LPMS-ME1
User Manual ver. 2.0

LP-RESEARCH Inc.
http://www.lp-research.com

Revisions

Date	Version	Changes
2019-11-22	ver. 2.0	• remove hardware information to the hardware manual, and rename the file to user manual • add the information of SPI
2017-10-02	ver. 1.12	• to change the file name of this document from Datasheet to Manual
2017-08-08	ver. 1.11	• to add “Command Lists” to Appendix
2017-05-25	ver. 1.10	• to add declaration of SPI interface reserved • to add example of “Magnetometer Calibration” • Correction of some typos
2016-10-21	ver. 1.9	• LPMS-ME1 package updated
2016-09-06	ver. 1.8	• to add example of “Set UART Baud Rate”
2016-08-31	ver. 1.7	• to add chapter of “Coordinate System”
2016-08-30	ver. 1.6	• Correction on types and units of transmitted data • to add chapter of “Communication Modes” • to add communication examples
2016-08-29	ver. 1.5	• to add definitions of logic high and low levels • to add descriptions of pin 15 and 25 • to add “Power Consumption” on Specifications
2016-08-25	ver. 1.4	• I2C registers information updated • to add more detailed information about data transmission
2016-08-18	ver. 1.3	• to add introductions about default UART baud rate • to add introductions about data types in different transmittal modes
2016-08-16	ver. 1.2	• I2C registers updated • LPMS-ME1 package updated
2016-07-18	ver. 1.1	• Initial release

Contents

1. Introduction

The LPMS-ME1 is a low cost, high performance inertial measurement unit (IMU) with 9-axis. It integrates multiple sensors including 3-axis accelerometer, 3-axis gyroscope and 3-axis magnetometer. After the correction and calculation through the unique algorithm of our company, it can provide precise data including Euler angles, quaternion and linear acceleration. In the meanwhile, the size of LPMS-ME1 is very small, which means it is easy to assemble, convenient for you to embed it in your system and good for your design and development.

Key Features:

- MEMS miniature inertial measurement unit (IMU)
- Integration of 3-axis gyroscope, accelerometer and magnetometer in one unit
- Real-time, on-device calculation of sensor orientation and linear acceleration
- Power Supply: 3.3~5.5V
- Interfaces: UART, I2C, SPI
- Size: PLCC-28 (12.0x12.0x2.6mm)

Applications:

- Human motion capture
- Internet of Things (IOT) devices
- Sports performance evaluation
- Drone fight control

2. Communication Protocol

2.1 LPBUS Protocol

LPBUS is a communication protocol based on the industry standard MODBUS protocol. It is the default communication format used by LPMS devices.

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 device). Later in this manual we will show a description of all supported commands to the sensor, their type and transported data.

GET Commands: Data from the client is read using GET requests. A GET request usually contains no data. The answer from the client to a GET request contains the requested data.

SET Commands: Data registers of the client are written using SET requests. A SET command from the host contains the data to be set. The answer from the client is either ACK (acknowledged) for a successful write, or NACK (not acknowledged) for a failure to set the register occurred.

Notes: Please refer to the Appendix for detailed command lists.

2.2 Communication Modes

LPMS devices have two communication modes including Streaming Mode and Command Mode.

In streaming mode, a LPMS device keeps transmitting measurement data at a preset frequency.

In command mode, a LPMS device is communicated by sending commands, which can be used to set up the parameters and get measurement data of the device.

The default communication mode of LPMS-ME1 is streaming mode when powered up. (The default output data rate is 100 Hz; and please refer to Table 2-3 for the default transmitted data types.)

Figure 2.1 shows the flowchart for changing the parameters of LPMS devices.

Notes:

- Only 4 commands are executable in streaming mode, see Figure 2.1.

- Command WRITE_REGISTERS must be executed after changing sensor parameters, or all changes fail to set up after power down, see Step 4 in Figure 2.1.

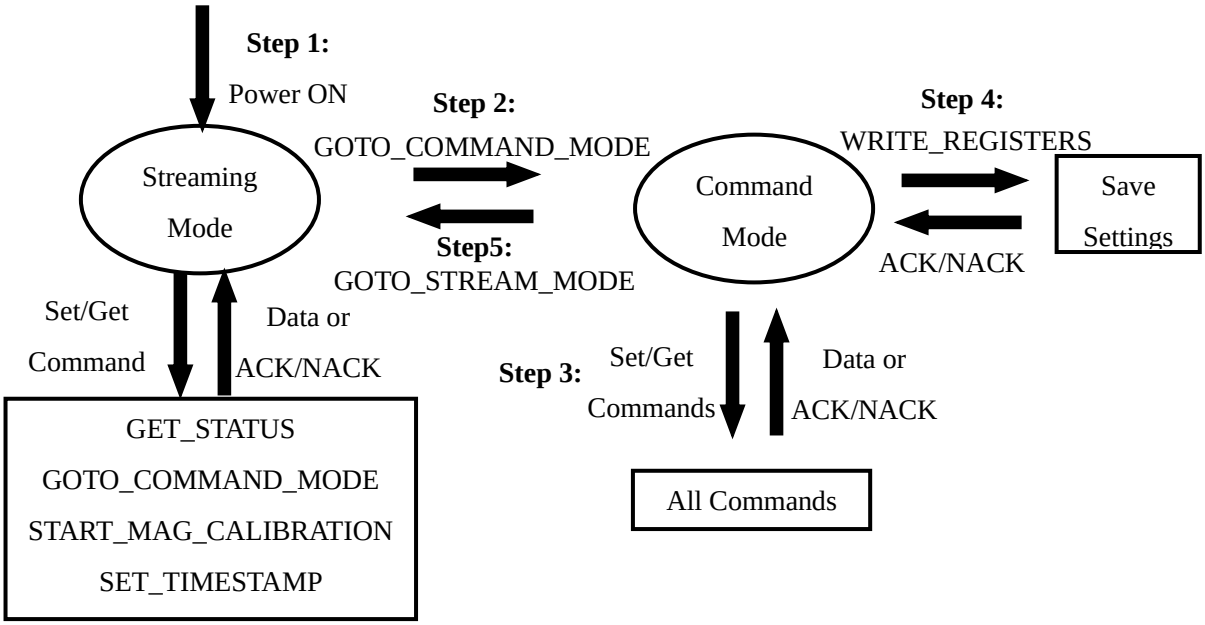

Figure 2.1. Flowchart of sensor parameters setting

2.3 LPBUS Packet Structure

Table 2-1 LPBUS Packet Structure

Byte#	Name	Description
0	Packet start	3Ah
1	Sensor ID byte 1	Low byte of the Sensor ID to be communicated with. The default value of sensor ID is 1. A host can send out a GET / SET request to the sensor by using relative sensor ID, and the client answers to request also with the same ID. This ID can be adjusted by sending a SET command to the sensor.
2	Sensor ID byte 2	High byte of the Sensor ID.
3	Command # byte 1	Low byte of the command number.
4	Command # byte 2	High byte of the command number.
5	Packet data length byte 1	Low byte of the packet data length in bytes.
6	Packet data length byte 2	High byte of the packet data length in bytes.
x	Packet data (***n*** bytes)	If packet data length ***n*** not equal to zero, ***x* = 6+1, 6+2…6+*n***. Otherwise ***x*** = none, the data field is empty.

7+*n*	LRC byte 1	Low byte of LRC check-sum. LRC is calculated in the following way: LRC = sum(Packet Byte#1 to #x)
8+n	LRC byte 2	High byte of LRC check-sum.
9+n	Termination byte 1	0Dh
10+n	Termination byte 2	0Ah

The Packet data is sent in **little-endian format**, low order byte first, high order byte last. There are two types of data format for the packet data:

- 32-bit float
- 16-bit integer

In default setting, sensor data is in 32-bit float format (except timestamp, always 32-bit integer), Table 2-2 shows the data format and order of each senor data type inside a packet. Please refer to Table 2-5 for the definition of each data format identifier.

Table 2-2 Data Format in 32-bit Float Data Transmission Mode

Chunk#	Format identifier	Sensor data type
1	UInt32	Timestamp counter. (400Hz update rate, 0.0025s)
2	Vector3f	Calibrated gyroscope data (rad/s)
3	Vector3f	Calibrated accelerometer data (g)
4	Vector3f	Calibrated magnetometer data (μT)
5	Vector3f	Angular velocity (rad/s)
6	Vector4f	Orientation quaternion (normalized)
7	Vector3f	Euler angle data (rad)
8	Vector3f	Linear acceleration data (g)

If users change the sensor setting to 16-bit integer data transmission mode, data values are transmitted to the host with pre-scale factor in order to increase precision. Table 2-3 shows the data format, sensor data order and relative pre-scale factor in 16-bit data transmission mode.

Table 2-3 Data Format in 16-bit Integer Data Transmission Mode

Chunk#	Data type	Sensor data type	Factor
1	UInt32	Timestamp counter (400Hz update rate, 0.0025s)	none
2	Vector3i16	Calibrated gyroscope data (rad/s)	1000
3	Vector3i16	Calibrated accelerometer data (g)	1000
4	Vector3i16	Calibrated magnetometer data (μT)	100
5	Vector3i16	Angular velocity (rad/s)	1000

6	Vector4i16	Orientation quaternion (normalized)	10000
7	Vector3i16	Euler angle data (rad)	10000
8	Vector3i16	Linear acceleration data (g)	1000

Table 2-4 Data Format Identifier Definition

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 element 32-bit float vector
Vector3i16	3 element 16-bit signed integer vector
Vector4f	4 element 32-bit float vector
Vector4i16	4 element 16-bit signed integer vector
Matrix3x3f	3x3 element float value matrix

The sensor data is sent at the order showed in Table 2-2 and Table 2-3, totally 8 types of data from #1 to #8. The timestamp data is a fixed output which cannot be disabled by users. The data output of types from #2 to #8 can be enabled or disabled by users. If there is any data type is disabled, the following data type will be rolled forwards.

In default setting, the sensor outputs the following data in order (total 7 types of data):

1. Timestamp
2. Calibrated gyroscope data
3. Calibrated accelerometer data
4. Calibrated magnetometer data
5. Orientation quaternion
6. Euler angle data
7. Linear acceleration data

2.4 Communication Examples

In this section we will show a few communication examples using LPBUS protocol.

Go to Command Mode

(HOST -> SENSOR)

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

Reply data (SENSOR -> HOST)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	00h	Command no. LSB (REPLY_ACK=00h)
4	00h	Command no. MSB
5	00h	Data length LSB (REPLY_ACK reply = no data)
6	00h	Data length MSB
7	01h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet end 1
10	0Ah	Packet end 2

Go to Streaming Mode

(HOST -> SENSOR)

Packet byte no.	Content	Meaning
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 command = no data)
6	00h	Data length MSB
7	08h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet end 1
10	0Ah	Packet end 2

Reply data (SENSOR -> HOST)

Packet byte no.	Content	Meaning
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 (REPLY_ACK reply = no data)
6	00h	Data length MSB
7	01h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet end 1
10	0Ah	Packet end 2

Get Sensor Configuration

Get request (HOST -> SENSOR)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	04h	Command no. LSB (04h = GET_CONFIG)
4	00h	Command no. MSB
5	00h	Data length LSB (GET_CONFIG command = no data)
6	00h	Data length MSB
7	05h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet end 1
10	0Ah	Packet end 2

Reply data (SENSOR -> HOST)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	04h	Command no. LSB (04h = GET_CONFIG)
4	00h	Command no. MSB
5	04h	Data length LSB (32-bit integer = 4 bytes)
6	00h	Data length MSB
7	xxh	Configuration data byte 1 (LSB)
8	xxh	Configuration data byte 2
9	xxh	Configuration data byte 3
10	xxh	Configuration data byte 4 (MSB)
11	xxh	Check sum LSB
12	xxh	Check sum MSB
13	0Dh	Packet end 1
14	0Ah	Packet end 2

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

Get Gyroscope Range

Get request (HOST -> SENSOR)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	1Ah	Command no. LSB (1Ah = GET_GYR_RANGE)
4	00h	Command no. MSB
5	00h	Data length LSB (GET_GYR_RANGE command = no data)
6	00h	Data length MSB
7	1Bh	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet end 1
10	0Ah	Packet end 2

Reply data (SENSOR -> HOST)

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

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

Set Accelerometer Range

Set request (HOST -> SENSOR)

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

Reply data (SENSOR -> HOST)

Packet byte no.	Content	Meaning
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 (REPLY_ACK reply = no data)
6	00h	Data length MSB
7	01h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet end 1
10	0Ah	Packet end 2

Get Sensor Data

Get request (HOST -> SENSOR)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	09h	Command no. LSB (09h = GET_SENSOR_DATA)
4	00h	Command no. MSB
5	00h	Data length LSB (GET_SENSOR_DATA command = no data)
6	00h	Data length MSB
7	0Ah	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet end 1
10	0Ah	Packet end 2

Reply data (SENSOR -> HOST), 32-bit float data format

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	09h	Command no. LSB (09h = GET_SENSOR_DATA)
4	00h	Command no. MSB
5	50h	Data length LSB (50h = 80 bytes)
6	00h	Data length MSB
7-10	xxxxxxxxh	Timestamp
11-14	xxxxxxxxh	Gyroscope data x-axis
15-18	xxxxxxxxh	Gyroscope data y-axis
19-22	xxxxxxxxh	Gyroscope data z-axis
23-26	xxxxxxxxh	Accelerometer x-axis
27-30	xxxxxxxxh	Accelerometer y-axis
31-34	xxxxxxxxh	Accelerometer z-axis
35-38	xxxxxxxxh	Magnetometer x-axis
39-42	xxxxxxxxh	Magnetometer y-axis
43-46	xxxxxxxxh	Magnetometer z-axis

47-50	xxxxxxxxh	Orientation quaternion q0
51-54	xxxxxxxxh	Orientation quaternion q1
55-58	xxxxxxxxh	Orientation quaternion q2
59-62	xxxxxxxxh	Orientation quaternion q3
63-66	xxxxxxxxh	Euler angles x-axis
67-70	xxxxxxxxh	Euler angles y-axis
71-74	xxxxxxxxh	Euler angles z-axis
75-78	xxxxxxxxh	Linear acceleration x-axis
79-82	xxxxxxxxh	Linear acceleration y-axis
83-86	xxxxxxxxh	Linear acceleration z-axis
87	xxh	Check sum LSB
88	xxh	Check sum MSB
89	0Dh	Packet end 1
90	0Ah	Packet end 2

Notes:

1. The reply data above is in default setting.

2. xx = Value depends on the current configuration and measurement value.

If only accelerometer and quaternion data are enabled, reply data will be like the following.

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	09h	Command no. LSB (09h = GET_SENSOR_DATA)
4	00h	Command no. MSB
5	20h	Data length LSB (20h = 32 bytes)
6	00h	Data length MSB
7-10	xxxxxxxxh	Timestamp
11-14	xxxxxxxxh	Accelerometer x-axis
15-18	xxxxxxxxh	Accelerometer y-axis
19-22	xxxxxxxxh	Accelerometer z-axis
23-26	xxxxxxxxh	Orientation quaternion q0
27-30	xxxxxxxxh	Orientation quaternion q1
31-34	xxxxxxxxh	Orientation quaternion q2
35-38	xxxxxxxxh	Orientation quaternion q3

39	xxh	Check sum LSB
40	xxh	Check sum MSB
41	0Dh	Packet end 1
42	0Ah	Packet end 2

Save Settings to Sensor

Get request (HOST -> SENSOR)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	0Fh	Command no. LSB (0Fh =WRITE_REGISTERS)
4	00h	Command no. MSB
5	00h	Data length LSB (WRITE_REGISTERS command = no data)
6	00h	Data length MSB
7	10h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet end 1
10	0Ah	Packet end 2

Reply data (SENSOR -> HOST)

Packet byte no.	Content	Meaning
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 reply = no data)
6	00h	Data length MSB
7	01h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet end 1
10	0Ah	Packet end 2

Note: This command needs about 1~2s to get the reply data.

Get Sensor Status

Get request (HOST -> SENSOR)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	05h	Command no. LSB (05h = GET_STATUS)
4	00h	Command no. MSB
5	00h	Data length LSB (GET_STATUS command = no data)
6	00h	Data length MSB
7	06h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet end 1
10	0Ah	Packet end 2

Reply data (SENSOR -> HOST)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	05h	Command no. LSB (05h =GET_STATUS)
4	00h	Command no. MSB
5	04h	Data length LSB (32-bit integer = 4 bytes)
6	00h	Data length MSB
7-10	xxxxxxxxh	Status data
11	xxh	Check sum LSB
12	xxh	Check sum MSB
13	0Dh	Packet end 1
14	0Ah	Packet end 2

Note: Please refer to Appendix for the introduction of status register.

Gyroscope Calibration

Get request (HOST -> SENSOR)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	16h	Command no. LSB (16h = START_GYR_CALIBRATION)
4	00h	Command no. MSB
5	00h	Data length LSB (START_GYR_CALIBRATION command = no data)
6	00h	Data length MSB
7	17h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet end 1
10	0Ah	Packet end 2

Reply data (SENSOR -> HOST)

Packet byte no.	Content	Meaning
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 reply = no data)
6	00h	Data length MSB
7	01h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet end 1
10	0Ah	Packet end 2

Notes:

After sending this command, a ACK reply indicates a start of gyroscope calibration. During calibration, sensor has to be held still for about 10s. The calibration status can be checked by sending command GET_STATUS (Bit3 of the int32 reply data is for Gyroscope calibration status. "1" indicates calibration running while it is cleared by hardware after calibration finished).

Magnetometer Calibration

Get request (HOST -> SENSOR)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	11h	Command no. LSB (11h = START_MAG_CALIBRATION)
4	00h	Command no. MSB
5	00h	Data length LSB (START_MAG_CALIBRATION command = no data)
6	00h	Data length MSB
7	12h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet end 1
10	0Ah	Packet end 2

Reply data (SENSOR -> HOST)

Packet byte no.	Content	Meaning
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 reply = no data)
6	00h	Data length MSB
7	01h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet end 1
10	0Ah	Packet end 2

Notes:

This command is similar to gyroscope calibration command, a ACK reply indicates a start of calibration for about 10s. During calibration, sensor needs to be rotated around x, y and z axis continuously so as to create a map of environment magnetic field. You can also use command GET_STATUS for calibration status check (Bit 4 of reply data).

Set UART Baud Rate

Set request (HOST -> SENSOR)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	54h	Command no. LSB (54h = SET_UART_BAUDRATE)
4	00h	Command no. MSB
5	04h	Data length LSB (32-bit integer = 4 bytes)
6	00h	Data length MSB
7	07h	Set UART baud rate = 921600 bps; Please refer to Command "SET_UART_BAUDRATE" in Appendix for details.
8	00h	
9	00h	
10	00h	
11	60h	Check sum LSB
12	00h	Check sum MSB
13	0Dh	Packet end 1
14	0Ah	Packet end 2

Reply data (SENSOR -> HOST)

Packet byte no.	Content	Meaning
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 reply = no data)
6	00h	Data length MSB
7	01h	Check sum LSB
8	00h	Check sum MSB
9	0Dh	Packet end 1
10	0Ah	Packet end 2

Note: The new Baudrate setting will be activated from the next power on

3. Appendix

Summary

Acknowledged / Not-acknowledged Identifiers				
Identifier	**Name**	**Parameter**	**Response**	**Default**
0	REPLY_ACK			
1	REPLY_NACK			

Get Configuration and Status Info Commands				
Identifier	**Name**	**Parameter**	**Response**	**Default**
4 (04h)	GET_CONFIG	NONE	Int32	
5 (05h)	GET_STATUS[1]	NONE	Int32	

Mode Switching Commands				
Identifier	**Name**	**Parameter**	**Response**	**Default**
6 (06h)	GOTO_COMMAND_MODE[1]	NONE	ACK/NACK	
7 (07h)	GOTO_STREAM_MODE	NONE	ACK/NACK	

IMU ID Settings Command				
Identifier	**Name**	**Parameter**	**Response**	**Default**
20 (14h)	SET_IMU_ID	Int32	ACK/NACK	
21 (15h)	GET_IMU_ID	NONE	Int32	1

Gyroscope Settings Command				
Identifier	**Name**	**Parameter**	**Response**	**Default**
22 (16h)	START_GYR_CALIBRATION	NONE	ACK/NACK	
25 (19h)	SET_GYR_RANGE	Int32	ACK/NACK	
26 (1Ah)	GET_GYR_RANGE	NONE	Int32	2000dps

Accelerometer Settings Command				
Identifier	**Name**	**Parameter**	**Response**	**Default**
31 (1Fh)	SET_ACC_RANGE	Int32	ACK/NACK	
32 (20h)	GET_ACC_RANGE	NONE	Int32	4g

Magnetometer Settings Command				
Identifier	**Name**	**Parameter**	**Response**	**Default**

17 (11h)	START_MAG_CALIBRATION[1]	NONE	ACK/NACK	
33 (21h)	SET_MAG_RANGE	Int32	ACK/NACK	
34 (22h)	GET_MAG_RANGE	NONE	Int32	8Gauss

Data Transmission Commands				
Identifier	**Name**	**Parameter**	**Response**	**Default**
9 (09h)	GET_SENSOR_DATA	NONE		
10 (0Ah)	SET_TRANSMIT_DATA	Int32	ACK/NACK	
11 (0Bh)	SET_STREAM_FREQ	Int32	ACK/NACK	
66 (42h)	SET_TIMESTAMP[1]	Int32	ACK/NACK	
84 (54h)	SET_UART_BAUDRATE	Int32	ACK/NACK	
85 (55h)	GET_UART_BAUDRATE	NONE	Int32	

Register Value Save and Reset Command				
Identifier	**Name**	**Parameter**	**Response**	**Default**
15 (0Fh)	WRITE_REGISTERS	NONE	ACK/NACK	
16 (10h)	RESTORE_FACTORY_DEFAULTS	NONE	ACK/NACK	

Reference Setting and Offset Reset Command				
Identifier	**Name**	**Parameter**	**Response**	**Default**
18 (12h)	SET_ORIENTATION_OFFSET	Int32	ACK/NACK	
82 (52h)	RESET_ORIENTATION_OFFSET	NONE	ACK/NACK	

Filter Settings Command				
Identifier	**Name**	**Parameter**	**Response**	**Default**
41(29h)	SET_FILTER_MODE	Int32	ACK/NACK	
42(2Ah)	GET_FILTER_MODE	NONE	Int32	1
43(2Bh)	SET_FILTER_PRESET	Int32	ACK/NACK	
44(2Ch)	GET_FILTER_PRESET	NONE	Int32	3

Device Info				
Identifier	**Name**	**Parameter**	**Response**	**Default**
90(5Ah)	GET_SERIAL_NUMBER	NONE	Char[24]	
92(5Ch)	GET_FIRMWARE_INFO	NONE	Char[16]	

[1]**Note:** These commands are executable in both streaming mode and command mode. Other commands are executable only when the sensor is in command mode.

Acknowledged and Not-acknowledged Identifiers

Identifier	0
Name	REPLY_ACK
Description	Confirms a successful SET command.

Identifier	1
Name	REPLY_NACK
Description	Reports an error during processing a SET command.

Configuration and Status Commands

Identifier	4 (0x04)
Name	GET_CONFIG
Description	Get the current value of the configuration register of the sensor. The configuration word is read-only. The different parameters are set by their respective SET commands. E.g. SET_TRANSMIT_DATA for defining which data is transmitted from the sensor.
Parameter	NONE
Response:	Int32
Data format	

Bit	Reported State / Parameter
0 - 2	Stream frequency setting (see SET_STREAM_FREQ)
3 - 8	Reserved
9	Reserved
10	Magnetometer data transmission enabled
11	Accelerometer data transmission enabled
12	Gyroscope data transmission enabled
13	Temperature output enabled
14	Reserved
15	Reserved
16	Angular velocity output enabled
17	Euler angle data transmission enabled
18	Quaternion orientation output enabled
19	Reserved
20	Reserved
21	Linear acceleration output enabled
22	16-bit data output mode enabled
23	Reserved
24	Magnetometer compensation enabled
25	Accelerometer compensation enabled
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Gyroscope auto-calibration enabled
31	Reserved

Identifier	5 (0x05)
Name	GET_STATUS
Description	Get the current value of the status register of the sensor. The status word is read-only
Parameter	NONE
Response:	Int32
Data format	

Bit	Indicated state
0	COMMAND mode enabled
1	STREAM mode enabled
2	Reserved
3	Gyroscope calibration running
4	Magnetometer calibration running
5	Gyroscope initialization failed
6	Accelerometer initialization failed
7	Magnetometer initialization failed
8	Reserved
9	Gyroscope unresponsive
10	Accelerometer unresponsive
11	Magnetometer unresponsive
12	Flash write failed
13-31	Reserved

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, and some commands cannot be performed until the sensor receives the GOTO_COMMAND_MODE command.
Parameter	NONE
Response:	ACK (success) or NACK (error)

IMU ID Setting Command

Identifier	20 (0x14)
Name	SET_IMU_ID
Description	Set sensor ID
Parameter	Int32
Response:	ACK (success) or NACK (error)

Identifier	21 (0x15)
Name	GET_IMU_ID
Description	Get sensor ID
Parameter	None
Response:	Int32

Gyroscope Settings Command

Identifier	22 (0x16)
Name	START_GYR_CALIBRATION
Description	Start the calibration of the gyroscope sensor
Parameter	NONE
Response:	ACK (success) or NACK (error)

Identifier	25 (0x19)
Name	SET_GYR_RANGE
Description	Set the current range of the gyroscope
Parameter	Int32 **Range (deg/s)** / **Identifier** **125** / 125 **245** / 245 **500** / 500 **1000** / 1000 **2000** / 2000
Response:	ACK (success) or NACK (error)

Identifier	26 (0x1A)
Name	GET_GYR_RANGE
Description	Get current gyroscope range.
Parameter	NONE
Response:	Int32 **Range (deg/s)** / **Identifier** **125** / 125 **245** / 245 **500** / 500 **1000** / 1000 **2000** / 2000

Accelerometer Settings Command

Identifier	31 (0x1F)
Name	SET_ACC_RANGE
Description	Set the current range of the accelerometer
Parameter	Int32 **Range** / **Identifier** **2g** / 2 **4g** / 4 **8g** / 8 **16g** / 16
Response:	ACK (success) or NACK (error)

Identifier	32 (0x20)
Name	GET_ACC_RANGE
Description	Get the current range of the accelerometer
Parameter	NONE
Response:	Int32 **Range** / **Identifier** **2g** / 2 **4g** / 4 **8g** / 8 **16g** / 16

Magnetometer Settings Command

Identifier	17 (0x11)
Name	START_MAG_CALIBRATION
Description	Start the calibration of the magnetometer sensor
Parameter	NONE
Response:	ACK (success) or NACK (error)

Identifier	33 (0x21)
Name	SET_MAG_RANGE
Description	Set the current range of the gyroscope
Parameter	Int32 **Range** / **Identifier** **4 Gauss** 4 **8 Gauss** 6 **12 Gauss** 12 **16 Gauss** 16
Response:	ACK (success) or NACK (error)

Identifier	34 (0x22)
Name	GET_MAG_RANGE
Description	Get current magnetometer range.
Parameter	NONE
Response:	Int32 **Range** / **Identifier** **4 Gauss** 4 **8 Gauss** 6 **12 Gauss** 12 **16 Gauss** 16

Data Transmission Commands

Identifier	**9 (0x09)**
Name	GET_SENSOR_DATA
Description	Retrieves the latest set of sensor data. A data packet will be composed as defined by SET_TRANSMIT_DATA. The currently set format can be retrieved with the sensor configuration word.
Parameter	NONE
Response:	See the LPBUS protocol explanation for a description of the measurement data format.

<table>
<tr><td>Identifier</td><td>10 (0x0A)</td></tr>
<tr><td>Name</td><td>SET_TRANSMIT_DATA</td></tr>
<tr><td>Description</td><td>Set the current transmit data</td></tr>
<tr><td>Parameter</td><td>Int32
<table>
<tr><th>Bit</th><th>Reported State / Parameter</th></tr>
<tr><td>0 - 9</td><td>0</td></tr>
<tr><td>10</td><td>Magnetometer data transmission enabled</td></tr>
<tr><td>11</td><td>Accelerometer data transmission enabled</td></tr>
<tr><td>12</td><td>Gyroscope data transmission enabled</td></tr>
<tr><td>13</td><td>Temperature output enabled</td></tr>
<tr><td>14</td><td>0</td></tr>
<tr><td>15</td><td>0</td></tr>
<tr><td>16</td><td>Angular velocity output enabled</td></tr>
<tr><td>17</td><td>Euler angle data transmission enabled</td></tr>
<tr><td>18</td><td>Quaternion orientation output enabled</td></tr>
<tr><td>19</td><td>0</td></tr>
<tr><td>20</td><td>0</td></tr>
<tr><td>21</td><td>Linear acceleration output enabled</td></tr>
<tr><td>22</td><td>16-bit data output mode enabled</td></tr>
<tr><td>23</td><td>0</td></tr>
<tr><td>24</td><td>Magnetometer compensation enabled</td></tr>
<tr><td>25</td><td>Accelerometer compensation enabled</td></tr>
<tr><td>26-31</td><td>0</td></tr>
</table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>11 (0x0B)</td></tr>
<tr><td>Name</td><td>SET_STREAM_FREQ</td></tr>
<tr><td>Description</td><td>Set the current streaming frequency</td></tr>
<tr><td>Parameter</td><td>Int32
<table>
<tr><th>Frequency (Hz)</th><th>Identifier</th><th>Bit : 0~2
(GET_CONFIG return data)</th></tr>
<tr><td>5</td><td>5</td><td>000</td></tr>
<tr><td>10</td><td>10</td><td>001</td></tr>
<tr><td>25</td><td>25</td><td>010</td></tr>
<tr><td>50</td><td>50</td><td>011</td></tr>
<tr><td>100</td><td>100</td><td>100</td></tr>
<tr><td>200</td><td>200</td><td>101</td></tr>
<tr><td>400</td><td>400</td><td>110</td></tr>
</table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

Identifier	66 (0x42)
Name	SET_TIMESTAMP
Description	Set the current sensor timestamp counter. Counter updates at 400Hz, i.e. setting timestamp counter equates to setting the timestamp to 1s.
Parameter	Int32
Response:	ACK (success) or NACK (error)

Identifier	84 (0x54)
Name	SET_UART_BAUDRATE
Description	Set the current UART baudrate
Parameter	Int32 **Baud rate** / **Identifier** **19200** 0 **38400** 1 **57600** 2 **115200** 3 **230400** 4 **256000** 5 **460800** 6 **921600** 7
Response:	ACK (success) or NACK (error)

Identifier	85 (0x55)
Name	GET_UART_BAUDRATE
Description	Get the current UART baudrate
Parameter	NONE
Response:	Int32 **Baud rate** / **Identifier** **19200** 0 **38400** 1 **57600** 2 **115200** 3 **230400** 4 **256000** 5 **460800** 6 **921600** 7

Register Value Save and Reset Command

Identifier	15 (0x0F)
Name	WRITE_REGISTERS
Description	Write the currently set parameters to flash memory.
Parameter	NONE
Response:	ACK (success) or NACK (error)

Identifier	16 (0x10)
Name	RESTORE_FACTORY_DEFAULTS
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)

Reference Setting and Offset Reset Command

Identifier	18 (0x12)
Name	SET_OFFSET
Description	Sets the orientation offset using one of the three offset methods.
Parameter	Int32 **Mode** / **Value** **Object reset** / 0 **Heading reset** / 1
Response:	ACK (success) or NACK (error)

Identifier	82 (0x52)
Name	RESET_ORIENTATION_OFFSET
Description	Reset the orientation offset to 0 (unity quaternion).
Parameter	NONE
Response:	ACK (success) or NACK (error)

Filter Settings Command

Identifier	41 (0x29)
Name	SET_ FILTER_MODE
Description	Set the sensor filter mode
Parameter	Int32 **Mode** / **Value** **Gyroscope only** 0 **Accelerometer + gyroscope (Kalman filter)** 1 **Accelerometer+ gyroscope+ magnetometer (Kalman filter)** 2 **Accelerometer + gyroscope (DCM filter)** 3 **Accelerometer + gyroscope + Magnetometer (DCM filter)** 4
Response:	ACK (success) or NACK (error)

Identifier	42 (0x2A)
Name	GET_ FILTER_MODE
Description	Get the sensor filter mode
Parameter	NONE
Response:	Int32 **Mode** / **Value** **Gyroscope only** 0 **Accelerometer + gyroscope (Kalman filter)** 1 **Accelerometer+ gyroscope+ magnetometer (Kalman filter)** 2 **Accelerometer + gyroscope (DCM filter)** 3 **Accelerometer + gyroscope + Magnetometer (DCM filter)** 4

Identifier	43 (0x2B)
Name	SET_ FILTER_PRESET
Description	Set one of the filter parameter presets for accelerometer and magnetometer covariance strength
Parameter	Int32 **Correction strength** / **Value** **Weak** 0 **Medium** 1 **Strong** 2 **Dynamic** 3
Response:	ACK (success) or NACK (error)

Identifier	44 (0x2C)
Name	GET_ FILTER_PRESET
Description	Get current filter preset
Parameter	NONE
Response:	Int32 **Correction strength** / **Value** **Dynamic** 0 **Strong** 1 **Medium** 2 **Weak** 3

Device Info

Identifier	90 (0x5A)
Name	GET_SERIAL_NUMBER
Description	Get sensor serial number
Parameter	NONE
Response:	Char[24] Character array of length 24

Identifier	92 (0x5C)
Name	GET_FIRMARE_INFO
Description	Get firmware info
Parameter	NONE
Response:	Char[16] Firmware name - version

www.lp-research.com