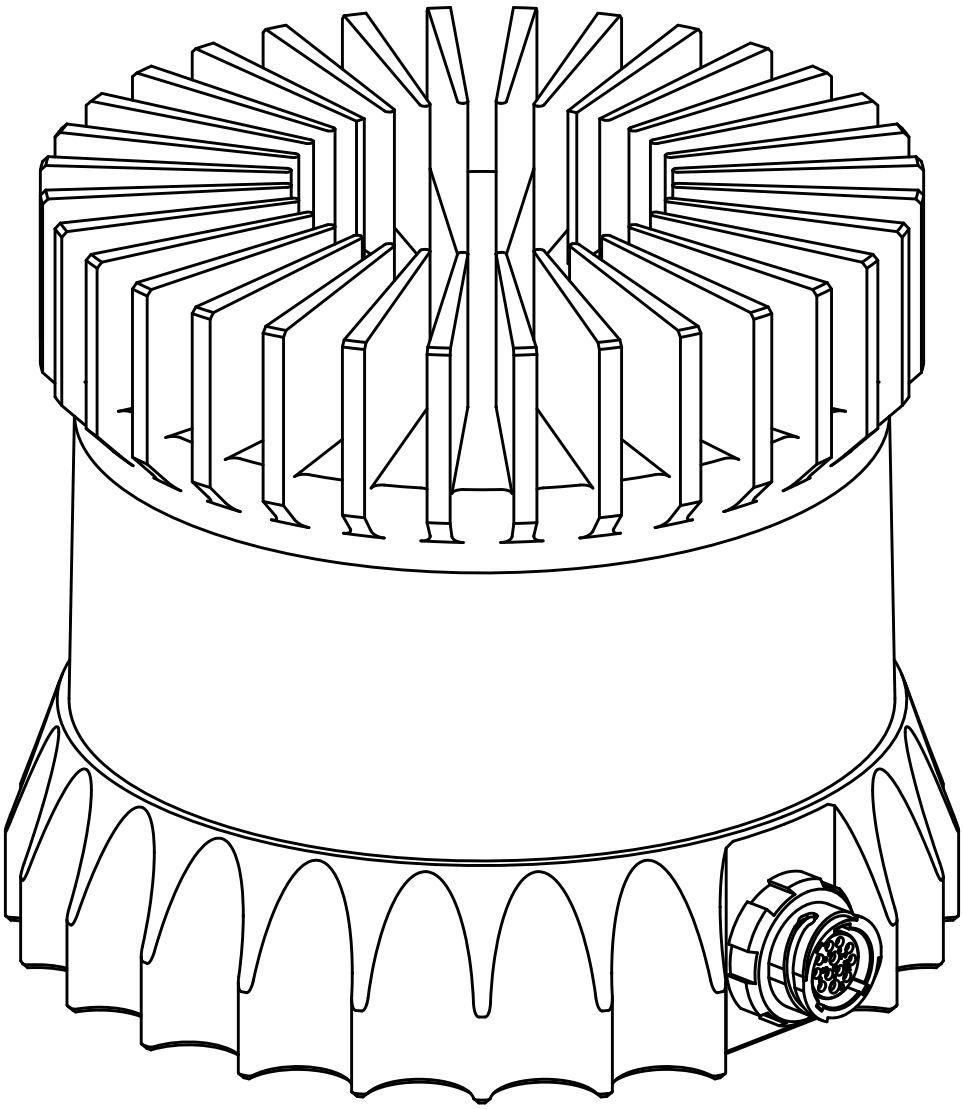


Hardware User Guide

Release v1.13.0

OS1-16/64 High Resolution Imaging Lidar

Nov 07, 2019

Hardware User Guide

1 Mechanical Interface **3**
- 1.1 Included Components . . . 3
- 1.2 Exterior Mechanical Dimensions - 840_101396 Sensor . . . 4
- 1.3 Exterior Mechanical Dimensions - 840_101855 Sensor . . . 5
- 1.4 Heat Sink Requirements . . . 5

2 Electrical Interface **6**
- 2.1 Interface Box . . . 6
- 2.2 Direct Cable Connection . . . 7

3 Drivers & Interface **8**
- 3.1 Ethernet . . . 8
- 3.2 Digital Input/Output . . . 8

4 Updating Firmware **12**

5 Troubleshooting **12**

6 Hardware Errata **15**
- 6.1 MULTIPURPOSE_IO Errata . . . 15

1 Mechanical Interface

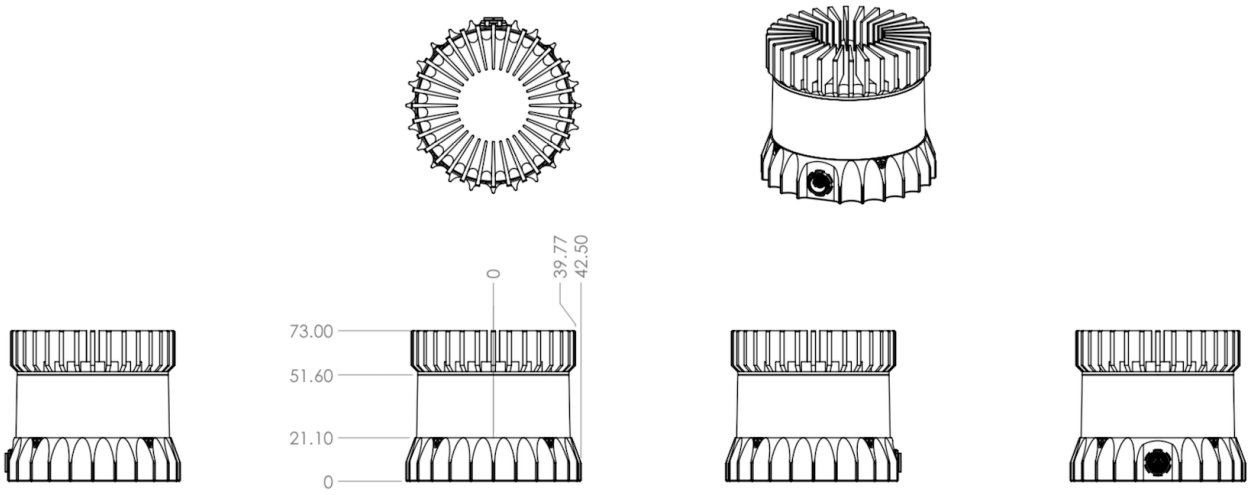
1.1 Included Components

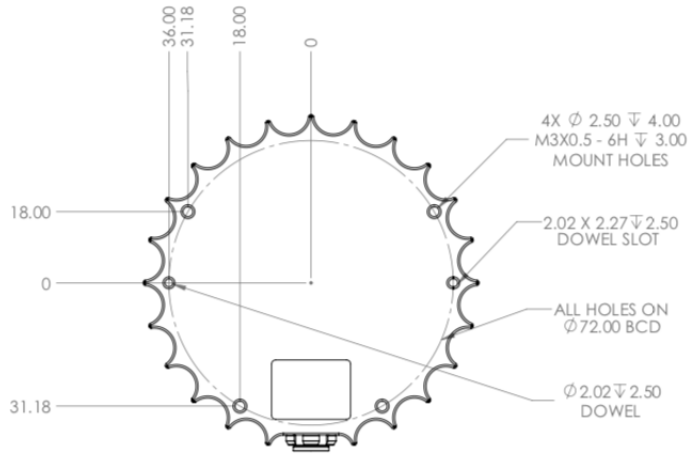
→

The OS1 is shipped with the following items

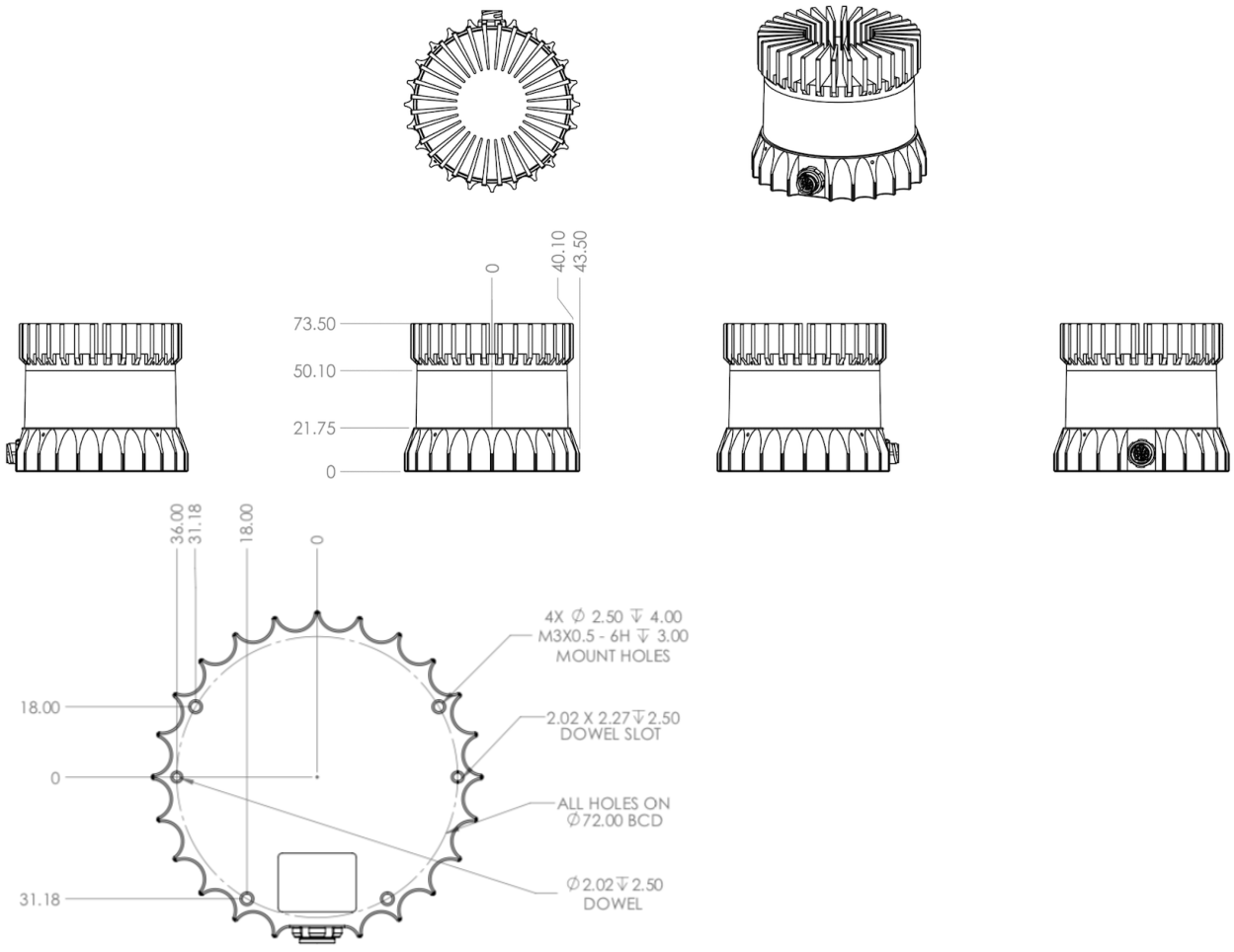
- OS1-16 or OS1-64 sensor
- Sensor to interface box cable/connector
- Interface box
- Interface box AC/DC power supply (2 meters)
- RJ45 cable (1 meter)
- Optional: Heat sink

1.2 Exterior Mechanical Dimensions - 840_101396 Sensor





1.3 Exterior Mechanical Dimensions - 840_101855 Sensor



The sensor has 4 x M3 mounting holes and 2 x 2.0 mm dowel pin holes.

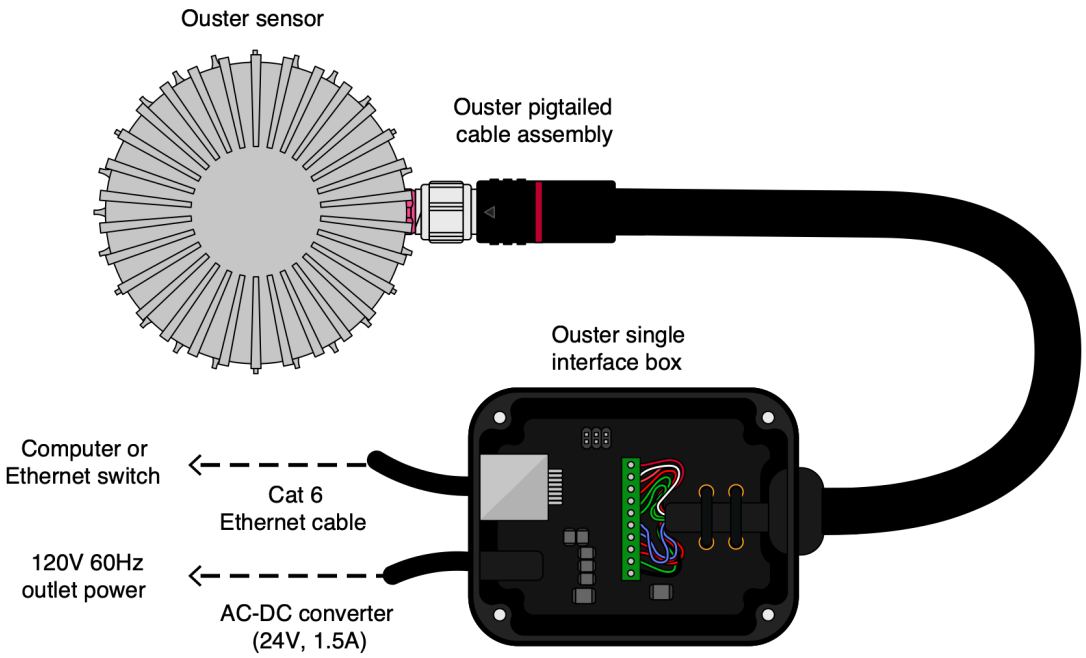
1.4 Heat Sink Requirements

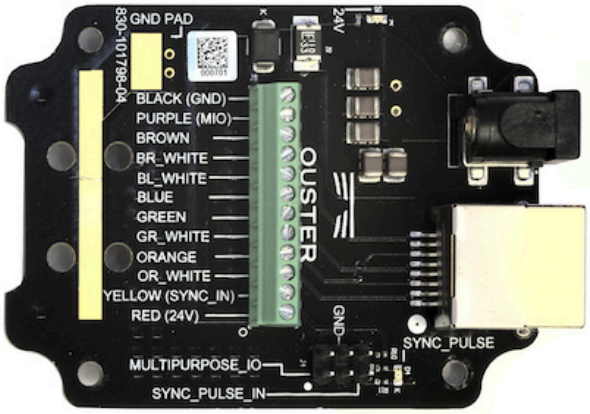
The sensor typically consumes 14-20 W. The recommended temperature of the heatsink is < 25ºC above ambient. Thus, the sensor can operate without performance degradation at 45-50ºC, or higher, with a heat sink. A heat sink is required to achieve the full temperature spec of the sensor, though Ouster internal testing has shown that a heat sink may not be required for use on drones or other moderately or highly convective environments.

2 Electrical Interface

2.1 Interface Box

The Interface Box that accompanies the OS1 sensor is designed to allow the sensor to be operated for test and evaluation purposes. It breaks out the OS1 sensor cable into power and Ethernet connectors.

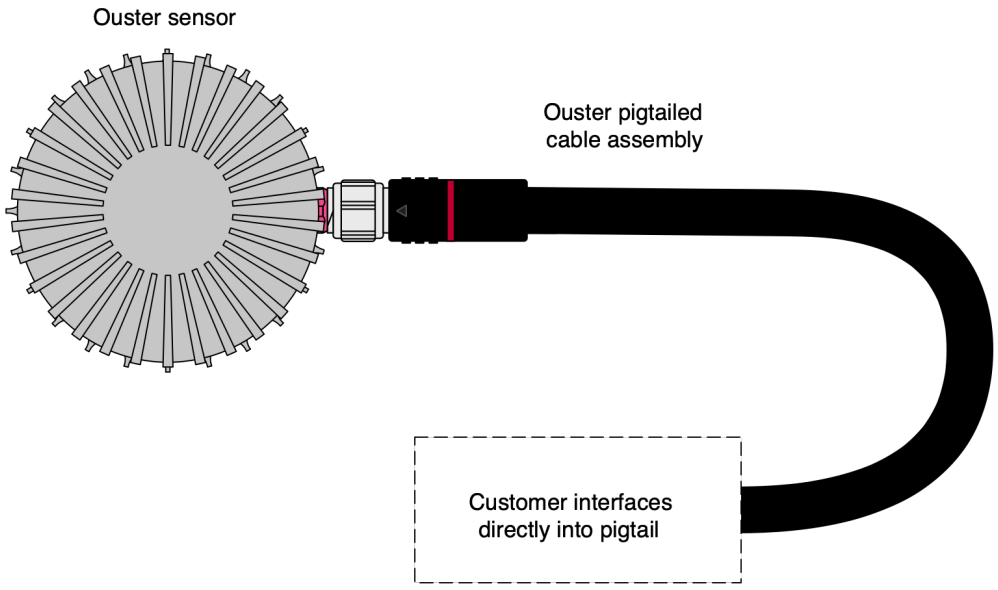




It terminates the interface cable from the sensor, allows it to be powered up and provides access to the Sensor Gigabit Ethernet Interface via a standard RJ45 connector. DC Power to the sensor is provided to the interface box by the accompanying 24 V DC supply.

2.2 Direct Cable Connection

The OS1 can be operated without the use of an interface box; however, Ouster is not responsible for any errors in wiring as a result of bypassing the interface box and this activity may result in a voiding of your warranty if it results in damage to the sensor.



3 Drivers & Interface

3.1 Ethernet

The sensor will automatically turn on, start scanning, obtain an IP address, and start taking measurements when provided power by the Interface Box. However it will only stream UDP data packets after receiving a destination IP address on TCP Port 7501. Instructions can be found in the OS1 Software User Guide and Ouster's drivers on GitHub (https://github.com/ouster-lidar).

Note: If the sensor is not connected to gigabit Ethernet, it will stop sending data and will output an error code. This can be caused by faulty cabling, incorrect wiring, using an older Ethernet switch, or any system-level issue impeding a 1000 Mb/s + full duplex link.

3.2 Digital Input/Output

SYNC_PULSE_IN

`SYNC_PULSE_IN` is a dedicated input channel that can be driven from `SYNC_PULSE_IN` pin of the Interface Box. This channel expects an input pulse sequence which can be used for time synchronization. See the OS1 Software User Guide for more information on configuring this input. Any references to pulse

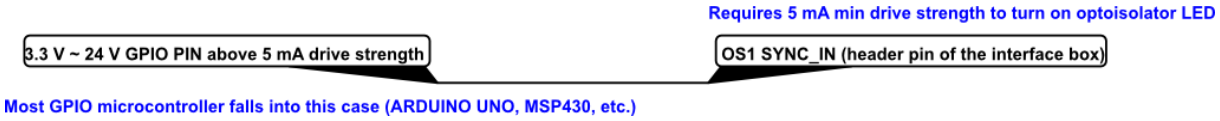
polarity in the OS1 Software User Guide references the signal polarity on the SYNC_PULSE_IN pin of the OS1 sensor. A wide valid range of input pulses can activate the high-speed current source (10 mA at full turn on) on the channel's optoisolator.

Table1: SYNC_PULSE_IN Interface Requirements

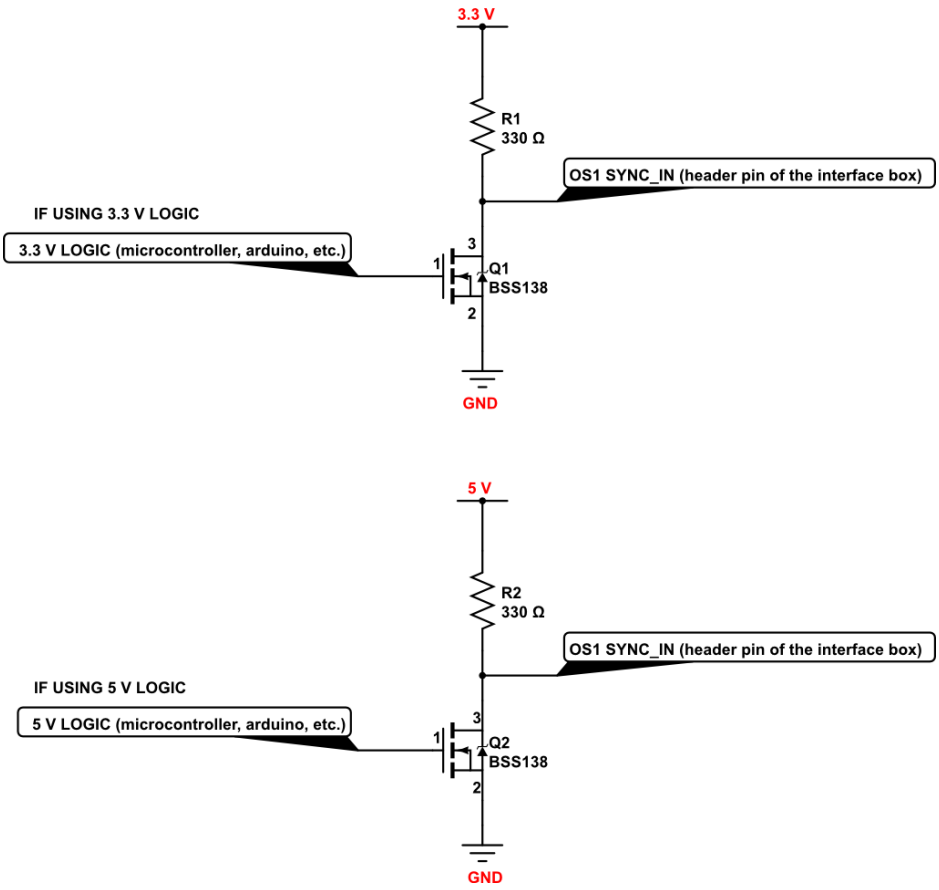
Parameter	Min Voltage	Max Voltage	Min Driver Current
LOGIC LOW	0 V	1 V	N/A
LOGIC HIGH	3.3 V	15 V	5 mA

SYNC_PULSE_IN Interface Requirements were tested with 2 m cable interface box connection at 2 MHz.

- When GPIO has 5 mA drive strength minimum, GPIO can be directly connected to the SYNC_PULSE_IN pin of the interface box header. This is the most common case and has been tested to work on common Arduino microcontroller series. Typical common logic levels of 3.3 V, 5 V GPIO of microcontrollers can produce drive strength of 5 mA min (Arduino, MSP430, etc.).



- If the 5 mA drive strength minimum cannot be met, a buffer circuit is required to drive SYNC_PULSE_IN. Example circuits are provided for common 3.3 V and 5 V logic.



MULTIPURPOSE_IO (M_IO)

`MULTIPURPOSE_IO` (`M_IO`) is a configurable input or output channel connected to the `MULTIPURPOSE_IO` pin of the Interface Box. Detailed information on how to configure this channel using the OS1 TCP interface can be found in the OS1 Software User Guide. By default this channel is disabled.

When configured in **OUTPUT** mode, this channel sends a pulse sequence that can be used for time synchronization or event triggering outside the OS1. For a full description of output pulse triggering options, see the OS1 Software User Guide. This output is an optoisolated open collector circuit, relying on an externally provided pull-up resistor. This resistor is not provided on the Ouster Interface Box.

Table2: MULTIPURPOSE_IO - OUTPUT Interface Requirements

Parameter	Min	Max
Pull Up Voltage	N/A	15 V
Sinking Current	N/A	25 mA

Output is an optoisolated open collector. However current OS1 units have an earlier version of this port which requires an external circuit to utilize this port in output mode. See the below MULTIPURPOSE_IO ERRATA section for more details.

When configured in **INPUT** mode, this channel can accept a standard NMEA $GPRMC UART message. These messages are a common way for GPS systems to share timestamp information in UTC allowing the OS1 to synchronize with GPS time. More information on this packet structure and supported baud rates can be found in the OS1 Software User Guide.

Table3: MULTIPURPOSE_IO - INPUT Interface Requirements

Parameter	Min Voltage	Max Voltage	Min Driver Current
LOGIC LOW	0 V	1 V	N/A
LOGIC HIGH	1.7 V	15 V	10 mA

Above are tested with 2 m cable interface box connection at 2 MHz.

Note: Please note that these input channels are being renamed. Interface boxes may be labeled with the deprecated name."PPS_MASTER" has been renamed to `MULTIPURPOSE_IO`, and `PPS_SLAVE` has been renamed to `SYNC_PULSE_IN`

---

4 Updating Firmware

Sensor firmware can be updated with an Ouster-provided firmware file from www.ouster.com/resources by accessing the sensor over http - e.g. http://os1-991900123456.local/ and uploading the file as prompted.

Always check your firmware version before attempting an update. Only update to a equal or higher version number. Do not roll back firmware to lower numbered versions without having been instructed by Ouster support.

5 Troubleshooting

Starting from firmware v1.11, the sensor HTTP server page http://os1-991900123456.local/ has `Dashboard`, `Diagnostics`, `Documentation` and `Reset Configuration` buttons:

- `Dashboard`: Current page that lists some basic sensor information, and allows sensor firmware upgrade.
- `Diagnostics`: Diagnostic information and system journal that can be downloaded.
- `Documentation`: Sensor User Guide
- `Reset Configuration`: Sensor factory configuration that can be reset to if desired. This will erase any custom configuration that you set on the sensor previously.

Many initial problmes with the OS1 are associated with the sensor not properly being assigned an IP address by a network switch or DHCP server on a client computer. Check your networking settings, the steps in `drivers-interface`, and that all wires are firmly connected if you suspect this problem.

Note: If the sensor is not connected to gigabit Ethernet, it will stop sending data and will output an error code if the cabling is wired incorrectly and fails to achieve a 1000 Mb/s + full duplex link.

To check for hardware errors, use the `get_sensor_info` command as described above.

If the watchdog is triggered (various temperature limits exceeded, uplink/downlink status), an error code will be appended to the end of TCP command `get_sensor_info`. The sensor has a limited size historical buffer that will record the initial errors detected by the sensor with the code itself, timestamp, and an info field (temperature that caused it to trip for temperature failures, true/false for uplink/-downlink).

Example showing active and logged forced temperature sensor failures occuring at timestamps 1569712873477772800, 1569712879991844096, 1569712884968876544 (nanoseconds). The first logged error then resolves itself at 1569713260229536000. The example has been JSON formatted:

```
{
   "active":[
```

(continues on next page)

(continued from previous page)


```
    {
       "category":"OVERTEMP",
       "level":"ERROR",
       "realtime":"1569712879991844096",
       "active":true,
       "msg":"Unit internal temperature out of bounds; please ensure proper heat sinking.",
       "cursor":1,
       "id":"0x01000001",
       "msg_verbose":""
    },
    {
       "category":"OVERTEMP",
       "level":"ERROR",
       "realtime":"1569712884968876544",
       "active":true,
       "msg":"Unit internal temperature out of bounds; please ensure proper heat sinking.",
       "cursor":2,
       "id":"0x01000002",
       "msg_verbose":""
    }
 ],
 "next_cursor":4,
 "log":[
    {
       "category":"OVERTEMP",
       "level":"ERROR",
       "realtime":"1569712873477772800",
       "active":true,
       "msg":"Unit internal temperature out of bounds; please ensure proper heat sinking.",
       "cursor":0,
       "id":"0x01000000",
       "msg_verbose":""
    },
    {
       "category":"OVERTEMP",
       "level":"ERROR",
       "realtime":"1569712879991844096",
       "active":true,
       "msg":"Unit internal temperature out of bounds; please ensure proper heat sinking.",
       "cursor":1,
       "id":"0x01000001",
       "msg_verbose":""
    },
    {
       "category":"OVERTEMP",
       "level":"ERROR",
       "realtime":"1569712884968876544",
       "active":true,
       "msg":"Unit internal temperature out of bounds; please ensure proper heat sinking.",
       "cursor":2,
       "id":"0x01000002",
       "msg_verbose":""
    },
    {
```


(continues on next page)

(continued from previous page)


```
        "category": "OVERTEMP",
        "level": "ERROR",
        "realtime": "1569713260229536000",
        "active": false,
        "msg": "Unit internal temperature out of bounds; please ensure proper heat sinking.",
        "cursor": 3,
        "id": "0x01000000",
        "msg_verbose": ""
      }
   ]
}
```

6 Hardware Errata

6.1 MULTIPURPOSE_IO Errata

Current OS1 sensors require an external compensation circuit when the MULTIPURPOSE_IO port is configured as an OUTPUT. Without this circuit, the resulting waveform is a low amplitude pulse train regardless of the pull up voltage. The following example circuit can correct the output waveform. An upcoming hardware fix will eliminate the need for this buffer circuit.

Buffer circuit fix for MULTIPURPOSE_IO in OUTPUT mode:

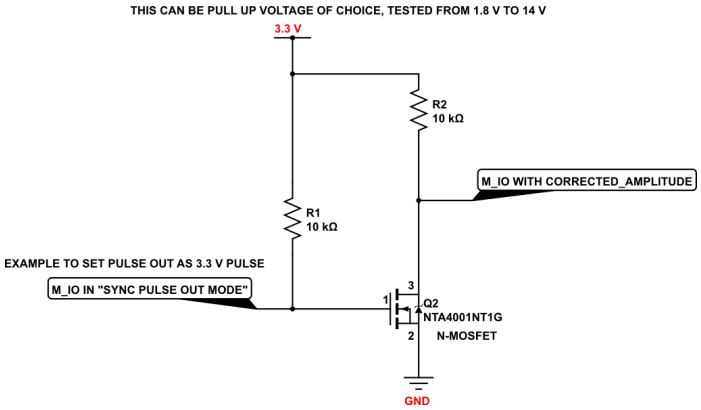


Table4: MULTIPURPOSE_IO - OUTPUT Buffer Circuit Example

Parameter	M_IO IN SYNC_PULSE_OUT_MODE	SYNC_PULSE_OUT_MODE with corrected amplitude
Example Waveform LOGIC LOW	0 V	0 V
Example Waveform LOGIC HIGH	1.2 V	3.3 V