LPMS-ME1
Manual ver. 1.12

LP-RESEARCH Inc.

http://www.lp-research.com

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. And 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[1]
- Size: PLCC-28 (12.0x12.0x2.6mm)

Applications:

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

[1]Note: SPI communication not available yet.

2. Revisions

Date	Version	Changes
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

3. General Information

3.1 Block Diagram

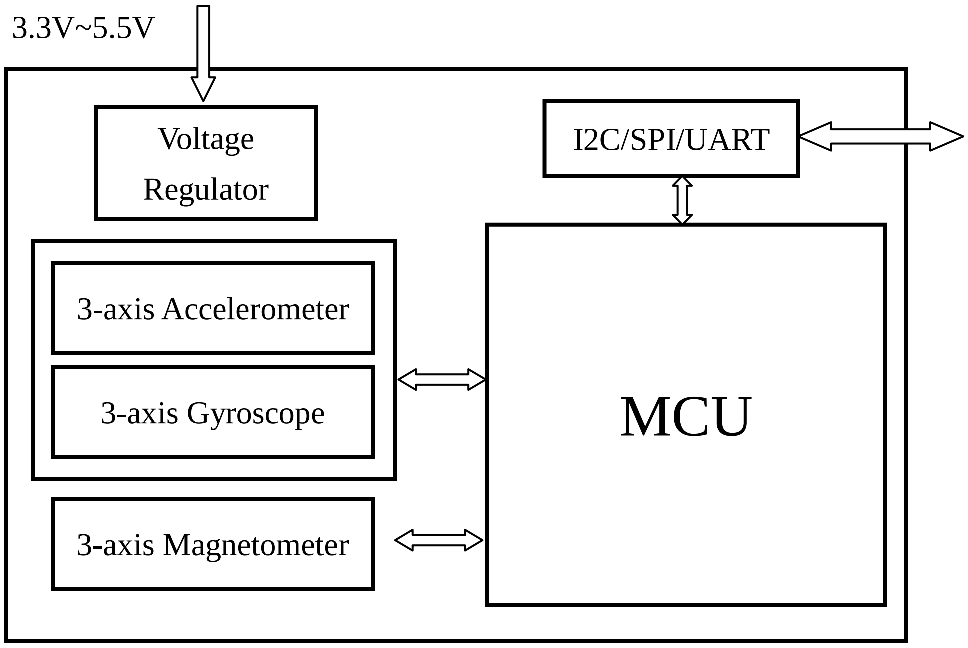

Figure 3.1. Block diagram of LPMS-ME1

3.2 Pin out

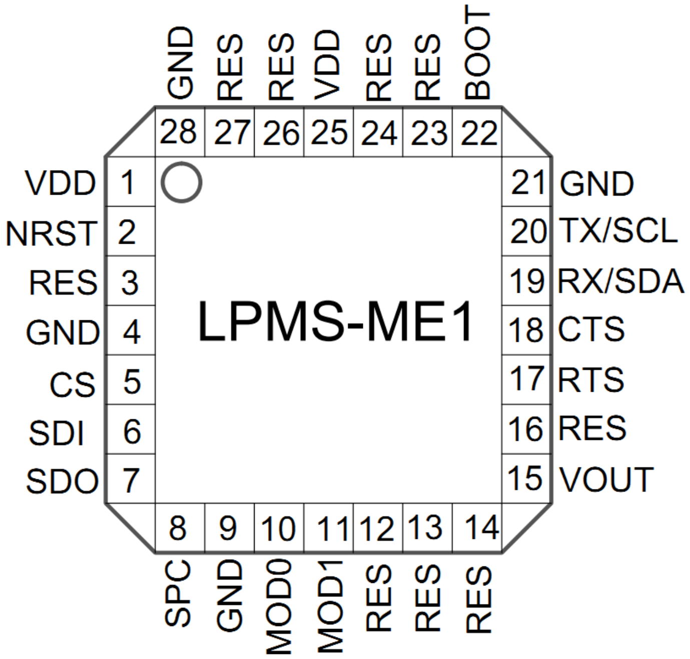

Figure 3.2. Pin out of LPMS-ME1

Table 3-1 Pin Descriptions

Pin#	Name	Function		Description
1, 25	VDD	Power		Power Input (3.3V~5.5V)
2	NRST	Reset Pin		Active low reset pin. During normal operation this pin must be driven high or left floating.
5	CS[1]	Chip select		SPI chip select (active low)
6	SDI[1]	Slave Data Input		SPI serial data input (slave)
7	SDO[1]	Slave Data Output		SPI serial data output (slave)
8	SPC[1]	Serial Clock		SPI serial clock
10	MOD0	Selection pins of interfaces		These pins determine the signal interface. See table 2-2 below.
11	MOD1			
15	VOUT	Power Output		3.3V output (current output < 20 mA)
17	RTS[2]	UART_RTS		Hardware flow control in UART full duplex mode (Ready-To-Send)
18	CTS[2]	UART_CTS		Hardware flow control in UART full duplex mode (Clear-To-Send)
19	RX/SDA	UART	UART_RX	Receiver data input
		I^2C	I^2C_SDA	I^2C serial data
20	TX/SCL	UART	UART_TX	Transmitter data output
		I^2C	I^2C_SCL	I^2C serial clock
22	BOOT	-		During normal operation this pin should connect to GND through a pull-down resistor
4, 9, 21, 28	GND	-		Connect to GND
3, 12, 13, 14, 16, 23, 24,26, 27	RES	-		Reserved pins[3]

Notes:

[1] SPI interface reserved.

[2] Hardware flow control of UART is not used (default).

[3] All reserved pins should be left floating.

[4] See Table 3-3 for definitions of logic high and low level.

Table 3-2 Selections of Communication Interfaces

MOD1	MOD0	Interfaces
0	0	UART (default)
0	1	SPI (Reserved)
1	0	I^{2}C (ADD0=0)
1	1	I^{2}C (ADD0=1)

Notes:

ADD0 is LSB of I^{2}C slave address;

MOD0/MOD1 should be driven high or low through a pull-up resistor.

Table 3-3 Definitions of Logic High and Low Level

Level	Value	Unit
Low	0~0.99	V
High	2.31~3.3	V

3.3 Typical Applications

UART Mode

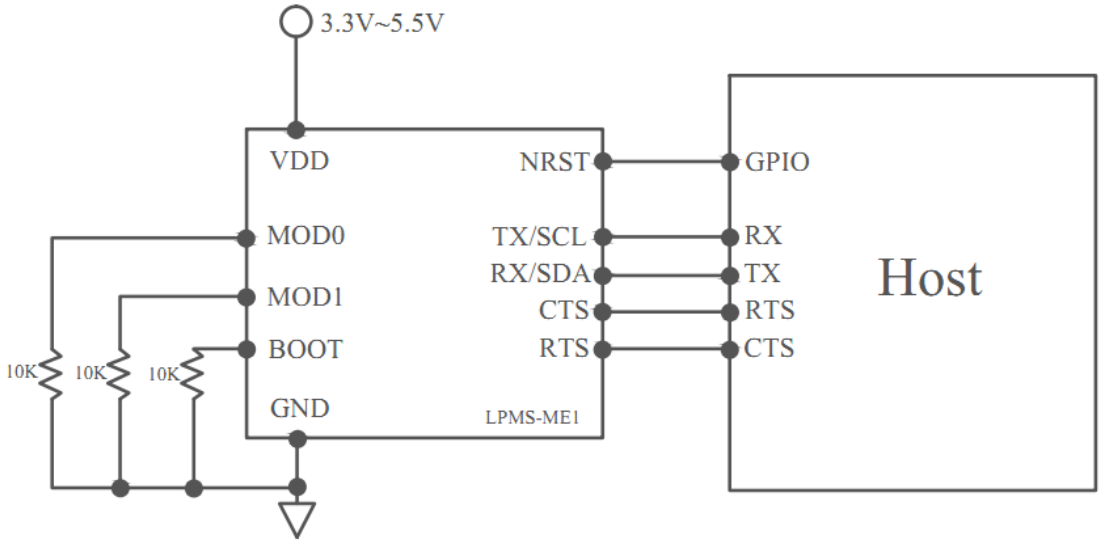

Figure 3.3. LPMS-ME1 typical application (UART mode)

Note: Only four pins including VDD, GND, TX and RX are needed to be connected for test.

I^{2}C Mode

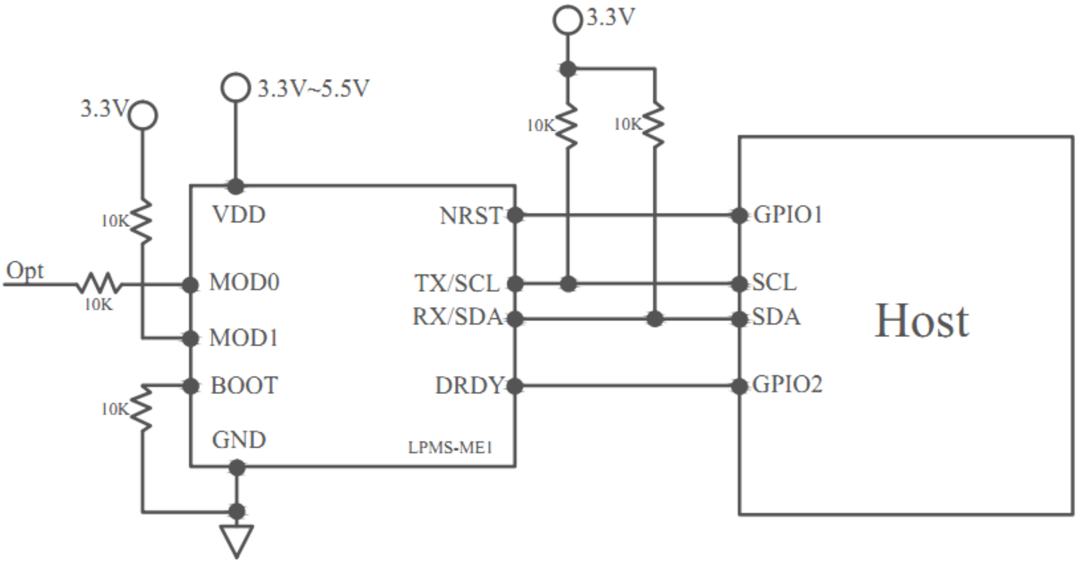

Figure 3.4. LPMS-ME1 typical application (I^{2}C mode)

Note: If Opt connects to logic high, ADD0 = 1; if Opt connects to logic low, ADD0 = 0.

SPI Mode (Reserved)

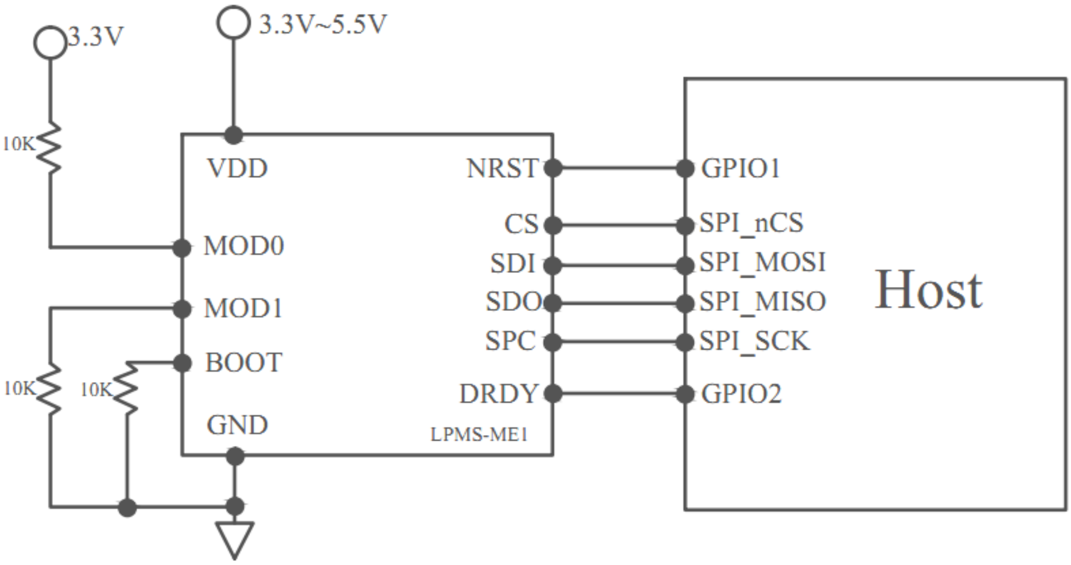

Figure 3.5. LPMS-ME1 typical application (SPI mode)

3.4 Coordinate System

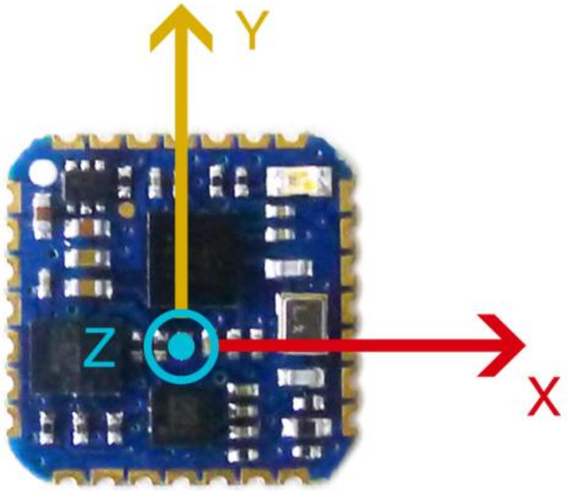

Figure 3.6.Coordinate system of LPMS-ME1

4. Communication Interfaces

4.1 UART

The universal asynchronous receiver transmitter (UART) is a common interface of asynchronous communication with up to 4.5Mbps baud rate for transmitting and receiving.LPMS-ME1 offers 4 pins (TX, RX, RTS and CTS) for UART configuration, its default baud rate is 115200 bps. The default configuration: 8 bits data length, 1 stop bit, no parity. Sequence diagrams of UART are shown as Figure 4.1 and Figure 4.2.

TX: Transmit data output.

RX: Receive data input.

RTS: "Request to send" indicates that the USART is ready to receive data (when low).

CTS: "Clear to send" blocks the data transmission at the end of the current transfer when high.

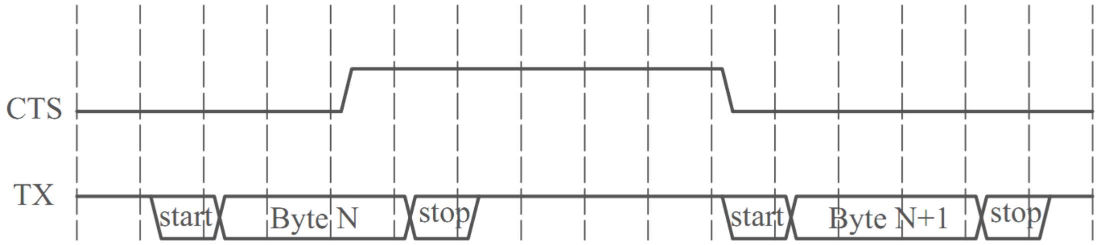

Figure 4.1. Sequence diagram of transmitter with CTS control

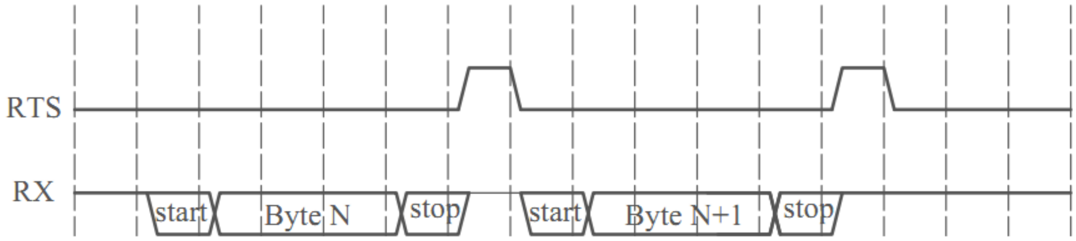

Figure 4.2. Sequence diagram of receiver with RTS control

Please refer to chapter “LPBUS Protocol” for more details about UART communication.

4.2 I^{2}C

I^{2}C Introduction

The I^{2}C (Inter-Integrated Circuit) bus is a two-wire serial bus, which handles communications between the microcontroller and the serial I^{2}C bus. The interface is connected to the I^{2}C bus by a data pin (SDA) and by a clock pin (SCL) for receiving and transmitting data. Generally, it can be connected with a Fast-mode (up to 400 kHz) I^{2}C bus.

When pin MOD1 is driven high, the interface of LPMS-ME1 is I^{2}C as a slave device. The slave address is determined by pin MOD0, as shown in Table 4-1.

Table 4-1 I^{2}C Slave Address

MOD0	I^{2}C slave address (7 bits)
0	0x32
1	0x33

See from Figure 4.3 to Figure 4.6 for time sequence diagrams of I^{2}C.

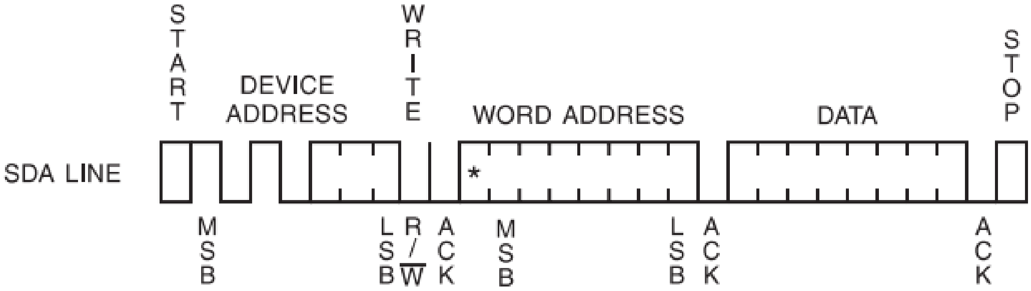

Figure 4.3. Write a register

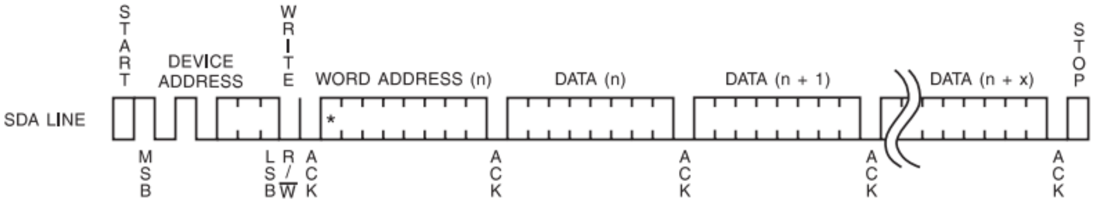

Figure 4.4. Write multi registers

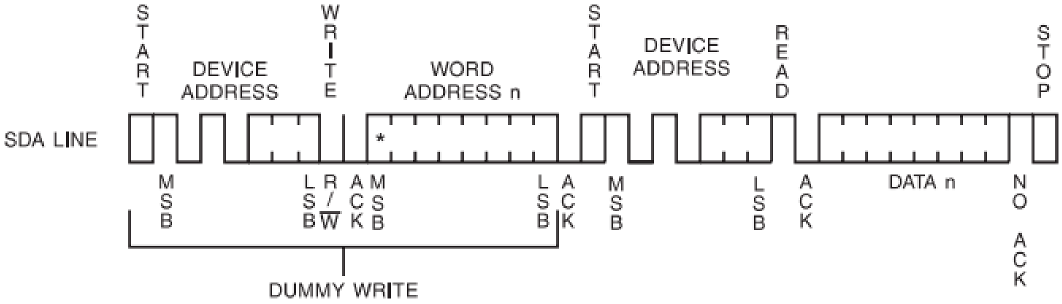

Figure 4.5. Read a register

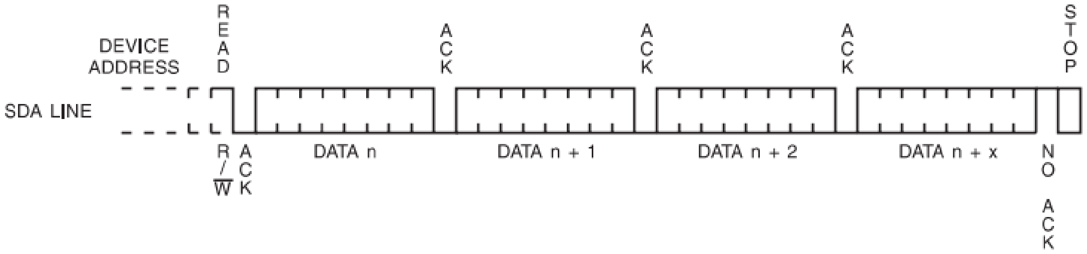

Figure 4.6. Read multi registers

I^{2}C Registers

Table 4-2 I^{2}C Registers

Address	Name	Type	Default Value
0x00	FUN_CONFIG	R	0x00
0x01	SYS_CONFIG	R/W	0x00
0x02	DATA_CTRL	R/W	0x03
0x03	DATA_ENABLE	R/W	0x37
0x04	CTRL_0_A	R/W	0x08
0x05	CTRL_1_G	R/W	0x0C
0x06	CTRL_2_M	R/W	0x20
0x07	STATUS	R	0x00
0x08	FILTER_CONFIG	R/W	0x01
0x09~0x19	-	-	-
0x20	TIMESTAMP_0	R	0x00
0x21	TIMESTAMP_1	R	0x00
0x22	TIMESTAMP_2	R	0x00
0x23	TIMESTAMP_3	R	0x00
0x24	ACC_X_0	R	0x00
0x25	ACC_X_1	R	0x00
0x26	ACC_X_2	R	0x00
0x27	ACC_X_3	R	0x00
0x28	ACC_Y_0	R	0x00
0x29	ACC_Y_1	R	0x00
0x2A	ACC_Y_2	R	0x00
0x2B	ACC_Y_3	R	0x00

0x2C	ACC_Z_0	R	0x00
0x2D	ACC_Z_1	R	0x00
0x2E	ACC_Z_2	R	0x00
0x2F	ACC_Z_3	R	0x00
0x30	GYR_X_0	R	0x00
0x31	GYR_X_1	R	0x00
0x32	GYR_X_2	R	0x00
0x33	GYR_X_3	R	0x00
0x34	GYR_Y_0	R	0x00
0x35	GYR_Y_1	R	0x00
0x36	GYR_Y_2	R	0x00
0x37	GYR_Y_3	R	0x00
0x38	GYR_Z_0	R	0x00
0x39	GYR_Z_1	R	0x00
0x3A	GYR_Z_2	R	0x00
0x3B	GYR_Z_3	R	0x00
0x3C	MAG_X_0	R	0x00
0x3D	MAG_X_1	R	0x00
0x3E	MAG_X_2	R	0x00
0x3F	MAG_X_3	R	0x00
0x40	MAG_Y_0	R	0x00
0x41	MAG_Y_1	R	0x00
0x42	MAG_Y_2	R	0x00
0x43	MAG_Y_3	R	0x00
0x44	MAG_Z_0	R	0x00
0x45	MAG_Z_1	R	0x00
0x46	MAG_Z_2	R	0x00
0x47	MAG_Z_3	R	0x00
0x48	EULER_X_0	R	0x00
0x49	EULER_X_1	R	0x00
0x4A	EULER_X_2	R	0x00
0x4B	EULER_X_3	R	0x00
0x4C	EULER_Y_0	R	0x00

0x4D	EULER_Y_1	R	0x00
0x4E	EULER_Y_2	R	0x00
0x4F	EULER_Y_3	R	0x00
0x50	EULER_Z_0	R	0x00
0x51	EULER_Z_1	R	0x00
0x52	EULER_Z_2	R	0x00
0x53	EULER_Z_3	R	0x00
0x54	QUAT_W_0	R	0x00
0x55	QUAT_W_1	R	0x00
0x56	QUAT_W_2	R	0x00
0x57	QUAT_W_3	R	0x00
0x58	QUAT_X_0	R	0x00
0x59	QUAT_X_1	R	0x00
0x5A	QUAT_X_2	R	0x00
0x5B	QUAT_X_3	R	0x00
0x5C	QUAT_Y_0	R	0x00
0x5D	QUAT_Y_1	R	0x00
0x5E	QUAT_Y_2	R	0x00
0x5F	QUAT_Y_3	R	0x00
0x60	QUAT_Z_0	R	0x00
0x61	QUAT_Z_1	R	0x00
0x62	QUAT_Z_2	R	0x00
0x63	QUAT_Z_3	R	0x00
0x64	LIN_ACC_X_0	R	0x00
0x65	LIN_ACC_X_1	R	0x00
0x66	LIN_ACC_X_2	R	0x00
0x67	LIN_ACC_X_3	R	0x00
0x68	LIN_ACC_Y_0	R	0x00
0x69	LIN_ACC_Y_1	R	0x00
0x6A	LIN_ACC_Y_2	R	0x00
0x6B	LIN_ACC_Y_3	R	0x00
0x6C	LIN_ACC_Z_0	R	0x00
0x6D	LIN_ACC_Z_1	R	0x00

0x6E	LIN_ACC_Z_2	R	0x00
0x6F	LIN_ACC_Z_3	R	0x00
0x70	TEMP_0	R	0x00
0x71	TEMP_1	R	0x00
0x72	TEMP_2	R	0x00
0x73	TEMP_3	R	0x00
0x74	WHO AM I	R	0x32
0x75	FIRMWARE_VERSION_0	R	-
0x76	FIRMWARE_VERSION_1	R	-

SYS_CONFIG

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-	-	-	-	-	-	-	LEDOn/Off

Bit0: LED control bit. (1: LED on; 0: LED off)

DATA_CTRL

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-	-	Reset Timestamp	-	Setting of output data rate			

Bit5 is set to 1 for resetting timestamp, and it will be cleared by hardware after data transmitting is finished.

Bit3~Bit0: to set the output data rate, see Table 4-3.

Table 4-3 Configuration of Output Data Rate

Bit3~Bit0	the Output Data Rate
0000	5 Hz
0001	10 Hz
0010	50 Hz
0011	100 Hz (default)
0100	200 Hz
0101	400 Hz

DATA_ENABLE

Bit7	Bit6	Bit5	Bit4
Temperature	Linear Acceleration	Quaternion	Euler angles
Bit3	Bit2	Bit1	Bit0
Magnetometer	Gyroscope	Accelerometer	Timestamp

This register enables the specified data output. (1: enable data output; 0: disable data output)

CTRL_0_A

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-	-	-	-	Acceleration range		-	-

Bit3and Bit2 determine the acceleration range, see Table 4-4.

Table 4-4 Configuration of Acceleration Range

Bit3~Bit2	Acceleration Range
00	±2 g
01	±16 g
10	±4 g (default)
11	±8 g

CTRL_1_G

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Gyroscope static bias calibration	-	-	-	Gyroscope range			-

Bit7 is set to 1 for the start of calibrating gyroscope static bias. This bit is cleared by hardware after calibration finished.

From Bit3 to Bit1 it determines the gyroscope range, see Table 4-5.

Table 4-5 Configuration of Gyroscope Range

Bit3~Bit1	Gyroscope Range
000	245 dps
010	500 dps
100	1000 dps
110	2000 dps (default)
001	125dps

CTRL_2_M

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Magnetometer calibration	Magnetometer range		-	-	-	-	-

Bit7 is set to 1 for the start of calibrating magnetometer. This bit is cleared by hardware after calibration finished.

Bit6 and Bit5 determine the magnetometer range, see Table 4-6

Table 4-6 Configuration of Magnetometer Range

Bit6~Bit5	Magnetometer Range
00	4 gauss
01	8 gauss (default)
10	12 gauss
11	16 gauss

FILTER_CONFIG

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-		Low Pass Filter Coefficient			Filter Mode		

From Bit5 to Bit3 determine the coefficient of Low Pass Filter, see Table 4-7.

From Bit2 to Bit0 determine the Filter Mode, see Table 4-8.

Table 4-7 Configuration of Low Pass Filter Coefficient

Bit5~Bit3	the Low Pass Filter coefficient
000	No filter implemented
001	0.1
010	0.05
011	0.01
100	0.005
101	0.001

Table 4-8 Configuration of Filter Mode

Bit2~Bit0	Filter Mode
000	GYR
001	GYR+ACC (Kalman)
010	GYR+ACC+MAG (Kalman)
011	GYR+ACC (DCM)

100	GYR+ACC_MAG (DCM)

FIRMWARE_VERSION_0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-	-	-	-	Revision number			

Bit3 ~ Bit0 of FIRMWARE_VERSION_0 determine the revision number of firmware version.

FIRMWARE_VERSION_1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Major number				Minor number			

Bit7 ~ Bit4 of FIRMWARE_VERSION_1 determine the major number of firmware version.

Bit3 ~ Bit0 determine the minor number of firmware version.

Example: If firmware version is 0.1.0, then FIRMWARE_VERSION_0 is 0x00 and FIRMWARE_VERSION_1 is 0x01.

4.3 SPI (Reserved)

The Serial Peripheral Interface (SPI) protocol supports full-duplex synchronous serial communication with external devices. The SPI interface of LPMS-ME1 have four pins including CS, SDI, SDO and SPC.LPMS-ME1 acts as a slave, see Figure 4.7.

CS: Slave select pin, controlled by Master.

SDI: Master Out / Slave In data. In the general case, this pin is used to receive data.

SDO: Master In / Slave Out data. In the general case, this pin is used to transmit data.

SPC: Serial Clock input pin.

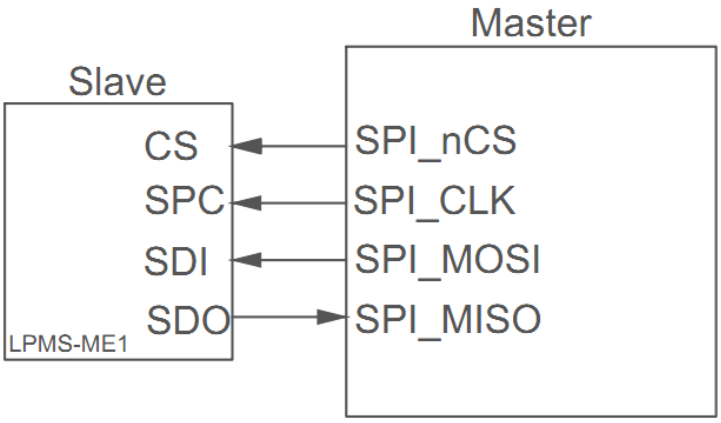

Figure 4.7. LPMS-ME1 (SPI Mode) connection example

5. Specifications

Table 5-1 Main Specifications

Parameter	Value
Name	LPMS-ME1
Size	12.0x12.0x2.6mm
Weight	0.3g
Orientation Measurement Range	360 °about all axes
Resolution	0.01 °
Accuracy	<0.5 °（static），<2 °（dynamic）
Available Output Data	Raw data/Euler angle/Quaternion
Power Source	3.3-5.5V
Output Data Rate	100Hz (Default, 5~400Hz Selectable)
Power Consumption (100Hz, UART)	<20mA @ 3.3V
Temperature Range	-40~+80℃
Communication Interfaces	UART/I^{2}C/SPI

Table 5-2 Acceleration Characteristics

Parameter	Typical Value	Unit
Measurement Range	±2/ ±4/ ±8/ ±16	g
Sensitivity	0.061/0.122/0.244/0.488	mg/LSB
Sensitivity change vs. temperature	±1	%
Typical zero-g level offset accuracy	±40	mg
Zero-g level change vs. temperature	±0.5	mg/ ℃
Noise density	90 (FS= ±2 g ODR = 104 Hz)	μg/√Hz

Table 5-3 Gyroscope Characteristics

Parameter	Typical Value	Unit
Measurement Range	±125/±245/±500/±1000/±2000	dps
Sensitivity	4.375/8.75/17.50/35/70	mdps/LSB
Sensitivity change vs. temperature	±1.5	%
Typical zero-g level offset accuracy	±10	dps
Zero-g level change vs. temperature	±0.05	dps/ ℃
Noise density	7	mdps/√Hz

Table 5-4 Magnetometer Characteristics

Parameter		Typical Value	Unit
Measurement Range		±4/±8/±12/±16	gauss
Sensitivity		6842/3421/2281/1711	LSB/gauss
Zero-gauss level		±1	gauss
RMS noise (FS = ±12gauss; Ultra-high-performance mode)	X-axis	3.2	mgauss
	Y-axis	3.2	mgauss
	Z-axis	4.1	mgauss
Non-linearity		±0.12	%FS

6. Communication Protocol

6.1 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.

6.2 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 use 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 6-3 for the default transmitted data types.)

Figure 6.1 shows the flowchart for changing the parameters of LPMS devices.

Notes:

- Only 4 commands are executable in streaming mode, see Figure 6.1.

- 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 6.1.

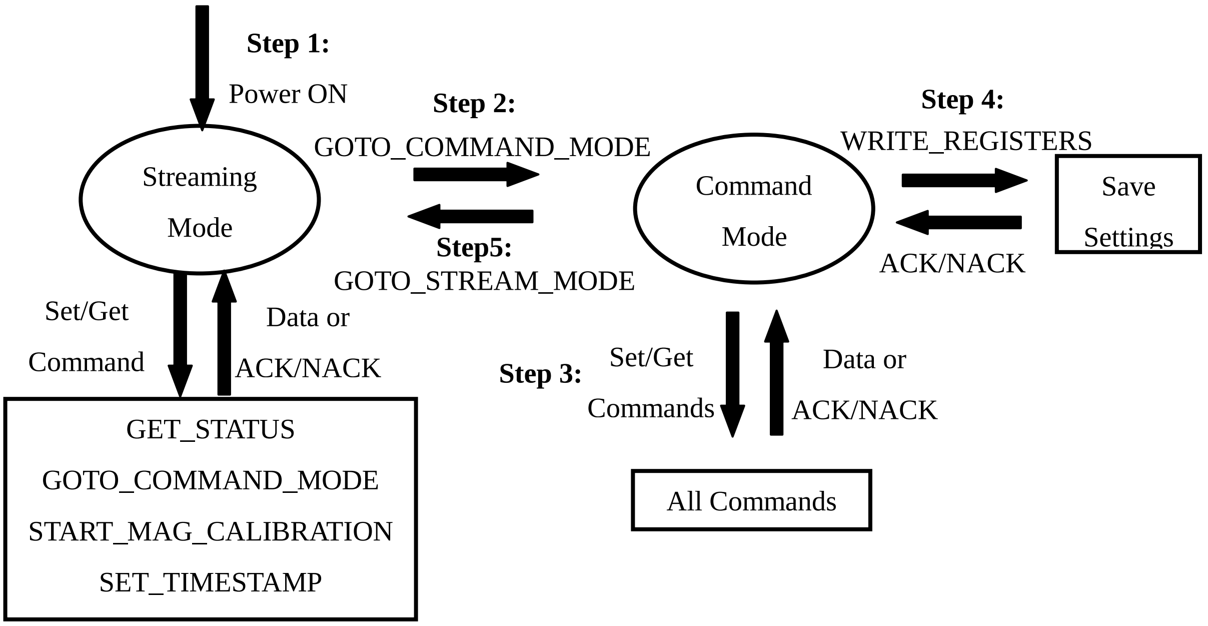

Figure 6.1. Flowchart of sensor parameters setting

6.3 LPBUS Packet Structure

Table 6-1 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 6-2 shows the data format and order of each senor data type inside a packet. Please refer to Table 6-5 for the definition of each data format identifier.

Table 6-2 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 6-3 shows the data format, sensor data order and relative pre-scale factor in 16-bit data transmission mode.

Table 6-3 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 6-4 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 6-2 and Table 6-3, 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

6.4 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

7. Package Information

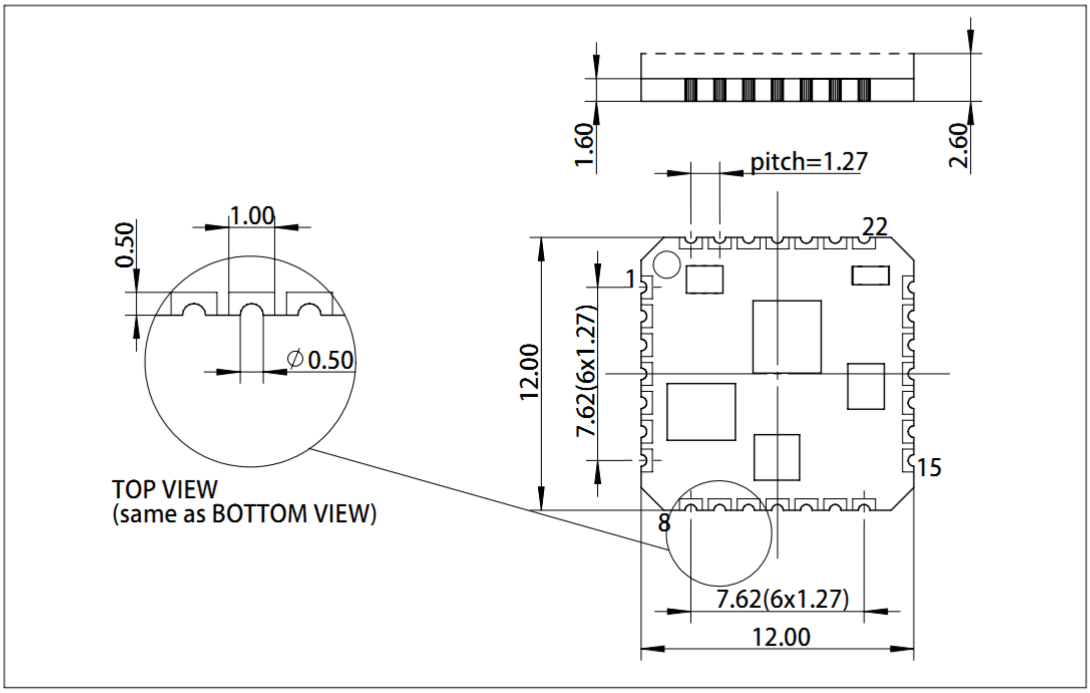

Figure 7.1. LPMS-ME1 Dimension (unit: mm)

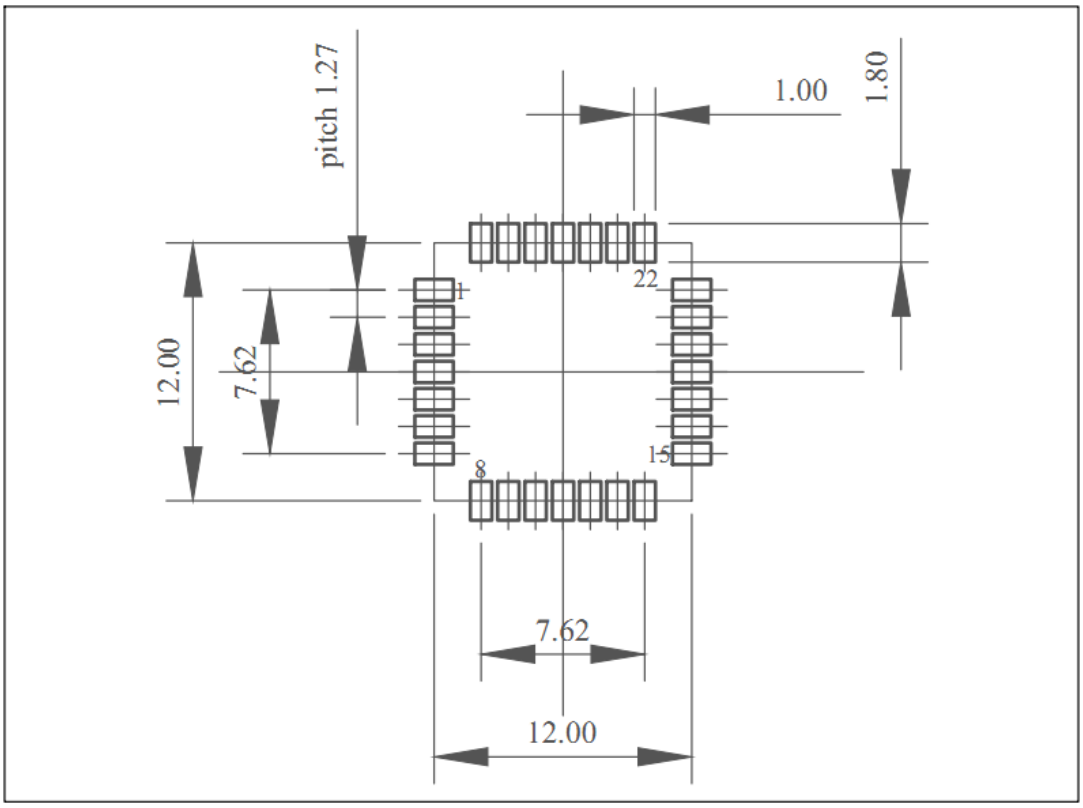

Figure 7.2. LPMS-ME1 Recommended Footprint (unit: mm)

8. Appendix

8.1 Firmware function / command list

Applies to LPMS-ME1 Firmware 2.0.8

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

<table>
<tr><td><b>Identifier</b></td><td>4 (0x04)</td></tr>
<tr><td><b>Name</b></td><td>GET_CONFIG</td></tr>
<tr><td><b>Description</b></td><td>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.</td></tr>
<tr><td><b>Parameter</b></td><td>NONE</td></tr>
<tr><td><b>Response:</b></td><td>Int32</td></tr>
<tr><td><b>Data format</b></td><td>
<table>
<tr><th>Bit</th><th>Reported State / Parameter</th></tr>
<tr><td><b>0 - 2</b></td><td>Stream frequency setting (see SET_STREAM_FREQ)</td></tr>
<tr><td><b>3 - 8</b></td><td>Reserved</td></tr>
<tr><td><b>9</b></td><td>Reserved</td></tr>
<tr><td><b>10</b></td><td>Magnetometer data transmission enabled</td></tr>
<tr><td><b>11</b></td><td>Accelerometer data transmission enabled</td></tr>
<tr><td><b>12</b></td><td>Gyroscope data transmission enabled</td></tr>
<tr><td><b>13</b></td><td>Temperature output enabled</td></tr>
<tr><td><b>14</b></td><td>Reserved</td></tr>
<tr><td><b>15</b></td><td>Reserved</td></tr>
<tr><td><b>16</b></td><td>Angular velocity output enabled</td></tr>
<tr><td><b>17</b></td><td>Euler angle data transmission enabled</td></tr>
<tr><td><b>18</b></td><td>Quaternion orientation output enabled</td></tr>
<tr><td><b>19</b></td><td>Reserved</td></tr>
<tr><td><b>20</b></td><td>Reserved</td></tr>
<tr><td><b>21</b></td><td>Linear acceleration output enabled</td></tr>
<tr><td><b>22</b></td><td>16-bit data output mode enabled</td></tr>
<tr><td><b>23</b></td><td>Reserved</td></tr>
<tr><td><b>24</b></td><td>Magnetometer compensation enabled</td></tr>
<tr><td><b>25</b></td><td>Accelerometer compensation enabled</td></tr>
<tr><td><b>26</b></td><td>Reserved</td></tr>
<tr><td><b>27</b></td><td>Reserved</td></tr>
<tr><td><b>28</b></td><td>Reserved</td></tr>
<tr><td><b>29</b></td><td>Reserved</td></tr>
<tr><td><b>30</b></td><td>Gyroscope auto-calibration enabled</td></tr>
<tr><td><b>31</b></td><td>Reserved</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td><b>Identifier</b></td><td>5 (0x05)</td></tr>
<tr><td><b>Name</b></td><td>GET_STATUS</td></tr>
<tr><td><b>Description</b></td><td>Get the current value of the status register of the sensor. The status word is read-only</td></tr>
<tr><td><b>Parameter</b></td><td>NONE</td></tr>
<tr><td><b>Response:</b></td><td>Int32</td></tr>
<tr><td><b>Data format</b></td><td>
<table>
<tr><th>Bit</th><th>Indicated state</th></tr>
<tr><td><b>0</b></td><td>COMMAND mode enabled</td></tr>
<tr><td><b>1</b></td><td>STREAM mode enabled</td></tr>
<tr><td><b>2</b></td><td>Reserved</td></tr>
<tr><td><b>3</b></td><td>Gyroscope calibration running</td></tr>
<tr><td><b>4</b></td><td>Magnetometer calibration running</td></tr>
<tr><td><b>5</b></td><td>Gyroscope initialization failed</td></tr>
<tr><td><b>6</b></td><td>Accelerometer initialization failed</td></tr>
<tr><td><b>7</b></td><td>Magnetometer initialization failed</td></tr>
<tr><td><b>8</b></td><td>Reserved</td></tr>
<tr><td><b>9</b></td><td>Gyroscope unresponsive</td></tr>
<tr><td><b>10</b></td><td>Accelerometer unresponsive</td></tr>
<tr><td><b>11</b></td><td>Magnetometer unresponsive</td></tr>
<tr><td><b>12</b></td><td>Flash write failed</td></tr>
<tr><td><b>13-31</b></td><td>Reserved</td></tr>
</table>
</td></tr>
</table>

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)

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

<table>
<tr><td><b>Identifier</b></td><td>26 (0x1A)</td></tr>
<tr><td><b>Name</b></td><td>GET_GYR_RANGE</td></tr>
<tr><td><b>Description</b></td><td>Get current gyroscope range.</td></tr>
<tr><td><b>Parameter</b></td><td>NONE</td></tr>
<tr><td><b>Response:</b></td><td>Int32
<table>
<tr><th>Range (deg/s)</th><th>Identifier</th></tr>
<tr><td><b>125</b></td><td>125</td></tr>
<tr><td><b>245</b></td><td>245</td></tr>
<tr><td><b>500</b></td><td>500</td></tr>
<tr><td><b>1000</b></td><td>1000</td></tr>
<tr><td><b>2000</b></td><td>2000</td></tr>
</table>
</td></tr>
</table>

Accelerometer Settings Command

<table>
<tr><td><b>Identifier</b></td><td>31 (0x1F)</td></tr>
<tr><td><b>Name</b></td><td>SET_ACC_RANGE</td></tr>
<tr><td><b>Description</b></td><td>Set the current range of the accelerometer</td></tr>
<tr><td><b>Parameter</b></td><td>Int32
<table>
<tr><th>Range</th><th>Identifier</th></tr>
<tr><td><b>2g</b></td><td>2</td></tr>
<tr><td><b>4g</b></td><td>4</td></tr>
<tr><td><b>8g</b></td><td>8</td></tr>
<tr><td><b>16g</b></td><td>16</td></tr>
</table>
</td></tr>
<tr><td><b>Response:</b></td><td>ACK (success) or NACK (error)</td></tr>
</table>

<table>
<tr><td><b>Identifier</b></td><td>32 (0x20)</td></tr>
<tr><td><b>Name</b></td><td>GET_ACC_RANGE</td></tr>
<tr><td><b>Description</b></td><td>Get the current range of the accelerometer</td></tr>
<tr><td><b>Parameter</b></td><td>NONE</td></tr>
<tr><td><b>Response:</b></td><td>Int32
<table>
<tr><th>Range</th><th>Identifier</th></tr>
<tr><td><b>2g</b></td><td>2</td></tr>
<tr><td><b>4g</b></td><td>4</td></tr>
<tr><td><b>8g</b></td><td>8</td></tr>
<tr><td><b>16g</b></td><td>16</td></tr>
</table>
</td></tr>
</table>

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)

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

<table>
<tr><td><b>Identifier</b></td><td>34 (0x22)</td></tr>
<tr><td><b>Name</b></td><td>GET_MAG_RANGE</td></tr>
<tr><td><b>Description</b></td><td>Get current magnetometer range.</td></tr>
<tr><td><b>Parameter</b></td><td>NONE</td></tr>
<tr><td><b>Response:</b></td><td>Int32
<table>
<tr><th>Range</th><th>Identifier</th></tr>
<tr><td><b>4 Gauss</b></td><td>4</td></tr>
<tr><td><b>8 Gauss</b></td><td>6</td></tr>
<tr><td><b>12 Gauss</b></td><td>12</td></tr>
<tr><td><b>16 Gauss</b></td><td>16</td></tr>
</table></td></tr>
</table>

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

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

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

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

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

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

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

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

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

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

<table>
<tr><td><b>Identifier</b></td><td>44 (0x2C)</td></tr>
<tr><td><b>Name</b></td><td>GET_ FILTER_PRESET</td></tr>
<tr><td><b>Description</b></td><td>Get current filter preset</td></tr>
<tr><td><b>Parameter</b></td><td>NONE</td></tr>
<tr><td><b>Response:</b></td><td>Int32
<table>
<tr><th>Correction strength</th><th>Value</th></tr>
<tr><td><b>Dynamic</b></td><td>0</td></tr>
<tr><td><b>Strong</b></td><td>1</td></tr>
<tr><td><b>Medium</b></td><td>2</td></tr>
<tr><td><b>Weak</b></td><td>3</td></tr>
</table>
</td></tr>
</table>

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

Tokyo – Guangzhou – Munich
www.lp-research.com