Ouster Sensor API Guide

Firmware v3.0.0 for all Ouster sensors

Ouster

Jan 27, 2023

TCP and HTTP APIs

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

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
udp_dest	String	"" (default)
udp_port_lidar	Integer	7502 (default)
udp_port_imu	Integer	7503 (default)
sync_pulse_in_polarity	Keyword	ACTIVE_HIGH (default) ACTIVE_LOW
sync_pulse_out_polarity	Keyword	ACTIVE_LOW (default) ACTIVE_HIGH
sync_pulse_out_frequency	Integer >= 1	1 (default)
sync_pulse_out_angle	Integer [0 ... 360]	360 (default)
sync_pulse_out_pulse_width	Integer >= 0	10 (default)
nmea_in_polarity	Keyword	ACTIVE_HIGH (default) ACTIVE_LOW
nmea_ignore_valid_char	integer [0 ... 1]	0 (default) 1
nmea_baud_rate	Keyword	BAUD_9600 (default) BAUD_115200
nmea_leap_seconds	Integer >= 0	0 (default)
azimuth_window	List	[0,360000] (default)
signal_multiplier	number [0.25, 0.5, 1 ... 3]	0.25 0.5 1 (default) 2 3
udp_profile_lidar	Keyword	LEGACY RNG19_RFL8_SIG16_NIR16 (default) RNG19_RFL8_SIG16_NIR16_DUAL RNG15_RFL8_NIR8
udp_profile_imu	Keyword	LEGACY (default)

continues on next page

Table 1 - continued from previous page

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

1.2 Description- Configurable Parameters

udp_dest

Description:

- Type: String
- Default: "169.254.26.118"
- 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 >= 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 180º 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 >= 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 >= 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 [0.25, 0.5, 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 for 0.25, 0.5 and 1: (Default) 360º, 2: 180º, 3: 120º.

  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: "RNG19_RFL8_SIG16_NIR16"
- **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`, `RNG19_RFL8_SIG16_NIR16` *[Default]*, `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 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://169.254.26.118/api/v1/sensor/metadata/lidar_intrinsics

 {
   "lidar_to_sensor_transform": [-1, 0, 0, 0, 0, -1, 0, 0, 0, 0, 1, 38.195, 0, 0, 0, 1]
 }
```

2.1 Sensor Metadata

GET /api/v1/sensor/metadata/sensor_info

`GET 169.254.26.118/api/v1/sensor/cmd/get_sensor_info`

Get the sensor information

```
GET /api/v1/sensor/metadata/sensor_info HTTP/1.1
Host: 169.254.26.118

 HTTP/1.1 200 OK
 Connection: keep-alive
 Content-Length: 285
 Content-Type: application/json

 {
     "build_date": "2023-1-15T15:56:07Z",
     "build_rev": "v3.0.0",
     "image_rev": "ousteros-image-prod-bootes-v3.0.0+0123456789",
     "initialization_id": 390072,
     "prod_line": "OS-1-128",
     "prod_pn": "860-105010-07",
     "prod_sn": "992244000006",
     "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 169.254.26.118/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: 169.254.26.118

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 724
Content-Type: application/json
Date: Thu, 28 Apr 2022 19:00:38 GMT
Server: nginx

{
   "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, 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": "RNG19_RFL8_SIG16_NIR16_DUAL"
}
```

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 single return profile (`RNG19_RFL8_SIG16_NIR16`).

`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 169.254.26.118/api/v1/sensor/metadata/beam_intrinsics`

Get the sensor beam intrinsics

```
GET /api/v1/sensor/metadata/beam_intrinsics HTTP/1.1
Host: 169.254.26.118
```

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 1895
Content-Type: application/json

{
    "beam_altitude_angles": [
        20.38, 20.12, 19.79, 19.45, 19.14, 18.85, 18.55, 18.2, 17.86, 17.58, 17.27, 16.93,
        16.58, 16.29, 15.98, 15.61, 15.27, 14.97, 14.66, 14.3, 13.96, 13.65, 13.33, 12.97,
        12.62, 12.31, 11.98, 11.63, 11.27, 10.96, 10.63, 10.26, 9.91, 9.59, 9.26, 8.89,
        8.54, 8.21, 7.87, 7.52, 7.15, 6.82, 6.47, 6.11, 5.76, 5.42, 5.08, 4.73, 4.36, 4.03,
        3.66, 3.31, 2.96, 2.62, 2.27, 1.91, 1.55, 1.22, 0.85, 0.51, 0.16, -0.2, -0.55, -0.91,
        -1.26, -1.62, -1.96, -2.3, -2.66, -3.02, -3.36, -3.72, -4.07, -4.42, -4.77, -5.11,
        -5.46, -5.82, -6.16, -6.49, -6.85, -7.21, -7.55, -7.88, -8.23, -8.59, -8.93, -9.25,
        -9.6, -9.96, -10.31, -10.63, -10.96, -11.32, -11.67, -11.97, -12.31, -12.68, -13,
        -13.32, -13.64, -14, -14.33, -14.63, -14.96, -15.31, -15.64, -15.94, -16.26,
        -16.62, -16.93, -17.22, -17.54, -17.9,  -18.22, -18.49, -18.8, -19.16, -19.47,
        -19.73, -20.04, -20.39, -20.7, -20.94, -21.25, -21.6, -21.9, -22.14
    ],
    "beam_azimuth_angles": [
        4.24, 1.41, -1.42, -4.23, 4.23, 1.41, -1.41, -4.23, 4.23, 1.41, -1.41, -4.21, 4.23,
        1.42, -1.4, -4.23, 4.24, 1.41, -1.4, -4.23, 4.24, 1.42, -1.4, -4.22, 4.23, 1.41,
        -1.41, -4.22, 4.23, 1.42, -1.4, -4.22, 4.24, 1.41, -1.4, -4.23, 4.23, 1.41, -1.41,
        -4.22, 4.23, 1.41, -1.41, -4.23, 4.23, 1.4, -1.42, -4.23, 4.23, 1.41, -1.42, -4.23,
        4.23, 1.4, -1.42, -4.24, 4.22, 1.41, -1.43, -4.24, 4.22, 1.4, -1.42, -4.24, 4.22,
        1.4, -1.42, -4.23, 4.22, 1.4, -1.4, -4.24, 4.22, 1.4, -1.42, -4.24, 4.22, 1.41,
        -1.41, -4.22, 4.22, 1.39, -1.42, -4.23, 4.22, 1.41, -1.41, -4.22, 4.23, 1.41,
        -1.41, -4.23, 4.23, 1.41, -1.41, -4.22, 4.23, 1.41, -1.41, -4.22, 4.22, 1.41,
        -1.41, -4.22, 4.23, 1.41, -1.4, -4.23, 4.22, 1.41, -1.41, -4.23, 4.22, 1.4, -1.41,
        -4.23, 4.22, 1.4, -1.41, -4.24, 4.22, 1.4, -1.42, -4.24, 4.22, 1.4, -1.42, -4.23
    ],
```

(continues on next page)

(continued from previous page)


```
    "beam_to_lidar_transform": [ 1, 0, 0, 15.805999755859375, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1 ],
    "lidar_origin_to_beam_origin_mm": 15.8059998
}
```


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 169.254.26.118/api/v1/sensor/metadata/imu_intrinsics
```

Get the sensor imu intrinsics


```
GET /api/v1/sensor/metadata/imu_intrinsics HTTP/1.1
Host: 169.254.26.118

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 91
Content-Type: application/json

  {
    "imu_to_sensor_transform": [1, 0, 0, 6.253, 0, 1, 0, -11.775, 0, 0, 1, 7.645, 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 169.254.26.118/api/v1/sensor/metadata/lidar_intrinsics
```

Get the sensor lidar intrinsics


```
GET /api/v1/sensor/metadata/lidar_intrinsics HTTP/1.1
Host: 169.254.26.118

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 86
Content-Type: application/json

  {
    "lidar_to_sensor_transform": [-1, 0, 0, 0, 0, -1, 0, 0, 0, 0, 1, 38.195, 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 169.254.26.118/api/v1/sensor/metadata/calibration_status`

Get the sensor calibration status

```
GET /api/v1/sensor/metadata/calibration_status HTTP/1.1
Host: 169.254.26.118

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 69
Content-Type: application/json

  {
    "reflectivity":
      {
        "timestamp": "2022-11-18T20:31:06",
        "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 169.254.26.118/api/v1/sensor/config`

Get sensor configuration parameter

```
GET /api/v1/sensor/config HTTP/1.1
Host: 169.254.26.118

HTTP/1.1 200 OK
Connection: keep-alive
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",
```

(continues on next page)

(continued from previous page)


```
        "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": "169.254.225.4",
        "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 169.254.26.118/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 169.254.26.118/api/v1/sensor/config "lidar_mode"="512x10"`

```
POST /api/v1/sensor/config HTTP/1.1
Host: 169.254.26.118

{ "lidar_mode": "512x10" }
```

Run *GET /api/v1/sensor/config* after `POST` command:


```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 712
Content-Type: application/json

    {
        "azimuth_window": [
            0,
```


(continues on next page)

(continued from previous page)


```
            360000
        ],
        "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": "169.254.225.4",
        "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 169.254.198.184/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 169.254.26.118/api/v1/sensor/config "lidar_mode"="511x10"
```

```
POST /api/v1/sensor/config HTTP/1.1

Host: 169.254.26.118

{"lidar_mode": "511x10"}
```

Run *GET /api/v1/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 169.254.26.118/api/v1/sensor/metadata
```

Get the sensor metadata information

```
GET /api/v1/sensor/metadata HTTP/1.1
Host: 169.254.26.118

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 4009
Content-Type: application/json

    {
        "beam_intrinsics": {
            "beam_altitude_angles": [
                20.38, 20.12, 19.79, 19.45, 19.14, 18.85, 18.55, 18.2, 17.86, 17.58, 17.27, 16.93,
                16.58, 16.29, 15.98, 15.61, 15.27, 14.97, 14.66, 14.3, 13.96, 13.65, 13.33, 12.97,
                12.62, 12.31, 11.98, 11.63, 11.27, 10.96, 10.63, 10.26, 9.91, 9.59, 9.26, 8.89,
                8.54, 8.21, 7.87, 7.52, 7.15, 6.82, 6.47, 6.11, 5.76, 5.42, 5.08, 4.73, 4.36, 4.03,
                3.66, 3.31, 2.96, 2.62, 2.27, 1.91, 1.55, 1.22, 0.85, 0.51, 0.16, -0.2, -0.55, -0.91,
                -1.26, -1.62, -1.96, -2.3, -2.66, -3.02, -3.36, -3.72, -4.07, -4.42, -4.77, -5.11,
                -5.46, -5.82, -6.16, -6.49, -6.85, -7.21, -7.55, -7.88, -8.23, -8.59, -8.93, -9.25,
```

(continues on next page)

(continued from previous page)


```
                -9.6, -9.96, -10.31, -10.63, -10.96, -11.32, -11.67, -11.97, -12.31, -12.68, -13,
                -13.32, -13.64, -14, -14.33, -14.63, -14.96, -15.31, -15.64, -15.94, -16.26,
                -16.62, -16.93, -17.22, -17.54, -17.9,  -18.22, -18.49, -18.8, -19.16, -19.47,
                -19.73, -20.04, -20.39, -20.7, -20.94, -21.25, -21.6, -21.9, -22.14
            ],
            "beam_azimuth_angles": [
                4.24, 1.41, -1.42, -4.23, 4.23, 1.41, -1.41, -4.23, 4.23, 1.41, -1.41, -4.21, 4.23,
                1.42, -1.4, -4.23, 4.24, 1.41, -1.4, -4.23, 4.24, 1.42, -1.4, -4.22, 4.23, 1.41,
                -1.41, -4.22, 4.23, 1.42, -1.4, -4.22, 4.24, 1.41, -1.4, -4.23, 4.23, 1.41, -1.41,
                -4.22, 4.23, 1.41, -1.41, -4.23, 4.23, 1.4, -1.42, -4.23, 4.23, 1.41, -1.42, -4.23,
                4.23, 1.4, -1.42, -4.24, 4.22, 1.41, -1.43, -4.24, 4.22, 1.4, -1.42, -4.24, 4.22,
                1.4, -1.42, -4.23, 4.22, 1.4, -1.4, -4.24, 4.22, 1.4, -1.42, -4.24, 4.22, 1.41,
                -1.41, -4.22, 4.22, 1.39, -1.42, -4.23, 4.22, 1.41, -1.41, -4.22, 4.23, 1.41,
                -1.41, -4.23, 4.23, 1.41, -1.41, -4.22, 4.23, 1.41, -1.41, -4.22, 4.22, 1.41,
                -1.41, -4.22, 4.23, 1.41, -1.4, -4.23, 4.22, 1.41, -1.41, -4.23, 4.22, 1.4, -1.41,
                -4.23, 4.22, 1.4, -1.41, -4.24, 4.22, 1.4, -1.42, -4.24, 4.22, 1.4, -1.42, -4.23
            ],
            "beam_to_lidar_transform": [ 1, 0, 0, 15.805999755859375, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1 ],
            "lidar_origin_to_beam_origin_mm": 15.8059998
        }
        "calibration_status": {
            "reflectivity": {
                "timestamp": "2022-11-18T20:31:06",
                "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": "169.254.225.4",
        "udp_port_imu": 7503,
        "udp_port_lidar": 7502,
        "udp_profile_imu": "LEGACY",
        "udp_profile_lidar": "RNG19_RFL8_SIG16_NIR16_DUAL"
    },
```


(continues on next page)

(continued from previous page)


```
        "imu_intrinsics": {
            "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, 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": "RNG19_RFL8_SIG16_NIR16_DUAL"
        },

        "lidar_intrinsics": {
            "lidar_to_sensor_transform": [
                -1, 0, 0, 0, 0, -1, 0, 0, 0, 0, 1, 38.195, 0, 0, 0, 1
            ]
        },
        "sensor_info": {
            "build_date": "2023-1-15T15:56:07Z",
            "build_rev": "v3.0.0",
            "image_rev": "ousteros-image-prod-bootes-v3.0.0+0123456789",
            "initialization_id": 390079,
            "prod_line": "OS-1-128",
            "prod_pn": "860-105010-07",
            "prod_sn": "992244000006",
            "status": "RUNNING"
        }
    }
```


status code: 200 No error

2.2 System

GET /api/v1/system/firmware

`GET 169.254.26.118/api/v1/system/firmware`

Get the firmware version of the sensor

```
GET /api/v1/system/firmware HTTP/1.1
Host: 169.254.26.118

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 104
Content-Type: application/json

    {
        "commit_pending": false,
        "fw": "ousteros-image-prod-bootes-v3.0.0+0123456789"
    }
```

>json string fw Running firmware image name and version.

statuscode: 200 No error

GET /api/v1/system/network

`GET 169.254.26.118/api/v1/system/network`

Get the system network configuration.

```
GET /api/v1/system/network HTTP/1.1
Host: 169.254.26.118

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 256
Content-Type: application/json

    {
        "carrier": true,
        "duplex": "full",
        "ethaddr": "bc:0f:a7:00:84:17",
        "hostname": "os-992244000006",
        "ipv4": {
            "link_local": "169.254.26.118/16",
            "override": null
        },
        "ipv6": {
            "link_local": "fe80::be0f:a7ff:fe00:8417/64"
        },
        "speed": 1000,
        "speed_override": null
    }
```

>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 169.254.26.118/api/v1/system/network/ipv4`

Get the IPv4 network configuration.

```
GET /api/v1/system/network/ipv4 HTTP/1.1
Host: 169.254.26.118
```

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 53
Content-Type: application/json


    {
        "link_local": "169.254.26.118/16",
        "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 169.254.26.118/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: 169.254.26.118
```

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 4
Content-Type: application/json

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

2.3 Time

GET /api/v1/time

`GET 169.254.26.118/api/v1/time`

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

```
GET /api/v1/time HTTP/1.1
Host: 169.254.26.118

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 2484
Content-Type: application/json

    {
        "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.fffe.008417",
                "grandmaster_priority1": 128,
                "grandmaster_priority2": 128,
                "observed_parent_clock_phase_change_rate": 2147483647,
                "observed_parent_offset_scaled_log_variance": 65535,
                "parent_port_identity": "bc0fa7.fffe.008417-0",
                "parent_stats": 0
            },
```

(continues on next page)

(continued from previous page)


```
            "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.fffe.008417-1",
                "port_state": "LISTENING",
                "version_number": 2
            },
            "profile": "default",
            "time_properties_data_set": {
                "current_utc_offset": 37,
                "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.fffe.008417",
                "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,
```


(continues on next page)

(continued from previous page)


```
                    "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": 297.397987312,
                "time_options": {
                    "internal_osc": 297,
                    "ptp_1588": 1651197874,
                    "sync_pulse_in": 1
                }
            }
        },
        "system": {
            "monotonic": 30131.822811617,
            "realtime": 1651197874.3271277,
            "tracking": {
                "frequency": -9.558,
                "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 169.254.26.118/api/v1/time/sensor

Get the sensor time information

```
GET /api/v1/time/sensor HTTP/1.1
Host: 169.254.26.118

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 775
Content-Type: application/json

    {
        "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"
```

(continues on next page)

(continued from previous page)


```
        },
        "timestamp": {
            "mode": "TIME_FROM_INTERNAL_OSC",
            "time": 376.510445056,
            "time_options": {
                "internal_osc": 376,
                "ptp_1588": 1651197953,
                "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 169.254.26.118/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: 169.254.26.118
```


```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 389
Content-Type: application/json

    {
        "monotonic": 30348.898799855,
        "realtime": 1651198091.4031146,
        "tracking": {
            "frequency": -9.558,
            "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": 3e-09,
            "update_interval": 0.0
        }
    }
```

>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 169.254.26.118/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: 169.254.26.118

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 1287
Content-Type: application/json

    {
        "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.fffe.008417",
            "grandmaster_priority1": 128,
```

(continues on next page)

(continued from previous page)


```
            "grandmaster_priority2": 128,
            "observed_parent_clock_phase_change_rate": 2147483647,
            "observed_parent_offset_scaled_log_variance": 65535,
            "parent_port_identity": "bc0fa7.fffe.008417-0",
            "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.fffe.008417-1",
            "port_state": "LISTENING",
            "version_number": 2
        },
        "profile": "default",
        "time_properties_data_set": {
            "current_utc_offset": 37,
            "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.fffe.008417",
            "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
        }
    }
```


>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 169.254.26.118/api/v1/time/ptp/profile`

Get the active PTP profile of the Ouster sensor

```
GET /api/v1/time/ptp/profile HTTP/1.1
 Host: 169.254.26.118

 HTTP/1.1 200 OK
 Connection: keep-alive
 Content-Length: 9
 Content-Type: application/json

 "default"
```

```
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 169.254.26.118/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: 169.254.26.118

   "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

2.4 Alerts, Diagnostics and Telemetry

GET /api/v1/sensor/alerts

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

`GET 169.254.26.118/api/v1/sensor/alerts`

Get the sensor lidar intrinsics

```
GET /api/v1/sensor/alerts HTTP/1.1
Host: 169.254.26.118

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 1983
Content-Type: application/json

    {
        "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. This Alert may occur on sensor startup if the
                        client is not listening at that time.",
                "msg_verbose": "Failed to send lidar UDP data to destination host 169.254.225.4:7502",
```

(continues on next page)

(continued from previous page)


```
            "realtime": "1651197616274601728"
        },
        {
            "active": true,
            "category": "UDP_TRANSMISSION",
            "cursor": 0,
            "id": "0x01000018",
            "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.",
            "msg_verbose": "Failed to send imu UDP data to destination host 169.254.225.4:7503",
            "realtime": "1651197615284695040"
        }
    ],
    "log": [
        {
            "active": true,
            "category": "UDP_TRANSMISSION",
            "cursor": 0,
            "id": "0x01000018",
            "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.",
            "msg_verbose": "Failed to send imu UDP data to destination host 169.254.225.4:7503",
            "realtime": "1651197615284695040"
        },
        {
            "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. This Alert may occur on sensor startup if the
                    client is not listening at that time.",
            "msg_verbose": "Failed to send lidar UDP data to destination host 169.254.225.4:7502",
            "realtime": "1651197616274601728"
        }
    ],
    "next_cursor": 2
}
```


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://169.254.26.118/api/v1/sensor/alerts

 {
     "next_cursor": 10,
     "log":
     [
      {
         "active": true,
         "msg_verbose": "Failed to send imu UDP data to destination
                         host 169.254.225.4:7503",
```

(continues on next page)

(continued from previous page)


```
        "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"
    },
    {
        "active": true,
        "msg_verbose": "Failed to send lidar UDP data to destination
                        host 169.254.225.4:7502",
        "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"
    },
```


(continues on next page)

(continued from previous page)


```
    {
        "active": true,
        "msg_verbose": "Failed to send imu UDP data to destination
                        host 169.254.225.4:7503",
        "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 169.254.225.4:7502",
        "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"
    },
```


(continues on next page)

(continued from previous page)


```
    {
        "active": true,
        "msg_verbose": "Failed to send imu UDP data to destination
                        host 169.254.225.4:7503",
        "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 169.254.225.4:7502",
        "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 169.254.225.4:7502",
        "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 169.254.225.4:7503",
        "cursor": 8,
        "id": "0x01000018",
        "category": "UDP_TRANSMISSION",
        "level": "WARNING",
        "msg": "Client machine announced it is not reachable on the provided IMU data port;
```


(continues on next page)

(continued from previous page)

```
                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 169.254.26.118/api/v1/diagnostics/dump

Get the diagnostics files of the sensor

```
GET /api/v1/diagnostics/dump HTTP/1.1
Host: 169.254.26.118
```

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: application/octet-stream
Transfer-Encoding: chunked
content-disposition: attachment; filename="os-992244000006_diagnostics-dump_b7d348c2
                                          -c763-11ec-accf-bc0fa7008417.bin"


+----------------------------------------+
| NOTE: binary data not shown in terminal |
+----------------------------------------+
```

statuscode: 200 No error

GET /api/v1/sensor/telemetry

GET 169.254.26.118/api/v1/sensor/telemetry

Get the sensor telemetry information

```
GET /api/v1/sensor/telemetry HTTP/1.1
Host: 169.254.26.118

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 150
Content-Type: application/json

    {
        "input_current_ma": 644,
        "input_voltage_mv": 23624,
        "internal_temperature_deg_c": 48,
        "phase_lock_status": "DISABLED",
        "timestamp_ns": 2093396806056
    }
```

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.

---

3 TCP API Guide (Important Notice)

Note: TCP API is subject to deprecation starting FW 3.0, please prepare to use HTTP API instead. TCP API will be completely deprecated and unavailable for use in the next 6 to 9 months.

3.1 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-992244000006.local 7501
get_sensor_info

 {"prod_line": "OS-1-128", "prod_pn": "860-105010-07", "prod_sn": "992244000006",
  "image_rev": "ousteros-image-prod-bootes-v3.0.0+0123456789",
  "build_rev": "v3.0.0", "build_date": "2023-1-15T15:56:07Z", "status": "RUNNING",
  "initialization_id": 1128503}
```

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 errors 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-9919xxxxxxxx/diag for support.

3.2 Sensor Config 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-992244000006.local 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-992244000006.local 7501
  get_config_param active
  {
    "udp_dest": "169.254.225.4",
    "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)

(continued from previous 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-992244000006.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
`reinitialize` or `reinit`	Restarts the sensor. Changes to lidar, multipurpose_io, and timestamp modes will only take effect after reinitialization. **Response on success:** `reinitialize` or `reinit`
`save_config_params`	Makes all current parameter settings persist after reboot. **Response on success:** `save_config_params`
`set_udp_dest_auto`	Set the destination of UDP traffic to the destination address that issued the command. **Response on success:** `set_udp_dest_auto`

Note: Refer to *Sensor Configuration* for detailed information on all configurable parameters using HTTP or TCP commands.

3.3 Sensor Status and Calibration

`get_sensor_info`

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


```
{
    "prod_line": "OS-1-128",
    "prod_pn": "860-105010-07",
    "prod_sn": "992244000006",
    "image_rev": "ousteros-image-prod-bootes-v3.0.0+0123456789",
    "build_rev": "v3.0.0",
    "build_date": "2023-1-15T15:56:07Z",
    "status": "RUNNING",
    "initialization_id": 1128503
}
```

`get_time_info`

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


```
{
   "timestamp":
     {
       "time": 3657.224442616,
       "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,
          "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":
```


(continues on next page)

(continued from previous page)


```
                {
                  "start_char_count": 0,
                  "char_count": 0,
                  "bit_count": 1,
                  "bit_count_unfiltered": 0
                }
            }
        }
    }
}
```


`get_beam_intrinsics`

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": [
    20.38, 20.12, 19.79, 19.45, 19.14, 18.85, 18.55, 18.2, 17.86, 17.58, 17.27, 16.93,
    16.58, 16.29, 15.98, 15.61, 15.27, 14.97, 14.66, 14.3, 13.96, 13.65, 13.33, 12.97,
    12.62, 12.31, 11.98, 11.63, 11.27, 10.96, 10.63, 10.26, 9.91, 9.59, 9.26, 8.89,
    8.54, 8.21, 7.87, 7.52, 7.15, 6.82, 6.47, 6.11, 5.76, 5.42, 5.08, 4.73, 4.36, 4.03,
    3.66, 3.31, 2.96, 2.62, 2.27, 1.91, 1.55, 1.22, 0.85, 0.51, 0.16, -0.2, -0.55, -0.91,
   -1.26, -1.62, -1.96, -2.3, -2.66, -3.02, -3.36, -3.72, -4.07, -4.42, -4.77, -5.11,
   -5.46, -5.82, -6.16, -6.49, -6.85, -7.21, -7.55, -7.88, -8.23, -8.59, -8.93, -9.25,
   -9.6, -9.96, -10.31, -10.63, -10.96, -11.32, -11.67, -11.97, -12.31, -12.68, -13,
   -13.32, -13.64, -14, -14.33, -14.63, -14.96, -15.31, -15.64, -15.94, -16.26,
   -16.62, -16.93, -17.22, -17.54, -17.9,  -18.22, -18.49, -18.8, -19.16, -19.47,
   -19.73, -20.04, -20.39, -20.7, -20.94, -21.25, -21.6, -21.9, -22.14],
  "beam_azimuth_angles": [
    4.24, 1.41, -1.42, -4.23, 4.23, 1.41, -1.41, -4.23, 4.23, 1.41, -1.41, -4.21, 4.23,
    1.42, -1.4, -4.23, 4.24, 1.41, -1.4, -4.23, 4.24, 1.42, -1.4, -4.22, 4.23, 1.41,
   -1.41, -4.22, 4.23, 1.42, -1.4, -4.22, 4.24, 1.41, -1.4, -4.23, 4.23, 1.41, -1.41,
   -4.22, 4.23, 1.41, -1.41, -4.23, 4.23, 1.4, -1.42, -4.23, 4.23, 1.41, -1.42, -4.23,
    4.23, 1.4, -1.42, -4.24, 4.22, 1.41, -1.43, -4.24, 4.22, 1.4, -1.42, -4.24, 4.22,
    1.4, -1.42, -4.23, 4.22, 1.4, -1.4, -4.24, 4.22, 1.4, -1.42, -4.24, 4.22, 1.41,
   -1.41, -4.22, 4.22, 1.39, -1.42, -4.23, 4.22, 1.41, -1.41, -4.22, 4.23, 1.41,
   -1.41, -4.23, 4.23, 1.41, -1.41, -4.22, 4.23, 1.41, -1.41, -4.22, 4.22, 1.41,
   -1.41, -4.22, 4.23, 1.41, -1.4, -4.23, 4.22, 1.41, -1.41, -4.23, 4.22, 1.4, -1.41,
   -4.23, 4.22, 1.4, -1.41, -4.24, 4.22, 1.4, -1.42, -4.24, 4.22, 1.4, -1.42, -4.23],
  "beam_to_lidar_transform": [ 1, 0, 0, 15.805999755859375, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1 ],
  "lidar_origin_to_beam_origin_mm": 15.8059998
}
```

`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, 38.195, 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 169.254.225.4:7503",
        "realtime": "39850161524"
    },
    {
        "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;
```


(continues on next page)

(continued from previous page)


```
            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 169.254.225.4:7502",
        "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 169.254.225.4:7502",
        "realtime": "414261086316"
    },
    {
        "active": true,
        "category": "UDP_TRANSMISSION",
```


(continues on next page)

(continued from previous page)


```
      "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 169.254.225.4:7503",
      "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 169.254.225.4:7502",
      "realtime": "40842065146"
    },
    {
      "active": true,
      "category": "UDP_TRANSMISSION",
      "cursor": 3,
      "id": "0x01000016",
      "level": "WARNING",
      "msg": "Could not send lidar data
```


(continues on next page)

(continued from previous page)


```
            UDP packet to host;
            check that network is up.",
        "msg_verbose": "Failed to send
            lidar UDP data to destination
            host 169.254.225.4:7502",
        "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
            169.254.225.4:7503",
        "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 169.254.225.4:7503",
        "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)

(continued from previous 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
    },
    {
```


(continues on next page)

(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 "RNG19_RFL8_SIG16_NIR16"].
- `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, 12, 4, -4, -12, 12,
   4, -4, -12],
  "pixels_per_column": 128,
  "udp_profile_imu": "LEGACY",
  "udp_profile_lidar": "RNG19_RFL8_SIG16_NIR16_DUAL"
}
```

`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": "2022-11-18T20:31:06"
   }
}
```


`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": 644,
    "input_voltage_mv": 23624,
    "internal_temperature_deg_c": 48,
    "phase_lock_status": "DISABLED",
    "timestamp_ns": 2093396806056
}
```

4 API Changelog

Version v3.0.0

Date 2023-01-26

Description

"Fixed"

- Bug in keep-alive behavior for HTTP 1.1.

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"

- Previously deprecated parameters that have now been removed:
  - `auto_start_flag`
  - `udp_ip`
  - `base_pn`, `base_sn` and `proto_rev`

"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.