` is equivalent to `operating_mode STANDBY` and `auto_start_flag 1` is equivalent to `operating_mode NORMAL`.

---

## 2 TCP API Guide

---

### 2.1 Querying Sensor Info and Intrinsic Calibration

---

The sensor can be queried and configured using a simple plaintext protocol over TCP on port 7501.

An example session using the unix netcat utility is shown below. Note: "xxx" refers to the sensor serial number. The hostname of the sensor can look like "os-xxx" or "os1-xxx".

```
$ nc os-122201000998.local 7501
get_sensor_info

{"prod_line": "OS-1-128", "prod_pn": "840-103575-06", "prod_sn": "122201000998",
 "image_rev": "ousteros-image-prod-aries-v2.4.0-omega.2+20220730010801.staging",
 "build_rev": "v2.4.0-omega.2", "build_date": "2022-07-29T23:56:15Z",
 "status": "RUNNING", "initialization_id": 9599937}
```

A sensor may have one of the following statuses:

Table 2: Sensor Status

Status	Description
INITIALIZING	When the sensor is booting and not yet outputting data
WARMUP	Sensor has gone into thermal warmup state
UPDATING	When the sensor is updating the FPGA firmware on the first reboot after a firmware upgrade
RUNNING	When the sensor has reached the final running state where it can output data
STANDBY	The sensor has been configured into a low-power state where sensor is on but not spinning
ERROR	Check error codes in the <code>errors</code> field for more information
UNCONFIGURED	An error with factory calibration that requires a manual power cycle or reboot

---

**Note:** If the sensor is set to **STANDBY** mode some of these commands will not return the expected values.

---

If the sensor is in an **ERROR** or **UNCONFIGURED** state, please contact [Ouster support](#) with the diagnostic file found at <http://os-9919xxxxxxx/diag> for support.

## 2.2 Sensor Configuration Parameter Interface

---

### `get_config_param`

**Description:** Returns all active or staged JSON-formatted sensor configuration.

Sensor configurations and operating modes can also be queried over TCP. Below is the command format:

- `get_config_param active <parameter>` will return the current active configuration parameter values.
- `get_config_param staged <parameter>` will return the parameter values that will take place after issuing a `reinitialize` command.

**Warning:** The command `get_config_txt` is deprecated and superseded by `get_config_param active`, which provides the same response. `get_config_txt` will be removed in a future firmware.

**Example 1:** Shows how a user can `get` information of a particular parameter below using the unix netcat utility:

```
$ nc os-991900123456 7501
get_config_param active lidar_mode
1024x10
```

**Example 2:** Shows how a user can `get` all the `<active>` `parameters` information below using the unix netcat utility:

```
$ nc os-991900123456 7501
get_config_param active
{
  "udp_dest": "169.254.111.53",
  "udp_port_lidar": 7502,
  "udp_port_imu": 7503,
  "udp_profile_lidar": "RNG19_RFL8_SIG16_NIR16_DUAL",
  "udp_profile_imu": "LEGACY",
  "columns_per_packet": 16,
  "timestamp_mode": "TIME_FROM_INTERNAL_OSC",
  "sync_pulse_in_polarity": "ACTIVE_HIGH",
  "nmea_in_polarity": "ACTIVE_HIGH",
  "nmea_ignore_valid_char": 0,
  "nmea_baud_rate": "BAUD_9600",
  "nmea_leap_seconds": 0,
  "multipurpose_io_mode": "OFF",
  "sync_pulse_out_polarity": "ACTIVE_HIGH",
  "sync_pulse_out_frequency": 1,
  "sync_pulse_out_angle": 360,
  "sync_pulse_out_pulse_width": 10,
  "operating_mode": "NORMAL",
  "lidar_mode": "1024x10",
  "azimuth_window": [0, 360000],
  "signal_multiplier": 1,
```

(continues on next page)

```
"phase_lock_enable": false,  
"phase_lock_offset": 0  
}
```

### set\_config\_param

`set_config_param <parameter> <value>` will set new values for configuration parameters, which will take effect after issuing the `reinitialize` command.

`reinitialize` will reinitialize the sensor so the staged values of the parameters will take effect immediately.

`save_config_params` will write new values of active parameters into a configuration file, so they will persist after sensor reset. In order to permanently change a parameter in the configuration file, first use `set_config_param` to update the parameter in a staging area, then use `reinitialize` to make that parameter active. Only after the parameter is made active will `save_config_params` capture it to persist after reset.

**Warning:** The command `write_config_txt` will be deprecated in a future firmware. The command `save_config_params` provides the same response.

While in **STANDBY** mode, we can set the config parameters, but it will not take effect until we switch the sensor back to **NORMAL** mode.

`set_udp_dest_auto` will automatically determine the sender's IP address at the time the command was sent, and set it as the destination of UDP traffic. This takes effect after issuing a `reinitialize` command. Using this command has the same effect as using `set_config_param udp_dest <ip address>`.

An example session using the unix netcat utility is shown below.

---

**Note:** In the example below, to distinguish between the command and expected response, a dash has been added before the expected response. The actual response will be without the dash.

---

```
$ nc os-991900123456.local 7501
set_config_param lidar_mode 512x20
-set_config_param
set_udp_dest_auto
-set_udp_dest_auto
reinitialize
-reinitialize
save_config_params
-save_config_params
```

---

**Note:** In the example below, see the **error** message when an invalid value is set i.e., **lidar\_mode = 511x10**.

---

```
$ nc os-122201000998.local 7501
set_config_param lidar_mode 511x10
error: '511x10' is not supported
```

The following commands will set sensor configuration parameters:

---

**Note:** Each of the following commands have two responses: \* **set\_config\_param** on Success \* **error:** Otherwise

---

**Table3:** Reinitialize, Save Sensor Config, and Auto Setting of Destination IP

Command	Command Description
<b>reinitialize</b> or <b>reinit</b>	Restarts the sensor. Changes to lidar, multipurpose_io, and timestamp modes will only take effect after reinitialization. <b>Response on success:</b> <b>reinitialize</b> or <b>reinit</b>
<b>save_config_params</b>	Makes all current parameter settings persist after reboot. <b>Response on success:</b> <b>save_config_params</b>
<b>set_udp_dest_auto</b>	Set the destination of UDP traffic to the destination address that issued the command. <b>Response on success:</b> <b>set_udp_dest_auto</b>

---

**Note:** Refer to [Sensor Configuration](#) for detailed information on all configurable parameters using TCP commands.

---

## 2.3 Sensor Status and Calibration

---

### `get_sensor_info`

**Description:** Returns JSON-formatted sensor metadata: serial number, hardware and software revision, and sensor status.

```
{
  "prod_line": "0S-1-128",
  "prod_pn": "840-103575-06",
  "prod_sn": "122201000998",
  "image_rev": "ousteros-image-prod-aries-v2.4.0-omega.2+20220730010801.staging",
  "build_rev": "v2.4.0-omega.2",
  "build_date": "2022-07-29T23:56:15Z",
  "status": "RUNNING",
  "initialization_id": 9599937
}
```

### `get_time_info`

**Description:** Returns JSON-formatted sensor timing configuration and status of udp `timestamp`, `sync_pulse_in`, and `multipurpose_io`.

```
{
  "timestamp":
  {
    "time": 3709.04727264,
    "mode": "TIME_FROM_INTERNAL_OSC",
    "time_options":
    {
      "ptp_1588": 3718,
      "sync_pulse_in": 1,
      "internal_osc": 3709
    }
  },
  "sync_pulse_in":
  {
    "locked": 0,
    "polarity": "ACTIVE_HIGH",
    "diagnostics":
    {
      "last_period_nsec": 0,
      "count": 1,
      "count_unfiltered": 0
    }
  },
  "multipurpose_io":
  {
    "mode": "OFF",
    "sync_pulse_out":
    {
      "polarity": "ACTIVE_HIGH",
      "frequency_hz": 1,

```

(continues on next page)



(continued from previous page)

```
    "angle_deg": 360,
    "pulse_width_ms": 10
  },
  "nmea":
  {
    "locked": 0,
    "polarity": "ACTIVE_HIGH",
    "ignore_valid_char": 0,
    "baud_rate": "BAUD_9600",
    "leap_seconds": 0,
    "diagnostics":
    {
      "decoding":
      {
        "utc_decoded_count": 0,
        "date_decoded_count": 0,
        "not_valid_count": 0,
        "last_read_message": ""
      },
      "io_checks":
      {
        "start_char_count": 0,
        "char_count": 0,
        "bit_count": 1,
        "bit_count_unfiltered": 0
      }
    }
  }
}
```

### get\_beam\_intrinsic

**Description:** Returns JSON-formatted beam altitude and azimuth offsets, in degrees. Length of arrays is equal to the number of channels in the sensor. Also returns distance between lidar origin and beam origin in mm, to be used for point cloud calculations.

```
{
  "beam_altitude_angles": [21.01, 20.72, 20.42,
    20.1, 19.79, 19.48, 19.17,
    18.84, 18.55, 18.22, 17.9, 17.6,
    17.27, 16.96, 16.65, 16.32, 15.97,
    15.65, 15.34, 15.01, 14.67, 14.35,
    14.01, 13.68, 13.33, 13.02, 12.67,
    12.34, 11.99, 11.65, 11.33, 10.98,
    10.64, 10.28, 9.949999999999999,
    9.609999999999999, 9.27, 8.92, 8.57,
    8.23, 7.88, 7.54, 7.18, 6.84, 6.47,
    6.13, 5.78, 5.45, 5.09, 4.73,
    4.41, 4.05, 3.69, 3.32, 2.98,
    2.63, 2.27, 1.93, 1.57, 1.22,
    0.85, 0.5, 0.15, -0.19, -0.55, -0.92,
```

(continues on next page)

```
-1.25, -1.63, -1.98, -2.31, -2.67,  
-3.04, -3.4, -3.74, -4.09, -4.45, -4.8,  
-5.15, -5.5, -5.87, -6.21, -6.57, -6.91,  
-7.25, -7.62, -7.95, -8.300000000000001,  
-8.65, -9.01, -9.35, -9.69, -10.05,  
-10.39, -10.74, -11.09, -11.42, -11.77,  
-12.11, -12.45, -12.8, -13.14, -13.47,  
-13.81, -14.15, -14.48, -14.82, -15.13,  
-15.47, -15.81, -16.15, -16.48, -16.8,  
-17.15, -17.48, -17.79, -18.12, -18.47,  
-18.77, -19.09, -19.42, -19.73, -20.06,  
-20.36, -20.69, -21, -21.32, -21.62, -21.94],  
"beam_azimuth_angles": [4.23, 1.41, -1.4,  
-4.21, 4.22, 1.42, -1.41, -4.22, 4.23,  
1.41, -1.42, -4.21, 4.23, 1.42, -1.4,  
-4.2, 4.23, 1.41, -1.39, -4.21, 4.25,  
1.43, -1.41, -4.22, 4.24, 1.44, -1.41,  
-4.22, 4.23, 1.42, -1.38, -4.22, 4.24,  
1.42, -1.4, -4.23, 4.26, 1.44, -1.41,  
-4.23, 4.24, 1.44, -1.41, -4.23, 4.24,  
1.42, -1.41, -4.24, 4.25, 1.42, -1.39,  
-4.22, 4.25, 1.41, -1.4, -4.23, 4.24,  
1.43, -1.41, -4.23, 4.24, 1.42, -1.41,  
-4.23, 4.25, 1.42, -1.4, -4.24, 4.24,  
1.44, -1.4, -4.24, 4.24, 1.43, -1.4, -4.24,  
4.25, 1.43, -1.41, -4.24, 4.25, 1.42, -1.42,  
-4.22, 4.24, 1.43, -1.4, -4.24, 4.25, 1.44,  
-1.41, -4.24, 4.25, 1.42, -1.41, -4.24,  
4.25, 1.42, -1.41, -4.25, 4.26, 1.43, -1.41,  
-4.26, 4.27, 1.43, -1.4, -4.24, 4.25, 1.43,  
-1.4, -4.24, 4.26, 1.42, -1.4, -4.25, 4.26,  
1.43, -1.41, -4.26, 4.26, 1.42, -1.42,  
-4.26, 4.25, 1.42, -1.42, -4.27],  
"lidar_origin_to_beam_origin_mm": 15.806  
}
```

`get_imu_intrinsics`

**Description:** Returns JSON-formatted IMU transformation matrix needed to transform to the Sensor Coordinate Frame.

```
{  
  "imu_to_sensor_transform":  
  [  
    1, 0, 0, 6.253, 0, 1, 0, -11.775,  
    0, 0, 1, 7.645, 0, 0, 0, 1  
  ]  
}
```

## get\_lidar\_intrinsics

**Description:** Returns JSON-formatted lidar transformation matrix needed to transform to the Sensor Coordinate Frame.

```
{
  "lidar_to_sensor_transform":
  [
    -1, 0, 0, 0, 0, -1, 0, 0, 0, 0,
    1, 36.18, 0, 0, 0, 0, 1
  ]
}
```

## get\_alerts

**Example 1:** `get_alerts <START_CURSOR><MODE>`

**Description:** Returns JSON-formatted sensor diagnostic information.

Two lists will be returned, an **active** list and a **log** list. The **active** list will contain alert-trigger events for alerts that are currently active. An alert-trigger event will by-definition always have its active attribute set to true. There is no limit on the number of alert-trigger events that are displayed in the **active** event list. All currently active alert-trigger events will be displayed in the **active** event list.

The **log** list will contain all current and past alert-trigger and alert-clear events. An alert-clear event will by-definition have the exact same attributes and attribute values as its corresponding trigger event, with the exception of the realtime and cursor attributes which should have higher values, since an alert-clear event will always be received after an alert-trigger event. The **log** list has a length limit of 32 events with the oldest events automatically removed from the log list once a new event needs to be added to the **log** list and the **log** list length limit is reached, essentially acting as a FIFO (First In First Out) queue.

In addition to the **active** and **log** lists, `get_alerts` also returns a **next\_cursor** field. Every alert event has a cursor attribute, which increments for every alert event logged. This can be used to track the alert activity that has been viewed and reduce message bandwidth. To do this, users are recommended to save the **next\_cursor** field when calling `get_alerts` and then use that value as the **START\_CURSOR** argument on the next `get_alerts` call to fetch only new **log** entries.

A valid value for **MODE** is either `summary` or `default`.

---

**Note: Valid uses of get\_alerts:**

- Example: Calling get\_alerts with START\_CURSOR=1

```
get_alerts 1
```

- Example: Calling get\_alerts with START\_CURSOR=2 and MODE=summary

```
get_alerts 2 summary
```

**Invalid uses of get\_alerts:**

- Example: Calling get\_alerts with START\_CURSOR=summary and MODE=2

```
get_alerts summary 2
```

---

**Note:** The position of `START_CURSOR` and `MODE` is important to follow when using `TCP Commands` and is irrelevant when calling the equivalent `/api/v1/sensor/alerts` `HTTP endpoint`, meaning for an `HTTP GET` call the position is insensitive.

---

The example in the code block is for `get_alerts`.

```
{
  "log":
  [
    {
      "active": true,
      "category": "UDP_TRANSMISSION",
      "cursor": 0,
      "id": "0x01000018",
      "level": "WARNING",
      "msg": "Client machine announced
            it is not reachable on the
            provided not reachable on
            IMU data port; check that
            udp_dest and udp_port_imu
            configured on the sensor matches
            client IP and port.",
      "msg_verbose": "Failed to send
                     imu UDP data to destination
                     host 169.254.175.254:7503",
      "realtime": "39850161524"
    },
    {
      "active": true,
      "category": "UDP_TRANSMISSION",
      "cursor": 1,
      "id": "0x01000015",
      "level": "WARNING",
      "msg": "Client machine announced
            it is not reachable on the
```

(continues on next page)

```
        provided lidar data port;
        check that udp_dest and
        udp_port_lidar configured on
        the sensor matches client IP
        and port.",
    "msg_verbose": "Failed to send
        lidar UDP data to destination
        host 169.254.175.254:7502",
    "realtime": "40842065146"
},
{
    "active": true,
    "category": "ETHERNET_LINK_BAD",
    "cursor": 2,
    "id": "0x01000011",
    "level": "WARNING",
    "msg": "Ethernet link bad, please
        check network switch and
        harnessing can support
        1 Gbps Ethernet.",
    "msg_verbose": "Link transitioned
        to 0/Unknown",
    "realtime": "414257307390"
},
{
    "active": true,
    "category": "ETHERNET_LINK_BAD",
    "cursor": 2,
    "id": "0x01000011",
    "level": "WARNING",
    "msg": "Ethernet link bad, please
        check network switch and
        harnessing can support
        1 Gbps Ethernet.",
    "msg_verbose": "Link transitioned
        to 0/Unknown",
    "realtime": "414257307390"
},
{
    "active": true,
    "category": "UDP_TRANSMISSION",
    "cursor": 3,
    "id": "0x01000016",
    "level": "WARNING",
    "msg": "Could not send lidar data
        UDP packet to host; check that
        network is up.",
    "msg_verbose": "Failed to send
        lidar UDP data to destination
        host 169.254.175.254:7502",
    "realtime": "414261086316"
},
{
    "active": true,
```

```
"category": "UDP_TRANSMISSION",
"cursor": 4,
"id": "0x01000019",
"level": "WARNING",
"msg": "Could not send IMU UDP
      packet to host; check
      that network is up.",
"msg_verbose": "Failed to send imu
      UDP data to destination
      host 169.254.175.254:7503",
"realtime": "414266339945"
},
{
  "active": false,
  "category": "ETHERNET_LINK_BAD",
  "cursor": 5,
  "id": "0x01000011",
  "level": "WARNING",
  "msg": "Ethernet link bad,
        please check network switch
        and harnessing can support
        1 Gbps Ethernet.",
  "msg_verbose": "Link transitioned
        to 1000/Full",
  "realtime": "416337486469"
}
],
"next_cursor": 6,
"active":
[
  {
    "active": true,
    "category": "UDP_TRANSMISSION",
    "cursor": 1,
    "id": "0x01000015",
    "level": "WARNING",
    "msg": "Client machine announced
          it is not reachable on the
          provided lidar data port;
          check that udp_dest and
          udp_port_lidar configured
          on the sensor matches
          client IP and port.",
    "msg_verbose": "Failed to send
          lidar UDP data to destination
          host 169.254.175.254:7502",
    "realtime": "40842065146"
  },
  {
    "active": true,
    "category": "UDP_TRANSMISSION",
    "cursor": 3,
    "id": "0x01000016",
    "level": "WARNING",
```

```
"msg": "Could not send lidar data
  UDP packet to host;
  check that network is up.",
"msg_verbose": "Failed to send
  lidar UDP data to destination
  host 169.254.175.254:7502",
"realtime": "414261086316"
},
{
  "active": true,
  "category": "UDP_TRANSMISSION",
  "cursor": 0,
  "id": "0x01000018",
  "level": "WARNING",
  "msg": "Client machine announced
    it is not reachable on the provided
    not reachable on IMU data port;
    check that udp_dest and
    udp_port_imu configured on the
    sensor matches client IP
    and port.",
  "msg_verbose": "Failed to send imu
    UDP data to destination host
    169.254.175.254:7503",
  "realtime": "39850161524"
},
{
  "active": true,
  "category": "UDP_TRANSMISSION",
  "cursor": 4,
  "id": "0x01000019",
  "level": "WARNING",
  "msg": "Could not send IMU UDP
    packet to host; check that
    network is up.",
  "msg_verbose": "Failed to send
    imu UDP data to destination
    host 169.254.175.254:7503",
  "realtime": "414266339945"
}
]
}
```

**Example 2:** `get_alerts <START_CURSOR><MODE>`

**Description:** Returns JSON-formatted sensor diagnostic information. A valid value for **MODE** is either `summary` or `default`. Setting the **MODE** argument to `SUMMARY` provides a less verbose version of the `get_alerts <START_CURSOR>` command as referenced above.

---

**Note: Valid uses of get\_alerts:**

- Example: Calling `get_alerts` with `START_CURSOR=1`

```
get_alerts 1
```

- Example: Calling `get_alerts` with `START_CURSOR=2` and `MODE=summary`

```
get_alerts 2 summary
```

**Invalid uses of get\_alerts:**

- Example: Calling `get_alerts` with `START_CURSOR=summary` and `MODE=2`

```
get_alerts summary 2
```

---

**Note:** The order in which the `START_CURSOR` and `MODE` arguments are specified for the `get_alerts` TCP command must be followed when providing both arguments, with `START_CURSOR` preceding the `MODE`. The arguments can be specified in any order when calling the equivalent `/api/v1/sensor/alerts` HTTP endpoint.

---

The example shown in the code block is for `get_alerts summary`.

```
{
  "log":
  [
    {
      "cursor": 0,
      "id": "0x01000018",
      "realtime": "80395661548",
      "active": true
    },
    {
      "cursor": 1,
      "id": "0x01000015",
      "realtime": "81386289401",
      "active": true
    },
    {
      "cursor": 2,
      "id": "0x01000014",
      "realtime": "2730854039127",
      "active": true
    }
  ]
}
```

(continues on next page)



```
},
{
  "cursor": 3,
  "id": "0x01000014",
  "realtime": "2740849252064",
  "active": false
},
{
  "cursor": 4,
  "id": "0x01000011",
  "realtime": "8835209059341",
  "active": true
},
{
  "cursor": 5,
  "id": "0x01000016",
  "realtime": "8835240707086",
  "active": true
},
{
  "cursor": 6,
  "id": "0x01000019",
  "realtime": "8835245794318",
  "active": true
},
{
  "cursor": 7,
  "id": "0x01000011",
  "realtime": "8837298086954",
  "active": false
},
{
  "cursor": 8,
  "id": "0x01000015",
  "realtime": "10742075130316",
  "active": false
},
{
  "cursor": 9,
  "id": "0x01000016",
  "realtime": "10742075535820",
  "active": false
},
{
  "cursor": 10,
  "id": "0x01000018",
  "realtime": "10742075793868",
  "active": false
},
{
  "cursor": 11,
  "id": "0x01000019",
  "realtime": "10742076051916",
  "active": false
}
```

(continued from previous page)

```
    },
    {
      "cursor": 12,
      "id": "0x01000015",
      "realtime": "10799083353303",
      "active": true
    },
    {
      "cursor": 13,
      "id": "0x01000018",
      "realtime": "10799092477143",
      "active": true
    },
    {
      "cursor": 14,
      "id": "0x01000016",
      "realtime": "11782640857349",
      "active": true
    }
  ],
  "next_cursor": 15,
  "active":
  [
    {
      "cursor": 12,
      "id": "0x01000015",
      "realtime": "10799083353303"
    },
    {
      "cursor": 14,
      "id": "0x01000016",
      "realtime": "11782640857349"
    },
    {
      "cursor": 13,
      "id": "0x01000018",
      "realtime": "10799092477143"
    }
  ]
}
```

### get\_lidar\_data\_format

**Description:** Returns JSON-formatted response that describes the structure of a lidar packet.

- **columns\_per\_frame:** Number of measurement columns per frame. This can be 512, 1024, or 2048, depending upon the set lidar mode.
- **columns\_per\_packet:** Number of measurement blocks contained in a single lidar packet. Currently in v2.2.0 and earlier, this is 16. **Note:** This is not user configurable.
- **pixel\_shift\_by\_row:** Offset in terms of pixel count. Can be used to destagger image. Varies by lidar mode. Length of this array is equal to the number of channels of the sensor. **Note:** The

new value is centered around zero with both positive and negative values such that the destaggered image corresponds to the sensor's azimuth angles. The old destagger values introduced an offset to shift the *pixel\_shift\_by\_row* values so they are all non-negative.

- `pixels_per_column`: Number of channels of the sensor.
- `column_window`: Index of measurement blocks that are active. Default is [0, lidar\_mode-1], e.g. [0,1023]. If there is an azimuth window set, this parameter will reflect which measurement blocks of data are within the region of interest.
- `udp_profile_lidar` - Lidar data profile format [Default LEGACY].
- `udp_profile_lidar` - Lidar data profile format [Default LEGACY].
- `udp_profile_imu` - IMU data profile format [Default LEGACY].

---

**Note:** This command only works when the sensor is in **RUNNING** status.

---

```
{
  "column_window": [0, 1023],
  "columns_per_frame": 1024,
  "columns_per_packet": 16,
  "pixel_shift_by_row": [
    12, 4, -4, -12, 12, 4, -4,
    -12, 12, 4, -4, -12, 12, 4,
    -4, -12, 12, 4, -4, -12, 12,
    4, -4, -12, 12, 4, -4, -12,
    12, 4, -4, -12, 12, 4, -4,
    -12, 12, 4, -4, -12, 12, 4,
    -4, -12, 12, 4, -4, -12, 12,
    4, -4, -12, 12, 4, -4, -12,
    12, 4, -4, -12, 12, 4, -4,
    -12, 12, 4, -4, -12, 12, 4,
    -4, -12, 12, 4, -4, -12, 12,
    4, -4, -12, 12, 4, -4, -12,
    12, 4, -4, -12, 12, 4, -4,
    -12, 12, 4, -4, -12, 12, 4,
    -4, -12],
  "pixels_per_column": 128,
  "udp_profile_imu": "LEGACY",
  "udp_profile_lidar": "LEGACY"
}
```

### get\_calibration\_status

**Description:** Returns JSON-formatted calibration status of the sensor reflectivity. `valid`: true/false depending on calibration status. `timestamp`: if valid is true; time at which the calibration was completed.

```
{
  "reflectivity":
  {
    "valid": true,
    "timestamp": "2021-10-05T00:02:36"
  }
}
```

### get\_telemetry

**Description:** Returns JSON-formatted response that provides sensor system state information. This includes the FPGA **Timestamp** in `ns` (Nanoseconds) at which the information was collected from the FPGA, **Lidar Input Voltage** in `mv` (Millivolt), **Lidar Input Current** in `ma` (Milliamp), **Internal Temperature** of the sensor in `°C` (Degree Celsius) and **Phase Lock status** namely `LOCKED`, `LOST`, `DISABLED`.

---

**Note:** Using `get_telemetry`, Internal temperature can only be measured with **Rev 06** and above sensors.

---

---

**Note:** **Phase lock** output will not indicate loss of lock if the PTP source is lost.

---

```
{
  "input_current_ma": 758,
  "input_voltage_mv": 23606,
  "internal_temperature_deg_c": 45,
  "phase_lock_status": "DISABLED",
  "timestamp_ns": 2962666299310
}
```

# 3 HTTP API Reference Guide

---

This reference guide documents the interface for HTTP API and is accessible via `/api/v1` on the sensor hosted HTTP server.

The sensor can be queried and configured using an HTTP GET requests. This can be done using several different tools such as HTTPie, cURL, Advanced REST Client, etc.

Here is an example using **curl** command:

```
$ curl --request GET --url http://169.254.198.184/api/v1/sensor/metadata/lidar_intrinsics

{
  "lidar_to_sensor_transform": [-1, 0, 0, 0, 0, -1, 0, 0, 0, 0, 1, 36.18, 0, 0, 0, 1]
}
```

## 3.1 Sensor Metadata

---

### GET /api/v1/sensor/metadata/sensor\_info

GET 169.254.198.184/api/v1/sensor/metadata/sensor\_info  
Get the sensor information

```
GET /api/v1/sensor/metadata/sensor_info HTTP/1.1
Host: 169.254.198.184

HTTP/1.1 200 OK
content-length: 283
content-type: application/json

{
  "build_date": "2022-07-29T23:56:15Z",
  "build_rev": "v2.4.0-omega.2",
  "image_rev": "ousteros-image-prod-aries-v2.4.0-omega.2+20220730010801.staging",
  "initialization_id": 9599937,
  "prod_line": "OS-1-128",
  "prod_pn": "840-103575-06",
  "prod_sn": "122201000998",
  "status": "RUNNING"
}
```

**statuscode:** 200 No error

**Description:** Returns JSON-formatted response that includes serial number, product number, FW image revision and sensor status along with other parameters as shown is provided.

## GET /api/v1/sensor/metadata/lidar\_data\_format

GET 169.254.198.184/api/v1/sensor/metadata/lidar\_data\_format

Get the sensor lidar data format

```
GET /api/v1/sensor/metadata/lidar_data_format HTTP/1.1
```

```
Host: 169.254.198.184
```

```
HTTP/1.1 200 OK
```

```
content-type: application/json
```

```
content-length: 661
```

```
{
  "column_window": [0, 1023],
  "columns_per_frame": 1024,
  "columns_per_packet": 16,
  "pixel_shift_by_row": [
    12, 4, -4, -12, 12, 4, -4,
    -12, 12, 4, -4, -12, 12, 4,
    -4, -12, 12, 4, -4, -12, 12,
    4, -4, -12, 12, 4, -4, -12,
    12, 4, -4, -12, 12, 4, -4,
    -12, 12, 4, -4, -12, 12, 4,
    -4, -12, 12, 4, -4, -12, 12,
    4, -4, -12, 12, 4, -4, -12,
    12, 4, -4, -12, 12, 4, -4,
    -12, 12, 4, -4, -12, 12, 4,
    -4, -12, 12, 4, -4, -12, 12,
    4, -4, -12, 12, 4, -4, -12,
    12, 4, -4, -12, 12, 4, -4,
    -12, 12, 4, -4, -12, 12, 4,
    -4, -12],
  "pixels_per_column": 128,
  "udp_profile_imu": "LEGACY",
  "udp_profile_lidar": "LEGACY"
}
```

**statuscode:** 200 No error

**Description:** Returns JSON-formatted response that describes the structure of a lidar packet.

**columns\_per\_frame:** Number of measurement columns per frame. This can be 512, 1024, or 2048, depending upon the set lidar mode.

**columns\_per\_packet:** Number of measurement blocks contained in a single lidar packet. Currently in v2.2.0 and earlier, this is 16. **Note:** This is not user configurable.

**pixel\_shift\_by\_row:** Offset in terms of pixel count. Can be used to destagger image. Varies by lidar mode. Length of this array is equal to the number of channels of the sensor.

`pixels_per_column`: Number of channels of the sensor.

`column_window`: Index of measurement blocks that are active. Default is [0, lidar\_mode-1], e.g. [0,1023]. If there is an azimuth window set, this parameter will reflect which measurement blocks of data are within the region of interest.

`udp_profile_lidar`: Lidar data profile format. Default LEGACY.

`udp_profile_lidar`: Lidar data profile format. Default LEGACY.

`udp_profile_imu`: IMU data profile format. Default LEGACY.

**NOTE:** This command only works when the sensor is in **RUNNING** status.

## GET /api/v1/sensor/metadata/beam\_intrinsics

GET 169.254.198.184/api/v1/sensor/metadata/beam\_intrinsics

Get the sensor beam intrinsics

```
GET /api/v1/sensor/metadata/beam_intrinsics HTTP/1.1
Host: 169.254.198.184

HTTP/1.1 200 OK
content-type: application/json
content-length: 4506

{
  "beam_azimuth_angles": [4.2300000000000004, 1.4099999999999999, -1.3999999999999999, -4.21,
    4.2199999999999998, 1.4199999999999999, -1.4099999999999999, -4.2199999999999998, 4.2300000000000004,
    1.4099999999999999, -1.4199999999999999, -4.21, 4.2300000000000004, 1.4199999999999999,
    -1.3999999999999999, -4.2000000000000002, 4.2300000000000004, 1.4099999999999999, -1.3899999999999999,
    -4.21, 4.25, 1.4299999999999999, -1.4099999999999999, -4.2199999999999998, 4.2400000000000002,
    1.4399999999999999, -1.4099999999999999, -4.2199999999999998, 4.2300000000000004, 1.4199999999999999,
    -1.3799999999999999, -4.2199999999999998, 4.2400000000000002, 1.4199999999999999, -1.3999999999999999,
    -4.2300000000000004, 4.2599999999999998, 1.4399999999999999, -1.4099999999999999, -4.2300000000000004,
    4.2400000000000002, 1.4399999999999999, -1.4099999999999999, -4.2300000000000004, 4.2400000000000002,
    1.4199999999999999, -1.4099999999999999, -4.2400000000000002, 4.25, 1.4199999999999999,
    -1.3899999999999999, -4.2199999999999998, 4.25, 1.4099999999999999, -1.3999999999999999,
    -4.2300000000000004, 4.2400000000000002, 1.4299999999999999, -1.4099999999999999, -4.2300000000000004,
    4.2400000000000002, 1.4199999999999999, -1.4099999999999999, -4.2300000000000004, 4.25,
    1.4199999999999999, -1.3999999999999999, -4.2400000000000002, 4.2400000000000002, 1.4399999999999999,
    -1.3999999999999999, -4.2400000000000002, 4.2400000000000002, 1.4299999999999999, -1.3999999999999999,
    -4.2400000000000002, 4.25, 1.4299999999999999, -1.4099999999999999, -4.2400000000000002, 4.25,
    1.4199999999999999, -1.4199999999999999, -4.2199999999999998, 4.2400000000000002, 1.4299999999999999,
    -1.3999999999999999, -4.2400000000000002, 4.25, 1.4399999999999999, -1.4099999999999999,
    -4.2400000000000002, 4.25, 1.4199999999999999, -1.4099999999999999, -4.2400000000000002, 4.25,
    1.4199999999999999, -1.4099999999999999, -4.25, 4.2599999999999998, 1.4299999999999999,
    -1.4099999999999999, -4.2599999999999998, 4.2699999999999996, 1.4299999999999999, -1.3999999999999999,
    -4.2400000000000002, 4.25, 1.4299999999999999, -1.3999999999999999, -4.2400000000000002,
    4.2599999999999998, 1.4199999999999999, -1.3999999999999999, -4.25, 4.2599999999999998,
    1.4299999999999999, -1.4099999999999999, -4.2599999999999998, 4.2599999999999998, 1.4199999999999999,
    -1.4199999999999999, -4.2599999999999998, 4.25, 1.4199999999999999, -1.4199999999999999,
    -4.2699999999999996],
  "beam_altitude_angles": [21.010000000000002, 20.719999999999999,
    20.420000000000002, 20.100000000000001, 19.789999999999999, 19.48, 19.170000000000002, 18.84,
```

(continues on next page)

(continued from previous page)

```
18.550000000000001, 18.219999999999999, 17.899999999999999, 17.600000000000001, 17.27,
16.960000000000001, 16.649999999999999, 16.32, 15.970000000000001, 15.65, 15.34, 15.01, 14.67, 14.35,
14.01, 13.68, 13.33, 13.02, 12.67, 12.34, 11.99, 11.65, 11.33, 10.98, 10.640000000000001,
10.279999999999999, 9.949999999999993, 9.609999999999994, 9.269999999999996, 8.919999999999999,
8.570000000000003, 8.230000000000004, 7.879999999999999, 7.54, 7.179999999999997,
6.839999999999999, 6.469999999999998, 6.129999999999999, 5.780000000000002, 5.450000000000002,
5.089999999999999, 4.730000000000004, 4.410000000000001, 4.049999999999998, 3.689999999999999,
3.319999999999998, 2.98, 2.629999999999999, 2.27, 1.929999999999999, 1.570000000000001, 1.22,
0.849999999999998, 0.5, 0.149999999999999, -0.19, -0.550000000000004, -0.920000000000004,
-1.25, -1.629999999999999, -1.98, -2.310000000000001, -2.669999999999999, -3.04,
-3.399999999999999, -3.740000000000002, -4.089999999999999, -4.450000000000002,
-4.799999999999998, -5.150000000000004, -5.5, -5.870000000000001, -6.21, -6.570000000000003,
-6.910000000000001, -7.25, -7.620000000000001, -7.950000000000002, -8.300000000000007,
-8.650000000000004, -9.009999999999998, -9.349999999999996, -9.689999999999995,
-10.050000000000001, -10.390000000000001, -10.74, -11.09, -11.42, -11.77, -12.109999999999999,
-12.449999999999999, -12.800000000000001, -13.140000000000001, -13.470000000000001,
-13.81, -14.15, -14.48, -14.82, -15.130000000000001, -15.470000000000001, -15.81,
-16.149999999999999, -16.48, -16.800000000000001, -17.149999999999999, -17.48, -17.789999999999999,
-18.120000000000001, -18.469999999999999, -18.77, -19.09, -19.420000000000002, -19.73,
-20.059999999999999, -20.359999999999999, -20.690000000000001, -21, -21.32, -21.620000000000001,
-21.940000000000001],
"lidar_origin_to_beam_origin_mm": 15.805999999999999
}
```

**status code:** 200 No error

**Description:** Returns JSON-formatted beam altitude and azimuth offsets, in degrees. Length of arrays is equal to the number of channels in the sensor. Also returns distance between lidar origin and beam origin in mm, to be used for point cloud calculations.

### GET /api/v1/sensor/metadata/imu\_intrinsics

GET 169.254.198.184/api/v1/sensor/metadata/imu\_intrinsics  
Get the sensor imu intrinsics

```
GET /api/v1/sensor/metadata/imu_intrinsics HTTP/1.1
Host: 169.254.198.184

HTTP/1.1 200 OK
content-type: application/json
content-length: 117
{
  "imu_to_sensor_transform": [1, 0, 0, 6.253000000000001,
    0, 1, 0, -11.775, 0, 0, 1, 7.644999999999996, 0, 0, 0, 1]
}
```

**status code:** 200 No error

**Description:** Returns JSON-formatted IMU transformation matrix needed to transform to the Sensor Coordinate Frame.



## GET /api/v1/sensor/metadata/lidar\_intrinsics

GET 169.254.198.184/api/v1/sensor/metadata/lidar\_intrinsics

Get the sensor lidar intrinsics

```
GET /api/v1/sensor/metadata/lidar_intrinsics HTTP/1.1
Host: 169.254.198.184

HTTP/1.1 200 OK
content-type: application/json
content-length: 85

{
  "lidar_to_sensor_transform": [-1, 0, 0, 0, 0, -1, 0, 0, 0, 0, 1, 36.18, 0, 0, 0, 1]
}
```

**status code:** 200 No error

**Description:** Returns JSON-formatted lidar transformation matrix needed to transform to the Sensor Coordinate Frame.

## GET /api/v1/sensor/metadata/calibration\_status

GET 169.254.198.184/api/v1/sensor/metadata/calibration\_status

Get the sensor calibration status

```
GET /api/v1/sensor/metadata/calibration_status HTTP/1.1
Host: 169.254.198.184

HTTP/1.1 200 OK
content-type: application/json
content-length: 69

{
  "reflectivity":
  {
    "timestamp": "2021-10-05T00:02:36",
    "valid": true
  }
}
```

**status code:** 200 No error

**Description:** Returns JSON formatted calibration status of the sensor reflectivity. **valid:** true/false depending on calibration status. **timestamp:** if valid is true; time at which the calibration was completed.

## GET /api/v1/sensor/config

GET 169.254.198.184/api/v1/sensor/config  
Get sensor configuration parameter

```
GET /api/v1/sensor/config HTTP/1.1
Host: 169.254.198.184

HTTP/1.1 200 OK
content-length: 715
content-type: application/json

{
  "azimuth_window": [
    0,
    360000
  ],
  "columns_per_packet": 16,
  "lidar_mode": "1024x10",
  "multipurpose_io_mode": "OFF",
  "nmea_baud_rate": "BAUD_9600",
  "nmea_ignore_valid_char": 0,
  "nmea_in_polarity": "ACTIVE_HIGH",
  "nmea_leap_seconds": 0,
  "operating_mode": "NORMAL",
  "phase_lock_enable": false,
  "phase_lock_offset": 0,
  "signal_multiplier": 1,
  "sync_pulse_in_polarity": "ACTIVE_HIGH",
  "sync_pulse_out_angle": 360,
  "sync_pulse_out_frequency": 1,
  "sync_pulse_out_polarity": "ACTIVE_HIGH",
  "sync_pulse_out_pulse_width": 10,
  "timestamp_mode": "TIME_FROM_INTERNAL_OSC",
  "udp_dest": "169.254.198.184",
  "udp_port_imu": 7503,
  "udp_port_lidar": 7502,
  "udp_profile_imu": "LEGACY",
  "udp_profile_lidar": "RNG19_RFL8_SIG16_NIR16_DUAL"
}
```

**status code:** 200 No error

**Description:** Please refer to [Description- Configurable Parameters](#) section for detailed description on sensor configurable parameters.

## POST /api/v1/sensor/config

### Description

- Currently the lidar\_mode=1024x10, to change the lidar mode to "512X10", use the command: `http POST 169.254.198.184/api/v1/sensor/config "parameter"="value"`.
- Note: To identify all parameters that can be changed using this command please refer to [get\\_config\\_param](#).

**Example 1:** Valid sensor configuration to change lidar\_mode shown below:

```
POST http POST 169.254.198.184/api/v1/sensor/config "lidar_mode"="512x10"
```

```
POST /api/v1/sensor/config HTTP/1.1
Host: 169.254.198.184

HTTP/1.1 204 No Content
```

Sensor config after **POST** command:

```
HTTP/1.1 200 OK
content-length: 767
content-type: application/json
{
  "auto_start_flag": 1,
  "azimuth_window": [90000, 270000],
  "columns_per_packet": 16,
  "lidar_mode": "512x10",
  "multipurpose_io_mode": "OFF",
  "nmea_baud_rate": "BAUD_9600",
  "nmea_ignore_valid_char": 0,
  "nmea_in_polarity": "ACTIVE_HIGH",
  "nmea_leap_seconds": 0,
  "operating_mode": "NORMAL",
  "phase_lock_enable": false,
  "phase_lock_offset": 0,
  "signal_multiplier": 1,
  "sync_pulse_in_polarity": "ACTIVE_HIGH",
  "sync_pulse_out_angle": 360,
  "sync_pulse_out_frequency": 1,
  "sync_pulse_out_polarity": "ACTIVE_HIGH",
  "sync_pulse_out_pulse_width": 10,
  "timestamp_mode": "TIME_FROM_INTERNAL_OSC",
  "udp_dest": "169.254.198.184",
  "udp_port_imu": 7503,
  "udp_port_lidar": 7502,
  "udp_profile_imu": "LEGACY",
  "udp_profile_lidar": "RNG19_RFL8_SIG16_NIR16_DUAL"
}
```

**status code:** 204 server has successfully fulfilled the request.

**Example 2:** Invalid sensor configuration to change lidar\_mode shown below:

- Currently the lidar\_mode=1024x10, to change the lidar mode to "512X10", use the command: `http POST 169.254.216.107/api/v1/sensor/config "parameter"="value"`.
- Note: An incorrect value for a valid key i.e., "lidar\_mode" is given to show the error message that would be prompted. In this case Lidar\_mode is set to **511x10**.

`POST http POST 169.254.216.107/api/v1/sensor/config "lidar_mode"="511x10"`

```
POST /api/v1/sensor/config HTTP/1.1
Host: 169.254.216.107
```

Sensor config after `POST` command:

```
HTTP/1.1 400 Bad Request
content-length: 102
content-type: application/json

{
  "error": {
    "title": "While processing key 'lidar_mode' encountered error: '511x10' is not supported"
  }
}
```

**status code:** 400 Server cannot or will not process the request due to something that is perceived to be a client error.

## GET /api/v1/sensor/metadata

`GET 169.254.198.184/api/v1/sensor/metadata`  
Get the sensor metadata information

```
GET /api/v1/sensor/metadata HTTP/1.1
Host: 169.254.198.184

HTTP/1.1 200 OK
content-length: 4163
content-type: application/json

{
  "beam_intrinsic": {
    "beam_altitude_angles": [
      20.95, 20.67, 20.36, 20.03,
      19.73, 19.41, 19.11, 18.76,
      18.47, 18.14, 17.82, 17.5,
      17.19, 16.86, 16.53, 16.2,
      15.89, 15.56, 15.23, 14.9,
      14.57, 14.23, 13.9, 13.57,
      13.25, 12.91, 12.57, 12.22,
      11.9, 11.55, 11.2, 10.87,
      10.54, 10.18, 9.84, 9.51,
      9.15, 8.81, 8.47, 8.11,
      7.78, 7.43, 7.08, 6.74,
```

(continues on next page)

```
6.39, 6.04, 5.7, 5.34,  
4.98, 4.64, 4.29, 3.93,  
3.58, 3.24, 2.88, 2.53,  
2.17, 1.82, 1.47, 1.12,  
0.78, 0.41, 0.07, -0.28,  
-0.64, -0.99, -1.35, -1.7,  
-2.07, -2.4, -2.75, -3.11,  
-3.46, -3.81, -4.15, -4.5,  
-4.86, -5.22, -5.57, -5.9,  
-6.27, -6.61, -6.97, -7.3,  
-7.67, -8.01, -8.35, -8.69,  
-9.05, -9.38, -9.71, -10.07,  
-10.42, -10.76, -11.09, -11.43,  
-11.78, -12.12, -12.46, -12.78,  
-13.15, -13.46, -13.8, -14.12,  
-14.48, -14.79, -15.11, -15.46,  
-15.79, -16.12, -16.45, -16.76,  
-17.11, -17.44, -17.74, -18.06,  
-18.39, -18.72, -19.02, -19.32,  
-19.67, -19.99, -20.27, -20.57,  
-20.92, -21.22, -21.54, -21.82
```

```
],  
"beam_azimuth_angles": [  
4.21, 1.41, -1.4, -4.22, 4.22,  
1.41, -1.4, -4.23, 4.21,  
1.4, -1.42, -4.2, 4.22,  
1.41, -1.4, -4.23, 4.21,  
1.41, -1.41, -4.21, 4.22,  
1.4, -1.41, -4.2, 4.22,  
1.42, -1.4, -4.2, 4.22,  
1.41, -1.42, -4.21, 4.22,  
1.41, -1.4, -4.21, 4.2,  
1.4, -1.4, -4.22, 4.21,  
1.41, -1.41, -4.21, 4.22,  
1.41, -1.4, -4.21, 4.21,  
1.41, -1.4, -4.21, 4.2,  
1.41, -1.4, -4.21, 4.2,  
1.4, -1.41, -4.21, 4.22,  
1.4, -1.4, -4.21, 4.22,  
1.42, -1.4, -4.2, 4.2,  
1.42, -1.4, -4.22, 4.22,  
1.41, -1.4, -4.2, 4.23,  
1.41, -1.4, -4.2, 4.21,  
1.41, -1.4, -4.21, 4.21,  
1.41, -1.4, -4.21, 4.22,  
1.41, -1.39, -4.21, 4.23,  
1.41, -1.39, -4.22, 4.23,  
1.4, -1.4, -4.2, 4.21,  
1.41, -1.41, -4.2, 4.22,  
1.42, -1.39, -4.22, 4.24,  
1.41, -1.41, -4.22, 4.23,  
1.41, -1.39, -4.21, 4.23,  
1.41, -1.39, -4.2, 4.23,  
1.4, -1.39, -4.2, 4.22,
```

```
    1.42, -1.39, -4.2
  ],
  "lidar_origin_to_beam_origin_mm": 15.806
},
"calibration_status": {
  "reflectivity": {
    "timestamp": "2022-01-08T10:00:38",
    "valid": true
  }
},
"config_params": {
  "azimuth_window": [
    0,
    360000
  ],
  "columns_per_packet": 16,
  "lidar_mode": "1024x10",
  "multipurpose_io_mode": "OFF",
  "nmea_baud_rate": "BAUD_9600",
  "nmea_ignore_valid_char": 0,
  "nmea_in_polarity": "ACTIVE_HIGH",
  "nmea_leap_seconds": 0,
  "operating_mode": "NORMAL",
  "phase_lock_enable": false,
  "phase_lock_offset": 0,
  "signal_multiplier": 1,
  "sync_pulse_in_polarity": "ACTIVE_HIGH",
  "sync_pulse_out_angle": 360,
  "sync_pulse_out_frequency": 1,
  "sync_pulse_out_polarity": "ACTIVE_HIGH",
  "sync_pulse_out_pulse_width": 10,
  "timestamp_mode": "TIME_FROM_INTERNAL_OSC",
  "udp_dest": "169.254.162.214",
  "udp_port_imu": 7503,
  "udp_port_lidar": 7502,
  "udp_profile_imu": "LEGACY",
  "udp_profile_lidar": "LEGACY"
},
"imu_intrinsic": {
  "imu_to_sensor_transform": [
    1,
    0,
    0,
    6.253,
    0,
    1,
    0,
    -11.775,
    0,
    0,
    1,
    7.645,
    0,
    0,
  ]
}
```

```
    0,
    1
  ]
},
"lidar_data_format": {
  "column_window": [
    0,
    1023
  ],
  "columns_per_frame": 1024,
  "columns_per_packet": 16,
  "pixel_shift_by_row": [12, 4, -4, -12, 12,
    4, -4, -12, 12, 4, -4, -12, 12, 4, -4,
    -12, 12, 4, -4, -12, 12, 4, -4, -12, 12,
    4, -4, -12, 12, 4, -4, -12, 12, 4, -4,
    -12, 12, 4, -4, -12, 12, 4, -4, -12, 12,
    4, -4, -12, 12, 4, -4, -12, 12, 4, -4,
    -12, 12, 4, -4, -12, 12, 4, -4, -12, 12,
    4, -4, -12, 12, 4, -4, -12, 12, 4, -4,
    -12, 12, 4, -4, -12, 12, 4, -4, -12, 12,
    4, -4, -12, 12, 4, -4, -12, 12, 4, -4,
    -12, 12, 4, -4, -12, 12, 4, -4, -12, 12,
    4, -4, -12
  ],
  "pixels_per_column": 128,
  "udp_profile_imu": "LEGACY",
  "udp_profile_lidar": "LEGACY"
},
"lidar_intrinsics": {
  "lidar_to_sensor_transform": [
    -1, 0, 0, 0, 0, -1, 0, 0, 0, 0, 1, 36.18, 0, 0, 0, 1
  ]
},
"sensor_info": {
  "build_date": "2022-07-29T23:56:15Z",
  "build_rev": "v2.4.0-omega.2",
  "image_rev": "ousteros-image-prod-aries-v2.4.0-omega.2+20220730010801.staging",
  "initialization_id": 9599937,
  "prod_line": "OS-1-128",
  "prod_pn": "840-103575-06",
  "prod_sn": "122201000998",
  "status": "RUNNING"
}
}
```

**status code:** 200 No error

## 3.2 System

---

### GET /api/v1/system/firmware

GET 192.0.2.123/api/v1/system/firmware  
Get the firmware version of the sensor

```
GET /api/v1/system/firmware HTTP/1.1
Host: 192.0.2.123
```

```
HTTP/1.1 200 OK
Host: 192.0.2.123
content-type: application/json; charset=UTF-8

{
  "fw": "ousteros-image-prod-aries-v2.0.0"
}
```

>**json string fw** Running firmware image name and version.

**statuscode:** 200 No error

### GET /api/v1/system/network

GET 192.0.2.123/api/v1/system/network  
Get the system network configuration.

```
GET /api/v1/system/network HTTP/1.1
Host: 192.0.2.123
```

```
HTTP/1.1 200 OK
content-type: application/json; charset=UTF-8

{
  "carrier": true,
  "duplex": "full",
  "ethaddr": "bc:0f:a7:00:01:2c",
  "hostname": "os-991900123456",
  "ipv4": {
    "addr": "192.0.2.123/24",
    "link_local": "169.254.245.183/16",
    "override": null
  },
  "ipv6": {
    "link_local": "fe80::be0f:a7ff:fe00:12c/64"
  },
  "speed": 1000
}
```

>**json boolean carrier:** State of Ethernet link, **true** when physical layer is connected.

>**json string duplex:** Duplex mode of Ethernet link, **half** or **full**.



**>json string ethaddr:** Ethernet hardware (MAC) address.

**>json string hostname:** Hostname of the sensor, also used when requesting *DHCP* address and registering mDNS hostname.

**>json object ipv4:** See *ipv4 object*

**>json string ipv6.link\_local:** Link-local IPv6 address.

**>json integer speed:** Ethernet physical layer speed in Mbps, should be 1000 Mbps.

**statuscode:** 200 No error

## GET /api/v1/system/network/ipv4

GET 192.0.2.123/api/v1/system/network/ipv4  
Get the IPv4 network configuration.

```
GET /api/v1/system/network/ipv4 HTTP/1.1
Host: 192.0.2.123
```

```
HTTP/1.1 200 OK
content-type: application/json; charset=UTF-8

{
  "addr": "192.0.2.123/23",
  "link_local": "169.254.245.183/16",
  "override": null
}
```

**>json string addr:** Current global or private IPv4 address.

**>json string link\_local:** Link-local IPv4 address.

**>json string override:** Static IP override value, this should match *addr*. This value will be *null* when unset and operating in *DHCP* or *link-local* modes.

**statuscode:** 200 No error

## GET /api/v1/system/network/ipv4/override

GET 192.0.2.123/api/v1/system/network/ipv4/override  
Get the current IPv4 static IP address override.

```
GET /api/v1/system/network/ipv4/override HTTP/1.1
Host: 192.0.2.123
```

```
HTTP/1.1 200 OK
content-type: application/json; charset=UTF-8

null
```

**>json string** Static IP override value, this should match `addr`. This value will be `null` when unset and operating in *DHCP* mode.

**statuscode:** 200 No error

### PUT /api/v1/system/network/ipv4/override

PUT 192.0.2.123/api/v1/system/network/ipv4/override

Override the default dynamic behavior and set a static IP address.

---

**Note:** The sensor will reset the network configuration after a short sub second delay (to allow for the HTTP response to be sent). After this delay the sensor will only be reachable on the newly set IPv4 address.

The sensor needs to be reachable either by *link-local* or dynamic *DHCP* configuration or by an existing static IP override from the host reconfiguring the sensor.

---

**Warning:** If an unreachable network address is set, the sensor will become unreachable. Tools such as *avahi-browse*, *dns-sd*, or *mDNS browser* can help with finding a sensor on a network.

Static IP override should only be used in special use cases. The *link-local* configuration is recommended where possible.

```
PUT /api/v1/system/network/ipv4/override HTTP/1.1
Content-Type: application/json
Host: 192.0.2.123

{"192.0.2.100/24"}
```

**<json string:** Static IP override value with subnet mask

**>json string:** Static IP override value that system will set after a short delay.

**statuscode:** 200 No error

### DELETE /api/v1/system/network/ipv4/override

DELETE 192.0.2.123/api/v1/system/network/ipv4/override

Delete the static IP override value and return to dynamic configuration.

---

**Note:** The sensor will reset the network configuration after a short sub second delay (to allow for the HTTP response to be sent). After this delay the sensor will only be reachable on the newly set IPv4 address.

The sensor may be unreachable for several seconds while a *link-local* lease is obtained from the network or client machine.

---

```
DELETE /api/v1/system/network/ipv4/override HTTP/1.1
Host: 192.0.2.123
```

**statuscode:** 204 No error, no content

## 3.3 Time

### GET /api/v1/time

GET 192.0.2.123/api/v1/time

Get the system time configuration for all timing components of the sensor.

```
GET /api/v1/time HTTP/1.1
Host: 192.0.2.123

HTTP/1.1 200 OK
content-type: application/json; charset=UTF-8

{
  "ptp": {
    "current_data_set": {
      "mean_path_delay": 0.0,
      "offset_from_master": 0.0,
      "steps_removed": 0
    },
    "parent_data_set": {
      "gm_clock_accuracy": 254,
      "gm_clock_class": 255,
      "gm_offset_scaled_log_variance": 65535,
      "grandmaster_identity": "bc0fa7.ffffe.003aa6",
      "grandmaster_priority1": 128,
      "grandmaster_priority2": 128,
      "observed_parent_clock_phase_change_rate": 2147483647,
      "observed_parent_offset_scaled_log_variance": 65535,
      "parent_port_identity": "bc0fa7.ffffe.003aa6-0",
      "parent_stats": 0
    },
    "port_data_set": {
      "announce_receipt_timeout": 3,
      "delay_mechanism": 1,
      "log_announce_interval": 1,
      "log_min_delay_req_interval": 0,
      "log_min_pdelay_req_interval": 0,
      "log_sync_interval": 0,
      "peer_mean_path_delay": 0,
      "port_identity": "bc0fa7.ffffe.003aa6-1",
      "port_state": "LISTENING",
      "version_number": 2
    },
    "profile": "default",
    "time_properties_data_set": {
      "current_utc_offset": 37,
```

(continues on next page)

```
"current_utc_offset_valid": 0,
"frequency_traceable": 0,
"leap59": 0,
"leap61": 0,
"ptp_timescale": 1,
"time_source": 160,
"time_traceable": 0
},
"time_status_np": {
  "cumulative_scaled_rate_offset": 0.0,
  "gm_identity": "bc0fa7.ffff.003aa6",
  "gm_present": false,
  "gm_time_base_indicator": 0,
  "ingress_time": 0,
  "last_gm_phase_change": "0x0000'0000000000000000.0000",
  "master_offset": 0,
  "scaled_last_gm_phase_change": 0
}
},
"sensor": {
  "multipurpose_io": {
    "mode": "OFF",
    "nmea": {
      "baud_rate": "BAUD_9600",
      "diagnostics": {
        "decoding": {
          "date_decoded_count": 0,
          "last_read_message": "",
          "not_valid_count": 0,
          "utc_decoded_count": 0
        },
        "io_checks": {
          "bit_count": 1,
          "bit_count_unfiltered": 0,
          "char_count": 0,
          "start_char_count": 0
        }
      },
      "ignore_valid_char": 0,
      "leap_seconds": 0,
      "locked": 0,
      "polarity": "ACTIVE_HIGH"
    },
    "sync_pulse_out": {
      "angle_deg": 360,
      "frequency_hz": 1,
      "polarity": "ACTIVE_HIGH",
      "pulse_width_ms": 10
    }
  },
  "sync_pulse_in": {
    "diagnostics": {
      "count": 1,
      "count_unfiltered": 0,

```

(continued from previous page)

```
        "last_period_nsec": 0
    },
    "locked": 0,
    "polarity": "ACTIVE_HIGH"
},
"timestamp": {
    "mode": "TIME_FROM_INTERNAL_OSC",
    "time": 311.36525506,
    "time_options": {
        "internal_osc": 311,
        "ptp_1588": 320,
        "sync_pulse_in": 1
    }
}
},
"system": {
    "monotonic": 320.78890018,
    "realtime": 320.788918614,
    "tracking": {
        "frequency": 3.943,
        "last_offset": 0.0,
        "leap_status": "not synchronised",
        "ref_time_utc": 0.0,
        "reference_id": "00000000",
        "remote_host": "",
        "residual_frequency": 0.0,
        "rms_offset": 0.0,
        "root_delay": 1.0,
        "root_dispersion": 1.0,
        "skew": 0.0,
        "stratum": 0,
        "system_time_offset": -1e-09,
        "update_interval": 0.0
    }
}
}
```

**>json string:** See sub objects for details.

**statuscode:** 200 No error

### GET /api/v1/time/sensor

GET 169.254.198.184/api/v1/time/sensor

Get the sensor time information

```
GET /api/v1/time/sensor HTTP/1.1
Host: 169.254.198.184

HTTP/1.1 200 OK
content-type: application/json
content-length: 773
```

(continues on next page)

```

{
  "multipurpose_io": {
    "mode": "OFF",
    "nmea": {
      "baud_rate": "BAUD_9600",
      "diagnostics": {
        "decoding": {
          "date_decoded_count": 0,
          "last_read_message": "",
          "not_valid_count": 0,
          "utc_decoded_count": 0
        },
        "io_checks": {
          "bit_count": 1,
          "bit_count_unfiltered": 0,
          "char_count": 0,
          "start_char_count": 0
        }
      },
      "ignore_valid_char": 0,
      "leap_seconds": 0,
      "locked": 0,
      "polarity": "ACTIVE_HIGH"
    },
    "sync_pulse_out": {
      "angle_deg": 360,
      "frequency_hz": 1,
      "polarity": "ACTIVE_HIGH",
      "pulse_width_ms": 10
    }
  },
  "sync_pulse_in": {
    "diagnostics": {
      "count": 1,
      "count_unfiltered": 0,
      "last_period_nsec": 0
    },
    "locked": 0,
    "polarity": "ACTIVE_HIGH"
  },
  "timestamp": {
    "mode": "TIME_FROM_INTERNAL_OSC",
    "time": 27784.88125111,
    "time_options": {
      "internal_osc": 27784,
      "ptp_1588": 27795,
      "sync_pulse_in": 1
    }
  }
}

```

**status code:** 200 No error

**Description:** Returns JSON-formatted sensor timing configuration and status of udp `timestamp`, `sync_pulse_in`, and `multipurpose_io`. For more information on these parameters refer to the `get_time_info` TCP command.

## GET /api/v1/time/system

GET 192.0.2.123/api/v1/time/system

Get the operating system time status. These values relate to the sensor operating system clocks, and not clocks related to hardware timestamp data from the lidar sensor.

```
GET /api/v1/time/system HTTP/1.1
Host: 192.0.2.123
```

```
HTTP/1.1 200 OK
content-type: application/json; charset=UTF-8
```

```
{
  "monotonic": 345083.599570944,
  "realtime": 1551814510.730453,
  "tracking": {
    "frequency": -6.185,
    "last_offset": -3.315e-06,
    "leap_status": "normal",
    "ref_time_utc": 1551814508.1982567,
    "reference_id": "70747000",
    "remote_host": "ptp",
    "residual_frequency": -0.019,
    "rms_offset": 4.133e-06,
    "root_delay": 1e-09,
    "root_dispersion": 0.000128737,
    "skew": 1.14,
    "stratum": 1,
    "system_time_offset": 4.976e-06,
    "update_interval": 2
  }
}
```

**>json float monotonic: Monotonic time of operating system. This timestamp** never counts backwards and is the time since boot in seconds.

**>json float realtime: Time in seconds since the Unix epoch, should match** wall time if synchronized with external time source.

**>json object tracking:** Operating system time synchronization tracking status. See [chronyc tracking documentation](#) for more information.

**statuscode:** 200 No error

System `tracking` fields of interest:

**rms\_offset:** Long-term average of the offset value.

**system\_time\_offset:** Time delta (in seconds) between the estimate of the operating system time and the current true time.

**last\_offset:** Estimated local offset on the last clock update.

**ref\_time\_utc:** UTC Time at which the last measurement from the reference source was processed.

**remote\_host:** This is either **ptp** if the system is synchronizing to a **PTP** time source or the address of a remote NTP server the system has selected if the sensor is connected to the Internet.

## GET /api/v1/time/ptp

GET 192.0.2.123/api/v1/time/ptp

Get the status of the *PTP* time synchronization daemon.

---

**Note:** See the [IEEE 1588-2008 standard](#) for more details on the standard management messages.

---

```
GET /api/v1/time/ptp HTTP/1.1
Host: 192.0.2.123
```

```
HTTP/1.1 200 OK
content-type: application/json; charset=UTF-8

{
  "current_data_set": {
    "mean_path_delay": 30110,
    "offset_from_master": 224159,
    "steps_removed": 1
  },
  "parent_data_set": {
    "gm_clock_accuracy": 33,
    "gm_clock_class": 6,
    "gm_offset_scaled_log_variance": 20061,
    "grandmaster_identity": "001747.ffffe.700038",
    "grandmaster_priority1": 128,
    "grandmaster_priority2": 128,
    "observed_parent_clock_phase_change_rate": 2147483647,
    "observed_parent_offset_scaled_log_variance": 65535,
    "parent_port_identity": "001747.ffffe.700038-1",
    "parent_stats": 0
  },
  "port_data_set": {
    "announce_receipt_timeout": 3,
    "delay_mechanism": 1,
    "log_announce_interval": 1,
    "log_min_delay_req_interval": 0,
    "log_min_pdelay_req_interval": 0,
    "log_sync_interval": 0,
    "peer_mean_path_delay": 0,
    "port_identity": "bc0fa7.ffffe.00012c-1",
    "port_state": "SLAVE",
    "version_number": 2
  },
  "time_properties_data_set": {
```

(continues on next page)



(continued from previous page)

```
"current_utc_offset": 37,
"current_utc_offset_valid": 1,
"frequency_traceable": 1,
"leap59": 0,
"leap61": 0,
"ptp_timescale": 1,
"time_source": 32,
"time_traceable": 1
},
"time_status_np": {
  "cumulative_scaled_rate_offset": 0,
  "gm_identity": "001747.ffff.700038",
  "gm_present": true,
  "gm_time_base_indicator": 0,
  "ingress_time": 1551814546772493800,
  "last_gm_phase_change": "0x0000'0000000000000000.0000",
  "master_offset": 224159,
  "scaled_last_gm_phase_change": 0
}
}
```

>**json object current\_data\_set** Result of the PMC **GET CURRENT\_DATA\_SET** command.

>**json object parent\_data\_set** Result of the PMC **GET PARENT\_DATA\_SET** command.

>**json object port\_data\_set** Result of the PMC **GET PORT\_DATA\_SET** command.

>**json object time\_properties\_data\_set** Result of the PMC **GET TIME\_PROPERTIES\_DATA\_SET** command.

>**json object time\_status\_np** Result of the PMC **GET TIME\_STATUS\_NP** command. This is a linuxptp non-portable command.

**statuscode:** 200 No error

Fields of interest:

**current\_data\_set.offset\_from\_master** Offset from master time source in nanoseconds as calculated during the last update from master.

**parent\_data\_set.grandmaster\_identity** This should match the local grandmaster clock. If this displays the sensor's clock identity (derived from Ethernet hardware address) then this indicates the sensor is not properly synchronized to a grandmaster.

**parent\_data\_set** Various information about the selected master clock.

**port\_data\_set.port\_state** This value will be **SLAVE** when a remote master clock is selected. See **parent\_data\_set** for selected master clock.

**port\_data\_set** Local sensor *PTP* configuration values. Grandmaster clock needs to match these for proper time synchronization.

**time\_properties\_data\_set** *PTP* properties as given by master clock.

**time\_status\_np.gm\_identity** Selected grandmaster clock identity.

**time\_status\_np.gm\_present** True when grandmaster has been detected. This may stay true even if grandmaster goes off-line. Use `port_data_set.port_state` to determine up-to-date synchronization status. When this is false then the local clock is selected.

**time\_status\_np.ingress\_time** Indicates when the last *PTP* message was received. Units are in nanoseconds.

**time\_status\_np** Linux *PTP* specific diagnostic values. The [Red Hat manual](#) provides some more information on these fields

## GET /api/v1/time/ptp/profile

GET 192.0.2.123/api/v1/time/ptp/profile  
Get the active PTP profile of the Ouster sensor

```
GET /api/v1/time/ptp/profile HTTP/1.1
Content-Type: application/json
Host: 192.0.2.123
```

```
HTTP/1.1 200 OK
content-length: 9
content-type: application/json; charset=UTF-8

"gptp"
```

**>json string:** Active PTP profile.

**statuscode:** 200 No error

## PUT /api/v1/time/ptp/profile

PUT 192.0.2.123/api/v1/time/ptp/profile  
Change the PTP profile of the Ouster sensor

```
PUT /api/v1/time/ptp/profile HTTP/1.1
Content-Type: application/json
Host: 192.0.2.123

"gptp"
```

```
HTTP/1.1 200 OK
content-length: 9
content-type: application/json; charset=UTF-8

"gptp"
```

**<json string:** PTP profile to be activated, valid options are "default", "gptp", and "automotive-slave"

**>json string:** Active PTP profile.

**statuscode:** 200 No error

## 3.4 Alerts, Diagnostics and Telemetry

---

### GET /api/v1/sensor/alerts

**Example 1:** Using HTTPie command GET /api/v1/sensor/alerts

```
GET 169.254.198.184/api/v1/sensor/alerts
```

Get the sensor lidar intrinsic

```
GET /api/v1/sensor/alerts HTTP/1.1
Host: 169.254.198.184

HTTP/1.1 200 OK
content-length: 3941
content-type: application/json

{
  "active": [
    {
      "active": true,
      "category": "UDP_TRANSMISSION",
      "cursor": 8,
      "id": "0x01000018",
      "level": "WARNING",
      "msg": "Client machine announced it is not reachable on the provided not reachable on
            IMU data port; check that udp_dest and udp_port_imu configured on the sensor
            matches client IP and port.",
      "msg_verbose": "Failed to send imu UDP data to destination host 169.254.28.205:7503",
      "realtime": "1309574110356"
    }
  ],
  "log": [
    {
      "active": true,
      "category": "UDP_TRANSMISSION",
      "cursor": 0,
      "id": "0x01000018",
      "level": "WARNING",
      "msg": "Client machine announced it is not reachable on the provided not reachable on
            IMU data port; check that udp_dest and udp_port_imu configured on the sensor
            matches client IP and port.",
      "msg_verbose": "Failed to send imu UDP data to destination host 169.254.28.205:7503",
      "realtime": "87914851559"
    },
    {
      "active": true,
      "category": "UDP_TRANSMISSION",
      "cursor": 1,
      "id": "0x01000015",
      "level": "WARNING",
      "msg": "Client machine announced it is not reachable on the provided lidar data port;
            check that udp_dest and udp_port_lidar configured on the sensor matches client
            IP and port.",
      "msg_verbose": "Failed to send lidar UDP data to destination host 169.254.28.205:7502",
      "realtime": "88906828916"
    }
  ]
}
```

(continues on next page)

```
},
{
  "active": false,
  "category": "UDP_TRANSMISSION",
  "cursor": 2,
  "id": "0x01000015",
  "level": "WARNING",
  "msg": "Client machine announced it is not reachable on the provided lidar data port;
        check that udp_dest and udp_port_lidar configured on the sensor matches client
        IP and port.",
  "msg_verbose": "Cleared by reinitialization.",
  "realtime": "171640501848"
},
{
  "active": false,
  "category": "UDP_TRANSMISSION",
  "cursor": 3,
  "id": "0x01000018",
  "level": "WARNING",
  "msg": "Client machine announced it is not reachable on the provided not reachable on
        IMU data port; check that udp_dest and udp_port_imu configured on the sensor matches
        client IP and port.",
  "msg_verbose": "Cleared by reinitialization.",
  "realtime": "171640962692"
},
{
  "active": true,
  "category": "UDP_TRANSMISSION",
  "cursor": 4,
  "id": "0x01000018",
  "level": "WARNING",
  "msg": "Client machine announced it is not reachable on the provided not reachable on
        IMU data port; check that udp_dest and udp_port_imu configured on the sensor
        matches client IP and port.",
  "msg_verbose": "Failed to send imu UDP data to destination host 169.254.28.205:7503",
  "realtime": "188178225997"
},
{
  "active": true,
  "category": "UDP_TRANSMISSION",
  "cursor": 5,
  "id": "0x01000015",
  "level": "WARNING",
  "msg": "Client machine announced it is not reachable on the provided lidar data port;
        check that udp_dest and udp_port_lidar configured on the sensor matches client
        IP and port.",
  "msg_verbose": "Failed to send lidar UDP data to destination host 169.254.28.205:7502",
  "realtime": "189169539737"
},
{
  "active": false,
  "category": "UDP_TRANSMISSION",
  "cursor": 6,
  "id": "0x01000015",
```

(continued from previous page)

```
"level": "WARNING",
"msg": "Client machine announced it is not reachable on the provided lidar data port;
      check that udp_dest and udp_port_lidar configured on the sensor matches client
      IP and port.",
"msg_verbose": "Cleared by reinitialization.",
"realtime": "1293004145514"
},
{
  "active": false,
  "category": "UDP_TRANSMISSION",
  "cursor": 7,
  "id": "0x01000018",
  "level": "WARNING",
  "msg": "Client machine announced it is not reachable on the provided not reachable on
        IMU data port; check that udp_dest and udp_port_imu configured on the sensor
        matches client IP and port.",
  "msg_verbose": "Cleared by reinitialization.",
  "realtime": "1293004551057"
},
{
  "active": true,
  "category": "UDP_TRANSMISSION",
  "cursor": 8,
  "id": "0x01000018",
  "level": "WARNING",
  "msg": "Client machine announced it is not reachable on the provided not reachable on
        IMU data port; check that udp_dest and udp_port_imu configured on the sensor
        matches client IP and port.",
  "msg_verbose": "Failed to send imu UDP data to destination host 169.254.28.205:7503",
  "realtime": "1309574110356"
}
],
"next_cursor": 9
}
```

**status code:** 200 No error

**Description:** Returns JSON-formatted sensor diagnostic information.

Two lists will be returned, an **active** list and a **log** list. The **active** list will contain alert-trigger events for alerts that are currently active. An alert-trigger event will by-definition always have its active attribute set to true. There is no limit on the number of alert-trigger events that are displayed in the **active** event list. All currently active alert-trigger events will be displayed in the **active** event list.

The **log** list will contain all current and past alert-trigger and alert-clear events. An alert-clear event will by-definition have the exact same attributes and attribute values as its corresponding trigger event, with the exception of the realtime and cursor attributes which should have higher values, since an alert-clear event will always be received after an alert-trigger event. The **log** list has a length limit of 32 events with the oldest events automatically removed from the log list once a new event needs to be added to the **log** list and the **log** list length limit is reached, essentially acting as a FIFO (First In First Out) queue.

In addition to the **active** and **log** lists, `get_alerts` also returns a **next\_cursor** field. Every alert event has

a cursor attribute, which increments for every alert event logged. This can be used to track the alert activity that has been viewed and reduce message bandwidth. To do this, users are recommended to save the **next\_cursor** field when calling `get_alerts` and then use that value as the **START\_CURSOR** argument on the next `get_alerts` call to fetch only new **log** entries.

A valid value for **MODE** is either `summary` or `default`.

---

**Note: Valid uses of get\_alerts:**

- Example: Calling `get_alerts` with `START_CURSOR=1`  
`get_alerts 1`
- Example: Calling `get_alerts` with `START_CURSOR=2` and `MODE=summary`  
`get_alerts 2 summary`

**Invalid uses of get\_alerts:**

- Example: Calling `get_alerts` with `START_CURSOR=summary` and `MODE=2`  
`get_alerts summary 2`

---

**Note:** The order in which the `START_CURSOR` and `MODE` arguments are specified for the `get_alerts` TCP command must be followed when providing both arguments, with `START_CURSOR` preceding the `MODE`. The arguments can be specified in any order when calling the equivalent `/api/v1/sensor/alerts` HTTP endpoint.

---

**Example2:** Using cURL command `GET /api/v1/sensor/alerts`

```
$ curl --request GET --url http://169.254.216.107/api/v1/sensor/alerts

{
  "next_cursor": 10,
  "log":
  [
    {
      "active": true,
      "msg_verbose": "Failed to send imu UDP data to destination
                    host 169.254.117.239:7503",
      "cursor": 0,
      "id": "0x01000018",
      "category": "UDP_TRANSMISSION",
      "level": "WARNING",
      "msg": "Client machine announced it is not reachable on the
            provided IMU data port; check that udp_dest and
            udp_port_imu configured on the sensor matches client IP and port.
            This Alert may occur on sensor startup if the client is not
            listening at that time.",
      "realtime": "36290112071"
    },
    {
```

(continues on next page)

```
"active": true,
"msg_verbose": "Failed to send lidar UDP data to destination
               host 169.254.117.239:7502",
"cursor": 1,
"id": "0x01000015",
"category": "UDP_TRANSMISSION",
"level": "WARNING",
"msg": "Client machine announced it is not reachable on the provided
        lidar data port; check that udp_dest and udp_port_lidar configured
        on the sensor matches client IP and port.
        This Alert may occur on sensor startup if the client is not
        listening at that time.",
"realtime": "37280855516"
},
{
  "active": false,
  "msg_verbose": "Cleared by reinitialization.",
  "cursor": 2,
  "id": "0x01000015",
  "category": "UDP_TRANSMISSION",
  "level": "WARNING",
  "msg": "Client machine announced it is not reachable on the provided
          lidar data port; check that udp_dest and udp_port_lidar configured
          on the sensor matches client IP and port.
          This Alert may occur on sensor startup if the client is not listening
          at that time.",
  "realtime": "1324837582611"
},
{
  "active": false,
  "msg_verbose": "Cleared by reinitialization.",
  "cursor": 3,
  "id": "0x01000018",
  "category": "UDP_TRANSMISSION",
  "level": "WARNING",
  "msg": "Client machine announced it is not reachable on the provided
          IMU data port; check that udp_dest and udp_port_imu configured
          on the sensor matches client IP and port.
          This Alert may occur on sensor startup if the client is not
          listening at that time.",
  "realtime": "1324837951308"
},
{
  "active": true,
  "msg_verbose": "Failed to send imu UDP data to destination
               host 169.254.117.239:7503",
"cursor": 4,
"id": "0x01000018",
"category": "UDP_TRANSMISSION",
"level": "WARNING",
"msg": "Client machine announced it is not reachable on the provided IMU data port;
        check that udp_dest and udp_port_imu configured on the sensor matches client
        IP and port. This Alert may occur on sensor startup if the client is not
        listening at that time.",
```

```

    "realtime": "1337796738411"
  },
  {
    "active": true,
    "msg_verbose": "Failed to send lidar UDP data to destination
                  host 169.254.117.239:7502",
    "cursor": 5,
    "id": "0x01000015",
    "category": "UDP_TRANSMISSION",
    "level": "WARNING",
    "msg": "Client machine announced it is not reachable on the provided lidar data port;
          check that udp_dest and udp_port_lidar configured on the sensor matches
          client IP and port. This Alert may occur on sensor startup if the client
          is not listening at that time.",
    "realtime": "1342789095053"
  },
  {
    "active": false,
    "msg_verbose": "Cleared by reinitialization.",
    "cursor": 6,
    "id": "0x01000015",
    "category": "UDP_TRANSMISSION",
    "level": "WARNING",
    "msg": "Client machine announced it is not reachable on the provided lidar data port;
          check that udp_dest and udp_port_lidar configured on the sensor matches client
          IP and port. This Alert may occur on sensor startup if the client is not
          listening at that time.",
    "realtime": "2376756278143"
  },
  {
    "active": false,
    "msg_verbose": "Cleared by reinitialization.",
    "cursor": 7,
    "id": "0x01000018",
    "category": "UDP_TRANSMISSION",
    "level": "WARNING",
    "msg": "Client machine announced it is not reachable on the provided IMU data port;
          check that udp_dest and udp_port_imu configured on the sensor matches
          client IP and port. This Alert may occur on sensor startup if the client
          is not listening at that time.",
    "realtime": "2376756665275"
  },
  {
    "active": true,
    "msg_verbose": "Failed to send imu UDP data to destination
                  host 169.254.117.239:7503",
    "cursor": 8,
    "id": "0x01000018",
    "category": "UDP_TRANSMISSION",
    "level": "WARNING",
    "msg": "Client machine announced it is not reachable on the provided IMU data port;
          check that udp_dest and udp_port_imu configured on the sensor matches
          client IP and port. This Alert may occur on sensor startup if the client
          is not listening at that time.",

```



```
    "realtime": "2389805137860"
  },
  {
    "active": true,
    "msg_verbose": "Failed to send lidar UDP data to destination
                  host 169.254.117.239:7502",
    "cursor": 9,
    "id": "0x01000015",
    "category": "UDP_TRANSMISSION",
    "level": "WARNING",
    "msg": "Client machine announced it is not reachable on the provided lidar
          data port; check that udp_dest and udp_port_lidar configured on the sensor
          matches client IP and port. This Alert may occur on sensor startup if the
          client is not listening at that time.",
    "realtime": "2390795752261"
  }
],
  "active":
  [
    {
      "active": true,
      "msg_verbose": "Failed to send lidar UDP data to destination
                    host 169.254.117.239:7502",
      "cursor": 9,
      "id": "0x01000015",
      "category": "UDP_TRANSMISSION",
      "level": "WARNING",
      "msg": "Client machine announced it is not reachable on the provided lidar data port;
            check that udp_dest and udp_port_lidar configured on the sensor matches
            client IP and port. This Alert may occur on sensor startup if the client is
            not listening at that time.",
      "realtime": "2390795752261"
    },
    {
      "active": true,
      "msg_verbose": "Failed to send imu UDP data to destination
                    host 169.254.117.239:7503",
      "cursor": 8,
      "id": "0x01000018",
      "category": "UDP_TRANSMISSION",
      "level": "WARNING",
      "msg": "Client machine announced it is not reachable on the provided IMU data port;
            check that udp_dest and udp_port_imu configured on the sensor matches client
            IP and port. This Alert may occur on sensor startup if the client is not
            listening at that time.",
      "realtime": "2389805137860"
    }
  ]
}
```

## GET /api/v1/diagnostics/dump

GET 192.0.2.123/api/v1/diagnostics/dump

Get the diagnostics files of the sensor

```
GET /api/v1/diagnostics/dump HTTP/1.1
Host: 192.0.2.123
```

```
HTTP/1.1 200 OK
content-disposition: attachment; filename="192.0.2.123_diagnostics-dump_29811b9e-2afc-11eb-ae01-
bc0fa700190c.bin"
content-type: application/octet-stream

{binary data}
```

**statuscode:** 200 No error

## GET /api/v1/sensor/telemetry

GET 169.254.198.184/api/v1/sensor/telemetry

Get the sensor telemetry information

```
GET /api/v1/sensor/telemetry HTTP/1.1
Host: 169.254.198.184
```

```
HTTP/1.1 200 OK
content-length: 150
content-type: application/json

{
  "input_current_ma": 758,
  "input_voltage_mv": 23606,
  "internal_temperature_deg_c": 45,
  "phase_lock_status": "DISABLED",
  "timestamp_ns": 2962666299310
}
```

**status code:** 200 No error

**Description:** Returns JSON-formatted response that provides sensor system state information. This includes the FPGA **Timestamp** in **ns** (Nanoseconds) at which the information was collected from the FPGA, **Lidar Input Voltage** in **mv** (Millivolt), **Lidar Input Current** in **ma** (Milliamp), **Internal Temperature** of the sensor in **°C** (Degree Celsius) and **Phase Lock status** namely **LOCKED**, **LOST**, **DISABLED**..

---

### Note:

- **Internal temperature** can only be measured with Rev 06 and above sensors.
  - **Phase lock** output will not indicate loss of lock if the PTP source is lost.
-

# 4 API Changelog

---

## Version v2.4.0

Date 2022-08-31

### Description

#### "Added"

- New HTTP Commands to help configure and read sensor config parameters (Refer to [Sensor Metadata](#) for more information).
  - GET /api/v1/sensor/config
  - POST /api/v1/sensor/config
- New optional argument for both TCP/HTTP commands <MODE> = "summary" (Refer to [get\\_alerts](#) for more information).

#### "Removed"

- `auto_start_flag` has been deprecated.
- `udp_ip` has been deprecated.
- `base_pn`, `base_sn` and `proto_rev` have been deprecated.

#### "Fixed"

- Pixel shift by row has been updated. Please refer to [Sensor Status and Calibration](#).
- Bug in keep-alive behavior for HTTP 1.1.

## Version v2.3.0

Date 2022-04-15

### Description

#### "Added"

- Add additional options for config parameter `udp_profile_lidar`, refer to [Description- Configurable Parameters](#).
- Add new TCP command `get_telemetry`, refer to [Sensor Status and Calibration](#).
- Add the following GET HTTP Commands (Refer to [Sensor Metadata](#) for more information):
  - /api/v1/sensor/metadata/sensor\_info
  - /api/v1/sensor/metadata/lidar\_data\_format
  - /api/v1/sensor/metadata/beam\_intrinsics
  - /api/v1/sensor/metadata/imu\_intrinsics
  - /api/v1/sensor/metadata/lidar\_intrinsics

- /api/v1/sensor/metadata/calibration\_status
- /api/v1/sensor/metadata
- /api/v1/sensor/telemetry
- /api/v1/time/sensor
- /api/v1/sensor/alerts

#### **Version v2.2.0**

**Date** 2021-12-18

#### **Description**

##### **"Added"**

- Add config parameter udp\_profile\_lidar, udp\_profile\_imu and their documentation
- Add initialization\_id to get\_sensor\_info TCP command
- Add columns\_per\_packet to get\_config\_param TCP command

##### **"Changed"**

- The fields base\_pn, base\_sn, and proto\_rev in get\_sensor\_info TCP command have been cleared

#### **Version v2.1.3**

**Date** 2021-10-22

#### **Description**

##### **"Added"**

- Added PN support

#### **Version v2.1.2**

**Date** 2021-07-16

#### **Description**

##### **"Added"**

- Add support for minor hardware revisions

#### **Version v2.1.1**

**Date** 2021-06-21

#### **Description**

##### **"Added"**

- Add configuration parameter `signal_multiplier` and its documentation

### **"Removed"**

- Remove deprecated TCP command `set_data_dst_ip`
- Remove deprecated TCP command `get_data_dst_ip`
- Remove deprecated TCP command `set_udp_port_lidar`
- Remove deprecated TCP command `set_udp_port_imu`
- Remove deprecated TCP command `get_lidar_mode`
- Remove deprecated TCP command `set_lidar_mode`
- Remove deprecated TCP command `get_config_file_path`
- Remove deprecated TCP command `set_auto_start_flag`
- Remove deprecated TCP command `get_auto_start_flag`
- Remove deprecated TCP command `get_watchdog_status`

### **"Changed"**

- Fixed azimuth\_window parameter logic behavior. Notable changes:
  - [0,0] now outputs only a single column instead of all columns.
  - [1,2] results in sensor startup failure and an alert because there are no valid output columns.

## **Version v2.0.0**

**Date** 2020-11-20

### **Description**

#### **"Added"**

- Add TCP command `get_lidar_data_format`.
- Add in `azimuth_window` documentation.
- Add in commands `phase_lock_enable` and `phase_lock_offset` and their documentation.
- Add in verbose responses to parameter validation for TCP commands.
- Add in command `save_config_params` which supersedes the deprecated command `write_config_txt`, which will be deleted in future firmware.
- Add in command `get_config_param active` in favor of the deprecated command `get_config_txt`, which will be deleted in future firmware.
- Add in new STANDBY and WARMUP statuses.
- Add in parameter `operating_mode` in favor of the deprecated parameter `auto_start_flag`, which will be deleted in future firmware.
- Add in parameter `udp_dest` in favor of the deprecated parameter `udp_ip`, which will be deleted in future firmware. This is to be consistent with the `set_udp_dest_auto` parameter and to reflect that valid values can be hostnames in addition to ip addresses.
- Add in HTTP GET `api/v1/diagnostic/dump` endpoint.

### **"Removed"**

- Remove deprecated TCP command `set_udp_ip`.

### **"Changed"**

- TCP command `get_beam_intrinsics` now returns: 1) `lidar_origin_to_beam_origin_mm`, distance between the lidar origin and the beam origin in millimeters; and 2) beam altitude and azimuth angle arrays with padded zeros removed.
- `azimuth_window` parameter now in terms of millidegrees and implemented CCW.
- Deprecate `api/v1/system/time/` HTTP API and its sub-APIs and replace with `api/v1/time/`

## **Version v1.13.0**

### **Date**

### **Description**

#### **"Added"**

- Add TCP command `set_udp_dest_auto`
- TCP command `get_alerts`, includes more descriptive errors for troubleshooting

#### **"Changes"**

- Packet Status now called Azimuth Data Block Status and is calculated differently
- Packets with bad CRC are now dropped upstream and replaced with 0 padded packets to ensure all packets are sent for each frame.
- Return format of TCP command `get_time_info` updated

#### **"Removed"**

- Removed reference to `window_rejection_enable`

## **Version v1.12.0**

### **Date**

### **Description**

#### **"Changes"**

- Corrected IMU axis directions to match Sensor Coordinate Frame.
- Sensor Coordinate Frame section of sensor user manual for details on sensor coordinate frame. This change inverts IMU X, Y, and Z axis relative to v1.11.0.

## **Version v1.11.0**

### **Date** 2019-03-25

### **Description**

- Add section on HTTP API commands.

- TCP Port now hard-coded to 7501; port is no longer configurable.
- Update to SYNC\_PULSE\_IN and MULTIPURPOSE\_IO interface and configuration parameters (see details below).

#### Configuration parameters name changes:

- `pps_in_polarity` changed to `sync_pulse_in_polarity`
- `pps_out_mode` changed to `multipurpose_io_mode`
- `pps_out_polarity` changed to `sync_pulse_out_polarity`
- `pps_rate` changed to `sync_pulse_out_frequency`
- `pps_angle` changed to `sync_pulse_out_angle`
- `pps_pulse_width` changed to `sync_pulse_out_pulse_width`

#### New configuration parameters:

- `nmea_in_polarity`
- `nmea_ignore_valid_char`
- `nmea_baud_rate`
- `nmea_leap_seconds`

#### Configuration parameters option changes:

- `timestamp_mode` - `TIME_FROM_PPS` changed to `TIME_FROM_SYNC_PULSE_IN`
- `multipurpose_io_mode` (formerly `pps_out_mode`) - `OUTPUT_PPS_OFF` changed to `OFF` - `OUTPUT_FROM_PPS_IN_SYNCED` changed to `OUTPUT_FROM_SYNC_PULSE_IN` - Removed `OUTPUT_FROM_PPS_DEFINED_RATE` - Added `INPUT_NMEA_UART`

#### TCP command changes:

- **Added commands:**
  - `get_time_info`
- **Changed commands:**
  - `get_config_txt` (returned dictionary keys match parameter changes)
- **Removed commands:**
  - `set_pps_in_polarity`
  - `get_pps_out_mode`
  - `set_pps_out_mode`
  - `get_timestamp_mode`
  - `set_timestamp_mode`

Polarity changes: \* `sync_pulse_in_polarity` was corrected to match parameter naming. \*  
`sync_pulse_out_polarity` was corrected to match parameter naming.

**Version v1.10.0**

**Date** 2018-12-11

**Description**

**"Added"**

- Add `get_alerts`, `pps_rate` and `pps_angle` usage commands and expected output.

#### **"Removed"**

- Remove all references of `pulse_mode`.
- Remove TCP commands prior to v1.5.1.

#### **Version v1.9.0**

**Date** 2018-10-24

**Description**

#### **"Changes"**

- No TCP command change.

#### **Version v1.8.0**

**Date** 2018-10-11

**Description**

- `get_sensor_info` command gives `INITIALIZING`, `UPDATING`, `RUNNING`, `ERROR` and `UNCONFIGURED` status.

#### **Version v1.7.0**

**Date** 2018-09-05

**Description**

#### **"Changes"**

- No TCP command change.

#### **Version v1.6.0**

**Date** 2018-08-16

**Description**

#### **"Added"**

- Add `get_sensor_info` command gives `prod_line` info.