Interface Specification for ibeo LUX, ibeo LUX systems and ibeo Evaluation Suite

Version History

Date	Version	Changes
21.11.2008	1.0	Initial release
03.12.2008	1.1	Added section 9 with ibeo FUSION SYSTEM trace message description
04.12.2008	1.2	Correction of description of data type 2224 (ObjectList), Field ScanNumber removed.
09.12.2008	1.3	New object data type 0x2225 (replaces 0x2224)
27.1.2009	1.4	• Corrected reply to GetStatus command – reserved word removed • Another reserved word in"SetNTPTimestampSec" and "SetNTPTimestampFracSec" command messages included • FUSION SYSTEM and firmware version numbers integrated • FPGA and firmware version state are decimal coded
13.3.2009	1.5	Changed default IP-Address for LUX to 192.168.0.1
27.3.2009	1.6	Corrected sync phase offset in 0x2202.
6.4.2009	1.7	Comment added to set timestamp commands.
11.6.2009	1.8	Data type of parameter TCP/IP port fixed. Parameter name vehicle width fixed. Default IP configuration of Ibeo ECU changed.
16.6.2009	1.9	Corrected FUSION SYSTEM/ECU data type trace message. Clarified reset default parameters command.
31.7.2009	1.10	Steering ratio types added. Default IP address of Ibeo ECU changed. Format and text changes.
01.10.2009	1.11	ECU Set filter command added.
05.10.2009	1.12	Fixed reserved values in set NTP time stamp commands.

14.10.2009	1.13	Clarified echo and layer encoding in 0x2202.
20.11.2009	1.14	Added new/future scan data type 0x2205.
11.01.2010	1.15	Added scan flags in datatypes 0x2204 and 0x2205 Added scan flag mirror side in scan data type 0x2202
09.04.2010	1.16	Correction: Bounding box size is transmitted as width (Y-) and length (X-Coordinate) of bounding box, Datatype 0x2221
10.09.2010	1.17	In “ibeo LUX parameter list” add list of parameters that need a sensor restart to take effect.
05.10.2010	1.18	Replace commands SetNTPTimestampSec and SetNTPTimestampFracSec with command SetNTPTimestampSync. Add Customer Processing Switch 0 and Interface Flags to parameter list.
21.10.2010	1.19	More detailed description of SetFilterCommand for connection to an ECU.
05.11.2010	1.20	Added ibeo LUX vehicle state DataType 0x2805 with basic description
05.11.2010	1.21	More detailed description of 0x2805
16.11.2010	1.22	Description for ibeo ECU vehicle state data type (0x2806) added.
06.01.2011	1.23	Added description of transparency flag in scandata type 0x2205
29.03.2011	1.24	Added some description to the steer ratio polynom
05.07.2011	1.25	Some corrected mistakes concerning data type 0x2205, some more description
16.11.2011	1.26	Added description for yaw rate in date type LUX vehicle state (0x2805) Available since Firmware version 2.5.00.
28.11.2011	1.27	Follow-up to 1.26. Missing unit for yaw rate in data type LUX vehicle state (0x2805) added.
28.11.2011	1.28	Correction: Default IP address (ECU) 192.168.0.100
15.03.2012	1.29	Description for the sigma values added
26.04.2012	1.30	Table of Content updated
12.12.2012	1.31	Update fpga status, fpga error register and the fpga warning register
17.12.2012	1.32	Object box orientation in Datatype 0x2221 is not given in 1/32° but in 1/100°.
11.02.2013	1.33	Added DeviceType ParameterID to ethernet documentation

12.04.2013	1.34	Added Datatype Image
05.12.2013	1.35	Added new object 0x2281 and new vehicle state Data Type 0x2807
10.01.2014	1.36	Added new object 0x2281 data type
21.02.2014	1.37	Added description of validation, mobility, tracking model in Object Datatype 0x2280
25.04.2014	1.38	Updated descriptions for 0x2280 and 0x2281.
12.11.2014	1.39	Fixed Value description for 0x2806 and 0x2807
26.02.2015	1.40	Fix bad offset of (List of contour points) in 0x2280
09.09.2015	1.41	New name for the document, additional overview of datatypes and additional data types
10.09.2015	1.42	Addition of missing data types
11.09.2015	1.43	Fixes Carriage Way
12.10.2015	1.44	Addition flags for scans
29.04.2016	1.45	Added description for vehicle state 0x2808 and measurement list 0x2821
23.06.2016	1.46	Added description for point cloud plane 0x7510 and meta information list 0x7110. Added flag description to 0x2281
17.08.2016	1.47	Added description for CAN Messages over Ethernet 0x1002 and added description for Object Association 0x4001
30.05.2017	1.48	Added data types: - 0x9000 Time Record - 0x9001 GpsImu - 0x9002 Odometry - 0x6700 SystemMonitoringCANStatus - 0x6701 SystemMonitoringDeviceStatus - 0x6705 SystemMonitoringSystemStatus

Table of Content

1 Introduction

This document describes the data interface for:

- ibeo LUX sensor and ibeo Fusion System/ECU: this includes data received and transmitted from and to ibeo LUX sensors and ibeo LUX systems via the Ethernet connection and using *idc files (since the same data types are used for these files).

  Addressed systems are ibeo LUX sensors with the firmware version 2.6.08 and Fusion Systems/ECUs using the current Ibeo FUSION SYSTEM version 6.7.11.

- ibeo Evaluation Suite: considers additional data types that are generated by the Ibeo Evaluation Suite software and provided in *idc files.

  Addressed versions for the ibeo Evaluation Suite are:
  R2014-10
  R2015-08
  R2016-05

2 General information

2.1 Overview of the data types

This section presents an overview of all data types that can be produced by ibeo LUX sensor, ibeo Fusion Systems/ECUs, and the additional data types generated by the ibeo Evaluation Suite.

ibeo LUX sensor		
Data type name	**Data type ID**	**Details in this document**
ibeo LUX scan data:	0x2202	Section 3
ibeo LUX object data	0x2221	Section 4
ibeo LUX vehicle state	0x2805	Section 5
Ibeo LUX errors and warnings	0x2030	Section 6
ibeo LUX command interface	0x2010	Section 7

ibeo Fusion System/ECU		
Data type name	**Data type ID**	**Details in this document**
Ibeo FUSION SYSTEM/ECU scan data	0x2205	Section 9
Ibeo FUSION SYSTEM/ECU object data	0x2280	Section 11
Ibeo FUSION SYSTEM/ECU object data	0x2281	Section 12
Ibeo FUSION SYSTEM/ECU image	0x2403	Section 13
Ibeo FUSION SYSTEM/ECU vehicle state	0x2807	Section 16
Ibeo FUSION SYSTEM/ECU measurement list	0x2821	Section 21
Ibeo FUSION SYSTEM/ECU Configuration Info	0x6201	For internal use and debugging only

Ibeo FUSION SYSTEM/ECU GPS position (PositionWGS84)	0x2604	Section 22
Ibeo FUSION SYSTEM/ECU CAN messages[1]	0x1002	Section 30

ibeo Evaluation Suite		
Data type name	**Data type ID**	**Details in this document**
Ibeo FUSION SYSTEM/ECU measurement list	0x2821	Section 21
Ibeo FUSION SYSTEM/ECU scan data	0x2205	Section 9
Ibeo FUSION SYSTEM/ECU object data	0x2281	Section 12
Ibeo FUSION SYSTEM/ECU image	0x2403	Section 13
Ibeo FUSION SYSTEM/ECU vehicle state	0x2807	Section 16
Ibeo FUSION SYSTEM/ECU vehicle state	0x2808	Section 17
Ibeo frame end separator	0x1100	Section 23
Ibeo reference object data	0x2291	Section 24
Ibeo frame index	0x6130	Section 25
Ibeo labeled objects	0x6503	For manual corrections using ILV only. Not to be used by customers
Ibeo carriage way	0x6970	Section 26
Ibeo IDCTrailer	0x6120	Section 27
Ibeo point cloud plane	0x7510	Section 28
Ibeo meta information list	0x7110	Section 29
Ibeo object association	0x4001	Section 31

2.2 Ethernet configuration

ibeo LUX sensors and Ibeo ECUs use default Ethernet configurations until changed by the user.

ibeo LUX sensors use the default IP address 192.168.0.1
with the subnet mask 255.255.255.0.
The default port is 12002.

Ibeo ECUs use the default IP address 192.168.0.100
with the subnet mask 255.255.255.0.
Default port for data connection is 12002.
Standard ports for telnet and FTP are used.

2.3 Data encoding

Attention! See the data type description if little or big endian byte order is used!

[1] Only required when using the DuT Tools expansion module.

NTP64 timestamps represent the time encoded in 8 bytes. In order to decode NTP64 timestamps, the corresponding 8 bytes need to be interpreted as UINT64:
The higher 4 bytes are the number of seconds since 1.1.1900 - 0:00:00. The lower 4 bytes represent the fractional seconds with a resolution of 2-32 s.

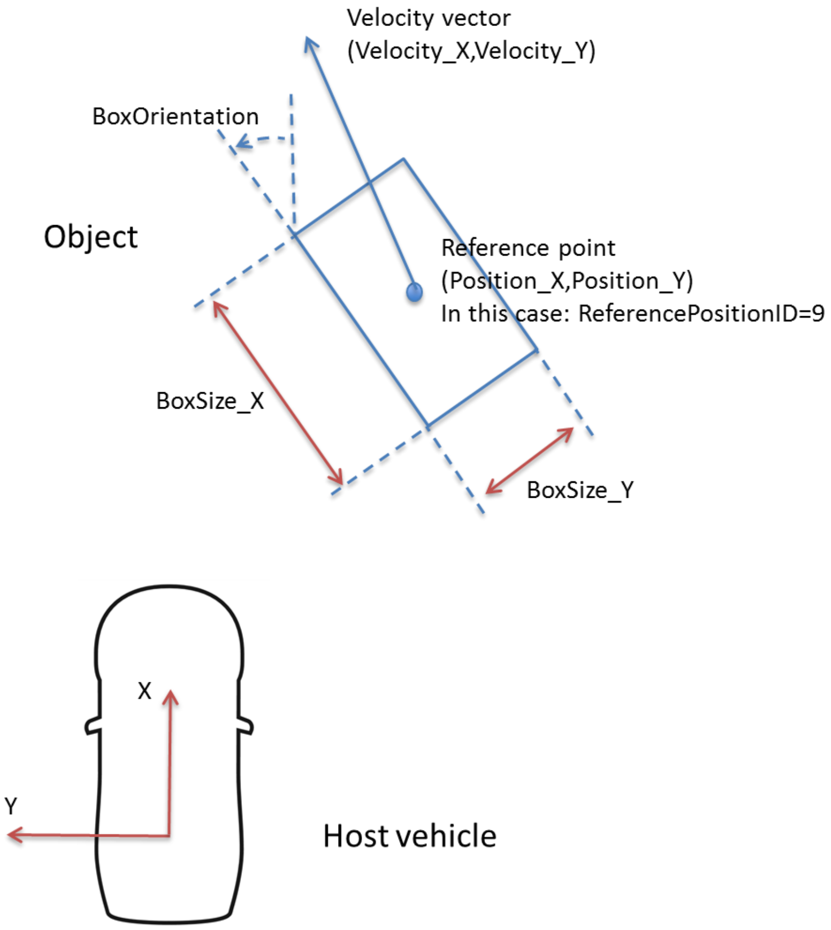
2.4 Ibeo data header

Each message always starts with an Ibeo data header. To resync just search for the "magic word".
The Ibeo data header is encoded in network byte order / big endian format.

Bytes	Offset	Ibeo data header	Data type	Description
4	0	Magic word (0xAFFEC0C2)	UINT32	The "magic word" is used for searching Ibeo messages and to distinguish between different versions.
4	4	Size of previous messages	UINT32	Helps to navigate backwards through a file. Unused in live data.
4	8	Size of this message	UINT32	Helps to read the message data. Size of message content without this header.
1	12	Reserved	UINT8	-
1	13	DeviceID	UINT8	ID of the connected device. Unused in data received directly from ibeo LUX sensors.
2	14	Data type	UINT16	Specifies the data type within this message.
8	16	NTP time	NTP64	Time when this message was created.
	24	Message data	-	Depending on data type.

2.5 Overview of coordinate frame and object properties

The object properties assume a coordinate system centered in the rear middle axle of the host vehicle, with the X-axis in the driving direction and Y-axis to the left (ISO 8855 / DIN 70000 coordinate system). The following figure illustrates the coordinate system and the properties for one example object


Velocity vector
(Velocity_X,Velocity_Y)
BoxOrientation
Object
Reference point
(Position_X,Position_Y)
In this case: ReferencePositionID=9
BoxSize_X
BoxSize_Y
X
Y
Host vehicle

3 ibeo LUX scan data: Data type 0x2202

Scan data available from ibeo LUX laserscanners (not available for ibeo LUX prototypes). Each scan data block starts with a header followed by the scan point list. The data is encoded in little endian format!

For angle information the unit angle ticks is used. An ibeo LUX typically uses 11520 ticks per rotation (see also Angle ticks per rotation below). Thus the angular resolution is 1/32°. This value is needed to convert angle ticks:

$$\text{angle} = 2\pi \frac{\text{angle ticks}}{\text{angle ticks per rotation}}$$

Angles are given in the ISO 8855 / DIN 70000 scanner coordinate system.

Bytes	Offset	Scan header:	Data type	Description
2	0	Scan number	UINT16	The number of this scan. The number will be increased from scan to scan.
2	2	Scanner status	bit field 16 bits	0x0001: motor on 0x0002: laser on 0x0004: internal feedback 0x0008: set frequency reached 0x0010: external sync signal detected 0x0020: sync ok 0x0040: sync master (instead of slave) 0x0080: reserved 0x0100: epw compensation on 0x0200: system compensation on 0x0400: start pulse compensation on 0x0800: reserved 0x1000: reserved 0x2000: reserved 0x4000: reserved 0x8000: upside down (Release FPGA Version 0x9604)
2	4	Sync phase offset	UINT16	Phase difference (conversion factor 409.6 ns) between sync signal and scanner mirror crossing the synchronization angle.
8	6	Scan start time NTP	NTP64	NTP time when the first/last measurement was done.
8	14	Scan end time NTP	NTP64	
2	22	Angle ticks per rotation	UINT16	Number of angle ticks per rotation.
2	24	Start angle	INT16	

2	26	End angle	INT16	Start/end angle in angle ticks of this scan.
2	28	Scan points	UINT16	Number of scan point transmitted in this scan.
2	30	Mounting position yaw angle	INT16	Rotation of the scanner around the axes of the reference coordinate system. All angles are given in angle ticks. Order of translation and rotation is essential: Yaw->Pitch->Roll->Translation. Scan data is given in the scanner coordinate system without any transformation.
2	32	Mounting position pitch angle	INT16	
2	34	Mounting position roll angle	INT16	
2	36	Mounting position x	INT16	Mounting position of the scanner relative to the reference coordinate system (ISO 8855 / DIN 70000 coordinate system). The origin is located on flat ground under the center of the rear axle. X-axis faces to the vehicle front resp. straight driving direction. Y-axis faces left. The mounting position is needed for ego motion compensation (only available if scanner x-y-plane is almost parallel to the ground). All coordinates are given in centimeters. Order of translation and rotation is essential (Rotation -> Translation). The mounting position is used for ego motion compensation, not to transform scan data but is available for further processing steps.
2	38	Mounting position y	INT16	
2	40	Mounting position z	INT16	
2	42	Flags	UINT16	Bit 0: ground labeled Bit 1: dirt labeled Bit 2: rain labeled Bit 3: reserved Bit 4..6: internal Bit 7..9: reserved Bit 10: mirror side (0=front, 1=rear)[2] Bit 11..15: reserved

[2] For use with 8-layer-scanners: 0 = front = mirror facing down, 1 = rear = mirror facing up

	44	Scan Point List	Scan Point	Array of scan points. See number of scan points above and point information below.

Bytes	Offset	Scan point:	Data type	Description
1	0	Layer	UINT4	Scan layer of this point (zero-based). Use the low nibble / bits 0…3 of this byte.
		Echo	UINT4	Echo number of this point (zero-based). Use the high nibble / bits 4…7 of this byte.
1	1	Flags	Bit field 8 bits	0x01: transparent point 0x02: clutter (atmospheric) 0x04: ground 0x08: dirt 0xF0: reserved
2	2	Horizontal angle	INT16	Angle of this point in angle ticks in the scanner coordinate system
2	4	Radial distance	UINT16	Distance of this point in the scanner coordinate system in cm
2	6	Echo pulse width	UINT16	Detected width of this echo pulse in cm
2	8	Reserved	UINT16	-
	10			

4 ibeo LUX object data: Data type 0x2221

Object data available from ibeo LUX laserscanners (not available for ibeo LUX prototypes).
Each data block starts with a header followed by the object list. Each object has a list of contour points.
The sigma values are calculated by Kalman filter by taking into account the object age.
The data is encoded in little endian format!

Bytes	Offset	Object header:	Data type	Description
8	0	Scan start timestamp	NTP64	Time stamp of the first measurement of the scan these objects are updated with.
2	8	Number of objects	UINT16	The number of objects transmitted in this message.
	10	List of objects	Object	Array of objects.

Bytes	Offset	Object: content	Data type	Description
2	0	Object ID	UINT16	ID of this object from tracking.
2	2	Object age	UINT16	Number of scans this object has been tracked for.
2	4	Object prediction age	UINT16	Number of scans this object has currently been predicted for without measurement update. Set to 0 as soon as a measurement update is available.
2	6	Relative timestamp	UINT16	Timestamp of this object relative to the scan start time in ms. The time is based on the object reference point.
4	8	Reference point	Point2D	Depending on tracking, this is the tracked object reference point (e.g. center of gravity) in cm. See below for Point2D.
4	12	Reference point sigma	Point2D	Standard deviation of the estimated reference point position in cm.
4	16	Closest point	Point2D	Unfiltered position of the closest object point in cm.
4	20	Bounding box center	Point2D	Center and size in cm of a rectangle in the reference coordinate system containing all object points. Width/length extend in Y-/X-coordinate.
2	24	Bounding box width	UINT16	
2	26	Bounding box length	UINT16	
4	28	Object box center	Point2D	Box center in the reference coordinate system in cm.
4	32	Object box size	Size2D	

2	36	Object box orientation	INT16	Box size in cm and orientation in 1/100° in the object coordinate system (box rotated by orientation in reference coordinate system).
4	38	Absolute velocity	Point2D	Velocity of this object in cm/s with ego motion taken into account. This velocity is based on the reference coordinate system which is compensated by the ego motion. Value set to 0x8000 if invalid.
4	42	Absolute velocity sigma	Size2D	Standard deviation of the estimated absolute velocity in cm/s.
4	46	Relative velocity	Point2D	Velocity of this object in cm/s without ego motion compensation (sensor/vehicle is seen as stationary).
2	50	Classification	UINT16	Most likely class of this object: 0: unclassified 1: unknown small 2: unknown big 3: pedestrian 4: bike 5: car 6: truck 7…: reserved
2	52	Classification age	UINT16	Number of scans this object has been classified as current class for.
2	54	Classification certainty	UINT16	The higher this value is the more reliable is the assigned object class.
2	56	Number of contour points	UINT16	The number of objects transmitted in this message.
	58	List of contour points	Point2D	Array of contour points in cm.

Bytes	Offset	Point2D:	Data type	Description
2	0	Position x	INT16	X-part/coordinate of this value/point.
2	2	Position y	INT16	Y-part/coordinate of this value/point.
	4			

Bytes	Offset	Size2D	Data type	Description
2	0	Size x	UINT16	X-value/size/width.
2	2	Size y	UINT16	Y-value/size/length.
	4			

5 ibeo LUX vehicle state: Data type 0x2805

The vehicle state is calculated by ibeoLUX from received CAN-Data. CAN data parsers need to be configured in order to receive valid vehicle state information.

All angles, position and distances are given in the ISO 8855 / DIN 70000 scanner coordinate system.

Bytes	Offset	Vehicle State:	Data type	Description
8	0	Timestamp	NTP64	
2	8	Scan number	UINT16	For synchronisation with Scan
2	10	Error flags	UINT16	0x0001: Axle dist parameter is not set, i.e. is set to zero. 0x0100: Measurement of steering wheel angle not up-to-date 0x0200: Measurement of front wheel angle not up-to-date or could not be calculated using SteeringWheelAngle 0x0800: No CAN-data received
2	12	Longitudinal velocity	INT16	Longitudinal Velocity [0.01m/s]
2	14	Steering wheel angle	INT16	Angle by which the steering wheel is rotated compared to its middle position. [0.001rad]
2	16	Front wheel angle	INT16	Wheel angle (calculated from steering wheel angle if available) [0.0001 rad]
2	18	Reserved		
4	20	X position	INT32	Distance from origin in X-Direction [0.01m]
4	24	Y position	INT32	Distance from origin in Y-Direction [0.01m]
2	28	Course angle	INT16	Orientation at time timestamp [0.0001 rad]
2	30	Time difference	UINT16	Time difference between this and last vehicle state message [ms]
2	32	X difference	INT16	Distance driven in X during time difference [0.001m]
2	34	Y difference	INT16	Distance driven in Y during time difference [0.001m]
2	36	Heading difference	INT16	Difference in Heading during time difference [0.0001 rad]
2	38	Reserved		
2	40	Current yaw rate	INT16	Yaw rate from latest CAN-Message received. Available since firmware version 2.5.00. [0.0001 rad/s]
4	42	Reserved		
	46			

Vehicle state information is to be considered as invalid if at least one error flag is set except 0x0100 or 0x0200, because the steering or front wheel angle is not necessary for vehicle state estimation. Warning W-EgoMotion of warning register 2 (see paragraph 6.4) is thrown if vehicle state information is invalid.

6 ibeo LUX errors and warnings - Data type 0x2030

As soon as an ibeo LUX Laserscanner detects an error or wants to emit a warning, this message is sent. Errors and warning bits are reset after sending this message.
This message will be sent periodically as long as errors of warnings persist.
The data is encoded in little endian format!

Bytes	Offset	LUX error/warning	Data type	Description
2	0	Error register 1	bit field 16 bits	See below
2	2	Error register 2	bit field 16 bits	
2	4	Warning register 1	bit field 16 bits	
2	6	Warning register 2	bit field 16 bits	
2	8	reserved	bit field 16 bits	
2	10	reserved	bit field 16 bits	
2	12	reserved	bit field 16 bits	
2	14	reserved	bit field 16 bits	

6.1 Error register 1

Bytes	LUX error	Description	Comment
Bit 0	E-SP	internal error	contact support
Bit 1	E-Motor_1	motor fault	contact support
Bit 2	E-Buffer_1	scan buffer transmitted incompletely	decrease scan resolution/frequency/range; contact support
Bit 3	E-Buffer_2	Scan buffer overflow	decrease scan resolution/frequency/range; contact support
Bit 4		reserved	
Bit 5		reserved	
Bit 6		reserved	
Bit 7		reserved	
Bit8…9	E-Temp	Bit 9: APD Over Temperature Bit 8: APD Under Temperature Bit 8 and 9: APD Temperature Sensor defect	provide cooling provide heating contact support
Bit 10	E-Motor_2	motor fault	contact support
Bit 11	E-Motor_3	motor fault	contact support
Bit 12	E-Motor_4	motor fault	contact support
Bit 13	E-Motor_5	motor fault	contact support
Bit14…15		reserved	

6.2 Error register 2

Bytes	LUX error	Description	Comment
Bit 0	E-IF_internal_1	no scan data received.	contact support
Bit 1	E-IF_internal_2	internal communication error	contact support
Bit 2	E-IF_internal_3	incorrect scan data	contact support
Bit 3	E-Configuration_1	FPGA not configurable	contact support
Bit 4	E-Configuartion_2	incorrect configuration data	load correct configuration values
Bit 5	E-Configuration_3	configuration contains incorrect parameters	load correct configuration values
Bit 6	E-Timeout_1	data processing timeout	decrease scan resolution or scan frequency
Bit 7	E-Timeout_2	reset the computation of the environmental model	contact support
Bit8…15	reserved		

6.3 Warning register 1

Bytes	LUX warning	Description	Comment
Bit 0	W-CMD	internal communication error	
Bit 1		reserved	
Bit 2		reserved	
Bit 3	W-low_temperature	temperature too low	warning of insufficient temperature
Bit 4	W-high_temperature	temperature too high	warning of exceeding temperature
Bit 5	W-Motor_1	internal warning	
Bit 6		reserved	
Bit 7	W-Sync	syncronisation error	check syncronisation- and scan frequency
Bit 8…11		reserved	
Bit 12	W-SP_1	start pulse missing (laser 1)	(Release FPGA Version 0x9604)
Bit 13	W-SP_2	start pulse missing (laser 2)	(Release FPGA Version 0x9604)
Bit 14…15		reserved	

6.4 Warning register 2

Bytes	LUX warning	Description	Comment
Bit0	W-IF_CAN	CAN Interface blocked	check CAN bus and CAN connection
Bit1	E-IF_ETH	Ethernet Interface blocked	check Ethernet connection
Bit2	W-CANdata	incorrect CAN message received	check CAN data
Bit3	W-IF_internal_1	incorrect scan data	contact support
Bit4	W-ETHdata	unknown or incomplete data	check Ethernet data
Bit5	W-Command	incorrect or forbidden command received	check command
Bit6	W-Flash	memory access failure	restart ibeo LUX, contact support
Bit7	W-Overflow_1	internal overflow	contact support
Bit8	W-EgoMotion	vehicle data update missing	check CAN vehicle data
Bit9	W-Mounting Position	incorrect mounting parameters	correct mounting position according to OM
Bit10	W-CalcFrequency	no object computation due to scan frequency.	set the scan frequency to 12.5 Hz to receive objects
Bit11…15	reserved		

7 ibeo LUX command interface

For sending commands to the ibeo LUX the data type 0x2010 is used. The data is encoded in little endian format!

Bytes	Offset	LUX command	Data type	Description
2	0	Command ID	UINT16	See detailed list of commands and according options/parameters.
2	2	Reserved	UINT16	Unused, but these 2 bytes must be sent for all commands.
	4	Command Data	-	Depending on command. May be completely missing for some commands.

The ibeo LUX replies to a command with a dedicated reply message. The data type used is 0x2020. The data is encoded in little endian format!

Bytes	Offset	LUX reply:	Data type	Description
2	0	Reply ID	UINT16	If a command succeeded, the reply ID is equal to the corresponding command ID. If a command failed, the reply ID is the command ID + 0x8000. Thus, the most significant bit indicates a failed command.
	2	Reply data	-	Depending on the corresponding command this reply is related to. May be completely missing for some commands and if a command failed. See detailed command description below.

7.1 ibeo LUX commands and command replies – data types 0x2010/ 0x2020

7.1.1 Reset

Bytes	Offset	LUX command	Data type	Description
2	0	0x0000	UINT16	ID - Reset DSP
2	2	Reserved	UINT16	-

In case of command Reset no reply is sent.

7.1.2 Get Status

Bytes	Offset	LUX command	Data type	Description
2	0	0x0001	UINT16	Status request
2	2	Reserved	UINT16	-

Bytes	Offset	LUX reply	Data type	Description
2	0	0x0001	UINT16	Status request
2	2	Firmware version	UINT16	e. g. 0x1230 = version 1.2.3, 0x123B = version 1.2.3b
2	4	FPGA version	UINT16	e. g. 0x1230 = version 1.2.3, 0x123B = version 1.2.3b
2	6	Scanner status	UINT16	Bit field, with the following meaning for every bit: 0x0001: motor on 0x0002: laser on 0x0004: internal feedback 0x0008: set frequency reached 0x0010: external sync signal detected 0x0020: sync ok 0x0040: sync master (instead of slave) 0x0080: reserved 0x0100: epw compensation on 0x0200: system compensation on 0x0400: start pulse compensation on 0x0800: reserved 0x1000: reserved 0x2000: reserved 0x4000: reserved 0x8000: upside down (Release FPGA Version 0x9604)
4	8		UINT32	reserved / internal
2	12	temperature	UINT16	T[°C] = - (temperature - 579.2364) / 3.63
2	14	serial number 0	UINT16	YYCW (z. B. YYCW = 0x0740 = year '07, calendar week 40)
2	16	serial number 1	UINT16	Counter of serial number
2	18		UINT16	reserved / internal
6	20	FPGA time stamp	[3] * UINT16	YYYY MMDD hhmm (FPGA version state decimal coded)
6	26	DSP time stamp	[3] * UINT16	YYYY MMDD hhmm (Firmware version state decimal coded)

7.1.3 SaveConfig

Bytes	Offset	LUX command	Data type	Description
2	0	0x0004	UINT16	Current sensor configuration will be saved permanently. Multiple SetParameter

				commands may be sent before saving the changes permanently.
2	2	Reserved	UINT16	-

The command SaveConfig will be acknowledged by the same command ID without command reply data.

7.1.4 Set Parameter

Bytes	Offset	LUX command	Data type	Description
2	0	0x0010	UINT16	Set a single Parameter by its index to the sensor memory. Parameter is set only temporarily until a SaveConfig command (see 7.1.3) is sent.
2	2	Reserved	UINT16	-
2	4	Parameter index	UINT16	Refer to ibeo LUX parameter list (see 7.2)
4	6	Parameter	UINT32	Set parameter accordingly to parameter list. If e.g. a 2 byte value is set, use the first 2 bytes. Fill the remaining 2 bytes with 0.

The command Set Parameter will be acknowledged by the same command ID without any command reply data.

7.1.5 Get Parameter

Bytes	Offset	LUX command	Data type	Description
2	0	0x0011	UINT16	Read a single Parameter with its index from the LUX.
2	2	Reserved	UINT16	-
2	4	Parameter index	UINT16	Refer to LUX parameter list (see 7.2)

Bytes	Offset	LUX reply	Data type	Description
2	0	0x0011	UINT16	Read a single Parameter by its index from the LUX.
2	2	Parameter index	UINT16	Refer to ibeo LUX parameter list (see 7.2)
4	4	Parameter	UINT32	

7.1.6 Reset Default Parameters

Bytes	Offset	LUX command	Data type	Description

2	0	0x001A	UINT16	Resets all parameters to the factory defaults.
2	2	Reserved	UINT16	-

The command Reset Default Parameters will be acknowledged by the same command ID without any command reply data.
Send SaveConfig command (see 7.1.3) to reset default parameters permanently after this command.

7.1.7 Start Measure

Bytes	Offset	LUX command	Data type	Description
2	0	0x0020	UINT16	Starts the measurement with the current settings.
2	2	Reserved	UINT16	-

The command Start Measure will be acknowledged by the same command ID without any command reply data.

7.1.8 Stop Measure

Bytes	Offset	LUX command	Data type	Description
2	0	0x0021	UINT16	Stops the measurement.
2	2	Reserved	UINT16	-

The command Stop Measure will be acknowledged by the same command ID without any command reply data.

7.1.9 SetNTPTimestampSync

Bytes	Offset	LUX command	Data type	Description
2	0	0x0034	UINT16	sets the second of NTPtimestamp.
2	2	Reserved	UINT16	-
2	4	Reserved	UNIT16	-
4	6	Seconds	UINT32	Seconds (NTP format).
4	6	Fractional seconds	UINT32	Fractional Seconds (NTP format).

The command SetNTPTimestampSync will be acknowledged by the same command ID without any command reply data.

7.2 ibeo LUX parameter list

This table gives an overview of available ibeo LUX parameters. Please refer to 7.1.4 and 7.1.5 for details on getting and setting these parameters.

IP address, subnet mask and standard gateway encode the data as UINT32 value, which is built like that: aa.bb.cc.dd = 0xaabbccdd. Due to little endian byte order, this value must be sent as 0xddccbbaa.

A change of following parameters only take effect after a sensor restart (cycle power or use command reset):

- IP address	(ID 0x1000)
- TCP Port	(ID 0x1001)
- Subnet Mask	(ID 0x1002)
- Standard gateway	(ID 0x1003)
- CAN Baud Rate	(ID 0x1011)
- Interface Flags	(ID 0x1019)

Bytes	Parameter index	LUX parameter	Data type	Description
4	0x1000	IP address	UINT32	Valid: all
2	0x1001	TCP Port	UINT16	Valid: all
4	0x1002	Subnet Mask	UINT32	Valid: all
4	0x1003	Standard gateway	UINT32	Valid: all
2	0x1004	Customer Processing Switch 0	UINT16	bit true: process false: do not process. bit0: dirt + rain detection. bit3/bit2: timeSyncMode: - 00: NMEA over RS232 - 01: ETH - 10: CAN - 11: AUTO
4	0x1010	CAN Base ID	UINT32	Valid: value <= 0x7F0
2	0x1011	CAN Baud Rate	UINT16	in kBaud - next matching value (1000 kBaud, 500 kBaud, 250 kBaud, 125 kBaud) will be used.
2	0x1012	Data Output Flag	16 bit field	Bit true: disable output, false: enable output. 0xFFFF is invalid. bit0: ETH scan data bit1: reserved/internal bit2: ETH object data bit3: ETH vehicle data bit4: ETH errors/warnings bit5: CAN errors/warnings bit6: CAN object data bit7...15: reserved/internal

Bytes	Parameter index	LUX parameter	Data type	Description
2	0x1013	maxObjectsViaCAN	UINT16	<= 65 (max. number of objects) limited by tracking and CAN bus capacity.
2	0x1014	ContourPointDensity	UINT16	Valid: < 3 0: closest point only 1: low density 2: high density
2	0x1015	ObjectPriorizationCriterion	UINT16	Valid: < 2 Used to reduce transmitted objects via CAN. Decision which objects are discarded is based on this criterion. 0: Radial 1: Look ahead
2	0x1016	CAN object data options	16 bit field	Valid: all bit 0: 0 = absolute velocities, 1 = relative velocities bit 1: 0 = boxes are object boxes, 1 = boxes are bounding boxes bits 2…15: reserved
2	0x1017	Minimum Object Age	UINT16	Valid: all Minimum tracking age (number of scans) of an object to be transmitted.
2	0x1018	Maximum Prediction Age	UINT16	Valid: all Maximum prediction age (number of scans) of an object to be transmitted.
2	0x1019	Interface Flags	UINT16	RS232 Baud rate: 0 = default (currently 57600 Baud) 1 = 2400 Baud 2 = 4800 Baud 3 = 9600 Baud 4 = 19200 Baud 5 = 38400 Baud 6 = 57600 Baud 7 = 115200 Baud 8 = 921600 Baud 9 = 6250000 Baud >9 = default
2	0x1100	Start angle	INT16	In 1/32°, in the sensor coordinate system. Valid: 1600…-1919. Start angle > end angle!

Bytes	Parameter index	LUX parameter	Data type	Description
2	0x1101	End angle	INT16	In 1/32°, in the sensor coordinate system. Valid: 1599…-1920. Start angle > end angle!
2	0x1102	Scan frequency	UINT16	In 1/256 Hz. Valid: 3200 (12.5 Hz) 6400 (25.0 Hz) 12800 (50.0 Hz)
2	0x1103	Sync angle offset	INT14 (!) 16 bits used	In 1/32° in the sensor coordinate system. Valid: -5760…+5759 (-180°…+180°). Bits 14 and 15 are ignored!
2	0x1104	angular resolution type	UINT16	0: focused 1: constant 2: reserved
2	0x1105	angleTicksPerRotation	UINT16	11520 (read only), constant for ibeo LUX
2	0x1200	SensorMounting_X	INT16	In cm, related to vehicle reference point, rear axle. Order of translation and rotation is essential (Rotation -> Translation).
2	0x1201	SensorMounting_Y	INT16	In cm, related to vehicle reference point, rear axle. Order of translation and rotation is essential (Rotation -> Translation).
2	0x1202	SensorMounting_Z	INT16	In cm, related to vehicle reference point, rear axle. Order of translation and rotation is essential (Rotation -> Translation).
2	0x1203	SensorMounting_Yaw	INT16	In 1/32°, order of translation and rotation is essential (Yaw->Pitch->Roll-> Translation).
2	0x1204	SensorMounting_Pitch	INT16	In 1/32°, order of translation and rotation is essential (Yaw->Pitch->Roll-> Translation).
2	0x1205	SensorMounting_Roll	INT16	In 1/32°, order of translation and rotation is essential (Yaw->Pitch->Roll-> Translation).
2	0x1206	VehicleFrontToFrontAxle	UINT16	valid: all; in cm
2	0x1207	FrontAxleToRearAxle	UINT16	valid: all; in cm

Bytes	Parameter index	LUX parameter	Data type	Description
2	0x1208	RearAxleToVehicleRear	UINT16	valid: all; in cm
2	0x1209	VehicleWidth	UINT16	valid: all; in cm
2	0x120A	steerRatioType	UINT16	0: Transmission ratio Front wheel angle = $\frac{x}{1.095(s_3x^3 + s_2x^2 + s_1x + s_0)}$ (right before calculation a transform from rad to deg is done, so the coefficients need to be adjusted for calculation with degrees, this option is deprecated and will be removed soon, make sure there is no division by zero, x = steering wheel angle in degree(it is converted from deg to rad internally)) 1: Transfer function Front wheel angle = $s_3x^3 + s_2x^2 + s_1x + s_0$ (this will be the only transmission function in the future) x = steering wheel angle in radian)
4	0x120C	SteerRatioPoly0 (s0)	Float32	valid: all
4	0x120D	SteerRatioPoly1 (s1)	Float32	valid: all
4	0x120E	SteerRatioPoly2 (s2)	Float32	valid: all
4	0x120F	SteerRatioPoly3 (s3)	Float32	valid: all
2	0x1210	Vehicle Motion Data Flags	16 bit field	Bit 0: Vehicle Motion data expected: 1=true, 0=false Bits 1 to 15: reserved

Bytes	Parameter index	LUX parameter	Data type	Description
2	0x3301	DeviceType	UINT16	TypeIbeoLUX3 0x06 TypeIbeoLUX4 0x10 TypeIbeoLUX4HD 0x11 TypeIbeoLUX4_8Lines 0x18 TypeIbeoLUX4HD_8Lines 0x19

7.3 Example

This example shows how to set the IP address via Ethernet 192.168.0.200.

Bytes	Offset	Ibeo data header Big endian byte order!	Data type	Content
4	0	Magic word	UINT32	0xAFFEC0C2
4	4	Size of previous message	UINT32	Not mandatory. Set e.g. to 0: 0x00000000
4	8	Size of this message	UINT32	0x000000XX
1	12	Reserved	UINT8	0x00
1	13	Device ID	UINT8	Not mandatory. Set e.g. to 7: 0x07
2	14	Data type: ibeo LUX command	UINT16	0x2010
8	16	NTP timestamp	UINT64	Not mandatory. Set e.g. to 0: 0x0000000000000000
Bytes	Offset	Message data Little endian byte order!	Data type	Content
2	24	Command ID: Set parameter	UINT16	0x0010 (send encoded as 0x1000)
2	26	Reserved	UINT16	0x0000
2	28	Parameter index: IP address	UINT16	0x1000 (send encoded as 0x0010)
4	30	Parameter data (here: 192.168.0.200)	UINT32	0xC0A800C8 (send encoded as 0xC800A8C0)
	34			

8 Ibeo FUSION SYSTEM/ECU scan data - Data type 0x2204

Scan data available from Ibeo FUSION SYSTEM and Ibeo AppBase2 (ECU) are sent as data type 0x2204 until FUSION SYSTEM version 2.1 and Ibeo Laserview 1.5. Please see data type 0x2205 for later versions.
Each scan data block starts with a header followed by the scanner info list and the scan point list. Each scan point has a device ID which refers to a sensor in the sensor info list. The data is encoded in network byte order / big endian format.

Bytes	Offset	Scan header	Data type	Description
8	0	Scan start time	NTP64	NTP time when the first measurement was done.
4	8	Scan end time offset	UINT32	Time difference between last and first measurement in us.
4	12	Flags	Bit field: 32 bits	Bit 0: ground labeled Bit 1: dirt labeled Bit 2: rain labeled Bits 3…8: reserved Bit 9: fused scan Bit 10: mirror side (0 = front, 1 = rear) Bit 11: coordinate system (0 = scanner coordinates, 1 = vehicle coordinates)
2	16	Scan number	UINT16	The number of this scan. The number will be increased from scan to scan. Overflow occurs after 2^{16} scans.
2	18	Scan points	UINT16	Number of scan points transmitted in this scan.
1	20	Number of scanner infos	UINT8	Number of scanner infos transmitted in this scan.
3	21	Reserved	3 bytes	-
	24	Scanner info list	Scanner info	Array of scanner infos. See number of scanner infos above and scanner info below.
	24 + scanner infos * 40	Scan point List	Scan point	Array of scan points. See number of scan points above and point information below.
	24 + scanner infos * 40 + scan points * 28			

Bytes	Offset	Scanner info	Data type	Description
1	0	Device ID	UINT8	Device ID of this scanner.
1	1	Scanner type	UINT8	3 = Alasca XT 4 = ECU 5 = ibeo LUX prototype 6 = ibeo LUX
2	2	Scan number	UINT16	The scan number coming from the scanner device. The number will be increased from scan to scan. Overflow occurs after 2^{16} scans.
4	4	Reserved	4 bytes	-
4	8	Start angle	FLOAT32	Field of view of this scanner given in its local coordinate system. In radians normalized to [−π, +π[.
4	12	End angle	FLOAT32	
4	16	Yaw angle	FLOAT32	Mounting angles relative to vehicle coordinate system. In radians normalized to [−π, +π[.
4	20	Pitch angle	FLOAT32	
4	24	Roll angle	FLOAT32	
4	28	Offset x	FLOAT32	Mounting position relative to vehicle coordinate system. In meters.
4	32	Offset y	FLOAT32	
4	36	Offset z	FLOAT32	
	40			

Bytes	Offset	Scan point:	Data type	Description
4	0	X position	FLOAT32	X position of this scan point in m.
4	4	Y position	FLOAT32	Y position of this scan point in m.
4	8	Z position	FLOAT32	Z position of this scan point in m.
4	12	Echo width	FLOAT32	Echo width of this scan point in m.
1	16	Device ID	UINT8	ID of the device measuring this point.
1	17	Layer	UINT8	Scan layer of this point (zero-based).
1	18	Echo	UINT8	Echo number of this point (zero-based).
1	19	Reserved	1 byte	-
4	20	Timestamp (µs)	UINT32	Time offset in µs when this scan point was measured based on the scan start time.
2	24	Flags	Bit field: 16 bits	0x0001: ground 0x0002: dirt 0x0004: rain/snow/spray/fog/… 0xFFF8: reserved
2	26	Reserved	2 bytes	-
	28			

9 Ibeo FUSION SYSTEM/ECU scan data - Data type 0x2205

Scan data available from Ibeo FUSION SYSTEM and Ibeo AppBase2 (ECU) is sent as data type 0x2205 using FUSION SYSTEM version 2.2 and later and Ibeo Laserview 1.6 and later. Please see data type 0x2204 for earlier versions.
Each scan data block starts with a header followed by the scanner info list and the scan point list. Each scan point has a device ID that refers to a sensor in the sensor info list.
The data is encoded in network byte order / big endian format.

Bytes	Offset	Scan header	Data type	Description
8	0	Scan start time	NTP64	NTP time when the first measurement was done.
4	8	Scan end time offset	UINT32	Time difference between last and first measurement in us.
4	12	Flags	Bit field: 32 bits	Bit 0: ground labeled Bit 1: dirt labeled Bit 2: rain labeled Bits 3…8: reserved Bit 9: fused scan Bit 10: mirror side (0 = front, 1 = rear) Bit 11: coordinate system (0 = scanner coordinates, 1 = vehicle coordinates)
2	16	Scan number	UINT16	The number of this scan. The number will be increased from scan to scan. Overflow occurs after 2^{16} scans.
2	18	Scan points	UINT16	Number of scan points transmitted in this scan.
1	20	Number of scanner infos	UINT8	Number of scanner infos transmitted in this scan.
3	21	Reserved	3 bytes	-
number of scanner infos * 148	24	Scanner info list	Scanner info	Array of scanner infos. See number of scanner infos above and scanner info below.
number of scan points * 28	24 + number of scanner infos * 148	Scan point List	Scan point	Array of scan points. See number of scan points above and point information below.
	24 + scanner infos * 148 + number of scan points * 28			

Bytes	Offset	Scanner info	Data type	Description
1	0	Device ID	UINT8	Device ID of this scanner.
1	1	Scanner type	UINT8	3 = Alasca XT 4 = ECU 5 = ibeo LUX prototype 6 = ibeo LUX
2	2	Scan number	UINT16	The scan number coming from the scanner device. The number will be increased from scan to scan. Overflow occurs after 2^{16} scans.
4	4	Reserved	4 bytes	-
4	8	Start angle	FLOAT32	Field of view of this scanner given in its local coordinate system. In radians normalized to [−π, +π[.
4	12	End angle	FLOAT32	
8	16	Scan start time	NTP64	NTP time (based on computer time on which the Ibeo software runs) when the first measurement of this scanner was done.
8	24	Scan end time	NTP64	NTP time (based on computer time on which the Ibeo software runs)when the last measurement of this scanner was done.
8	32	Scan start time from device	NTP64	NTP time (as received from the sensor) when the first measurement of this scanner was done.
8	40	Scan end time from device	NTP64	NTP time (as received from the sensor) when the first measurement of this scanner was done.
4	48	Scan frequency	FLOAT32	Scan frequency of this scanner in Hz.
4	52	Beam tilt	FLOAT32	Angle the scanner measurement is pitched relatively to sensor x-y plane. This value is valid for measuring in x-direction resp. 0° in the scanner coordinate system. In radians normalized to [−π, +π]. Beam is pitched downwards if values are positive and vice versa.

4	56	Scan flags	Bit field: 32 bits	Bit 0: Ground detection was performed Bit 1: Dirt detection was performed Bit 2: Clutter detection was performed Bits 3…8: Reserved Bit 9: Scan is result from scan data fusion Bit 10: Mirror side Bits 11..31: Reserved
4	60	Yaw angle	FLOAT32	Mounting angles relative to vehicle coordinate system. In radians normalized to [−π, +π].
4	64	Pitch angle	FLOAT32	
4	68	Roll angle	FLOAT32	
4	72	Offset x	FLOAT32	Mounting position relative to vehicle coordinate system. In meters.
4	76	Offset y	FLOAT32	
4	80	Offset z	FLOAT32	
8	84	Resolution 1	Resolution Info	Scan resolution for different sectors of the scanner field of view. Resolutions can be the same for all sectors (constant angular resolution) or different (e.g. focused angular resolution). Please see resolution info description below.
8	92	Resolution 2	Resolution Info	
8	100	Resolution 3	Resolution Info	
8	108	Resolution 4	Resolution Info	
8	116	Resolution 5	Resolution Info	
8	124	Resolution 6	Resolution Info	
8	132	Resolution 7	Resolution Info	
8	140	Resolution 8	Resolution Info	
	148			

Bytes	Offset	Resolution Info:	Data type	Description
4	0	Resolution start angle	FLOAT32	Starting from this angle the given resolution is valid until the next resolution start angle or the scan end. In radians normalized to [−π, +π[. Valid only if resolution value is > 0.
4	4	Resolution	FLOAT32	Resolution for this sector. In radians normalized to [−π, +π[. Valid only if > 0.
	8			

Bytes	Offset	Scan point:	Data type	Description
4	0	X position	FLOAT32	X position of this scan point in m.
4	4	Y position	FLOAT32	Y position of this scan point in m.
4	8	Z position	FLOAT32	Z position of this scan point in m.
4	12	Echo width	FLOAT32	Echo width of this scan point in m.
1	16	Device ID	UINT8	ID of the device measuring this point.
1	17	Layer	UINT8	Scan layer of this point (zero-based).
1	18	Echo	UINT8	Echo number of this point (zero-based).
1	19	Reserved	1 byte	-
4	20	Timestamp (µs)	UINT32	Time offset in µs when this scan point was measured based on the scan start time.
2	24	Flags	Bit field: 16 bits	0x0001: ground 0x0002: dirt 0x0004: rain/snow/spray/fog/… 0x0008: flag road marking 0x0080: flag curbstone 0x0088: flag guard rail 0x1000: transparent Else : reserved
2	26	Reserved	2 bytes	-
	28			

10 Ibeo FUSION SYSTEM/ECU object data - Data type 0x2225

Object data available from Ibeo FUSION SYSTEM and Ibeo AppBase2 (ECU). Each data block starts with a header followed by the object list. Each object has a list of contour points. The sigma values are calculated using a Kalman filter by taking into account the object age. All positions and angles are given in the vehicle / reference coordinate system. Data is encoded in network byte order / big endian format.

Bytes	Offset	Object header	Data type	Description
8	0	Mid-scan timestamp	NTP64	Mid-scan timestamp is the absolute timestamp when the scanner mirror crossed the middle of the corresponding scan. Used for synchronization purpose.
2	8	Number of objects	UINT16	The number of objects transmitted in this message.
	10	List of objects	Object	Array of objects.

Bytes	Offset	Object	Data type	Description
2	0	Object ID	UINT16	ID of this object from tracking.
2	2	Reserved	UINT16	-
4	4	Object age	UINT32	Number of scans this object has been tracked for.
8	8	Timestamp NTP	NTP64	Time when this object was observed. More precisely: the reference point of this object.
2	16	Object hidden status age	UINT16	Number of scans this object has only been predicted without measurement updates.
1	18	Classification	UINT8	Most likely class of this object: 0: unclassified 1: unknown small 2: unknown big 3: pedestrian 4: bike 5: car 6: truck 7…: reserved
1	19	Classification certainty	UINT8	The higher this value is the more reliable is the assigned object class.
4	20	Classification age	UINT32	Number of scans this object has been classified as current class.
8	24	Bounding box center	Point2D	Center point of the bounding box of this object. See below for definition of Point2D.

8	32	Bounding box size	Point2D	Size of the bounding box (a rectangle parallel to vehicle coordinate system).
8	40	Object box center	Point2D	Center point (tracked) of this object.
8	48	Object box center sigma	Point2D	Standard deviation of the object box center point.
8	56	Object box size	Point2D	Size of the object box in the object coordinate system (vehicle coordinate system rotated around z axis by object course angle).
8	64	Reserved	8 bytes	-
4	72	Yaw angle	FLOAT32	Orientation or heading of the object in radians.
4	76	Reserved	4 bytes	-
8	80	Relative velocity	Point2D	Velocity of this object in m/s relative to the ego vehicle. Ego motions are not taken into account here.
8	88	Relative velocity sigma	Point2D	Standard deviation of the relative velocity.
8	96	Absolute velocity	Point2D	Velocity of this object in m/s with ego motion taken into account. Inform about the object velocity in the ‘real world’.
8	104	Absolute velocity sigma	Point2D	Standard deviation of the absolute velocity.
18	112	Reserved	18 bytes	-
1	130	Number of contour points	UINT8	Number of contour points transmitted for this object.
1	131	Index of closest point	UINT8	Closes contour point of this object as index of the point list.
	132	List of contour points	Point2D	Array of contour points (Point2D) in m.

Bytes	Offset	Point2D	Data type	Description
4	0	Position x	FLOAT32	X-part/coordinate of this value/point.
4	4	Position y	FLOAT32	Y-part/coordinate of this value/point.
	8			

11 Ibeo FUSION SYSTEM/ECU object data - Data type 0x2280

Object data available from Ibeo FUSION SYSTEM and Ibeo ECU connected with Ibeo Laserscanners. Each data block starts with the IbeoDataHeader followed by the object list. For each object list this header is preceded.

Each object has a list of contour points. Subtypes are described below.

All positions and angles are given in the vehicle / reference coordinate system. Data is encoded in network byte order / big endian format.

In general, positions, lengths, distances and sizes are coded in meters. In general, angles are coded in radians.

Note: depending on the configuration of the fusion system, the ECU can provide object data of type 0x2281. Please refer to Section 12 for detail information about this data type.

ECU Object Data List:

Offset	Bytes	Object header	Data type	Description
0	8	Mid-scan timestamp	NTP64	Mid-scan timestamp is the absolute timestamp when the scanner mirror crossed the middle of the corresponding scan. Avg between in start and end scan time stamp. Not set! from 5.7.5
8	2	Number of objects	UINT16	The number of objects transmitted in this message.
10		List of objects	Object	Array of objects.

Offset	Bytes	Object	Data type	Description
0	2	Object ID	UINT16	ID of this object from tracking.
2	2	Flags	UINT16	Bit 6: 0 = tracked by dyn. model 1 = tracked by static model Bit 7: 0 = mobility of dynamic obj. not (yet) detected 1 = mobility of dynamic obj. successfully detected Bit 8: 0 = motion model not validated 1 = motion model validated For a detailed description of these flags and combinations please see Section 11.1

4	4	Object age	UINT32	Number of scans this object has been tracked for.
8	8	Timestamp NTP	NTP64	Timestamp of the last measurement (COG of Segment) that was used for updating this object.
16	2	Object Prediction Age	UINT16	Number of update cycles that this object has only been predicted without measurement updates.
18	1	Classification	UINT8	Most likely class of this object: 0: unclassified 1: unknown small 2: unknown big 3: pedestrian 4: bike 5: car 6: truck 7…: reserved 12: underdrivable
19	1	Classification Quality	UINT8	The higher this value is the more reliable is the assigned object class. Range: 0..100
20	4	Classification age	UINT32	Time that this object has been classified as current class in ms.
24	8	Reserved	Point2DFloat	
32	8	Reserved	Point2DFloat	
40	8	Object box center	Point2DFloat	Center point of this object box.
48	8	Object box center sigma	Point2DFloat	Standard deviation of the object box center point. **Not available yet!**
56	8	Object box size	Point2DFloat	Size of the object box in the object coordinate system (vehicle coordinate system rotated around z axis by object course angle). X value corresponds to the object length, Y value corresponds to the object width.
64	8	Reserved	8 bytes	-
72	4	Object course angle	FLOAT32	Orientation or heading of the object box [rad].
76	4	Object Course Angle Sigma	Float32	Uncertainty (standard deviation) of the course angle.
80	8	Relative velocity	Point2DFloat	Velocity of this object in m/s relative to the ego vehicle in the ego vehicle coordinate system.

88	8	Relative velocity sigma	Point2DFloat	Standard deviation of the relative velocity.
96	8	Absolute velocity	Point2DFloat	Absolute velocity of this object in m/s.
104	8	Absolute velocity sigma	Point2DFloat	Standard deviation of the absolute velocity.
112	18	Reserved	18 bytes	-
130	1	Number of contour points	UINT8	Number of contour points transmitted for this object.
131	1	Index of closest point	UINT8	Closest contour point of this object as index of the point list. The IDs of the list are zero-based.
132	2	Reference point location	UINT16	The reference point can be located at the following points: 0: Center of gravity 1: Front/Left 2: Front/Right 3: Rear/Right 4: Rear/Left 5: Front/Center 6: Right/Center 7: Rear/Center 8: Left/Center 9: Object Center *0xFF: unknown*
134	8	Reference point coordinate	Point2DFloat	Depending on tracking this is the tracked object reference point, i.e position of reference point in m.
142	8	Reference point coordinate sigma	Point2DFloat	Standard deviation of the estimated reference point position in m.
150	4	Reference point position correlation coefficient	FLOAT32	Pearson's product-moment coefficient. **Not available yet!**
154	8	Reserved	8 bytes	
162	2	Object priority	UINT16	Value determining priority of the object. The higher the number, the higher the object priority. Priority is based on (1) motion classification and (2) distance.
164	4	Object existence measurement	Float32	**Not available yet!**
170		List of contour points	Point2DFloat	Array of contour points

Bytes	Offset	Point2DFloat	Data type	Description
4	0	Position x	Float32	X-part/coordinate of this value/point.
4	4	Position y	Float32	Y-part/coordinate of this value/point.
	8			

11.1 Definition of object flags

Valid	Mobile detected	Static. Model/ Dynamic model	Description	Customer IF relevant	Put out for evaluation
0	0	0	Unvalidated object hypotheses	no	yes
0	0	1		no	yes
0	1	0	n.a.	No	No
0	1	1		No	No
1	0	0		no	no
1	**0**	**1**	**Validated stationary object ("a priori stationary")**	**Yes**	**Yes**
1	**1**	**0**	**validated dynamic object with validated track ("moving")**	**Yes**	**Yes**
1	**1**	**1**	**validated object, dynamic before, which is now stationary ("stopped")**	**Yes**	**Yes**

12 Ibeo FUSION SYSTEM/ECU object data - Data type 0x2281

Object data are available from the Ibeo FUSION SYSTEM and the Ibeo ECU connected with Ibeo laserscanners. Each data block starts with the IbeoDataHeader followed by the object list.
The IbeoDataHeader precedes each object list. The IbeoDataHeader is described in Section 2.4 Ibeo Data Header.

All positions and angles are given in the vehicle / reference coordinate system.
Data is encoded in network byte order / big endian format.

In general, positions, lengths, distances and sizes are coded in meters. In general, angles are coded in radians.

ECU Object Data List:

Offset	Bytes	Object header	Data type	Description
0	8	Mid-scan timestamp	NTP64	Mid-scan timestamp is the absolute timestamp when the scanner mirror crossed the middle of the corresponding scan. Avg between in start and end scan time stamp. Not set! from 5.7.5
8	1	Object List ID	UINT8	Unique object list identifier to match object list with its source of computation. Not used. Default 255.
9	1	Device Type	UNIT8	Device Type that created this object list e.g. laserscanner, radar, camera. 0: unknown 1: Laserscanner 2: ECU 3: CAN bus 4: Camera 5: GPS 99: other **Not set yet! This should be an enumeration.**
10	2	Device Interface Version	UINT16	Device Interface Version of Software that creates this object list. Only 14 Bits can be used (0-16383) **Not set yet! This should be an enumeration.**
12	1	Flags	UINT8	Bit 4: 0 = online object list 1 = post processed object list

13	1	Reserved	-	-
14	2	Number of objects	UINT16	The number of objects transmitted in this message.
16		List of objects	Object	Array of objects.

Offset	Bytes	Object	Data type	Description
0	4	Object ID	UINT32	ID of this object from tracking.
4	2	Flags	UINT16	Bit 6: 0 = tracked by dyn. model 1 = tracked by static model Bit 7: 0 = mobility of dynamic obj. not (yet) detected 1 = mobility of dynamic obj. successfully detected Bit 8: 0 = motion model not validated 1 = motion model validated For a detailed description of these flags and combinations please see Section 11.1
6	4	Object Age	UINT32	Number of scans this object has been tracked for.
10	8	Timestamp NTP	NTP64	Timestamp of the last measurement (COG of Segment) that was used for updating this object.
18	2	Object Prediction Age	UINT16	Number of update cycles that this object has only been predicted without measurement updates.
20	1	Classification	UINT8	Most likely class of this object: 0: unclassified 1: unknown small 2: unknown big 3: pedestrian 4: bike 5: car 6: truck 7…: reserved 12: underdrivable
21	1	Classification Quality	UINT8	The higher this value is the more reliable is the assigned object class. Range: 0..100

22	4	Classification Age	UINT32	Time that this object has been classified as current class in ms.
26	8	Object Box Size	Point2DFloat	Size of the object box in the object coordinate system (vehicle coordinate system rotated around z axis by object course angle). X value corresponds to the object length, Y value correcsponds to the object width.
34	8	Object Box Size Sigma	Point2DFloat	Standard deviation (uncertainty) of the objectBox estimate. Not Set.
42	4	Object Course Angle	Float32	Orientation or heading of the object box [rad].
46	4	Object Course Angle Sigma	Float32	Uncertainty (standard deviation) of the course angle.
50	8	Relative Velocity	Point2DFloat	Velocity of this object in m/s relative to the ego vehicle in the ego vehicle coordinate system.
58	8	Relative Velocity Sigma	Point2DFloat	Standard deviation of the relative velocity.
66	8	Absolute velocity	Point2DFloat	Absolute velocity of this object in m/s.
74	8	Absolute velocity sigma	Point2DFloat	Standard deviation of the absolute velocity.
82	4	Object Height	Float32	The height of this object in [m]. **Not available yet!**
86	4	Object Height Sigma	Float32	The height of this object in [m]. **Not available yet!**
90	8	Motion Reference Point	Point2DFloat	Motion reference point of this object. All motion information is related to this point. Not used
98	8	Motion Reference Point Sigma	Point2DFloat	The standard deviation of the motion reference point of this object. Not used
106	4	Object Longitudinal Acceleration	Float32	Longitudinal acceleration of this object [meter/seconds^2] in direction of the velocity vector. Not used yet.
110	4	Object Longitudinal Acceleration Sigma	Float32	Standard deviation (uncertainty) of the accelration estimate. Not used yet.
114	4	Object Yaw Rate	Float32	Yaw rate of this object in [rad/sec] Not used.

118	4	Object Yaw Rate Sigma	Float32	Standard deviation (uncertainty) of the yaw rate estimate. Not used.
122	1	Number of contour points	UINT8	Number of contour points transmitted for this object.
123	1	Index of closest point	UINT8	Closest contour point of this object as index of the point list.
124	2	Reference point location	UINT16	The reference point can be located at the following points: 0: Center of gravity 1: Front/Left 2: Front/Right 3: Rear/Right 4: Rear/Left 5: Front/Center 6: Right/Center 7: Rear/Center 8: Left/Center 9: Object Center *0xFF: unknown*
126	8	Reference point coordinate	Point2DFloat	Depending on tracking this is the tracked object reference point, i.e position of reference point [m].
134	8	Reference point sigma	Point2DFloat	Standard deviation of the estimated reference point position [m].
142	4	Reference point position correlation coefficient	Float32	Pearson’s product-moment coefficient. Range: -1..1
146	8	Center of gravity	Point2DFloat	Center of gravity of the tracked Object.
154	2	Object Priority	UINT16	Value determining priority. Value depends on performed algorithm for tracking processings. **Not available yet!**
156	4	Object existence measurement	FLOAT32	**Not available yet!**
160	4	Reserved	-	-
164		List of contour points	Point2DFloat	Array of contour points.
		Dynamic Object Properties	Property List	Dynamic array of additional object properties.

Offset	Bytes	Point2DFloat	Data type	Description
0	4	Position x	FLOAT32	X-part/coordinate of this value/point.

4	4	Position y	FLOAT32	Y-part/coordinate of this value/point.
8				

Offset	Bytes	Property List	Data type	Description
0	2	Property List Length	UINT16	Number of Properties in this list.
2		Array of Properties	Property	-

Offset	Bytes	Property	Data type	Description
0	2	Object Property Key	UINT16	300: AEB Target Selection Flags. 00: Object was not processed 11: Not in path, not AEB rele. 12: In path, not AEB rele. 21: Not in path, subtarget 22: In path, subtarget 32: In path, main target
2	1	Object Property Type	UINT8	Type of object property: 0: VOID 1: FLOAT 2: DOUBLE 3: INT8 4: UINT8 5: INT16 6: UINT16 7: INT32 8: UINT32 9: INT64 10: UINT64 11: BOOL 12: STRING (std)
3		Data	-	Value size: See property type.

13 Ibeo FUSION SYSTEM/ECU measurement list: Data type 0x2821

Measurement List data to represent dynamic measurements provided by an Ibeo sensor system.

All angles, position and distances are given in the ISO 8855 / DIN 70000 coordinate system. Please refer to Section 2.5 for an overview of Ibeo's coordinate system.

Offset	Bytes	Property List	Data type	Description
	4	Reserved	-	-
	8	Timestamp	NTP64	
		List Name	NULL terminated string	Name of the measurement list. Used to identify source of the measurements: "CAN" "XSens" (GPS/IMU)
		Group name	NULL terminated string	Intended for group selection
	2	Measurement list length	UINT16	Number of measurements in this list.
		Array of Properties	Property	-

Offset	Bytes	Property	Data type	Description
0	2	Object Property Key	UINT16	1: longitudinal velocity [m/s] 2: yaw rate [rad/s] 3: steering wheel angle [rad] 4: cross acceleration [m/s²] 5: steering angle [rad] 6: longitudinal acceleration [m/s²] 7: roll angle [rad] 8: pitch angle [rad] 43: vertical acceleration [m/s²] 44: pitch rate [rad/s] 45: roll rate [rad/s] 49: velocity north [m/s] 50: velocity west [m/s] 51: velocity up [m/s] 60: latitude [rad] 61: longitude [rad] 62: altitude [m] 67: yaw angle [rad] 70: UTC hours 71: UTC minutes 72: UTC seconds 73: UTC miliseconds 79: UTC days 80: UTC months 81: UTC years

2	1	Object Property Type	UINT8	Type of object property: 0: VOID (0 bytes) 1: FLOAT (4 bytes) 2: DOUBLE (8 bytes) 3: INT8 (1 bytes) 4: UINT8 (1 bytes) 5: INT16 (2 bytes) 6: UINT16 (2 bytes) 7: INT32 (4 bytes) 8: UINT32 (4 bytes) 9: INT64 (8 bytes) 10: UINT64 (8 bytes) 11: BOOL (1 bytes) 12: STRING (std)
3		Data	-	Value size: See property type.

14 Ibeo FUSION SYSTEM/ECU image: Data type 0x2403

Bytes	Offset	Image	Data type	Description
2	0	image format	UINT16	0 : JPEG , 1 : MJPEG, 2 : GRAY8, 3 : YUV420, 4 : YUV422
4	2	timestamp microseconds	UINT32	since power-on
8	6	NTP timestamp	NTP64	seconds; fractional seconds
1	14	device ID	UINT8	each IBEO device has a system wide unique id
24	15	mountingPosition	Mounting Position	Mounting position of the camera.
8	39	horizontal opening angle	DOUBLE64	radians
8	47	vertical opening angle	DOUBLE64	radians
2	55	image width	UINT16	pixel line count
2	57	image height	UINT16	pixel column count
4	59	compressed size	UINT32	size in bytes of the following image buffer
compressed size	63	Reserved	CAHR[]	image buffer
	39 + compressed size			

Mounting Position

Bytes	Offset	Point2D	Data type	Description
4	0	Yaw angle	FLOAT32	Mounting angles relative to vehicle coordinate system. In radians normalized to [−π, +π[.
4	4	Pitch angle	FLOAT32	
4	8	Roll angle	FLOAT32	
4	12	Offset x	FLOAT32	Mounting position relative to vehicle coordinate system. In meters.
4	16	Offset y	FLOAT32	
4	20	Offset z	FLOAT32	
	24			

15 Ibeo FUSION SYSTEM/ECU vehicle state: Data type 0x2806

Vehicle state data available from Ibeo FUSION SYSTEM and Ibeo AppBase2 (ECU).

All angles, position and distances are given in the ISO 8855 / DIN 70000 coordinate system. Please refer to Section 2.5 for an overview of Ibeo's coordinate system.

Bytes	Offset	Vehicle State:	Data type	Description
4	0	Reserved		
8	4	Timestamp	NTP64	Time stamp when this vehicle state was estimated
4	12	DistanceX	INT32	Distance from origin in x-direction [1/10mm]
4	16	DistanceY	INT32	Distance from origin in y-direction [1/10mm]
4	20	Course angle	FLOAT32	Orientation [rad]
4	24	Longitudinal velocity	FLOAT32	Longitudinal velocity [m/s]
4	28	Yaw rate	FLOAT32	Current yaw rate of vehicle [rad/s]
4	32	Steering wheel angle	FLOAT32	Angle by which the steering wheel is rotated compared to its middle position.
4	36	Reserved		
4	40	Front wheel angle	FLOAT32	Angle by which the front wheel is rotated compared to the vehicle's x-axis [rad]
2	44	Reserved		
4	46	Vehicle Width [3]	FLOAT32	Vehicle width in [m]
4	50	Reserved		
4	54	Distance: Vehicle's 3 front to front axle	FLOAT32	Distance: front axle to vehicle's front [m]
4	58	Distance: rear axle to front axle 3	FLOAT32	Distance: vehicle's rear axle to vehicle's front axle [m]
4	62	Distance: rear axle to vehicle's rear 3	FLOAT32	Distance: vehicle's rear axle to vehicle's rear [m]
4	66	Reserved		

4	70	SteerRatioPoly0 (s0) 3	FLOAT32	Coefficients for transfer function of steering wheel angle (x) to calculate front wheel angle (y):
				y = $\frac{x}{1.095(s_3x^3+s_2x^2+s_1x+s_0)}$ if s_0=0
				y = $s_3x^3+s_2x^2+s_1x+s_0$ if s_0≠0

4	74	SteerRatioPoly1 (s1) 3	FLOAT32	
4	78	SteerRatioPoly2 (s2) 3	FLOAT32	
4	82	SteerRatioPoly3 (s3) 3	FLOAT32	
	86			

[3] Static value of vehicle which needs to be configured in IbeoLUX or ECU

16 Ibeo FUSION SYSTEM/ECU vehicle state: Data type 0x2807

Vehicle state data available from Ibeo FUSION SYSTEM and Ibeo AppBase2 (ECU).

All angles, position and distances are given in the ISO 8855 / DIN 70000 coordinate system. Please refer to Section 2.5 for an overview of Ibeo's coordinate system.

Bytes	Offset	Vehicle State:	Data type	Description
4	0	Reserved		
8	4	Timestamp	NTP64	Time stamp when this vehicle state was estimated
4	12	DistanceX	INT32	Distance from origin in x-direction [1/10mm]
4	16	DistanceY	INT32	Distance from origin in y-direction [1/10mm]
4	20	Course angle	FLOAT32	Orientation [rad]
4	24	Longitudinal velocity	FLOAT32	Longitudinal velocity [m/s]
4	28	Yaw rate	FLOAT32	Current yaw rate of vehicle [rad/s]
4	32	Steering wheel angle	FLOAT32	Angle by which the steering wheel is rotated compared to its middle position.
4	36	Reserved		
4	40	Front wheel angle	FLOAT32	Angle by which the front wheel is rotated compared to the vehicle's x-axis [rad]
2	44	Reserved		
4	46	Vehicle Width [4]	FLOAT32	Vehicle width in [m]
4	50	Reserved		
4	54	Distance: Vehicle's 3 front to front axle	FLOAT32	Distance: front axle to vehicle's front [m]
4	58	Distance: rear axle to front axle 3	FLOAT32	Distance: vehicle's rear axle to vehicle's front axle [m]
4	62	Distance: rear axle to vehicle's rear 3	FLOAT32	Distance: vehicle's rear axle to vehicle's rear [m]
4	66	Reserved		

4	70	SteerRatioPoly0 (s0) 3	FLOAT32	Coefficients for transfer function of steering wheel angle (x) to calculate front wheel angle (y):
				y = $\frac{x}{1.095(s_3x^3+s_2x^2+s_1x+s_0)}$ if $s_0=0$
				y = $s_3x^3+s_2x^2+s_1x+s_0$ if $s_0\neq0$

4	74	SteerRatioPoly1 (s1) 3	FLOAT32	
4	78	SteerRatioPoly2 (s2) 3	FLOAT32	
4	82	SteerRatioPoly3 (s3) 3	FLOAT32	
4	86	Longitudinal Acceleration	FLOAT 32	current longitudinal acceleration of vehicle [m/s²]

[4] Static value of vehicle which needs to be configured in IbeoLUX or ECU

	90

17 Ibeo FUSION SYSTEM/ECU vehicle state: Data type 0x2808

Vehicle state data available from Ibeo FUSION SYSTEM and Ibeo AppBase2 (ECU).

All angles, position and distances are given in the ISO 8855 / DIN 70000 coordinate system. Please refer to Section 2.5 for an overview of Ibeo's coordinate system.

Bytes	Offset	Vehicle State:	Data type	Description
4	0	Reserved	-	-
8	4	Timestamp	NTP64	Time stamp when this vehicle state was estimated
2	12	Source	UINT16	Bit 0: CAN Bit 1: GPS Bit 2: IMU
2	14	reserved	-	-
8	16	Latitude	DOUBLE	Latitude [deg]
4	24	Reserved	-	-
8	28	Longitude	DOUBLE	Longitude [deg]
4	36	Reserved	-	-
4	40	Altitude	DOUBLE	Altitude [deg]
4	44	Reserved	-	-
8	48	Reserved	DOUBLE	-
4	56	Sigma X	FLOAT32	Standard deviation of the longitude position [m]
8	60	Reserved	-	-
4	68	Sigma Y	FLOAT32	Standard deviation of the latitude position [m]
4	72	Reserved	-	-
4	76	Reserved	-	-
4	80	Position Correlation Coefficient XY	FLOAT32	Position Correlation Coefficient XY
4	84	Reserved	-	-
4	88	Reserved	-	-
4	92	Course angle	FLOAT32	Direction of current motion [rad], north = 0.
4	96	Sigma Course Angle	FLOAT32	Standard deviation of the course angle [rad]
4	100	Heading angle	FLOAT32	Heading of the vehicle body [rad], north = 0.
4	104	Sigma heading angle	FLOAT32	Standard deviation of the heading angle [rad]
4	108	Reserved	-	-
4	112	Reserved	-	-
4	116	Reserved	-	-
4	120	Reserved	-	-

4	124	Longitudinal velocity	FLOAT32	Longitudinal velocity [m/s]
4	128	Sigma longitudinal velocity	FLOAT32	Standard deviation of the longitudinal velocity [m/s]
4	132	Yaw rate	FLOAT32	Current yaw rate of vehicle [rad/s]
4	136	Sigma yaw rate	FLOAT32	Standard deviation of the yaw rate [rad/s]
4	140	Longitudinal acceleration	FLOAT32	Longitudinal acceleration [m/s²]
4	144	Sigma longitudinal acceleration	FLOAT32	Standard deviation of the longitudinal acceleration [m/s²]
4	148	Cross acceleration	FLOAT32	Cross acceleration [m/s²]
4	152	Sigma cross acceleration	FLOAT32	Standard deviation of the cross acceleration [m/s²]
4	156	Steer angle	FLOAT32	Angle by which the front wheel is rotated compared to the vehicle's x-axis [rad]
4	160	Steering wheel angle	FLOAT32	Angle by which the steering wheel is rotated compared to its middle position.
4	164	Vehicle Width [5]	FLOAT32	Vehicle width in [m]
4	168	Reserved	-	-
4	172	Distance vehicle front to front axle	FLOAT32	Distance front axle to vehicle's front [m]
4	176	Distance rear axle to front axle	FLOAT32	Distance: vehicle's rear axle to vehicle's front axle [m]
4	180	Distance: rear axle to vehicle's rear	FLOAT32	Distance: vehicle's rear axle to vehicle's rear [m]
4	184	Reserved	-	-

4	188	SteerRatioPoly0 (s0)	FLOAT32	Coefficients for transfer function of steering wheel angle (x) to calculate front wheel angle (y):
				y = $\frac{x}{1.095(s_3x^3+s_2x^2+s_1x+s_0)}$ if $s_0=0$
				y = $s_3x^3+s_2x^2+s_1x+s_0$ if $s_0\neq0$

4	192	SteerRatioPoly1 (s1)	FLOAT32	
4	196	SteerRatioPoly2 (s2)	FLOAT32	
4	200	SteerRatioPoly3 (s3)	FLOAT32	
	204			

[5] Static value of vehicle which needs to be configured in IbeoLUX or ECU

19 Ibeo FUSION SYSTEM/ECU Trace Messages - Data types 0x6400 - 0x6440

Software modules that are deploying the Ibeo FUSION SYSTEM for communication can sent trace messages consisting of a character string. Trace Messages are distributed with four different data types dependent on their priority:

- Data type 0x6400 - Error
- Data type 0x6410 - Warning
- Data type 0x6420 - Note
- Data type 0x6430 – Debug Info

Bytes	Offset	Data	Data type	Description
1	0	Trace level	UINT8	Gives the trace level of this message. Currently this accords to the data type. 1=error, 2=warning, 3=note, 4=debug
Size of string	1	Trace message	String	Contains warnings and errors received from connected sensors. E.g. *“IbeoLUX3 "ibeoLUX": DSP warning: Invalid vehicle motion data. To avoid this warning please uncheck 'Vehicle Motion Data expected' in device configuration. Code: 0x0100”*
1	Size of string + 1	0x00	UINT8	End of string byte 0x00.
	Size of string + 2			

20 Ibeo FUSION SYSTEM/ECU Set Filter Command – Data type 0x2010

Before the Ibeo FUSION SYSTEM or ECU sends data after connecting to it (default port 12002), a filter command must be sent.
Data is encoded in network byte order / big endian format.

Bytes	Offset	Data	Data type	Description
2	0	Command ID: Set filter	UINT16	0x0005
2	2	2 x Number of following data type ranges (= 2 * n)	UINT16	The number of following entries in the filter list.
2 * n	4	List of data type ranges	Data type range	Range of data types to receive
	4 + 2 * n			

Data type range:

Bytes	Offset	Data	Data type	Description
2	0	Start of data type range.	UINT16	
2	2	End of data type range.	UINT16	
	4			

Range of data types to receive all scan data types:	0x0000, 0xffff.
Range of data types to receive all scan data types:	0x2202, 0x220f.
Range of data types to receive all object data types:	0x2220, 0x222f.

Example 1: Receive all data types:

Bytes	Offset	Data
2	0	0x0005
2	2	0x0002
2	4	0x0000
2	6	0xFFFF

Complete hex data of this example command looks like this:

```
af fe c0 c2 00 00 00 00 00 00 00 08 00 00 20 10 00 00 00 00 00 00 00 00
00 05 00 02 00 00 ff ff
```

First line is the Ibeo data header, second the command data.

Example 2: Receive all scan data types:

Bytes	Offset	Data
2	0	0x0005
2	2	0x0002
2	4	0x2202
2	6	0x220f

Complete hex data of this example command looks like this:

```
af fe c0 c2 00 00 00 00 00 00 00 08 00 00 20 10 00 00 00 00 00 00 00 00
00 05 00 02 2202 220f
```

The command will be replied by Ibeo FUSION SYSTEM or ECU with data type 0x2020. The reply data is 0x0005 (the received command ID).

21 Ibeo FUSION SYSTEM/ECU measurement list: Data type 0x2821

Data is encoded in network byte order / big endian format

Bytes	Offset	Data	Data type	Description
12	0	timeStamp	DeviceSpecificTime	Timestamp at which the vehicle had this state
2..n	12	listName	String	Null-terminated ASCII-character array
2..m	14..n	groupName	String	Null-terminated ASCII-character array
2	16..n+m	numberOfMeasurements	UNIT16	
		Measurement		

Device Specific Time

Bytes	Offset	Data	Data type	Description
4	0	microseconds	UINT32	The internal microsecond counter of this device: Elapsed microseconds since the power-on of this device
4	4	NTP seconds	UINT32	NTP timestamp: Number of seconds elapsed since 1900-01-01 00:00:00
4	8	NTP fractional seconds	UINT32	NTP Fractional part: 1/(2^32) seconds fragment
	12			

Measurement

Bytes	Offset	Data	Data type	Description
2	0	mkey	ENUM (UINT16)	measurement key, defined by application
1	2	Type ID	UINT8	ID for type stored in measurement: 0x00: VOID 0x01: float32 0x02: double64 0x03: INT8 0x04: UINT8 0x05: INT16 0x06: UINT16 0x07: INT32 0x08: UINT32 0x09: INT64 0x0A: UINT64 0x0B: BOOL 0x0C: String (null terminated character array)
1..8 dependent on Type ID	3	value	1..8 dependent on Type ID	Value

22 Ibeo FUSION SYSTEM/ECU GPS position (PositionWGS84): Data type 0x2604

Bytes	Offset	Data	Data type	Description
4	0	Time since startup.	ÙINT32	Microseconds since device startup.
8	4	Timestamp	NTPTime	Time stamp of position measurement.
1	12	Device ID	UINT8	ID of the GPS device giving the position measurement.
8	13	Latitude	DOUBLE64	Latitude of the position, in [rad], within interval [-Pi/2, Pi/2]
8	21	Longitude	DOUBLE64	Longitude of the position, in [rad], within interval [-Pi, Pi]
8	29	Altitude MSL	DOUBLE64	Altitude of the position above the sea level, in [m]
8	37	Latitude Sigma	DOUBLE64	Uncertainty of the latitude, in [m]
8	45	Longitude Sigma	DOUBLE64	Uncertainty of the Longitude, in [m]
8	53	Altitude MSL Sigma	DOUBLE64	Uncertainty of the altitude above sea level in [m]
8	61	Course Angle	DOUBLE64	Course angle in [rad] relative to North. Interval: [0;2*Pi] The course angle is the angle the vehicle is traveling to. If you drift, it is different to the yaw angle, which is the direction of the vehicle is heading/looking at. Example: 0=North, Pi/2=West
8	69	Course Angle Sigma	DOUBLE64	Uncertainty of course angle [rad]
8	77	Yaw Angle	DOUBLE64	Yaw Angle in [rad], Interval [0;2*Pi]. The yaw angle is the angle the vehicle is heading/looking at. If you drift, it is different to the course angle, which is the direction of travelling or the track angle. Example: 0=North, Pi/2=West
8	85	Yaw Angle Sigma	DOUBLE64	Uncertainty of the estimated yaw angle.
32	93	Reserved		Reserved for internal use.

2	125	Source Type	ENUM16	Type of the device that created this measurement.
				Unknown = 0,
				GPS_SPS = 1,
				GPS_PPS = 2,
				GPS_SBAS = 3,

				GPS_SBAS_Omnistar_VBS = 4,
				GPS_SBAS_Omnistar_HP = 5,
				GPS_GBAS = 6,
				GPS_GBAS_RTK_Float = 7,
				_GPS_GBAS_RTK_Integer = 8,
				Imu = 9,
				LandmarkPositioning = 10,
				Manual = 11,

	127			

23 Ibeo frame end separator: Data type 0x1100

FrameEndSeparator is placed at the end of one frame (e.g. frame like ScanPtList with vehicleData, DeviceStatus and ObjectData).

Data is encoded in network byte order / big endian format.

Bytes	Offset	Data	Data type	Description
4	0	FrameId	UINT32	Frame-Number (0 bis 1-2^32) (overflow after 47721 h). The scanner will increment the FrameId by 1 with each new frame.
4	4	SizeOfThisFrame [Byte]	UINT32	Size of all data [in bytes] between the last separator and this one. (All data before this separator up to the previous separator, not incuding the separator itself. If this frame is the first one in the file, this value will be 0.
4	8	SizeOfNextFrame [Byte]	UINT32	Size of all data [in bytes] between this separator and the next one. In most cases this value is not known when writing this datatype to the harddrive. In this case it is set to 0.
8	12	creationTime	NTPTime	Timestamp of the moment of creation of this package. This should be the same as the IbeoDataHeader timestamp in case this package originated from the creating device as ScaLa-Sensor.
2	20	reserved01	UINT16	For future use

2	22	reserved02	UINT16	For future use
2	24	reserved03	UINT16	For future use
2	26	reserved04	UINT16	For future use
2	28	reserved05	UINT16	For future use
2	30	reserved06	UINT16	For future use

24 Ibeo reference object data - Data type 0x2291

Hint: This data type is only written by Evaluation Suite Versions <= R2015-08. Starting with Version 2016-05 the data type 0x2281 is written setting a flag indicating that the objects are post processed.

Reference object data is provided by the ibeo Evaluation Suite software. It is identical to the object data of type 0x2281. Each data block starts with the IbeoDataHeader followed by the object list.
For each object list this header is preceded. The IbeoDataHeader is described in Section 2.4 Ibeo Data Header.

All positions and angles are given in the vehicle / reference coordinate system.
Data is encoded in network byte order / big endian format.

In general, positions, lengths, distances and sizes are coded in meters. In general, angles are coded in radians.

ECU Object Data List:

Offset	Bytes	Object header	Data type	Description
0	8	Mid-scan timestamp	NTP64	Mid-scan timestamp is the absolute timestamp when the scanner mirror crossed the middle of the corresponding scan. Avg between in start and end scan time stamp. Not set! from 5.7.5
8	1	Object List ID	UINT8	Unique object list identifier to match object list with its source of computation. Not used. Default 255.
9	1	Device Type	UNIT8	Device Type that created this object list e.g. laserscanner, radar, camera. 0: unknown 1: Laserscanner 2: ECU 3: CAN bus 4: Camera 5: GPS 99: other **Not set yet! This should be an enumeration.**

10	2	Device Interface Version	UINT16	Device Interface Version of Software that creates this object list. Only 14 Bits can be used (0-16383) **Not set yet! This should be an enumeration.**
12	2	Reserved	-	-
14	2	Number of objects	UINT16	The number of objects transmitted in this message.
16		List of objects	Object	Array of objects.

Offset	Bytes	Object	Data type	Description
0	4	Object ID	UINT32	ID of this object from tracking.
4	2	Flags	UINT16	Bit 6: 0 = tracked by dyn. model 1 = tracked by static model Bit 7: 0 = mobility of dynamic obj. not (yet) detected 1 = mobility of dynamic obj. successfully detected Bit 8: 0 = motion model not validated 1 = motion model validated For a detailed description of these flags and combinations please see Section 11.1
6	4	Object Age	UINT32	Number of scans this object has been tracked for.
10	8	Timestamp NTP	NTP64	Timestamp of the last measurement (COG of Segment) that was used for updating this object.
18	2	Object Prediction Age	UINT16	Number of update cycles that this object has only been predicted without measurement updates.
20	1	Classification	UINT8	Most likely class of this object: 0: unclassified 1: unknown small 2: unknown big 3: pedestrian 4: bike 5: car 6: truck 7…: reserved 12: underdrivable

21	1	Classification Quality	UINT8	The higher this value is the more reliable is the assigned object class. Range: 0..100
22	4	Classification Age	UINT32	Time that this object has been classified as current class in ms.
26	8	Object Box Size	Point2DFloat	Size of the object box in the object coordinate system (vehicle coordinate system rotated around z axis by object course angle). X value corresponds to the object length, Y value correcsponds to the object width.
34	8	Object Box Size Sigma	Point2DFloat	Standard deviation (uncertainty) of the objectBox estimate. Not Set.
42	4	Object Course Angle	Float32	Orientation or heading of the object box [rad].
46	4	Object Course Angle Sigma	Float32	Uncertainty (standard deviation) of the course angle.
50	8	Relative Velocity	Point2DFloat	Velocity of this object in m/s relative to the ego vehicle in the ego vehicle coordinate system.
58	8	Relative Velocity Sigma	Point2DFloat	Standard deviation of the relative velocity.
66	8	Absolute velocity	Point2DFloat	Absolute velocity of this object in m/s.
74	8	Absolute velocity sigma	Point2DFloat	Standard deviation of the absolute velocity.
82	4	Object Height	Float32	The height of this object in [m]. **Not available yet!**
86	4	Object Height Sigma	Float32	The height of this object in [m]. **Not available yet!**
90	8	Motion Reference Point	Point2DFloat	Motion reference point of this object. All motion information is related to this point. Not used
98	8	Motion Reference Point Sigma	Point2DFloat	The standard deviation of the motion reference point of this object. Not used
106	4	Object Longitudinal Acceleration	Float32	Longitudinal acceleration of this object [meter/seconds^2] in direction of the velocity vector.

				Not used yet.
110	4	Object Longitudinal Acceleration Sigma	Float32	Standard deviation (uncertainty) of the accelration estimate. Not used yet.
114	4	Object Yaw Rate	Float32	Yaw rate of this object in [rad/sec] Not used.
118	4	Object Yaw Rate Sigma	Float32	Standard deviation (uncertainty) of the yaw rate estimate. Not used.
122	1	Number of contour points	UINT8	Number of contour points transmitted for this object.
123	1	Index of closest point	UINT8	Closest contour point of this object as index of the point list.
124	2	Reference point location	UINT16	The reference point can be located at the following points: 0: Center of gravity 1: Front/Left 2: Front/Right 3: Rear/Right 4: Rear/Left 5: Front/Center 6: Right/Center 7: Rear/Center 8: Left/Center 9: Object Center *0xFF: unknown*
126	8	Reference point coordinate	Point2DFloat	Depending on tracking this is the tracked object reference point, i.e position of reference point [m].
134	8	Reference point sigma	Point2DFloat	Standard deviation of the estimated reference point position [m].
142	4	Reference point position correlation coefficient	Float32	Pearson’s product-moment coefficient. Range: -1..1
146	8	Center of gravity	Point2DFloat	Center of gravity of the tracked Object.

154	2	Object Priority	UINT16	Value determining priority. Value depends on performed algorithm for tracking processings. **Not available yet!**
156	4	Object existence measurement	FLOAT32	**Not available yet!**
160	4	Reserved	-	-
164		List of contour points	Point2DFloat	Array of contour points.
		Dynamic Object Properties	Property List	Dynamic array of additional object properties.

Offset	Bytes	Point2DFloat	Data type	Description
0	4	Position x	FLOAT32	X-part/coordinate of this value/point.
4	4	Position y	FLOAT32	Y-part/coordinate of this value/point.
8				

Offset	Bytes	Property List	Data type	Description
0	2	Property List Length	UINT16	Number of Properties in this list.
2		Array of Properties	Property	-

Offset	Bytes	Property	Data type	Description
0	2	Object Property Key	UINT16	300: AEB Target Selection Flags. 00: Object was not processed 11: Not in path, not AEB rele. 12: In path, not AEB rele. 21: Not in path, subtarget 22: In path, subtarget 32: In path, main target
2	1	Object Property Type	UINT8	Type of object property: 0: VOID 1: FLOAT 2: DOUBLE 3: INT8 4: UINT8 5: INT16 6: UINT16 7: INT32 8: UINT32 9: INT64 10: UINT64 11: BOOL 12: STRING (std)
3		Data	-	Value size: See property type.

25 Ibeo frame index: Data type 0x6130

Data is encoded in network byte order / big endian format.

Bytes	Offset	Data	Data type	Description
1	0	majorVersion	UINT8	Major version of the FrameIndex
1	1	minorVersion	UINT8	Minor version of the FrameIndex
17	2	"FRAMEINDEX_HEADER"	CHAR[17]	
8	19	nbOfFrames	UINT64	Number of Frames in this Index
8	27	lastTimestamp	UINT64	Timestamp of the last Frame
size(FramingPolicy); (max 30 TriggerType)	35	framePolicy	FramePolicy	The FramingPolicy that has been used to generate the FrameIndex.
128-35-size(FramingPolicy)	Dyn	padding characters	Char[*]	Padding characters to fill up 128 bytes
nbOfFrames * size(FrameIndexEntry)	128	frameIndices	FrameIndexEntry[nbOfFrames]	FrameIndexEntries

FramingPolicy:

Bytes	Offset	Data UN	Data type	Description
1	0	triggerBelongsToNewFrame	UINT8 (bool)	True if the trigger belongs to the new frame, e.g. it defines the start of the new frame. If false the trigger belongs to the old frame and therefore defines the end of the old frame.
2	1	nbOfTriggers	UINT16	Number of triggers.
nbOfTriggers * size(TriggerType)	3		Trigger(nbOfTriggers)	Trigger that defines a frame start/end.

TriggerType:

Bytes	Offset	Data	Data type	Description
2	0	DataTyoe	UINT16	Datatype of the trigger definition
1	2	sourceDeviceID	UINT8	Source device id of the trigger definition

FrameIndexEntry

Bytes	Offset	Data	Data type	Description
8	0	filePosition	UINT64	Position of the frame in the file.
8	8	timestamp	UINT64	Timestamp of this frame
1	16	deviceId	UINT8	Device id that defined the frame

26 Ibeo carriage way: Data type 0x6970

Data is encoded in network byte order / big endian format.

Bytes	Offset	Data	Data type	Description
8	0	numberOfCarriageWays	UINT64	The number of CarriageWays stored in this list
N*CarriageWay	8	CarriageWay (see numberOfCarriageWays	CarriageWay	The Carriageways of this list

CarriageWay

Bytes	Offset	Data	Data type	Description
8	0	ID	UINT64	The unique id of this Carriageway
2	8	nationalId	UINT16	The national ID of the CarriageWay (e.g. 255 for street A255)
1	10	Type	UINT8	The type of the CarriageWay: 01 : Motorway 02: Trunk 03: Primary 04: Secondary 05: Tertiary 05: Residential 06: Service 07: Unclassified
8	11	numberOfCarriageWaySegments	UINT64	The number of segments of this CarriageWay
N*CarriageWaySegment	19	CarriageWaySegment (see numberOfCarriageWaySegments)	CarriageWaySegment	The segments of the CarriageWay

CarriageWaySegment

Bytes	Offset	Scan header	Data type	Description
8	0	ID	UINT64	The unique id of this CarriagewaySegment
1	8	number of lanes	UINT8	The number of lanes of the segment
8	9	Next segment ID	UINT64	The ID of the following segment
8	17	Previous segment ID	UINT64	The ID of the previous segment
N * Lane	25	Lane (see number of lanes)	Lane	The lanes of this segment

Lane

Bytes	Offset	Scan header	Data type	Description
1	0	ID	UINT8	The id of this lane (valid only in the current segment)
8	1	LaneID	UINT64	The unique id of the lane (valid for the whole CarriageWay)
1	9	type	UINT8	The type of the lane: 01: Forward 02: Backward 03: Exit 04: Oncoming 05: Breakdown 06: Unclassified
1	10	Next lane id	UINT8	The id of the following lane
1	11	Previous lane id	UINT8	The id of the previous lane
1	12	Left lane id	UINT8	The id of the left neighbouring lane
1	13	Right lane id	UINT8	The id of the right neighbouring lane
8	14	number of lane segments	UINT64	The number of LaneSegments
N * LaneSegment	22	LaneSegment (see number of lane segments)	LaneSegment	The LaneSegments of this Lane

LaneSegment

Bytes	Offset	Scan header	Data type	Description
8	0	ID	UINT64	The id of this LaneSegment (valid for the current Lane)
8	8	Next segment id	UINT64	The id of the following LaneSegment
8	16	Previous segment id	UINT64	The id of the previous LaneSegment
8	24	Left segment id	UINT64	The id of the left LaneSegment
8	32	Right segment id	UINT64	The id of the right LaneSegment
1	40	marking type left	UINT8	The marking type of the left lane marking: 01: Solid 02: Dashed 03: Unclassified 04: None
1	41	marking type right	UINT8	The marking type of the right lane marking: 01: Solid 02: Dashed 03: Unclassified

				04: None
1	42	border type left	UINT8	The left border type: 01: Lane change possible 02: Lane change not possible 03: Oncoming lane 04: Solid end of street 05: Flat end of street 05: Unclassified
1	43	border type right	UINT8	The right border type: 01: Right change possible 02: Right change not possible 03: Oncoming lane 04: Solid end of street 05: Flat end of street 05: Unclassified
1	44	next segment in new CarriageWaySegment	UINT8	Flag indicating whether the next LaneSegment is in a new CarriageWaySegment
1	45	previous segment in new CarriageWaySegment	UINT8	Flag indicating whether the previous LaneSegment is in a new CarriageWaySegment
8	46	LaneSupportPoint longitude	double	longitude position of the support point in WGS84 in deg
8	54	LaneSupportPoint latitude	double	latitude position of the support point in WGS84 in deg
8	62	LaneSupportPoint course angle	double	The course angle of the support point in deg (from north)
8	70	offset to left lane marking	Point2D	The offset from the start point to the left lane marking
8	78	offset to right lane marking	Point2D	The offset from the start point to the right lane marking

27 IDC Trailer: Data type 0x6120

Each IDC-file shall conclude with the IDC-Trailer. The IDC-Trailer does not contain any data, but marks the end of the IDC-File. So it includes simply the header with the data type number.

28 Ibeo point cloud plane: Data type 0x7510

The point cloud datatype is used for holding a collection of 3-dimensional points. The term plane is indicating that the data are stored with an offset to a reference system. The differentiation is made because the reference system can also hold geo-coordinates and therefore it is important, that this point cloud type does not store the geo-coordinates for each single point.

Data is encoded in network byte order / big endian format.

Offset	Bytes	PointCloudPlane	Data type	Description
0	2	PointsKind	UINT16	Kind/Class of all the points in the cloud. [enum] 0: Undefined 1: ScanPoint 2: LanePoint 3: CurbstonePoint 4: GuardrailPoint 5: RoadmarkingPoint 6: OffRoadMarkingPoint
2	2	PointsType	UINT16	Type of the point. Defines which fields are present in the serialization. [enum] 0: Point 1: PointWithEPW 2: PointWithFlags 3: PointWithEpwAndFlags
4	20	Reference point	GpsPoint	"Reference point for the points inside the points vector. To use the data locally only, set all components of the referencePoint to NaN." [struct]
24	4	planeOrientationYaw	float	Orientation of the tangential plane. Angle of the Y=0 axis with respect to north. [rad]
28	4	planeOrientationPitch	float	Pitch [rad]
32	4	planeOrientationRoll	float	Roll [rad]
36	8	Reserved	-	-
44	4	Number of points	UINT32	Number of points in the points vector
48	variable	points	Points[]	Vector of points. The points structure depends on the pointType.

Offset	Bytes	GpsPoint	Data type	Description
0	8	Longitude	Double	longitudinal position [rad]
8	8	Latitude	Double	latitudinal position [rad]
16	4	Altitude	Float	Altitude [m]

Offset	Bytes	PointType: Point	Data type	Description
0	12	Position	PlanePosition	Position of the point [struct]

Offset	Bytes	PointType: PointWithEPW	Data type	Description
0	12	Position	PlanePosition	Position of the point [struct]
12	4	Echo pulse width	Float	Echo pulse width of this measured point

Offset	Bytes	PointType: PointWithFlags	Data type	Description
0	12	Position	PlanePosition	Position of the point [struct]
12	4	Flags	UINT32	Flags [BitField]

Offset	Bytes	PointType: PointWithEpwAndFlags	Data type	Description
0	12	Position	PlanePosition	Position of the point [struct]
12	4	Echo pulse width	Float	Echo pulse width of this measured point
16	4	Flags	UINT32	Flags [BitField]

Offset	Bytes	PlanePosition	Data type	Description
0	4	X	Float	x position of the point [m]
4	4	Y	Float	y position of the point [m]
8	4	Z	Float	z position of the point [m]

29 Ibeo meta information list: Data type 0x7110

The meta information datatype is used to store configurations, version numbers, keywords and other meta information.

Data is encoded in network byte order / big endian format.

Offset	Bytes	MetaInformationList	Data type	Description
0	4	Number of elements	UINT32	Number of MetaInformations in the list
4	Variable	Elements	MetaInformation[]	Vector holding the elements

Offset	Bytes	MetaInformation	Data type	Description
0	2	Type of information	UINT16	Type of the information 0: UNDEFINED 1: VERSION_NUMBER 2: AB_CONFIG 3: AB_SYNC_METHOD 4: ECU_ID 5: KEYWORDS
2	8	Time stamp	NTPTime	TimeStamp of this meta information
10	4	Size of payload	UINT32	Number of bytes used by the payload following this field.
14	Variable	Subtype fields	MetaInformationSubtype	the subtype holding the information

Offset	Bytes	Subtype: VERSION_NUMBER	Data type	Description
0	2	Software type	UINT16	0: Undefined 1: Custom 2: AppBase 3: EVS 4: ILV
2	4	Version number	UINT32	The version number. Major: VersionNumber/1000000 Minor: (VersionNumber%1000000)/1000 Patch: VersionNumber%1000
6	Variable	Extra string	String	Custom version string

Offset	Bytes	Subtype: AB_CONFIG	Data type	Description
0	Variable	Config	String	The appbase configuration

Offset	Bytes	Subtype:	Data type	Description

		AB_SYNC_METHOD		
0	2	Sync method	UINT16	0: UNDEFINED 1: GPS_SYNC

Offset	Bytes	Subtype: ECU_ID	Data type	Description
0	Variable	Ecu ID	String	ID of the ECU

Offset	Bytes	Subtype: KEYWORDS	Data type	Description
0	Variable	keywords	String	Keywords for the IDC Semicolon separated list of keywords

Offset	Bytes	String	Data type	Description
0	2	Size	UINT16	Number of String characters
2	Variable	Characters	UINT8[]	The characters of the string

30 CAN Message over Ethernet: Data type 0x1002

Messages described below are tunneled over Ethernet in following format with data type 0x1002.

Bytes	Offset	Object header	Data type	Description
1	0	Version	UINT4	Low nibble / bits 0…3 of this byte. Set to 0.
		Length of CAN message	UINT4	High nibble / bits 4..7 of this byte. Length = 1..8
[1..8]	1	Data	UINT8[1..8]	
1	[2..9]	Message Type	UINT8	Set to 0x00
2	[3..10]	CAN-ID	UINT16	CAN-ID is an 11-bit-identifier. Additionally the most significant bit (Bit 15) needs to be set to 1.
4	[5..12]	Device time in microseconds	UINT32	Hardware time stamp. Total number of microseconds elapsed since start-up of CAN device.
8	[9..16]	Receiving time	NTP64	Time when message has been received by remote receiver.
1	[17..24]	Device ID	UINT8	ID of CAN Device
	[18..25]			

31 Object Association: Data type 0x4001

The byte order for data type 0x4001 is Big Endian.

Bytes	Offset	Object header	Data type	Description
1	0	referenceObjListId	UINT8	Details of the (reference) object list which contains all reference objects of the association.
1	1	referenceDeviceType	UINT8	
2	2	refrenceDeviceInferfaceVersion	UINT16	
1	4	dutObjListId	UINT8	Details of the object list which contains all DUT objects of the association.
1	5	dutDeviceType	UINT8	
2	6	dutDeviceInterfaceVersion	UINT16	
4	8	objectAssocitationCount	UINT32	Number of ObjectAssociation entries in list
8	12	reserved	CHAR[8]	All bytes set to 0xFF.
	20	ObjectAssociation[N]		ObjectAssociation list

The following table contains the Object Association information.

Bytes	Offset	Object header	Data type	Description
4	0	m_referenceObjectId	UINT32	Object ID of the reference object involved in association
4	4	m_dutObjectId	UINT32	Object ID of the DuT object involved in association
8	8	m_timestampFirst	NTP64	Timestamp when the association begins
8	16	m_timestampLast	NTP64	Timestamp when the association ends
4	24	Certainty	float	Certainty of the association, [0 .. 1]

32 TimeRecord: Data type 0x9000

The byte order for data type 0x9000 is Big Endian.
This data type contains the local system time as well as a time signal that is provided by an external device, e.g. a GPS receiver connected to the Ibeo ECU. This data can be used for offline transformation of the time base of the recorded data stream.

Bytes	Offset	Object header	Data type	Description
2	0	ExternalClockType	ClockType	Characteristics of the clock used to record times.
2	2	InternalClockType	ClockType	Characteristics of the clock used to record times.
4	4	VectorSize	UINT32	Number of times recorded.
	8	ExternalClockTimes	NTPTimeVector	Vector of times as represented by external clock.
		InternalClockTimes	NTPTimeVector	Vector of times as represented by internal clock.
		FixModes	FixModeVector	Vector representing fix modes used.
16		reserved	UINT32[4]	Reserved fields

Bytes	Offset	ClockType	Data type	Description
1	0	ClockID	UINT8	Unique ID representing each clock.
2	1	ClockName	UINT8	Enum fields : 0:Clock_Unkown 1:Clock_Laserscanner 2 :Clock_ECU 3 :Clock_CanBus 4 :Clock_Camera 5 :Clock_GpsImu 6 :Clock_Camera 7 :Clock_Other

Bytes	Offset	NTPTimeVector	Data type	Description
8* vectorSize		NTP64[vectorSize]		

Bytes	Offset	FixModeVector	Data type	Description

8* vectorSize		FixMode[vectorSize]		

Bytes	Offset	NTP64	Data type	Description
8	0	UINT64	4 Most Significant Bytes: seconds 4 Least Significant Bytes: fractionalSeconds	Number of seconds elapsed since 1900-01-01 00:00:00 Fractional part: 1/(2^32) seconds fragment

Bytes	Offset	FixMode	Data type	Description
1	0	FixMode	UINT8	0 : Not_Seen 1 : None 2 : Fix_2D 3 : Fix_3D

33 GpsImu: Data type 0x9001

The byte order for data type 0x9001 is Big Endian.
This data type contains unprocessed position and motion data provided by a connected GPS/IMU device, e.g. XSens MTi-G.

Bytes	Offset	Object header	Data type	Description
8	0	latitude	double	Latitude [rad]
8	8	latitudeSigma	double	Standard deviation of the latitude [m]
8	16	longitude	double	Longitude [rad]
8	24	longitudeSigma	double	Standard deviation of the longitude [m]
8	32	altitude	double	Altitude [m]
8	40	altitudeSigma	double	Standard deviation of the altitude [m]
8	48	courseAngle	double	Course angle in [rad] relative to North. Interval: [0;2*Pi] The course angle is the angle the vehicle is traveling to. If you drift, it is different to the yaw angle, which is the direction of the vehicle is heading/looking at. Example: 0=North, Pi/2=West
8	56	courseAngleSigma	double	Uncertainty of course angle [rad]
8	64	yawAngle	double	Yaw Angle in [rad], Interval [0;2*Pi]. The yaw angle is the angle the vehicle is heading/looking at. If you drift, it is different to the course angle, which is the direction of travelling or the track angle. Example: 0=North, Pi/2=West
8	72	yawAngleSigma	double	Uncertainty of the estimated yaw angle.
8	80	pitchAngle	double	Pitch angle [rad]
8	88	pitchAngleSigma	double	Uncertainty in pitch angle [rad].
8	96	rollAngle	double	Roll angle [rad]

8	104	rollAngleSigma	double	Uncertainty in roll angle [rad].
8	112	crossAcceleration	double	Cross acceleration [m/s^2]
8	120	crossAccelerationSigma	double	Standard deviation of the cross acceleration [m/s^2].
8	128	longitudinalAcceleration	double	Longitudinal acceleration [m/s^2].
8	136	longitudinalAccelerationSigma	double	Standard deviation of the longitudinal acceleration [m/s^2].
8	144	verticalAcceleration	double	Vertical acceleration [m/s^2].
8	152	verticalAccelerationSigma	double	Standard deviation of the vertical acceleration [m/s^2].
8	160	velocityNorth	double	Velocity in north direction [m/s^2].
8	168	velocityNorthSigma	double	Standard deviation of velocity in north direction [m/s^2].
8	176	velocityWest	double	Velocity in west direction [m/s^2].
8	184	velocityWestSigma	double	Standard deviation of velocity in west direction [m/s^2].
8	192	velocityUp	double	Velocity upwards [m/s^2].
8	200	velocityUpSigma	double	Standard deviation of velocity upwards [m/s^2].
8	208	velocityX	double	Velocity in X direction [m/s^2].
8	216	velocityXSigma	double	Standard deviation of velocity in X direction [m/s^2].
8	224	velocityY	double	Velocity in Y direction [m/s^2].
8	232	velocityYSigma	double	Standard deviation of velocity in Y direction [m/s^2].
8	240	velocityZ	double	Velocity in Z direction [m/s^2]
8	248	velocityZSigma	double	Standard deviation of velocity in Z direction [m/s^2].
8	256	rollRate	double	Current roll rate [rad/s]
8	264	rollRateSigma	double	
8	272	yawRate	double	Current yaw rate of vehicle [rad/s]
8	280	yawRateSigma	double	

8	288	pitchRate	double	Current pitch rate [rad/s]
8	296	pitchRateSigma	double	
8	304	gpsStatus	double	Status of the GPS
1	312	noOfSatellites	UINT8	Gives the total number of satellites involved.
8	313	gpsDilutionOfPrecisionX	double	Precision in X
8	321	gpsDilutionOfPrecisionY	double	Precision in Y
8	329	gpsDilutionOfPrecisionZ	double	Precision in Z
8	337	gpsDilutionOfPrecisionVertical	double	Precision in vertical direction
8	345	gpsDilutionOfPrecisionPosition	double	Precision in position
	353	timestamp	Timestamp	Unique timestamp associated with each measurement signal.
1	395	Source Enum	UINT8	Enum fields: 0:Source_CAN 1:Source_XSensIMU 2:Source_OxtsRt 3:Source_GenesysADMA 4:Source_SpatialDual 5:Source_TFC 6 :Source_VBOX3i 99:Source_Unknown
1	396	INSQuality	UINT8	Enum fields: 0: Unknown 1: INSQuality_GPS (only GPS) 2: INSQuality_DGPSCourse (pre-RTK) 3: INSQuality_DGPSPrecise (RTK)
16	397	reserved	UINT8[16]	Reserved fields

Bytes	Offset	Timestamp	Data type	Description

8	0	measurementTimeECU	NTP64	Time of measurement as represented in the ECU timebase.
8	8	receivedTimeECU	NTP64	Time when received in the system as represented in the ECU timebase.
8	16	rawDeviceTime	NTP64	Raw device time from the header.
8	24	measurementTime	NTP64	Time of measurement as represented in the timebase of external clock.
8	32	receivedTime	NTP64	Time received as represented in the timebase of external clock.
2	40	clockType	ClockType	Properties of external clock which provides timebase for measurement and received times listed above.

Bytes	Offset	ClockType	Data type	Description
1	0	ClockID	UINT8	Unique ID representing each clock.
1	1	ClockName	UINT8	Enum fields : 0:Clock_Unkown 1:Clock_Laserscanner 2 :Clock_ECU 3 :Clock_CanBus 4 :Clock_Camera 5 :Clock_GpsImu 6 :Clock_Other

Bytes	Offset	NTP64	Data type	Description
8	0	UINT64	4 Most Significant Bytes: seconds 4 Least Significant Bytes: fractionalSeconds	Number of seconds elapsed since 1900-01-01 00:00:00

				Fractional part: 1/(2^32) seconds fragment

34 Odometry: Data type 0x9002

The byte order for data type 0x9002 is Big Endian.
This data type contains unprocessed motion data, such as the velocity and the yaw rate of the vehicle that is provided to the system via CAN.

Bytes	Offset	Object header	Data type	Description
8	0	steeringAngle	double	Angle at which the vehicle is being steered.
8	8	steeringWheelAngle	double	Angle by which the steering wheel is rotated compared to its middle position. [0.001rad]
8	16	steeringWheelAngleVelocity	double	Velocity of the steering wheel angle
8	24	wheelSpeedFL	double	Speed of Front Left wheel [m/s]
8	32	wheelSpeedFR	double	Speed of Front Right wheel [m/s]
8	40	wheelSpeedRL	double	Speed of Rear Left wheel [m/s]
8	48	wheelSpeedRR	double	Speed of Rear Right wheel [m/s]
8	56	wheelCircumference	double	Circumference of the wheel
8	64	vehVelocity	double	Current velocity of the vehicle
8	72	vehAcceleration	double	Current acceleration of the vehicle [m/s$^{2]}$
8	80	vehYawRate	double	Current yaw rate of the vehicle [rad/s]
	88	timestamp	Timestamp	Unique timestamp associated with each measurement signal.
16		reserved	UINT32[4]	Reserved fields

Bytes	Offset	Timestamp	Data type	Description
8	0	measurementTimeECU	NTP64	Time of measurement as represented in the ECU timebase.
8	8	receivedTimeECU	NTP64	Time when received in the system as represented in the ECU timebase.
8	16	rawDeviceTime	NTP64	Raw device time from the header.
8	24	measurementTime	NTP64	Time of measurement as represented in the timebase of external clock.
8	32	receivedTime	NTP64	Time received as represented in the timebase of external clock.
2	40	clockType	ClockType	Properties of external clock which provides timebase for measurement and received times listed above.

Bytes	Offset	ClockType	Data type	Description
1	0	ClockID	UINT8	Unique ID representing each clock.
1	1	ClockName	UINT8	Enum fields : 0:Clock_Unkown 1:Clock_Laserscanner 2 :Clock_ECU 3 :Clock_CanBus 4 :Clock_Camera 5 :Clock_GpsImu 6 :Clock_Other

Bytes	Offset	NTP64	Data type	Description
8	0	UINT64	4 Most Significant Bytes: seconds	Number of seconds elapsed since 1900-01-01 00:00:00

			4 Least Significant Bytes: fractionalSeconds	Fractional part: 1/(2^32) seconds fragment

35 SystemMonitoringCANStatus: Data type 0x6700

The byte order for data type 0x6700 is Big Endian.
The SystemMonitoringCANStatus data type provides information about registered CAN frames. The registration itself is done in the Worker.xml configuration file in SystemMonitoringWorker configuration block. If registered CAN frames are not received by the perception ECU anymore, this datatype is also used to signal the loss of these CAN frames.

Bytes	Offset	Object header	Data type	Description
1	0	DeviceID	UINT8	Device ID of connected device
4	1	MessageID	UINT32	CAN message identifier
1	5	State	UINT8	State of the CAN message
2	6	StateInformation	UINT8	Detailed information about the CAN message state
8	8	updateMissing	UINT64	Timestamp (in ms since 01.01.1970) of the last system update

Object header	Data type	Value	State
DeviceState	UINT8	0	Initialization
		1	OK
		2	Warning
		3	Error

Object header	Data type	Bit	State
StateInformation	UINT8	0	Initialization state: 1 if not enough signals have been received for evaluation, 0 otherwise
		1	reserved
		2	reserved
		3	No state change: 1 if an error state was present in the past, 0 otherwise
		4	No state change: 1 if signal drop rate is too high (Default: 25 %)
		5	No state change: 1 if signal jitter is too high (Default: 50 %)

		6	reserved
		7	Error state: 1 if the CAN message currently has a timeout, 0 otherwise

36 SystemMonitoringDeviceStatus: Data type 0x6701

The byte order for data type 0x6701 is Big Endian.
The SystemMonitoringDeviceStatus data type provides information about connected devices: Scanner, Cameras, GPS, IMU. If devices have lost connection to the perception ECU, this datatype is also used to signal the loss of these devices.

Size	Offset	Object header	Data type	Description
1	0	DeviceID	UINT8	Device ID of connected device
1	1	DeviceType	UINT8	Device type of the connected device
1	2	TypeInformation	UINT8	Further type information about the connected device
1	3	State	UINT8	State of the device
2	4	StateInformation	UINT8	Detailed information about the device state
8	6	updateMissing	UINT64	Timestamp (in ms since 01.01.1970) of the last system update

Object header	Data type	Value	State
DeviceType	UINT8	0	Unknown
		1	Scanner
		2	Camera
		4	WGS84
		5	GPS
		6	IMU

Object header	Data type	Device type	Value	State
DeviceTypeInformation	UINT8	Scanner	0	undefined
		Scanner	1	ibeo LUX
		Scanner	2	ibeo ScaLa
		Scanner	100	Sick LD-MRS
		Scanner	101	Sick LMS
		Camera	0	undefined
		CAN	0	undefined

		WGS84	0	undefined
		GPS	0	undefined
		IMU	0	Undefined
		IMU	1	XSens
		IMU	2	OXTS
		IMU	3	Genesys

Object header	Data type	Value	State
State	UINT8	0	Initialization
		1	OK
		2	Warning
		3	Error

Object header	Data type	Bit	State
StateInformation	UINT8	0	Initialization state: 1 if not enough signals have been received for evaluation (<25 default), 0 otherwise
		1	reserved
		2	reserved
		3	No state change: 1 if an error state was present in the past, 0 otherwise
		4	No state change: 1 if signal drop rate is too high (Default: 25 %)
		5	No state change: 1 if signal jitter is too high (Default: 50 %)
		6	reserved
		7	Error state: 1 if the device currently has a timeout, 0 otherwise

37 SystemMonitoringSystemStatus: Data type 0x6705

The byte order for data type 0x6705 is Big Endian.
The SystemsMonitoringSystemStatus data type provides information about the current state of the perception ECU combined with the information about the entire fusion system.

Size	Offset	Object header	Data type	Description
8	0	lastUpdateTimestamp	UINT64	Timestamp (in ms since 01.01.1970) of the last system update
1	8	State	UINT8	State of the entire system
1	9	StateInformation	UINT8	Detailed information about the entire system
1	10	CurrentCPUUsage	UINT8	for current CPU usage (0 .. 100)
1	11	CurrentRAMUsage	UINT8	for current RAM usage (0 .. 100)
1	12	CurrentHDDUsage	UINT8	for current HDD usage (0 .. 100)
1	13	MemoryWarningLevelPercent	UINT8	Warning level of memory consumption, percent. Default: 80 %
1	14	MemoryErrorLevelPercent	UINT8	Error level of memory consumption, percent. Default: 95 %

Object header	Data type	Value	State
State	UINT8	0	Initialization
		1	OK
		2	Warning
		3	Error

Object header	Data type	Bit	State
StateInformation	UINT8	0	No state change: 1 if the latency of the System Monitor Worker is too high (> 1 second default), 0 otherwise
		1	reserved
		2	reserved
		3	Warning state: 1 if memory usage has reached warning level, 0 otherwise.
		4	reserved

		5	Error state: 1 if one or more devices/CAN messages have timeout, 0 otherwise
		6	Error state: 1 if memory usage has reached error level, 0 otherwise
		7	reserved