

## OS3D

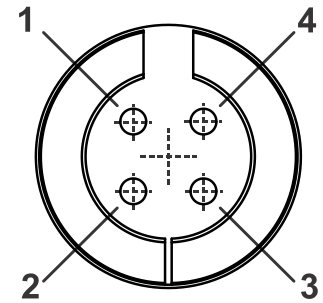
### MINIATURE 3D ORIENTATION SENSOR



Interface Control Document  
Rev. 1.14

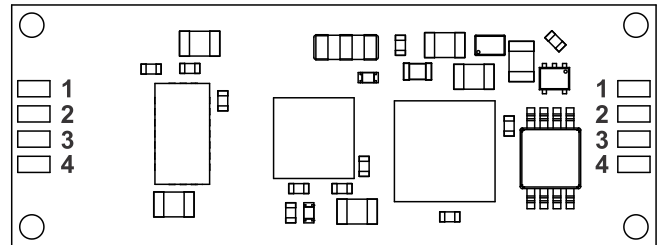
## 1. Hardware layer

The **Inertial Labs OS3D** can be connected to a host system via the 4-pins mail receptacle (Binder 09-9767-70-04 if equipped with a plastic case or Binder 09-3111-71-04 if equipped with an aluminum case) or by direct soldering to PCB contact pads. Both connection types are shown on the picture below. The **Inertial Labs LVDS-USB** converter is used for connection the **OS3D** to PC. See the Appendix A for the connection details. The direct UART TTL connection is also possible, see AN002 for details.



**Table 1.1 Hardware layer description.**

Num	Name	Function	Parameters
1	PWR	supply voltage	3.3V to 5.5V
2	GND	ground, shield	
3	A	A RS-485	1 Mbps, 120Ω
4	B	B RS-485	1 Mbps, 120Ω



This document describes communication layers according to ISO/OSI model.

## 2. Physical layer

**Table 2.1 Physical layer description.**

Parameter	Value	Units
Standard	TIA/EIA-485A (half-duplex)	
Default Baud Rate*	1000000	bps
Serial port settings	8-N-1	

\* could be changed according customer's requirements

## 3. Transport layer

The communications protocol is optimized to work with 16-bit words. Each word consists of two bytes, the first byte is the least significant, and the second is the most significant one. If the pause between bytes in packet is longer than 10uS, this packet is ignored by sensor.

**Table 3.1 Transport layer description.**

Index	Name	Description	Value
1	Header	Packet header	-
2	Length	Packet length in bytes (from Header to CS)	8-65534
3	Data1	Data	various types
XX	DataXX	Data	various types
XX	DataXX	Data	various types
N+2	DataN	Data	various types
N+3	CS	Checksum (Sum from Header to DataN)	0-65535
-	Pause	The pause allows a sensor to parse a packet	>20uS

In a request packet coming from an application host to a sensor the Header field could take on two values: Header = 0x55AA if broadcasting packet, or Header = (Address\*256) + (255-Address) if packet is intended to the sensor with certain address. Address can take on values from 0 to 255, Address = 85 is broadcast address. **Header** value is defined in **Status Buffer** of the sensor (see below).

In a response packet coming from a sensor to an application host the packet header is always 0x55AA;

## 4. Application layer

4.1 **Reset command** is issued by the application host to reset the device.

**Table 4.1 Reset command structure.**

Transport	Name	Value	Notation	Description
Data1	Cmd	0xFF00	unsigned word	Packet type

**Response:** no response. As the result the sensor is in the initial state and its output stream is stopped.

4.2 **GetIden command** is issued by the application host to request the sensor's ID string.

**Table 4.2.1 GetIden request structure.**

Transport	Name	Value	Notation	Description
Data1	Cmd	0x0100	unsigned word	Packet type

**Response:** the sensor returns its identification string.

**Table 4.2.2 GetIden response structure.**

Transport	Name	Value	Notation	Description
Data1	Cmd	0x0110	unsigned word	Command type
Data2...65	ID[64]		char	ID string

ID example: "OSv3m3 2010.04.03 V3.55" for OSv3 type; "OSv4m1 Ver 01.008 2012.07.26" for OSv4 type; "OSv6m1\_V1104 Oct 6 2015" for OSv6 type.

4.3 **GetData command** is issued by the application host to request the data from sensor. The application host can request the different combinations of the orientation data using the following GetData subtypes:

**GetDataR command** is issued by the application host to request the raw data form ADC (not calibrated).

**Table 4.3.1 GetDataR request structure.**

Transport	Name	Value	Notation	Description
Data1	Cmd	0x0200	unsigned word	Packet type

**Response:**

**Table 4.3.2 GetDataR response structure.**

Transport	Name	Value	Notation	Description
Data1	Cmd	0x0210	unsigned word	Command type
Data2	Cntr*	0-65535	unsigned word	Packet counter
Data3	Acc1	±32767	signed word	Raw acc. ch.1
Data4	Acc2	±32767	signed word	Raw acc. ch.2
Data5	Acc3	±32767	signed word	Raw acc. ch.3
Data6	Gyro1	±32767	signed word	Raw gyro ch.1
Data7	Gyro2	±32767	signed word	Raw gyro ch.2
Data8	Gyro3	±32767	signed word	Raw gyro ch.3
Data9	Mag1	±32767	signed word	Raw mag. ch.1
Data10	Mag2	±32767	signed word	Raw mag. ch.2
Data11	Mag3	±32767	signed word	Raw mag. ch.3
Data12	Temp	±32767	signed word	Raw temp.

\* Cntr is incremented each time the data packet is transferred; it can be used to control continuity of the sensor data stream.

**GetDataQ command** is issued by the application host to request the orientation quaternion. This command is not supported by OSv4 sensor. See GetIden command for sensor types.

**Table 4.3.3 GetDataQ request structure.**

Transport	Name	Value	Notation	Description
Data1	Cmd	0x0201	unsigned word	Packet type

**Response:**

The estimated attitude quaternion. The first part is the scalar value. The attitude is given as the body frame with respect to the local East North Up (ENU) frame.

**Table 4.3.4 GetDataQ response structure.**

Transport	Name	Value	Notation	Description
Data1	Cmd	0x0211	unsigned word	Command type
Data2	Cntr	0-65535	unsigned word	Packet counter
Data3	QuatW	-1.0 - 0.99997	signed fixed point Q1.15	Quaternion W
Data4	QuatX	-1.0 - 0.99997	signed fixed point Q1.15	Quaternion X
Data5	QuatY	-1.0 - 0.99997	signed fixed point Q1.15	Quaternion Y
Data6	QuatZ	-1.0 - 0.99997	signed fixed point Q1.15	Quaternion Z

**GetDataD command** is issued by the application host to request calibrated data & temperature. This command is not supported by OSv3 and OSv4 sensors. See GetIden command for sensor types.

**Table 4.3.5 GetDataD request structure.**

Transport	Name	Value	Notation	Description
Data1	Cmd	0x0202	unsigned word	Packet type

**Response:**

**Table 4.3.6 GetDataD response structure.**

Transport	Name	Value	Notation	Description
Data1	Cmd	0x0212	unsigned word	Command type
Data2	Cntr	0-65535	unsigned word	Packet counter
Data3	AccX	-1.0 - 0.99997	signed fixed point Q1.15	Acc. X <sup>1</sup>
Data4	AccY	-1.0 - 0.99997	signed fixed point Q1.15	Acc. Y <sup>1</sup>
Data5	AccZ	-1.0 - 0.99997	signed fixed point Q1.15	Acc. Z <sup>1</sup>
Data6	MagX	-1.0 - 0.99997	signed fixed point Q1.15	Mag. X <sup>2</sup>
Data7	MagY	-1.0 - 0.99997	signed fixed point Q1.15	Mag. Y <sup>2</sup>
Data8	MagZ	-1.0 - 0.99997	signed fixed point Q1.15	Mag. Z <sup>2</sup>
Data9	GyroX	-1.0 - 0.99997	signed fixed point Q1.15	Gyro X <sup>3</sup>
Data10	GyroY	-1.0 - 0.99997	signed fixed point Q1.15	Gyro Y <sup>3</sup>
Data11	GyroZ	-1.0 - 0.99997	signed fixed point Q1.15	Gyro Z <sup>3</sup>
Data12	Temp	-1.0 - 0.99997	signed fixed point Q1.15	Temperature <sup>4</sup>

- 1 – Conversion factor for **OSv3**: 0.5 = 1g; for **OSv6**: 0.0625 = 1g;
- 2 – Conversion factor for **OSv3**: 0.5 ≈ 0.5 Gauss; for **OSv6**: 0.0625 ≈ 0.5 Gauss;
- 3 – Conversion factor for **OSv3**: 1/1790.5 = 1 dps; for **OSv6**: pi/5760 = 1dps;
- 4 – Conversion formula for **OSv3**: T°C = -142\*Temp + 12; for **OSv6**: T°C = 96.4\*Temp + 33;

**GetDataF command** is issued by the application to request the quaternion, calibrated data & temperature. This command is not supported by OSv4 sensor. See GetIden command for sensor types.

**Table 4.3.7 GetDataF request structure.**

Transport	Name	Value	Notation	Description
Data1	Cmd	0x0203	unsigned word	Packet type

**Response:**

**Table 4.3.8 GetDataF response structure.**

Transport	Name	Value	Notation	Description
Data1	Cmd	0x0213	unsigned word	Command type
Data2	Cntr	0-65535	unsigned word	Packet counter
Data3	QuatW	-1.0 - 0.99997	signed fixed point Q1.15	Quaternion W
Data4	QuatX	-1.0 - 0.99997	signed fixed point Q1.15	Quaternion X
Data5	QuatY	-1.0 - 0.99997	signed fixed point Q1.15	Quaternion Y
Data6	QuatZ	-1.0 - 0.99997	signed fixed point Q1.15	Quaternion Z
Data7	AccX	-1.0 - 0.99997	signed fixed point Q1.15	Acc. X <sup>1</sup>
Data8	AccY	-1.0 - 0.99997	signed fixed point Q1.15	Acc. Y <sup>1</sup>
Data9	AccZ	-1.0 - 0.99997	signed fixed point Q1.15	Acc. Z <sup>1</sup>
Data10	MagX	-1.0 - 0.99997	signed fixed point Q1.15	Mag. X <sup>2</sup>
Data11	MagY	-1.0 - 0.99997	signed fixed point Q1.15	Mag. Y <sup>2</sup>
Data12	MagZ	-1.0 - 0.99997	signed fixed point Q1.15	Mag. Z <sup>2</sup>
Data13	GyroX	-1.0 - 0.99997	signed fixed point Q1.15	Gyro X <sup>3</sup>
Data14	GyroY	-1.0 - 0.99997	signed fixed point Q1.15	Gyro Y <sup>3</sup>
Data15	GyroZ	-1.0 - 0.99997	signed fixed point Q1.15	Gyro Z <sup>3</sup>
Data16	Temp	-1.0 - 0.99997	signed fixed point Q1.15	Temperature <sup>4</sup>

- 1 – Conversion factor for **OSv3**: 0.5 = 1g; for **OSv6**: 0.0625 = 1g;
- 2 – Conversion factor for **OSv3**: 0.5 ≈ 0.5 Gauss; for **OSv6**: 0.0625 ≈ 0.5 Gauss;
- 3 – Conversion factor for **OSv3**: 1/1790.5 = 1 dps; for **OSv6**: pi/5760 = 1dps;
- 4 – Conversion formula for **OSv3**: T°C = -142\*Temp + 12; for **OSv6**: T°C = 96.4\*Temp + 33;

**GetDataE command** is issued by the application to request the Euler angles. This command is not supported by OSv3 and OSv4 sensors. See GetIden command for sensor types.

**Table 4.3.9 GetDataE request structure.**

Transport	Name	Value	Notation	Description
Data1	Cmd	0x0204	unsigned word	Packet type

**Response:**

The estimated attitude Yaw, Pitch, and Roll angles measured in modified physical units: 1.0 = 180 deg. The attitude is given as a 3,1,2 Euler angle sequence describing the body frame with respect to the local East North Up (ENU) frame.

**Table 4.3.10 GetDataE response structure.**

Transport	Name	Value	Notation	Description
Data1	Cmd	0x0214	unsigned word	Command type
Data2	Cntr	0-65535	unsigned word	Packet counter
Data3	Yaw	-1.0 - 0.99997	signed fixed point Q1.15	Euler yaw angle <sup>2</sup>
Data4	Pitch	-1.0 - 0.99997	signed fixed point Q1.15	Euler pitch angle <sup>2</sup>
Data5	Roll	-1.0 - 0.99997	signed fixed point Q1.15	Euler roll angle <sup>2</sup>

**4.4 GetStat command** is issued by the application host to request the **Status Buffer**.

**Table 4.4.1 GetStat request structure.**

Transport	Name	Value	Notation	Description
Data1	Cmd	0x0300	unsigned word	Packet type

**Response:**

**Table 4.4.2 GetStat response structure.**

Transport	Name	Value	Notation	Description
Data1	Cmd	0x0310	unsigned word	Command type
Data2... Data33	StatBuff[32]	0-65535	unsigned word	Status buffer

See table 4.4.3 for **Status Buffer** structure.

**Status Buffer** contents the following status variables:

**Table 4.4.3 Status Buffer structure.**

Name	Notation	Description
AutoTx	unsigned word	AutoTx = 0xFFFF turns the sensor to auto transfer mode
ModeA	unsigned word	Defines the output data type in auto transfer mode
Period	unsigned word	Defines the data transferring period in auto transfer mode
Header	unsigned word	Device packet header (see the Table 3.1)
SN_H	unsigned word	MSW of the device's 32-bit serial number
SN_L	unsigned word	LSW of the device's 32-bit serial number
.....	unsigned word	Other variables defined by firmware

The period of data transferring in auto transfer mode is defined by **Period** variable in the following way  
 $Time(us) = Period$ . Valid range for **Period** is 500-65535 (2KHz - 15.26Hz);

The output data type is defined by the **ModeA** in the following way:

- 1000 (decimal value) – raw data (see **GetDataR response**);
- 1001 (decimal value) - quaternion (see **GetDataQ response**);
- 1002 (decimal value) - calibrated data & temperature (see **GetDataD response**);
- 1003 (decimal value) - quaternion, calibrated data & temperature (see **GetDataF response**);
- 1004 (decimal value) - Euler angles (see **GetDataE response**);

**4.5 SetVar command** is issued by the application host to define the variables in status buffer.

**Table 4.5.1 SetVar command structure.**

Transport	Name	Value	Notation	Description
Data1	Cmd+Addr*	0x0400+0x00XX	unsigned word	Pack.type+Addr
Data2	Value	0-65535	unsigned word	Variable value

\*Addr is the address of status variable in Status Buffer, for ex.: the Addr = 0 for AutoTx.

**Response:** this command doesn't have a response.

## 5. Software examples

### 5.1 The example of correct parsing the data packets

```
while(!Device->Stop)
{
    if( !ReadFile(hCom, CurBuf, 1, &Cntr, NULL ) || Cntr != 1 )
    {
        ClearCommError(hCom, &ldwErrors, &ICOMSTAT );
        continue;
    }

    /* parse byte read */
    switch(InCnt)
    {
    case 0 : // Idenf AA
        if( CurBuf[0] != 0xAA )
        {
            //printf("\n0xAA byte not found!");
            reset = true;
        }
        break;

    case 1 : // Idenf 55
        if( CurBuf[0] != 0x55 )
        {
            //printf("\n0x55 byte not found!");
            reset = true;
        }
        break;

    case 2 : // LenL
        {
            MessLen = CurBuf[0];
            break;
        }

    case 3 : // LenH
        {
            MessLen += (CurBuf[0] << 8);
            break;
        }

    default:
        {
            If( InCnt != ( MessLen - 1 ) )
            {
                break;
            }

            /* message reception completed, process the message */
            MessLen = (MessLen/2) - 1;
        }
    }
}
```

```
CSumm = 0;

/* checksum */
for( unsigned int i = 0; i < MessLen; i++ )
{
    CSumm += WBuf[i];
}
if( CSumm != WBuf[MessLen] )
{
    printf( "\nCheckSum failed" );
    reset = true;
    break;
}
}

if( reset )
{
    CurBuf = Buf;
    CSumm = 0;
    InCnt = 0;
    reset = false;
}
else
{
    InCnt++;
    CurBuf++;
}
}
}
```

## 5.2 The example of sending commands to sensor

- Send the **“Reset”** command to the sensor:  
**{0xAA,0x55,0x08,0x00,0x00,0xFF,0xB2,0x54}**
- Pause 1 S;
- Set the **GetDataQ** as the output data type in auto transfer mode:  
**{0xAA,0x55,0x0A,0x00,0x01,0x04,0xE9,0x03,0x9E,0x5D}**
- Pause 25 uS;
- Turn the sensor to auto transfer mode:  
**{0xAA,0x55,0x0A,0x00,0x00,0x04,0xFF,0xFF,0xB3,0x59}**
- Pause 25 uS;

Result: the sensor transfers a quaternion data packet.



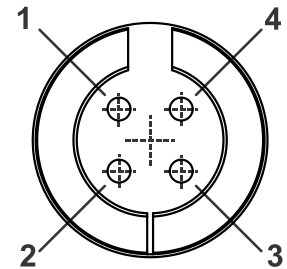
## 6. Appendix A - The OS3D possible interfaces

The OS3D sensor has built-in TIA/EIA-485A half-duplex transceiver. The host controller (user specific device or any PC) must also be equipped with a TIA/EIA-485A transceiver.

- The sensor can be equipped with 2 meter cable with mating to binder 09-9767-70-04 connector if ordered in plastic case. The final user may use the **Inertial Labs LVDS-USB** converter for connection the **OS3D** to PC. Please refer to the drawings below for the sensor's dimension and interface:

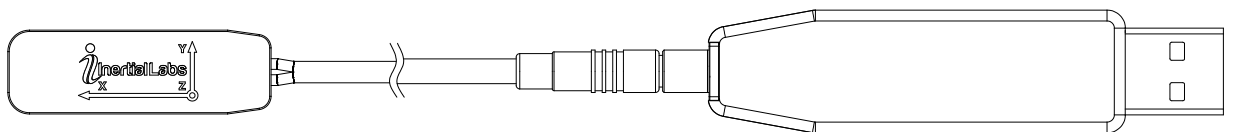
**Table 6.1 OS3D interface pinout**

Num	Name	Function	Parameters
1	PWR	supply voltage	3.3V to 5.5V
2	GND	ground, shield	
3	A	A RS-485	1 Mbps, 120Ω
4	B	B RS-485	1 Mbps, 120Ω



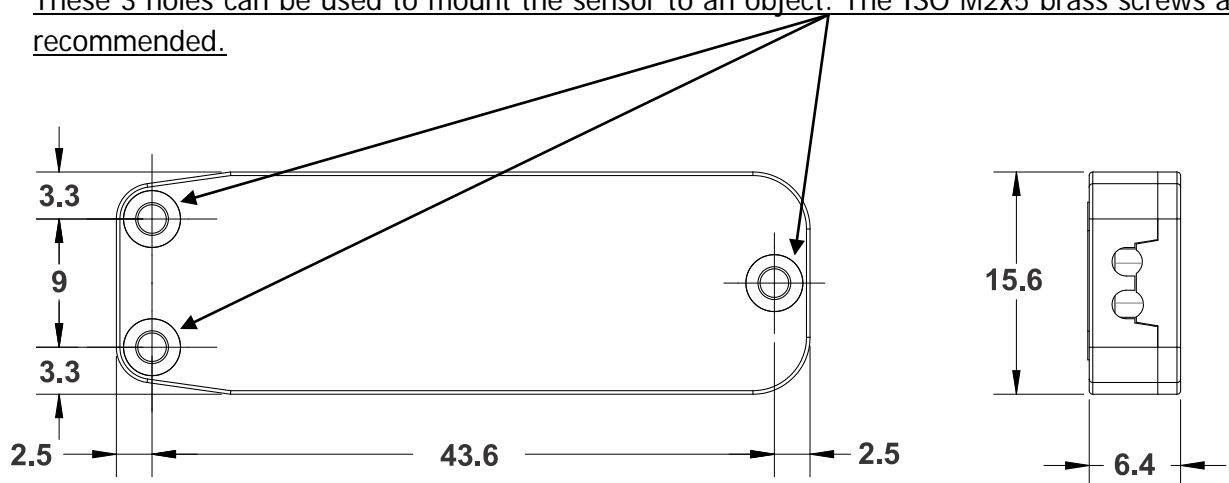
**Binder connector  
(view for pins side)**

### OS3D and the the Inertial Labs LVDS-USB converter



### OS3D mechanical drawings (plastic case) (mm)

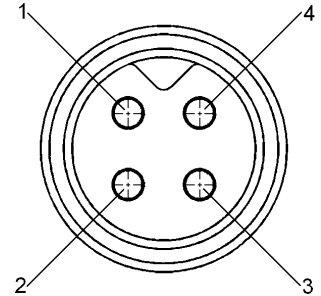
These 3 holes can be used to mount the sensor to an object. The ISO M2x5 brass screws are recommended.



- The sensor can be equipped with up to 5 meter cable with mating to binder 09-3111-71-04 connector if ordered in aluminum case. The final user may use the **Inertial Labs LVDS-USB** converter for connection the **OS3D** to PC. Please refer to the drawings below for the sensor's dimension and interface:

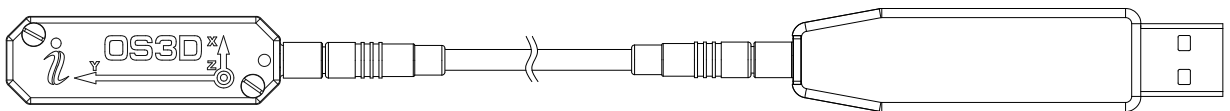
**Table 6.2 OS3D interface pinout**

Num	Name	Function	Parameters
1	PWR	supply voltage	3.3V to 5.5V
2	GND	ground, shield	
3	A	A RS-485	1 Mbps, 120Ω
4	B	B RS-485	1 Mbps, 120Ω



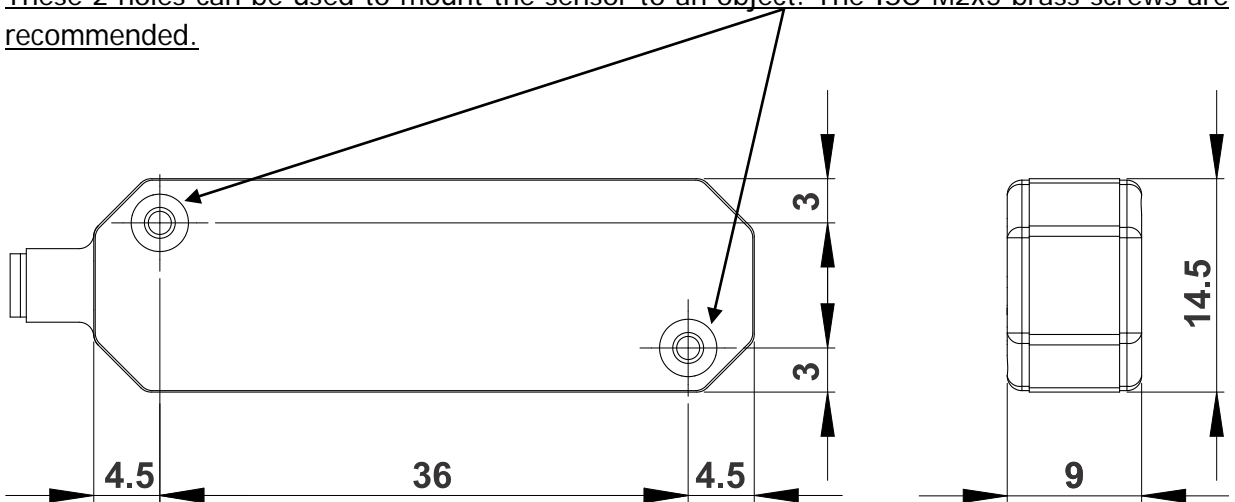
**Binder connector  
(view for pins side)**

#### OS3D and the the Inertial Labs LVDS-USB converter



#### OS3D mechanical drawings (aluminum case) (mm)

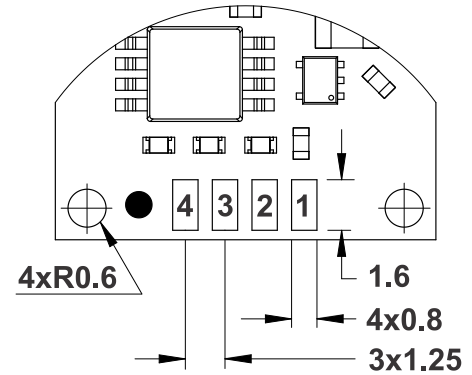
These 2 holes can be used to mount the sensor to an object. The ISO M2x5 brass screws are recommended.



- The customer is responsible for proper mechanical and electrical connection if a sensor was ordered as an OEM product. Please refer to the drawings below for the sensor's dimension and interface:

**Table 6.3 OS3D-OEM interface pinout**

Num	Name	Function	Parameters
1	PWR	supply voltage	3.3V to 5.5V
2	GND	ground, shield	
3	A	A RS-485	1 Mbps, 120Ω
4	B	B RS-485	1 Mbps, 120Ω



**OS3D-OEM soldering pads drawings (mm)**

**OS3D-OEM mechanical drawings (mm)**

