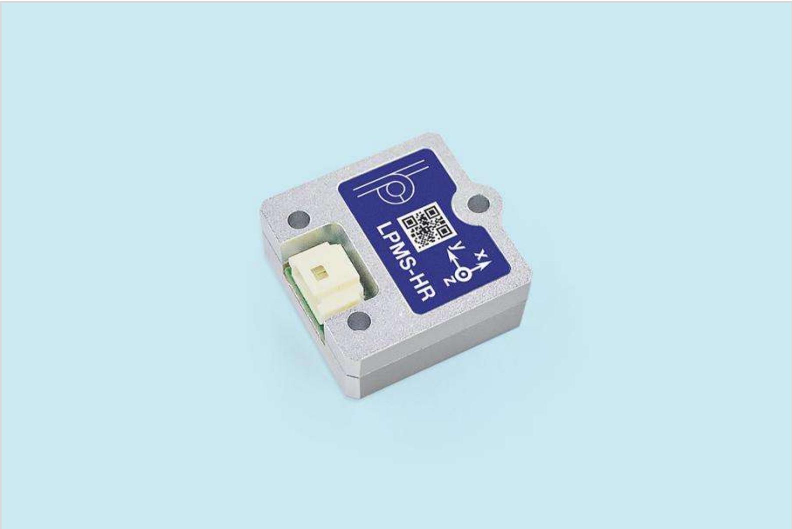
LPMS-HR Series
User Manual ver. 1.1



LP-Research Inc.

https://www.lp-research.com

Version History

Date	Version	Details
2025-10-11	ver. 1.0	• Initial release
2026-03-06	ver. 1.1	• Updated application fields

Table of Contents

1. Introduction

The LPMS-HR Series is a high-precision attitude sensor specifically developed for the robotics field. It integrates a 3-axis accelerometer and a 3-axis gyroscope. Through calibration and calculation using our proprietary algorithms, it provides precise data including Euler angles, quaternions, and acceleration information. Additionally, the module is compact and easy to assemble, facilitating user integration into their own systems for development and design.

Main Features:

- MEMS Micro Inertial Measurement Unit (IMU)
- Integrated 3-axis gyroscope, 3-axis accelerometer
- Real-time calculation of sensor attitude orientation, linear acceleration, angular velocity, and other data
- Power Input: 5V(USB), 5~36V(RS232/TTL)
- Communication Interfaces: USB, RS232, TTL
- Mechanical Dimensions: 22.4 x 22.4 x 11 mm

Application Fields:

- Mobile Robots
- Humanoid Robots
- Construction Machinery
- Platform Stabilization and Control

Table 1-1 LPMS-HR Series Product Model Overview

Product Model	**Communication Interface**		
	USB	**RS232**	**TTL**
LPMS-HR USB	✓	×	×
LPMS-HR RS232	×	✓	×
LPMS-HR TTL	×	×	✓

2. Output Characteristic Definition

2.1. Coordinate System

The LPMS-HR series coordinate system is a right-hand Cartesian coordinate system, corresponding to the coordinate system marked on the product. The directions of acceleration and gyroscope rotation are shown in Figure 2.1.

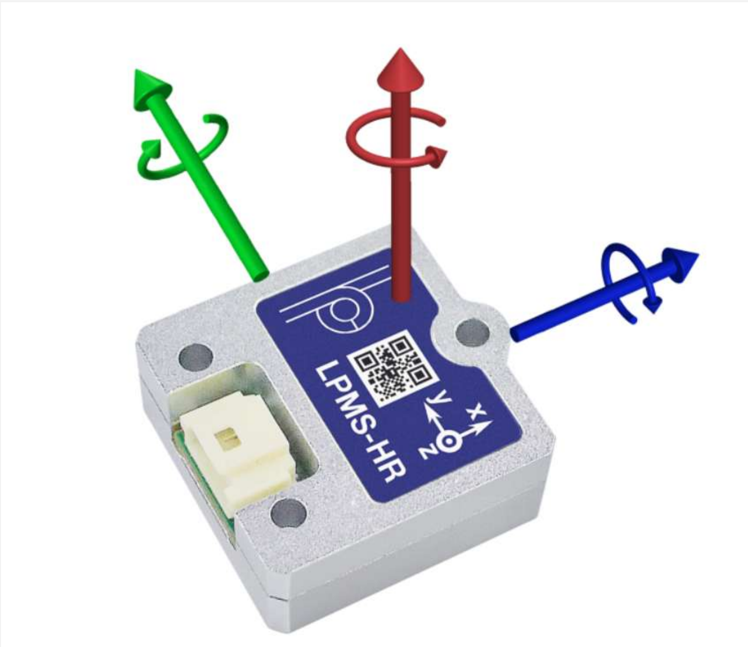


Fig. 2.1. LPMS-HR series reference coordinate system

2.2. Output Data Definition

The output data definitions for the LPMS-HR series sensor are as follows:

- **Acceleration:** Calibrated acceleration data outputs as [0,0,1g]
- **Gyroscope:** Calibrated gyroscope data is near zero.
- **Quaternion:** Quaternion output data is in the form q = (w, x, y, z)
- **Euler Angles:**

Ψ = yaw = Angle of rotation around the Z-axis, range -180°~180°

Θ = pitch = Angle of rotation around the Y-axis, range -90°~90°

Φ = roll = Angle of rotation around the X-axis, range -180°~180°

3. Communication Protocol

3.1. Communication Modes

The LPMS devices have two communication modes: Streaming Mode and Command Mode. In Streaming Mode, the sensor continuously sends data at a preset frequency (the data type and format can be configured). In Command Mode, communication with the sensor is done by sending commands, allowing configuration of sensor parameters. **The LPMS-HR series ships in Streaming Mode,** and it is possible to perform operations to save settings while in this mode.

Figure 3.1 shows the process for configuring sensor parameters. Typically, the sensor must first be put into Command Mode to modify or read parameters [1].

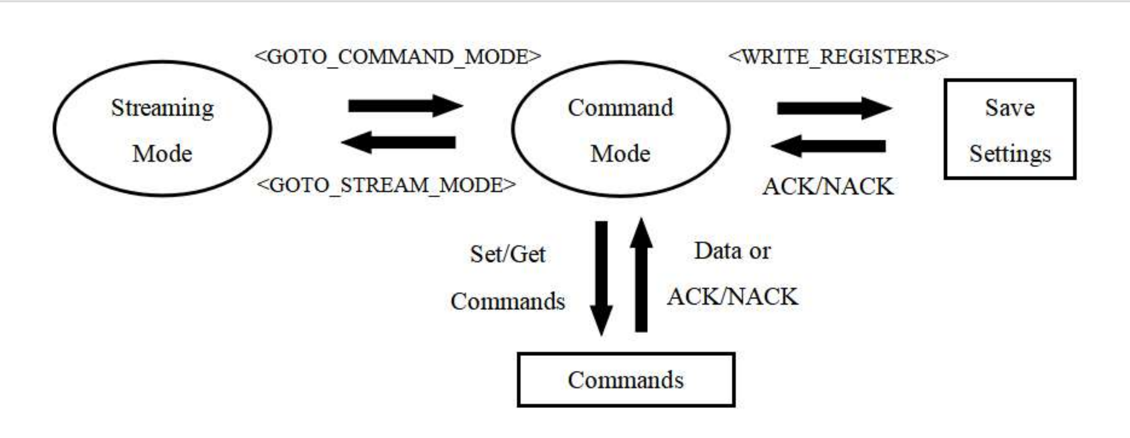


Fig. 3.1. Sensor parameter configuration process

3.2. LPBUS Protocol

The LPBUS is a communication protocol designed based on the industry standard MODBUS protocol. It is the default communication format used by LPMS sensors. Please refer to section 3.3 for the specific structure of the communication protocol packet.

An LPBUS communication packet has two basic command types, GET and SET, that are sent from a host (PC, mobile data logging unit etc.) to a client (LPMS sensor). Later in this manual, we will show a description of all supported commands to the sensor, their types and transported data.

GET Command: Reading data from the slave is achieved by sending a GET command. A GET request command typically does not contain any data; the requested

[Note 1] After modifying parameters in Command Mode, you must save the settings; otherwise, the changes will be lost after power cycling.

data is sent by the slave after receiving the GET command.

SET Command: Values in the slave's data registers are set by sending a SET command. A SET command contains the data to be set. The slave's return value is either ACK (indicating successful register write) or NACK (indicating register write failure).

Please refer to the appendix for the detailed command list.

3.3. LPBUS Packet Format

The structure of each data packet in the LPBUS communication protocol is shown in Table 3-1.

Table 3-1 Packet structure

Byte#	Name	Description
0	Packet Header	3Ah
1	Sensor ID (Low Byte)	Contains the low byte of the sensor's ID number. The default value of this ID is 01h. The host sends out a GET / SET request to a specific LPMS sensor by using this ID, and the client answers to the request also with the same ID. This ID can be adjusted by sending a SET command to the sensor firmware.
2	Sensor ID (High Byte)	Contains the high byte of the sensor's ID number.
3	Command No. (Low Byte)	Contains the low byte of the command number to be executed.
4	Command No. (High Byte)	Contains the high byte of the command number to be executed.
5	Data length (Low Byte)	Contains the low byte of the data length to be transmitted.
6	Data length (High Byte)	Contains the high byte of the data length to be transmitted.
x	Data (n bytes)[2]	If the data length **n** is not 0, then **x** = 6 + **n**; if **n** is 0, **x** is empty. If the data length is not 0, this field contains the data packet to be transmitted; otherwise, it is empty.
7+*n*	LRC (Low Byte)	Low byte of the LRC checksum. To ensure data

[Note 2] The data portion is typically transmitted in little-endian format, i.e., low byte first, high byte last.

		integrity during transmission, we use the LRC checksum method, calculated as follows: LRC = sum(Sensor ID, Command no., Data length, and Data field). The calculated LRC is usually compared with the LRC transmitted from the remote device. If the two LRCs are not equal, and an error is reported.
8+n	LRC (High Byte)	High byte of the LRC checksum.
9+n	Packet Footer (Low Byte)	0Dh
10+n	Packet Footer (High Byte)	0Ah

The data part of the packet can use either 32-bit floating point(default data format) or 16-bit signed integer format. When using the 16-bit signed integer format, the data is first multiplied by a scale factor before being packed and transmitted to improve transmitted, and the transmission order, please refer to Tables 3-2, 3-3, and 3-4.

Table 3-2 Data format identifier definitions

Identifier	Description
UInt32	32-bit unsigned integer value
Int32	32-bit signed integer value
Int16	16-bit signed integer value
Float32	32-bit float value
Vector3f[3]	3-element 32-bit float vector
Vector3i16[3]	3 element 16-bit signed integer vector
Vector4f[4]	4-element 32-bit float vector
Vector4i16[4]	4 element 16-bit signed integer vector
Matrix3x3f	3x3 element 32-bit float value matrix

[Note 3] When data is output as a 3-elenent vector, it is arranged in X, Y, Z order.
[Note 4] When data is output as a 4-elenent vector, it is arranged in W, X, Y, Z order.

Table 3-3 Data format of data part in packet (32-bit float mode)

Order[5]	Data Format Identifier	Sensor Data
1	UInt32	Timestamp (1000Hz update rate, i.e., 0.001s)
2	Vector3f	Raw accelerometer data (g)
3	Vector3f	Calibrated accelerometer data (g)
4	Vector3f	Raw gyroscope data (**dps (default)** / rad/s)
5	Vector3f	Bias calibrated gyroscope data (**dps (default)** / rad/s)
6	Vector3f	Align. calibrated gyroscope data (**dps (default)** / rad/s)
7	Vector3f	Angular Vel. (rad/s)
8	Vector4f	Quaternion (normalized unit)
9	Vector3f	Euler angle data(**degree (default)** / rad)
10	Vector3f	Linear accelerometer (g)
11	Vector3f	Temperature (℃)

Table 3-4 Data format of data part in packet (16-bit signed integer mode)

Order[5]	Data Format Identifier	Sensor Data	Scale Factor
1	UInt32	Timestamp (1000Hz update rate, i.e., 0.001s)	-
2	Vector3i16	Raw accelerometer data (g)	1000
3	Vector3i16	Calibrated accelerometer data (g)	1000
4	Vector3i16	Raw gyroscope data (**dps (default)** / rad/s)	**10**/100
5	Vector3i16	Bias calibrated gyroscope data (**dps (default)** / rad/s)	**10**/100
6	Vector3i16	Align. calibrated gyroscope data (**dps (default)** / rad/s)	**10**/100
7	Vector3i16	Angular Velocity (rad/s)	100
8	Vector4i16	Quaternion (normalized unit)	10000
9	Vector3i16	Euler angle data(**degree (default)** / rad)	**100**/10000
10	Vector3i16	Linear accelerometer (g)	1000
11	Vector3i16	Temperature (℃)	100

[Note 5] Data is output sequentially in this order by default. If a particular data item is not enabled for output, the subsequent enabled data items will shift forward accordingly. For example, if the data item at position 2 is set not to output, the data item originally at position 3 becomes the new position 2.

3.4. Data Conversion Examples

The follwing are examples of parsing specific data packets.

Default Output Data Types in 32-bit Float Mode

If the received data packet is (hex representation):
3A 01 00 09 00 54 00 6E F7 02 00 00 40 F2 3D 00 E0 C4 BE 00 78 6A BF 40 E3 66 3C 0E C6 A1 BE 5A 97 BC 3E 1A 37 84 BE E4 73 30 BE 67 5D FE BD B1 6F 41 3E 04 44 79 BF 7F 65 CB 3D 28 7C A3 BD A4 44 1D C3 74 BA D8 C0 1C 2C 50 C1 1A 37 84 BE E4 73 30 BE 67 5D FE BD 00 00 E3 41 E2 27 0D 0A
The parsing is as follows:

Description	Received Data	Raw Data	Converted Data	Scale Factor	Actual Value	Unit
Packet Header	3A					
Sensor ID	01 00	0001	1			
Command Number	09 00	0009	9			
Data Length	54 00	0054	84			
Timestamp	6E F7 02 00	0002F76E	194414	0.001	194.414	Sec.
Calibrated Acc X	00 40 F2 3D	3DF24000	0.118286	1	0.118286	g
Calibrated Acc Y	00 E0 C4 BE	BEC4E000	-0.384521	1	-0.384521	g
Calibrated Acc Z	00 78 6A BF	BF6A7800	-0.915894	1	-0.915894	g
Align. Calibrated Gyro X	40 E3 66 3C	3C66E340	0.014092	1	0.014092	dps
Align. Calibrated Gyro Y	0E C6 A1 BE	BEA1C60E	-0.315964	1	-0.315964	dps
Align. Calibrated Gyro Z	5A 97 BC 3E	3EBC975A	0.368342	1	0.368342	dps
Angular Vel. X	1A 37 84 BE	BE84371A	-0.258233	1	-0.258233	dps

Angular Vel. Y	E4 73 30 BE	BE3073E4	-0.172317	1	-0.172317	dps
Angular Vel. Z	67 5D FE BD	BDFE5D67	-0.124202	1	-0.124202	dps
Quaternion W	B1 6F 41 3E	3E416FB1	0.188903	1	0.188903	-
Quaternion X	04 44 79 BF	BF794404	-0.973694	1	-0.973694	-
Quaternion Y	7F 65 CB 3D	3DCB657F	0.099315	1	0.099315	-
Quaternion Z	28 7C A3 BD	BDA37C28	-0.079827	1	-0.079827	-
Euler Angle X	A4 44 1D C3	C31D44A4	-157.268127	1	-157.268127	deg
Euler Angle Y	74 BA D8 C0	C0D8BA74	-6.772760	1	-6.772760	deg
Euler Angle Z	1C 2C 50 C1	C1502C1C	-13.010769	1	-13.010769	deg
Linear Acc X	1A 37 84 BE	BE84371A	-0.258233	1	-0.258233	g
Linear Acc Y	E4 73 30 BE	BE3073E4	-0.172317	1	-0.172317	g
Linear Acc Z	67 5D FE BD	BDFE5D67	-0.124202	1	-0.124202	g
Temperature	00 00 E3 41	41E30000	28.375	1	28.375	℃
LRC Checksum*	E2 27	27 E2				
Packet Footer	0D 0A					

*LRC Checksum calculation:
01+00+09+00+54+00+6E+F7+02+00+00+40+F2+3D+00+E0+C4+BE+00+78+6A+BF+40+E3+66+3C+0E+C6+A1+BE+5A+97+BC+3E+1A+37+84+BE+E4+73+30+BE+67+5D+FE+BD+B1+6F+41+3E+04+44+79+BF+7F+65+CB+3D+28+7C+A3+BD+A4+44+1D+C3+74+BA+D8+C0+1C+2C+50+C1+1A+37+84+BE+E4+73+30+BE+67+5D+FE+BD+00+00+E3+41= 27E2

Default Output Data Types in 16-bit Signed Integer Mode

If the received data packet is (hex representation):
3A 01 00 09 00 2C 00 67 00 51 00 17 00 BF FD CB FC 03 00 06 00 FD FF FF FF 00 00 00 00 DC 26 B9 02 5B 01 91 02 39 03 57 01 0C 03 CA FE 34 FF 8E FF 74 0C E2 10 0D 0A
The parsing is as follows:

Description	Received Data	Raw Data	Converted Data	Scale Factor	Actual Value	Unit
Packet Header	3A					
Sensor ID	01 00	0001	1			
Command Number	09 00	0009	9			
Data Length	2C 00	002C	44			
Timestamp	67 00 51 00	00510067	5308519	0.001	5308.519	Sec.
Calibrated Acc X	17 00	0017	23	1/1000	0.023	g
Calibrated Acc Y	BF FD	FDBF	-577	1/1000	-0.577	g
Calibrated Acc Z	CB FC	FCCB	-821	1/1000	-0.821	g
Align. Calibrated Gyro X	03 00	0003	3	1/10	0.3	dps
Align. Calibrated Gyro Y	06 00	0006	6	1/10	0.6	dps
Align. Calibrated Gyro Z	FD FF	FFFD	-3	1/10	-0.3	dps
Angular Vel. X	FF FF	FFFF	-1	1/100	0.01	dps
Angular Vel. Y	00 00	0000	0	1/100	0.00	dps
Angular Vel. Z	00 00	0000	0	1/100	0.00	dps
Quaternion W	DC 26	26DC	9948	1/10000	0.9948	-
Quaternion X	B9 02	02B9	697	1/10000	0.0697	-
Quaternion Y	5B 01	015B	347	1/10000	0.0347	-
Quaternion Z	91 02	0291	657	1/10000	0.0657	-
Euler Angle X	39 03	0339	825	1/100	8.25	deg
Euler Angle Y	57 01	0157	343	1/100	3.43	deg
Euler Angle Z	0C 03	030C	780	1/100	7.8	deg

Linear Acc X	CA FE	FECA	-310	1/1000	-0.31	g
Linear Acc Y	34 FF	FF34	-204	1/1000	-2.04	g
Linear Acc Z	8E FF	FF8E	-114	1/1000	-1.14	g
Temperature	74 0C	0C74	3188	1/100	31.88	℃
LRC Checksum*	E2 10	10E2				
Packet Footer	0D 0A					

*LRC Checksum calculation:
01+00+09+00+2C+00+67+00+51+00+17+00+BF+FD+CB+FC+03+00+06+00+FD+FF+FF+FF+00+00+00+00+DC+26+B9+02+5B+01+91+02+39+03+57+01+0C+03+CA+FE+34+FF+8E+FF+74+0C=10E2

3.5. LPBUS Communication Examples

In this section, we will show a few practical examples of communication using the LPBUS protocol.

Goto Command Mode

Get request (Host -> Sensor)

Packet byte no.	Value	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	06h	Command no. LSB (06h = GOTO_COMMAND_MODE)
4	00h	Command no. MSB
5	00h	Data length LSB (GOTO_COMMAND_MODE data length is 0)
6	00h	Data length MSB
7	07h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet footer LSB
10	0Ah	Packet footer MSB

Reply data (Sensor -> Host)

Packet byte no.	Value	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	00h	Command no. LSB (00h = REPLY_ACK)
4	00h	Command no. MSB
5	00h	Data length LSB (ACK returned, no data)
6	00h	Data length MSB
7	01h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet footer LSB
10	0Ah	Packet footer MSB

Goto Streaming Mode

Get request (Host -> Sensor)

Packet byte no.	Value	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	07h	Command no. LSB (07h = GOTO_STREAMING_MODE)
4	00h	Command no. MSB
5	00h	Data length LSB (GOTO_STREAMING_MODE data length is 0)
6	00h	Data length MSB
7	08h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet footer LSB
10	0Ah	Packet footer MSB

Reply data (Sensor -> Host)

Packet byte no.	Value	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	00h	Command no. LSB (00h = REPLY_ACK)
4	00h	Command no. MSB
5	00h	Data length LSB (ACK returned, no data)
6	00h	Data length MSB
7	01h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet footer LSB
10	0Ah	Packet footer MSB

Request Gyroscope Range

Get request (Host -> Sensor)

Packet byte no.	Value	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	3Dh	Command no. LSB (3Dh = GET_GYR_RANGE)
4	00h	Command no. MSB
5	00h	Data length LSB (GET_GYR_RANGE data length is 0)
6	00h	Data length MSB
7	3Eh	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet footer LSB
10	0Ah	Packet footer MSB

Reply data (Sensor -> Host)

Packet byte no.	Value	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	3Dh	Command no. LSB (3Dh = GET_GYR_RANGE)
4	00h	Command no. MSB
5	04h	Data length LSB (32-bit integer is 4 bytes)
6	00h	Data length MSB
7	xxh[[6]	Range data byte 1 (LSB)
8	xxh[6]	Range data byte 2
9	xxh[6]	Range data byte 3
10	xxh[6]	Range data byte 4 (MSB)
11	xxh[6]	Check sum LSB
12	xxh[6]	Check sum MSB
13	0Dh	Packet footer LSB
14	0Ah	Packet footer MSB

[Note 6]: xx = Value depends on the current sensor configuration.

Set Accelerometer Range

Get request (Host -> Sensor)

Packet byte no.	Value	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	32h	Command no. LSB (32h = SET_ACC_RANGE)
4	00h	Command no. MSB
5	04h	Data length LSB (32-bit integer is 4 bytes)
6	00h	Data length MSB
7	08h	Range data byte 1 (Range indicator 8g = 8d)
8	00h	Range data byte 2
9	00h	Range data byte 3
10	00h	Range data byte 4
11	3Fh	Check sum LSB
12	00h	Check sum MSB
13	0Dh	Packet footer LSB
14	0Ah	Packet footer MSB

Reply data (Sensor -> Host)

Packet byte no.	Value	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	00h	Command no. LSB (00h = REPLY_ACK)
4	00h	Command no. MSB
5	00h	Data length LSB (ACK returned, no data)
6	00h	Data length MSB
7	01h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet footer LSB
10	0Ah	Packet footer MSB

Save Sensor Parameters

Get request (Host -> Sensor)[7]

Packet byte no.	Value	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	04h	Command no. LSB (04h =WRITE_REGISTERS)
4	00h	Command no. MSB
5	00h	Data length LSB (WRITE_REGISTERS data length is 0)
6	00h	Data length MSB
7	05h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet footer LSB
10	0Ah	Packet footer MSB

Reply data (Sensor -> Host)

Packet byte no.	Value	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	00h	Command no. LSB (00h = REPLY_ACK)
4	00h	Command no. MSB
5	00h	Data length LSB (ACK returned, no data)
6	00h	Data length MSB
7	01h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet footer LSB
10	0Ah	Packet footer MSB

[Note 7] Saving modified parameters takes some time. Data may not return immediately after sending this command; it takes about 1~2 seconds for data to return.

Reading Sensor Status

Get request (Host -> Sensor)

Packet byte no.	Value	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	08h	Command no. LSB (08h =GET_SENSOR_STATUS)
4	00h	Command no. MSB
5	00h	Data length LSB (GET_SENSOR_STATUS data length is 0)
6	00h	Data length MSB
7	09h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet footer LSB
10	0Ah	Packet footer MSB

Reply data (Sensor -> Host)[8]

Packet byte no.	Value	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	08h	Command no. LSB (08h =GET_SENSOR_STATUS)
4	00h	Command no. MSB
5	04h	Data length LSB
6	00h	Data length MSB
7-10	xxxxxxxxh	Sensor status data
11	xxh	Check sum LSB
12	xxh	Check sum MSB
13	0Dh	Packet footer LSB
14	0Ah	Packet footer MSB

[Note 8] Please refer to Appendix for details of reply data mapping.

Set UART Baudrate

Get request (Host -> Sensor)[9]

Packet byte no.	Value	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	82h	Command no. LSB (82h =SET_UART_BAUDRATE)
4	00h	Command no. MSB
5	04h	Data length LSB
6	00h	Data length MSB
7	00h	921600 = 0x000E1000
8	10h	
9	0Eh	
10	00h	
11	A5h	Check sum LSB
12	00h	Check sum MSB
13	0Dh	Packet footer LSB
14	0Ah	Packet footer MSB

Reply data (Sensor -> Host)

Packet byte no.	Value	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	00h	Command no. LSB (0d = REPLY_ACK)
4	00h	Command no. MSB
5	00h	Data length LSB (ACK returned, no data)
6	00h	Data length MSB
7	01h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet footer LSB
10	0Ah	Packet footer MSB

[Note 9] The baudrate change takes effect only after a restart.

3.6. ASCII Output Format

The UART communication interface of the LPMS-HR series supports ASCII output, configurable via the SET_UART_FORMAT command. In ASCII output format[10], the data is first multiplied by a scale factor (refer to Table 3-4; the factor for gyroscope-related data is **100**/1000), and then converted to an integer string for output. The output format is shown in Table 3-5.

Table 3-5 ASCII output format

Start character	Data1	,	Data 2	,	......	,	Data n	End character

The default start character is '$' and end character is '\n'. User can define both Start and End character via SET_UART_ASCII_CHARACTER command. Sensor data is streamed out in comma separated format.

For example, under **default settings**, the following string data is output, parsed as shown in Table 3-6:

Table 3-6 ASCII data packet parsing[11]

Data Item	Characters	Value
Start Character	$	-
Timestamp	31020	~ 31.02s
Raw Acc data X	-26	-0.026g
Raw Acc data Y	10	0.01g
Raw Acc data X	995	0.995g
......	......	
Raw Gyro data X	30	0.3dps
Raw Gyro data Y	6	0.06dps
Raw Gyro data Z	-12	-0.12dps
......	......	
Quaternion w	9996	0.9996
Quaternion x	59	0.0059
Quaternion y	142	0.0142
Quaternion z	-194	-0.0193
Euler Angle Roll	65	0.65°
Euler Angle Pitch	165	1.65°
Euler Angle Yaw	-221	-2.21°

[Note 10] In ASCII output format, user commands are still responded to using the LPBUS protocol format.
[Note 11] The data output in ASCII is converted based on 32-bit floating point data.

End Character	\n	"\n" is the newline character; if set to another character like "#", the string ends with "#".

4. Appendix

Summary

Acknowledged / Not-acknowledged Identifiers				
Identifier	**Name**	**Parameter**	**Response**	**Default**
0(00h)	REPLY_ACK			
1(01h)	REPLY_NACK			

Register Value Save and Reset Command				
Identifier	**Name**	**Parameter**	**Response**	**Default**
4 (04h)	WRITE_REGISTERS	NONE	ACK/NACK	
5 (05h)	RESTORE_FACTORY_VALUE	NONE	ACK/NACK	

Mode Switching Commands				
Identifier	**Name**	**Parameter**	**Response**	**Default**
6 (06h)	GOTO_COMMAND_MODE	NONE	ACK/NACK	
7 (07h)	GOTO_STREAM_MODE	NONE	ACK/NACK	

Sensor Status Command				
Identifier	**Name**	**Parameter**	**Response**	**Default**
8 (08h)	GET_SENSOR_STATUS	NONE	UInt32	1

Get Data Commands				
Identifier	**Name**	**Parameter**	**Response**	**Default**
9 (09h)	GET_IMU_DATA	NONE		

Device Info				
Identifier	**Name**	**Parameter**	**Response**	**Default**
20 (14h)	GET_SENSOR_MODEL	NONE	Char[24]	
21 (15h)	GET_FIRMWARE_INFO	NONE	Char[24]	
22 (16h)	GET_SERIAL_NUMBER	NONE	Char[24]	
23 (17h)	GET_FILTER_VERSION	NONE	Char[24]	

Data Transmission Commands				
Identifier	**Name**	**Parameter**	**Response**	**Default**
30 (1Eh)	SET_IMU_TRANSMIT_DATA	UInt32	ACK/NACK	
31 (1Fh)	GET_IMU_TRANSMIT_DATA	NONE	UInt32	

IMU ID Settings Commands				
Identifier	**Name**	**Parameter**	**Response**	**Default**
32 (20h)	SET_IMU_ID	Int32	ACK/NACK	
33 (21h)	GET_IMU_ID	NONE	Int32	1

Stream Frequency Commands				
Identifier	**Name**	**Parameter**	**Response**	**Default**
34 (22h)	SET_STREAM_FREQ	Int32	ACK/NACK	
35 (23h)	GET_STREAM_FREQ	NONE	Int32	100

Deg/Rad Output Commands				
Identifier	**Name**	**Parameter**	**Response**	**Default**
36 (24h)	SET_DEGRAD_OUTPUT	Int32	ACK/NACK	
37 (25h)	GET_DEGRAD_OUTPUT	NONE	Int32	0

Reference Setting and Offset Reset Commands				
Identifier	**Name**	**Parameter**	**Response**	**Default**
38 (26h)	SET_ORIENTATION_OFFSET	Int32	ACK/NACK	
39 (27h)	RESET_ORIENTATION_OFFSET	NONE	ACK/NACK	

Accelerometer Settings Commands				
Identifier	**Name**	**Parameter**	**Response**	**Default**
50 (32h)	SET_ACC_RANGE	Int32	ACK/NACK	
51 (33h)	GET_ACC_RANGE	NONE	Int32	12g

Gyroscope Settings Commands				
Identifier	**Name**	**Parameter**	**Response**	**Default**
60 (3Ch)	SET_GYR_RANGE	Int32	ACK/NACK	
61 (3Dh)	GET_GYR_RANGE	NONE	Int32	2000dps
62 (3Eh)	START_GYR_CALIBRATION	NONE	ACK/NACK	
64 (40h)	SET_ENABLE_GYR_AUTOCALIBRATION	Int32	ACK/NACK	
65 (41h)	GET_ENABLE_GYR_AUTOCALIBRATION	NONE	Int32	1
66(42h)	SET_GYR_THRESHOLD	Float32	ACK/NACK	
67(43h)	GET_GYR_THRESHOLD	NONE	Float32	0

Filter Settings Commands				
Identifier	**Name**	**Parameter**	**Response**	**Default**
90 (5Ah)	SET_FILTER_MODE	Int32	ACK/NACK	
91 (5Bh)	GET_FILTER_MODE	NONE	Int32	1
92 (5Ch)	SET_STATIC_SHUTDOWN	Int32	ACK/NACK	
93 (5Dh)	GET_STATIC_SHUTDOWN	NONE	Int32	1

UART Settings Command				
Identifier	**Name**	**Parameter**	**Response**	**Default**
130 (82h)	SET_UART_BAUDRATE	Int32	ACK/NACK	
131 (83h)	GET_UART_BAUDRATE	NONE	Int32	115200
132 (84h)	SET_UART_FORMAT	Int32	ACK/NACK	
133 (85h)	GET_UART_FORMAT	NONE	Int32	0
134 (86h)	SET_UART_ASCII_CHARACTER	Int8[4]	ACK/NACK	
135 (87h)	GET_UART_ASCII_CHARACTER	NONE	Int8[4]	‘$’‘\n’
136 (88h)	SET_LPBUS_DATA_PRECISION	Int32	ACK/NACK	
137 (89h)	GET_LPBUS_DATA_PRECISION	NONE	Int32	1

SensorData Timestamp Manipulation				
Identifier	**Name**	**Parameter**	**Response**	**Default**
152 (98h)	SET_TIMESTAMP	Int32	ACK/NACK	

Firmware Function / Command Code List				
Identifier	**Name**	**Parameter**	**Response**	**Default**
153 (99h)	SET_SYNC_MODE	Int32	ACK/NACK	
154 (9Ah)	GET_SYNC_MODE	NONE	Int32	**0**

Acknowledged and Not-acknowledged Identifiers

Identifier	0(0x00)
Name	REPLY_ACK
Description	Confirms a successful SET command.

Identifier	1(0x01)
Name	REPLY_NACK
Description	Reports an error during processing a SET command.

Register Value Save and Reset Command

Identifier	4 (0x04)
Name	WRITE_REGISTERS
Description	Write the currently set parameters to flash memory. Setting will be preserved after power cycling.
Parameter	NONE
Response:	ACK (success) or NACK (error)

Identifier	5 (0x05)
Name	RESTORE_FACTORY_VALUE
Description	Reset the LPMS parameters to factory default values. Please note that upon issuing this command, your currently set parameters will be erased.
Parameter	NONE
Response:	ACK (success) or NACK (error)

Mode Switching Commands

Identifier	6 (0x06)
Name	GOTO_COMMAND_MODE
Description	Switch to command mode. In command mode, the user can issue commands to the firmware to perform calibration, set parameters etc.
Parameter	NONE
Response:	ACK (success) or NACK (error)

Identifier	7 (0x07)
Name	GOTO_STREAM_MODE
Description	Switch to streaming mode. In this mode, data is continuously streamed from the sensor.
Parameter	NONE
Response:	ACK (success) or NACK (error)

Sensor Status Command

<table>
<tr><td>Identifier</td><td>8 (0x08)</td></tr>
<tr><td>Name</td><td>GET_SENSOR_STATUS</td></tr>
<tr><td>Description</td><td>Gets the status of the sensor, i.e. whether the sensor is in command mode or streaming mode.</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int32
<table><tr><td>Sensor status</td><td>Identifier</td></tr><tr><td>Command Mode</td><td>0</td></tr></table></td></tr>
</table>

	Streaming Mode	1	

Get Data Commands

Identifier	9 (0x09)
Name	GET_IMU_DATA
Description	Get the sensor data
Parameter	NONE
Response	

Device Info Commands

Identifier	20 (0x14)
Name	GET_SENSOR_MODEL
Description	Get the sensor model information
Parameter	NONE
Response	Char[24]

Identifier	21 (0x15)
Name	GET_FIRMWARE_INFO
Description	Get the firmware information
Parameter	NONE
Response	Char[24]

Identifier	22 (0x16)
Name	GET_SERIAL_NUMBER
Description	Get the serial number information
Parameter	NONE
Response	Char[24]

Identifier	23 (0x17)
Name	GET_FILTER_VERSION
Description	Gets version of sensor fusion core.
Parameter	NONE
Response	Char[24]

Data Transmission Commands

Identifier	30 (0x1E)
Name	SET_IMU_TRANSMIT_DATA
Description	Set the current transmitted data of sensor
Parameter	Int32 <table><tr><th>Bit</th><th>Reported State / Parameter</th></tr><tr><td>**0**</td><td>Accelerometer raw data transmission enabled</td></tr><tr><td>**1**</td><td>Accelerometer calibrated data transmission enabled</td></tr><tr><td>**2**</td><td>-</td></tr><tr><td>**3**</td><td>Gyro raw data transmission enabled</td></tr><tr><td>**4**</td><td>-</td></tr><tr><td>**5**</td><td>Gyro bias calibrated data transmission enabled</td></tr><tr><td>**6**</td><td>-</td></tr><tr><td>**7**</td><td>Gyro alignment calibrated data transmission enabled</td></tr><tr><td>**8-9**</td><td>-</td></tr><tr><td>**10**</td><td>Angular velocity transmission enabled</td></tr><tr><td>**11**</td><td>Quaternion orientation transmission enabled</td></tr><tr><td>**12**</td><td>Euler angle data transmission enabled</td></tr><tr><td>**13**</td><td>LinAcc data transmission enabled</td></tr><tr><td>**14-15**</td><td>-</td></tr><tr><td>**16**</td><td>Temperature data transmission enabled</td></tr><tr><td>**17-31**</td><td>-</td></tr></table>
Response:	ACK (success) or NACK (error)

Identifier	31 (0x1F)
Name	GET_IMU_TRANSMIT_DATA
Description	Get the current transmitted data from sensor
Parameter	NONE
Response:	Int32 <table><tr><th>Bit</th><th>Reported State / Parameter</th></tr><tr><td>**0**</td><td>Accelerometer raw data transmission enabled</td></tr><tr><td>**1**</td><td>Accelerometer calibrated data transmission enabled</td></tr><tr><td>**2**</td><td>-</td></tr><tr><td>**3**</td><td>Gyro raw data transmission enabled</td></tr><tr><td>**4**</td><td>-</td></tr><tr><td>**5**</td><td>Gyro bias calibrated data transmission enabled</td></tr><tr><td>**6**</td><td>-</td></tr><tr><td>**7**</td><td>Gyro alignment calibrated data transmission enabled</td></tr><tr><td>**8-9**</td><td>-</td></tr><tr><td>**10**</td><td>Angular velocity transmission enabled</td></tr></table>

11	Quaternion orientation transmission enabled
12	Euler angle data transmission enabled
13	LinAcc data transmission enabled
14-15	-
16	Temperature data transmission enabled
17-31	-

IMU ID Setting Command

Identifier	32 (0x20)
Name	SET_IMU_ID
Description	Set sensor ID
Parameter	Int32
Response:	ACK (success) or NACK (error)

Identifier	33 (0x21)
Name	GET_IMU_ID
Description	Get sensor ID
Parameter	None
Response:	Int32

Stream Frequency Commands

Identifier	34 (0x22)
Name	SET_STREAM_FREQ
Description	Set the current streaming frequency
Parameter	Int32
Response:	ACK (success) or NACK (error)

Parameter values:

Frequency (Hz)	**Identifier**
5	5
10	10
20	20
25	25
50	50
100	100
250	250
500	500
1000	1000

<table>
<tr><td>Identifier</td><td>35 (0x23)</td></tr>
<tr><td>Name</td><td>GET_STREAM_FREQ</td></tr>
<tr><td>Description</td><td>Get the current streaming frequency</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int32
<table>
<tr><th>Frequency (Hz)</th><th>Identifier</th></tr>
<tr><td>5</td><td>5</td></tr>
<tr><td>10</td><td>10</td></tr>
<tr><td>20</td><td>20</td></tr>
<tr><td>25</td><td>25</td></tr>
<tr><td>50</td><td>50</td></tr>
<tr><td>100</td><td>100</td></tr>
<tr><td>250</td><td>250</td></tr>
<tr><td>500</td><td>500</td></tr>
<tr><td>1000</td><td>1000</td></tr>
</table></td></tr>
</table>

Deg/Rad Output Commands

<table>
<tr><td>Identifier</td><td>36 (0x24)</td></tr>
<tr><td>Name</td><td>SET_DEGRAD_OUTPUT</td></tr>
<tr><td>Description</td><td>Set the current output unit of angle and rate</td></tr>
<tr><td>Parameter</td><td>Int32
<table>
<tr><th>Output unit</th><th>Identifier</th></tr>
<tr><td>degree or degree per second</td><td>0</td></tr>
<tr><td>radian or radian per second</td><td>1</td></tr>
</table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>37 (0x25)</td></tr>
<tr><td>Name</td><td>GET_DEGRAD_OUTPUT</td></tr>
<tr><td>Description</td><td>Get the current output unit of angle and rate</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int32
<table>
<tr><th>Output unit</th><th>Identifier</th></tr>
<tr><td>degree or degree per second</td><td>0</td></tr>
<tr><td>radian or radian per second</td><td>1</td></tr>
</table></td></tr>
</table>

Reference Setting and Offset Reset Command

<table>
<tr><td>Identifier</td><td>38 (0x26)</td></tr>
<tr><td>Name</td><td>SET_ORIENTATION_OFFSET</td></tr>
<tr><td>Description</td><td>Set the orientation offset (unity quaternion).</td></tr>
<tr><td>Parameter</td><td>Int32<table>
<tr><td>Offset Mode</td><td>Identifier</td></tr>
<tr><td>Object</td><td>0</td></tr>
<tr><td>Heading</td><td>1</td></tr>
<tr><td>Alignment</td><td>2</td></tr>
</table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

Identifier	39 (0x27)
Name	RESET_ORIENTATION_OFFSET
Description	Reset the orientation offset to 0 (unity quaternion).
Parameter	NONE
Response:	ACK (success) or NACK (error)

Accelerometer Settings Command

<table>
<tr><td>Identifier</td><td>50 (0x32)</td></tr>
<tr><td>Name</td><td>SET_ACC_RANGE</td></tr>
<tr><td>Description</td><td>Set the current range of the accelerometer</td></tr>
<tr><td>Parameter</td><td>Int32<table>
<tr><td>Range</td><td>Identifier</td></tr>
<tr><td>3g</td><td>3</td></tr>
<tr><td>6g</td><td>6</td></tr>
<tr><td>12g</td><td>12</td></tr>
<tr><td>24g</td><td>24</td></tr>
</table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>51 (0x33)</td></tr>
<tr><td>Name</td><td>GET_ACC_RANGE</td></tr>
<tr><td>Description</td><td>Get the current range of the accelerometer</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int32<table>
<tr><td>Range</td><td>Identifier</td></tr>
<tr><td>3g</td><td>3</td></tr>
<tr><td>6g</td><td>6</td></tr>
<tr><td>12g</td><td>12</td></tr>
</table></td></tr>
</table>

	24g	24

Gyroscope Settings Command

Identifier	60 (0x3C)
Name	SET_GYR_RANGE
Description	Set the current range of the gyroscope
Parameter	Int32 **Range (deg/s)** / **Identifier**: 125 / 125 250 / 250 500 / 500 1000 / 1000 2000 / 2000
Response:	ACK (success) or NACK (error)

Parameter (Int32):

Range (deg/s)	Identifier
125	125
250	250
500	500
1000	1000
2000	2000

Identifier	61 (0x3D)
Name	GET_GYR_RANGE
Description	Get current gyroscope range.
Parameter	NONE
Response:	UInt32

Range (deg/s)	Identifier
125	125
250	250
500	500
1000	1000
2000	2000

Identifier	62 (0x3E)
Name	START_GYR_CALIBRATION
Description	Start gyro static bias calibration.
Parameter	NONE
Response:	ACK (success) or NACK (error)

Filter Settings Command

<table>
<tr><td>Identifier</td><td>90 (0x5A)</td></tr>
<tr><td>Name</td><td>SET_FILTER_MODE</td></tr>
<tr><td>Description</td><td>Set the sensor filter mode</td></tr>
<tr><td>Parameter</td><td>Int32
<table>
<tr><td>Mode</td><td>Value</td></tr>
<tr><td>Gyroscope (Only)</td><td>0</td></tr>
<tr><td>Accelerometer + gyroscope (Kalman filter)</td><td>1</td></tr>
<tr><td>-</td><td>2</td></tr>
<tr><td>Accelerometer + gyroscope (DCM filter)</td><td>3</td></tr>
<tr><td>-</td><td>4</td></tr>
</table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>91 (0x5B)</td></tr>
<tr><td>Name</td><td>GET_FILTER_MODE</td></tr>
<tr><td>Description</td><td>Get the sensor filter mode</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int32
<table>
<tr><td>Mode</td><td>Value</td></tr>
<tr><td>Gyroscope (Only)</td><td>0</td></tr>
<tr><td>Accelerometer + gyroscope (Kalman filter)</td><td>1</td></tr>
<tr><td>-</td><td>2</td></tr>
<tr><td>Accelerometer + gyroscope (DCM filter)</td><td>3</td></tr>
<tr><td>-</td><td>4</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>Identifier</td><td>92 (0x5C)</td></tr>
<tr><td>Name</td><td>SET_STATIC_SHUTDOWN</td></tr>
<tr><td>Description</td><td>Set the current static non-integration status</td></tr>
<tr><td>Parameter</td><td>Int32
<table>
<tr><td>Status</td><td>Value</td></tr>
<tr><td>Disable</td><td>0</td></tr>
<tr><td>Enable</td><td>1</td></tr>
</table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>93 (0x5D)</td></tr>
<tr><td>Name</td><td>Get_STATIC_SHUTDOWN</td></tr>
<tr><td>Description</td><td>Get the current static non-integration status</td></tr>
<tr><td>Parameter</td><td>Int32<table><tr><td>Status</td><td>Value</td></tr><tr><td>Disable</td><td>0</td></tr><tr><td>Enable</td><td>1</td></tr></table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

UART Settings Command

<table>
<tr><td>Identifier</td><td>130 (0x82)</td></tr>
<tr><td>Name</td><td>SET_UART_BAUDRATE</td></tr>
<tr><td>Description</td><td>Set the current UART baudrate</td></tr>
<tr><td>Parameter</td><td>Int32<table><tr><td>Baudrate</td><td>Identifier</td></tr><tr><td>9600</td><td>9600</td></tr><tr><td>19200</td><td>19200</td></tr><tr><td>38400</td><td>38400</td></tr><tr><td>57600</td><td>57600</td></tr><tr><td>115200</td><td>115200</td></tr><tr><td>230400</td><td>230400</td></tr><tr><td>256000</td><td>256000</td></tr><tr><td>460800</td><td>460800</td></tr><tr><td>921600</td><td>921600</td></tr></table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>131 (0x83)</td></tr>
<tr><td>Name</td><td>GET_UART_BAUDRATE</td></tr>
<tr><td>Description</td><td>Get the current UART baudrate</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int32<table><tr><td>Baudrate</td><td>Identifier</td></tr><tr><td>9600</td><td>9600</td></tr><tr><td>19200</td><td>19200</td></tr><tr><td>38400</td><td>38400</td></tr><tr><td>57600</td><td>57600</td></tr><tr><td>115200</td><td>115200</td></tr><tr><td>230400</td><td>230400</td></tr><tr><td>256000</td><td>256000</td></tr><tr><td>460800</td><td>460800</td></tr></table></td></tr>
</table>

	921600	921600

Identifier	132 (0x84)
Name	SET_UART_FORMAT
Description	Set the UART output format
Parameter	Int32 **Format** / **Identifier** LPBUS / 0 ASCII / 1 MODBUS / 2
Response:	ACK (success) or NACK (error)

Identifier	133 (0x85)
Name	GET_UART_FORMAT
Description	Get the UART output format
Parameter	NONE
Response:	Int32 **Format** / **Identifier** LPBUS / 0 ASCII / 1 MODBUS / 2

Identifier	134 (0x86)
Name	SET_UART_ASCII_CHARACTER
Description	Set the ASCII start/stop character
Parameter	Int8[4] **Byte** / **Parameter** 0 / Start character 1 / stop character 2 / Reserved 3 / Reserved
Response:	ACK (success) or NACK (error)

<table>
<tr><td>Identifier</td><td>135 (0x87)</td></tr>
<tr><td>Name</td><td>GET_UART_ASCII_CHARACTER</td></tr>
<tr><td>Description</td><td>Get the ASCII start/stop character</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int8[4]<table>
<tr><th>Byte</th><th>Parameter</th></tr>
<tr><td>0</td><td>Start character</td></tr>
<tr><td>1</td><td>stop character</td></tr>
<tr><td>2</td><td>Reserved</td></tr>
<tr><td>3</td><td>Reserved</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>Identifier</td><td>136 (0x88)</td></tr>
<tr><td>Name</td><td>SET_LPBUS_DATA_PRECISION</td></tr>
<tr><td>Description</td><td>Set the current UART output data precision</td></tr>
<tr><td>Parameter</td><td>Int32<table>
<tr><th>Data Precision</th><th>Identifier</th></tr>
<tr><td>16bit Fixed point</td><td>0</td></tr>
<tr><td>32bit Float point</td><td>1</td></tr>
</table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>137 (0x89)</td></tr>
<tr><td>Name</td><td>GET_LPBUS_DATA_PRECISION</td></tr>
<tr><td>Description</td><td>Get the current UART output data precision</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int32<table>
<tr><th>Data Precision</th><th>Identifier</th></tr>
<tr><td>16bit Fixed point</td><td>0</td></tr>
<tr><td>32bit Float point</td><td>1</td></tr>
</table></td></tr>
</table>

Sensor Data Timestamp Manipulation

Identifier	152 (0x98)
Name	SET_TIMESTAMP
Description	Set the sensor data timestamp
Parameter	Int32
Response:	ACK (success) or NACK (error)

Firmware Function / Command Code List

Identifier	153 (0x99)
Name	SET_SYNC_MODE
Description	Set sync mode
Parameter	Int32 **Mode** / **Value** Disable / 0 Enable / 1
Response:	ACK (success) or NACK (error)

Identifier	154 (0x9A)
Name	GET_SYNC_MODE
Description	Get sync mode
Parameter	Int32 **Mode** / **Value** Disable / 0 Enable / 1
Response:	ACK (success) or NACK (error)

LP-Research Inc.

https://www.lp-research.com