

# LPMS-BE2

## 用户手册 ver. 1.2



广州阿路比电子科技有限公司

<https://www.alubi.cn>



## 历史版本

日期	版本号	更改
2022-08-02	ver. 1.2	<ul style="list-style-type: none"><li>• 更新产品图片</li></ul>
2022-05-06	ver. 1.1	<ul style="list-style-type: none"><li>• 更新网站信息、应用领域</li><li>• 增加输出特性介绍</li><li>• 增加数据包解析例子</li></ul>
2021-07-19	ver. 1.0	<ul style="list-style-type: none"><li>• 初始发布</li></ul>



## 目录

1. 产品介绍 .....	4 -
2. 输出特性介绍 .....	5 -
2.1 坐标系 .....	5 -
2.2 原始数据 .....	5 -
2.2.1 加速度计 .....	6 -
2.2.2 陀螺仪 .....	6 -
2.3 姿态数据 .....	6 -
2.3.1 四元数 .....	6 -
2.3.2 欧拉角 .....	6 -
2.4 坐标系转换 .....	6 -
2.4.1 Heading reset .....	7 -
2.4.2 Alignment reset .....	7 -
2.4.3 Object reset .....	7 -
3. 通信协议 .....	8 -
3.1 通信模式 .....	8 -
3.2 LPBUS 协议 .....	8 -
3.3 LPBUS 数据包结构 .....	9 -
3.4 数据转换例子 .....	12 -
32 位浮点型模式下输出默认数据类型 .....	12 -
16 位带符号整型模式下输出默认数据类型 .....	13 -
3.5 LPBUS 通信例程 .....	16 -
将传感器进入 Command Mode .....	16 -
将传感器进入 Streaming Mode .....	17 -
读取陀螺仪范围 .....	18 -
设置加速度计范围 .....	19 -
保存参数设置 .....	20 -
设置 UART 波特率 .....	22 -
3.6 ASCII 格式数据输出 .....	23 -
4. 附录 COMMAND LIST .....	24 -
Summary .....	24 -
Acknowledged and Not-acknowledged Identifiers .....	27 -
Register Value Save and Reset Command .....	27 -



<i>Mode Switching Commands</i> .....	- 27 -
<i>Sensor Status Command</i> .....	- 28 -
<i>GetData Commands</i> .....	- 28 -
<i>Device Info Commands</i> .....	- 28 -
<i>Data Transmission Commands</i> .....	- 29 -
<i>IMU ID Setting Command</i> .....	- 30 -
<i>Stream Frequency Commands</i> .....	- 30 -
<i>Deg/Rad Output Commands</i> .....	- 31 -
<i>Reference Setting and Offset Reset Command</i> .....	- 31 -
<i>Accelerometer Settings Command</i> .....	- 32 -
<i>Gyroscope Settings Command</i> .....	- 32 -
<i>Filter Settings Command</i> .....	- 33 -
<i>UART Settings Command</i> .....	- 33 -
<i>Sensor Data Timestamp Manipulation</i> .....	- 35 -



## 1. 产品介绍

LPMS-BE2 是一款低成本、高精度的 6 轴惯性测量单元。它整合了三轴加速度计、三轴陀螺仪传感器，通过我司独有的算法进行矫正和计算，能提供包括欧拉角、四元数和加速度等信息在内的精确数据。同时，该模块体积小巧，易于组装，方便用户嵌入到自己系统中进行开发和设计。

主要特征：

- MEMS 微型惯性测量单元（IMU）
- 集成三轴陀螺仪、三轴加速度计
- 实时计算传感器的姿态方向、线性加速度以及角速度等数据
- 电源输入：3.3-5.5V
- 通信接口：UART, I2C, SPI
- 封装尺寸：PLCC-32 (12.0x14.5x3.1mm)

应用领域：

- 服务机器人
- 移动小车

## 2. 输出特性介绍

### 2.1 坐标系

LPMS-BE2 输出的数据的解读与所用的参考坐标系有关,因此需先进行相关坐标系的介绍说明。

**Sensor coordinate system(S)**是固定在 LPMS-BE2 传感器上面的坐标系,属于右手直角坐标系,即产品屏蔽盖上所标记的坐标系,如图 2.1 所示。**Object coordinate system(O)**是固定 LPMS-BE2 传感器于所检测物体上的坐标系,当(S)与(O)不是完全重合时,可通过配置 LPMS-BE2 传感器调整坐标系(S),使得两坐标系重合,具体请参考下面的相关章节。

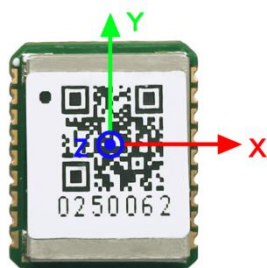


图 2.1. LPMS-BE2 参考坐标系(S)

**Global coordinate system(G)**是全球坐标系,同样属于右手直角坐标系,当算法运算有使用磁力计数据时, (G)的定义如下:

- X 轴指向地球磁场的北方(North)
- Y 轴指向地球磁场的西方(West)
- Z 轴指向上 (与重力加速度相反)

当算法运算没有使用磁力计数据时, (G)的定义如下:

- X 轴与传感器起电时(S)的 X 轴方向一致
- Y 轴方向根据坐标系右手法则所得 (已知 X 轴, Z 轴方向)
- Z 轴指向上 (与重力加速度相反)

### 2.2 原始数据

LPMS-BE2 的原始数据输出一般包括三轴的角速度和三轴的加速度数据。原始数据的参考坐标系为(S)或者(O), 其中角速度数据以右旋方向为正。



### 2.2.1 加速度计

当传感器在正轴的同一直线上加速时，将产生正加速度输出。当传感器静止且 Z 轴朝上时，校准后的加速度计数据输出为[0, 0, -1g]，表示由于重力而产生的加速度。

### 2.2.2 陀螺仪

陀螺仪的输出遵循右手规则，沿轴按右旋方式旋转将产生正角速度。当传感器处于静止状态时，校准后的陀螺仪数据输出应当是接近零的。

## 2.3 姿态数据

姿态数据表示的是由(G)旋转至(S)或者(O)的关系，共有两种表示方式：四元数和欧拉角。

### 2.3.1 四元数

由(G)旋转至(S)或者(O)可以等效成(G)绕某一转轴旋转一定的角度所得。写成向量的形式为  $q = (x, y, z, w)$ ，且  $|q|=1$ ，其中  $w$  为实部， $x, y, z$  为虚部。用四元数表示姿态的变化不会产生奇点。

### 2.3.2 欧拉角

欧拉角描述了刚体按照一个特定的旋转顺序连续旋转三次的运动。本文中说明的欧拉角使用 roll, pitch, yaw 表示，旋转顺序遵循航空公约（Z-Y-X 顺序）从(G)旋转至(S)或(O)。

- $\Psi = \text{yaw} =$  绕(G)的 Z 轴旋转的角度，范围为  $-180^\circ \sim 180^\circ$ （旋转后得坐标系(G')）
- $\Theta = \text{pitch} =$  绕(G')的 Y 轴旋转的角度，范围为  $-90^\circ \sim 90^\circ$ （旋转后得坐标系(G'')）
- $\Phi = \text{roll} =$  绕(G'')的 X 轴旋转的角度，范围为  $-180^\circ \sim 180^\circ$

注：由于欧拉角的定义，当(S)或(O)的 X 轴向上或向下时（即  $\text{pitch} = \pm 90^\circ$ ）会产生一个数学奇点，这时欧拉角数据不可信。而用四元数时这种奇异性则不存在。

## 2.4 坐标系转换

在某些情况下，可能 LPMS-BE2 的坐标轴与需要检测的对象的坐标轴不完全对齐，即(S)与(O)的坐标轴不对齐。另外，有时可能还需要根据具体情况检测在不同的坐标系((S')或(G'))下的数据。因此，我们提供了以下共三种坐标系调整模式使用。

### 2.4.1 Heading reset

通常情况下，保持(**G**)的 Z 轴垂直向上是非常重要的，但在某些情况下需要(**G**)的 X 轴指向一特定的方向，这时候就需要用到 Heading reset 了。当使用 Heading reset 后，(**G**)会变为(**G'**)，具体表现为 yaw 角变为 0 度，roll 和 pitch 角度数值保持不变，(**G'**)的定义如下：

- X 轴方向与(**S**)的 X 轴方向在水平面上的投影一致
- Y 轴方向根据坐标系右手法则所得
- Z 轴方向垂直向上（与重力加速度方向相反）

注：使用 Heading reset 后，yaw 可能不会是绝对的 0 度，尤其当(**S**)的 X 轴接近垂直时，这是因为当 pitch 接近 $\pm 90^\circ$ 时产生的奇异性令 yaw 角不稳定。

### 2.4.2 Alignment reset

当使用 Alignment reset 后，(**S**)会变为(**S'**)，具体表现为 roll 和 pitch 角变为 0 度，(**S'**)的定义如下：

- X 轴方向与(**S**)的 X 轴方向在水平面上的投影一致
- Y 轴方向根据坐标系右手法则所得
- Z 轴方向与(**G**)的 Z 轴方向一致

### 2.4.3 Object reset

使用 Object reset 是为了使得(**S**)与(**O**)重合。当使用 Object reset 后，(**S**)会变为(**S'**)，且(**S'**)、(**O**)重合，(**S'**)和(**O**)固定于传感器和被检测物体上随物体移动而移动，具体表现为 roll, pitch 和 yaw 角变为 0 度，(**S'**)的定义如下：

- X 轴方向与(**S**)的 X 轴方向在水平面上的投影一致
- Y 轴方向根据坐标系右手法则所得
- Z 轴方向垂直向上（与重力加速度方向相反）

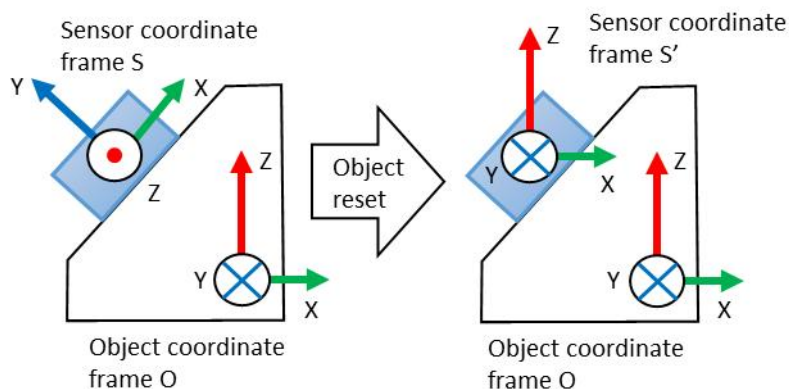


图 2.2. Object reset 示意图



### 3. 通信协议

#### 3.1 通信模式

LPMS 设备具有数据流模式（**Streaming Mode**）和命令模式（**Command Mode**）两种通信模式。数据流模式下，传感器以设定好的频率不断往外发送数据（发送的数据类型和格式可进行设置）。命令模式下，需通过发送指令与传感器进行通信，可对传感器的参数进行设置。**LPMS-BE2 出厂时为数据流模式**，可在该模式下进行保存设置操作。（注：通过上位机保存设置时，会保存为命令模式，上位机使用请参考其他相关手册。）

图 2.1 为对传感器进行参数设置的流程。通常首先要使传感器进入命令模式再进行参数的修改或者读取。

注：进入命令模式修改参数过后一定要保存设置，否则断电后修改无效。

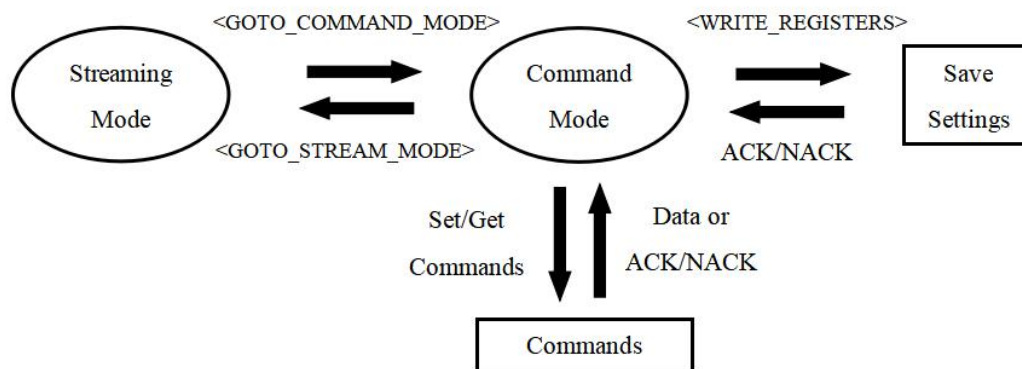


图 2.1 传感器参数设置流程

#### 3.2 LPBUS 协议

LPBUS 是基于工业标准的 MODBUS 协议所设计的通信协议。这是 LPMS 设备默认的通信方式。通信协议数据包结构具体请参照 2.3 节。

一个 LPBUS 通信包含有两项基本的指令形式：**GET** 和 **SET** 指令。指令均由主机（PC、移动数据记录单元等）发送给从机（LPMS 设备）。下面我们会对具体的指令进行详细的介绍，包括指令的类型和传输的数据。

**GET 指令：**读取从机的数据需通过发送 **GET** 指令实现。一个 **GET** 请求指令通常不包含任何数据，所请求的数据由从机收到 **GET** 指令后发出。

**SET 指令：**从机数据寄存器的值通过发送 **SET** 指令设置。一个 **SET** 指令包含要设置的数据，从机的返回值为 **ACK**（代表成功写入寄存器）或者 **NACK**（代表写入寄存器失败）。

详细的指令表请查看附录。



### 3.3 LPBUS 数据包结构

LPBUS 通信协议的每一个数据包的结构组成如表 2-1 所示。

表 2-1 数据包结构组成

字节#	名称	描述
0	包头	3Ah
1	Sensor ID（低位字节）	包含传感器的 Sensor ID 号的低位字节。ID 号默认值为 1。主机可使用该 ID 号发送 GET/SET 指令到特定的 LPMS 设备中，从机将返回相同的 ID 号。ID 号可通过发送 SET 指令修改。
2	Sensor ID（高位字节）	包含传感器的 Sensor ID 号的高位字节。
3	指令号（低位字节）	包含所要执行的指令号的低位字节。
4	指令号（高位字节）	包含所要执行的指令号的高位字节。
5	数据长度（低位字节）	包含所要传输的数据长度的低位字节。
6	数据长度（高位字节）	包含所要传输的数据长度的高位字节。
x	数据（n 个字节）	如果数据长度 n 不等于 0，那么 $x=6+n$ ；若 n 为 0，则 x 为空。 如果数据长度不为 0，则该数据含有要传输的数据包，反之该数据为空。
7+n	LRC（低位字节）	LRC 校验的低位字节。为了确保传输数据不失真，我们使用了 LRC 校验和的方法，具体计算如下所示： $LRC = \text{sum}(\text{Sensor ID}, \text{指令号}, \text{数据长度}, \text{数据})$ 。 计算出来的 LRC 通常与从远程设备传输过来的 LRC 进行比较，如果这两个 LRC 不相等则产生错误报告。
8+n	LRC（高位字节）	LRC 校验的高位字节。
9+n	包尾（低位字节）	0Dh
10+n	包尾（高位字节）	0Ah

数据包中的数据部分，通常以小端格式传输，即低位字节在前，高位字节在后。

数据包中的数据部分，数据格式有两种：

- 32 位浮点型数据格式
- 16 位带符号整型格式

传感器默认设置为 32 位浮点型数据格式（时间戳除外，其恒为 32 位无符号整型），如表 2-2 所示，其表示数据包中数据部分传输的数据类型、传输顺序及其数据格式。数据格式标识符定义请查看表 2-4。



表 2-2 数据包中数据部分的数据格式（32 位浮点型模式下）

顺序#	数据格式标识符	传感器数据
1	UInt32	时间戳 (500Hz 更新率,即 0.002s)
2	Vector3f	原始的加速度计数据 (g)
3	Vector3f	校准后的加速度计数据 (g)
4	Vector3f	原始的陀螺仪数据 ( <b>dps (默认)</b> / rad/s)
5	Vector3f	静止偏差校准后的陀螺仪数据 ( <b>dps (默认)</b> / rad/s)
6	Vector3f	坐标轴校准后的陀螺仪数据 ( <b>dps (默认)</b> / rad/s)
7	Vector3f	角速度 (rad/s)
8	Vector4f	四元数 (归一化单位)
9	Vector3f	欧拉角数据 ( <b>degree (默认)</b> / rad)
10	Vector3f	线性加速度数据 (g)
11	Vector3f	温度数据(°C)

如用户修改传感器为 16 位带符号整型数据格式，数据将先通过应用一乘法因子后，再进行数据打包传输，以提高数据传输精度，部分数据根据不同输出单位其乘法因子也会有所不同，具体请看表 2-3。

表 2-3 数据包中数据部分的数据格式（16 位带符号整型模式下）

顺序#	数据格式	传感器数据	乘法因子
1	UInt32	时间戳 (500Hz 更新率,即 0.002s)	无
2	Vector3i16	原始的加速度计数据 (g)	1000
3	Vector3i16	校准后的加速度计数据 (g)	1000
4	Vector3i16	原始的陀螺仪数据 ( <b>dps (默认)</b> / rad/s)	10/100
5	Vector3i16	静止偏差校准后的陀螺仪数据 ( <b>dps (默认)</b> / rad/s)	10/100
6	Vector4i16	坐标轴校准后的陀螺仪数据 ( <b>dps (默认)</b> / rad/s)	10/100
7	Vector3i16	角速度 (rad/s)	100
8	Vector3i16	四元数 (归一化单位)	10000
9	Vector3i16	欧拉角数据 ( <b>degree (默认)</b> / rad)	100/10000
10	Vector3i16	线性加速度数据 (g)	1000
11	Vector3i16	温度数据(°C)	100



表 2-4 数据格式标识符定义

标识符	描述
UInt32	32 位无符号整型
Int32	32 位带符号整型
Int16	16 位带符号整型
Float32	32 位浮点型
Vector3f	3 元素 32 位浮点型向量
Vector3i16	3 元素 16 位带符号整型向量
Vector4f	4 元素 32 位浮点型向量
Vector4i16	4 元素 16 位带符号整型向量
Matrix3x3f	3x3 浮点型矩阵

传感器数据包中的数据部分，数据类型的传输顺序规定为如表 2-2 和 2-3 中所示，即从顺序#1 到#11 所示的数据类型，最多 11 种。时间戳不能被用户修改，其恒为输出数据。顺序#2 到#11 中的数据，如有任何一数据类型没有被使能输出，则顺延传输下一个被使能的数据，顺序号相对地往前移。

**默认设置下，传感器只输出以下顺序的数据类型（共 8 种）：**

1. 时间戳
2. 原始的加速度计数据
3. 校准后的加速度计数据
4. 原始的陀螺仪数据
5. 静止偏差校准后的陀螺仪数据
6. 坐标轴校准后的陀螺仪数据
7. 四元数
8. 欧拉角数据



### 3.4 数据转换例子

以下将列举具体的数据包解析例子。

#### 32 位浮点型模式下输出默认数据类型

如果接收到的数据包为（十六进制表示）：

3A 01 00 09 00 5C 00 C6 A7 00 00 00 00 3C BC 00 00 80 BC 00 88 80 BF E5 39 2A  
BC 76 14 64 3B 96 F2 7E BF 29 5C 0F 3E 33 33 33 3F 3E 0A D7 BE 29 5C 0F 3E 33  
33 33 3F 3E 0A D7 BE 24 63 18 3E 6A AD 32 3F 64 A0 D2 BE AE 6E 7B 3F 35 A5 6A  
BC 06 79 47 BD ED 6C 39 3E 99 BB 1A 3F 2C 37 B9 40 55 E5 A6 C1 9A 24 0D 0A

则解析如下表所示：

描述	接收到的数据	原始数据	转换的数据	乘法因子	真实数据	单位
包头	3A					
Sensor ID	01 00	0001	1			
指令号	09 00	0009	9			
数据长度	5C 00	005C	92			
时间戳	C6 A7 00 00	0000A7C6	42950	0.002	85.9	seconds
原始的 加速度计数据 X	00 00 3C BC	BC3C0000	-0.01147161	1	-0.01147161	g
原始的 加速度计数据 Y	00 00 80 BC	BC800000	-0.015625	1	-0.015625	g
原始的 加速度计数据 Z	00 88 80 BF	BF808800	-1.00415	1	-1.00415	g
校准后的 加速度计数据 X	E5 39 2A BC	BC2A39E5	-0.01038978	1	-0.01038978	g
校准后的 加速度计数据 Y	76 14 64 3B	3B641476	0.003480223	1	0.003480223	g
校准后的 加速度计数据 Z	96 F2 7E BF	BF7EF296	-0.9958891	1	-0.9958891	g
原始的 陀螺仪数据 X	29 5C 0F 3E	3E0F5C29	0.14	1	0.14	dps
原始的 陀螺仪数据 Y	33 33 33 3F	3F333333	0.7	1	0.7	dps
原始的 陀螺仪数据 Z	3E 0A D7 BE	BED70A3E	-0.42	1	-0.42	dps



静止偏差校准后的陀螺仪数据 X	29 5C 0F 3E	3E0F5C29	0.14	1	0.14	dps
静止偏差校准后的陀螺仪数据 Y	33 33 33 3F	3F333333	0.7	1	0.7	dps
静止偏差校准后的陀螺仪数据 Z	3E 0A D7 BE	BED70A3E	-0.42	1	-0.42	dps
坐标轴校准后的陀螺仪数据 X	24 63 18 3E	3E186324	0.1488157	1	0.1488157	dps
坐标轴校准后的陀螺仪数据 Y	6A AD 32 3F	3F32AD6A	0.6979586	1	0.6979586	dps
坐标轴校准后的陀螺仪数据 Z	64 A0 D2 BE	BED2A064	-0.4113799	1	-0.4113799	dps
四元数 w	AE 6E 7B 3F	3F7B6EAE	0.9821576	1	0.9821576	-
四元数 x	35 A5 6A BC	BC6AA535	-0.01432161	1	-0.01432161	-
四元数 y	06 79 47 BD	BD477906	-0.0486994	1	-0.0486994	-
四元数 z	ED 6C 39 3E	3E396CED	0.1810796	1	0.1810796	-
欧拉角数据 X	99 BB 1A 3F	3F1ABB99	0.604425	1	0.604425	degree
欧拉角数据 Y	2C 37 B9 40	40B9372C	5.787985	1	5.787985	degree
欧拉角数据 Z	55 E5 A6 C1	C1A6E555	-20.86798	1	-20.86798	degree
LRC 校验和*	9A 24	249A				
包尾	0D 0A					

\*LRC 校验和计算：

01+00+09+00+5C+00+C6+A7+00+00+00+00+3C+BC+00+00+80+BC+00+88+80+BF  
 +E5+39+2A+BC+76+14+64+3B+96+F2+7E+BF+29+5C+0F+3E+33+33+33+3F+3E+  
 0A+D7+BE+29+5C+0F+3E+33+33+33+3F+3E+0A+D7+BE+24+63+18+3E+6A+AD+  
 32+3F+64+A0+D2+BE+AE+6E+7B+3F+35+A5+6A+BC+06+79+47+BD+ED+6C+39+  
 3E+99+BB+1A+3F+2C+37+B9+40+55+E5+A6+C1 = 249A

## 16 位带符号整型模式下输出默认数据类型

如果接收到的数据包为（十六进制表示）：

3A 01 00 09 00 30 00 D8 49 01 00 F5 FF F1 FF 16 FC F6 FF 04 00 1E FC 02 00 07 00  
 FE FF 02 00 07 00 FE FF 02 00 07 00 FE FF A5 24 15 FF 45 FE 64 0D 4F 00 39 02 56  
 F0 D8 14 0D 0A



则解析如下表所示:

描述	接收到的数据	原始数据	转换的数据	乘法因子	真实数据	单位
包头	3A					
Sensor ID	01 00	0001	1			
指令号	09 00	0009	9			
数据长度	30 00	0030	48			
时间戳	D8 49 01 00	000149D8	84440	0.002	168.88	seconds
原始的 加速度计数据 X	F5 FF	FFF5	-11	1/1000	-0.011	g
原始的 加速度计数据 Y	F1 FF	FFF1	-15	1/1000	-0.015	g
原始的 加速度计数据 Z	16 FC	FC16	-1002	1/1000	-1.002	g
校准后的 加速度计数据 X	F6 FF	FFF6	-10	1/1000	-0.01	g
校准后的 加速度计数据 Y	04 00	0004	4	1/1000	0.004	g
校准后的 加速度计数据 Z	1E FC	FC1E	-994	1/1000	-0.994	g
原始的 陀螺仪数据 X	02 00	0002	2	1/10	0.2	dps
原始的 陀螺仪数据 Y	07 00	0007	7	1/10	0.7	dps
原始的 陀螺仪数据 Z	FE FF	FFFE	-2	1/10	-0.2	dps
静止偏差校准后的 陀螺仪数据 X	02 00	0002	2	1/10	0.2	dps
静止偏差校准后的 陀螺仪数据 Y	07 00	0007	7	1/10	0.7	dps
静止偏差校准后的 陀螺仪数据 Z	FE FF	FFFE	-2	1/10	-0.2	dps
坐标轴校准后的 陀螺仪数据 X	02 00	0002	2	1/10	0.2	dps
坐标轴校准后的 陀螺仪数据 Y	07 00	0007	7	1/10	0.7	dps



坐标轴校准后的 陀螺仪数据 Z	FE FF	FFFE	-2	1/10	-0.2	dps
四元数 w	A5 24	24A5	9381	1/10000	0.9381	-
四元数 x	15 FF	FF15	-235	1/10000	-0.0235	-
四元数 y	45 FE	FE45	-443	1/10000	-0.0443	-
四元数 z	64 0D	0D64	3428	1/10000	0.3428	-
欧拉角数据 X	4F 00	004F	79	1/100	0.79	degree
欧拉角数据 Y	39 02	0239	569	1/100	5.69	degree
欧拉角数据 Z	56 F0	F056	-4010	1/100	-40.10	degree
LRC 校验和*	D8 14	14D8				
包尾	0D 0A					

\*LRC 校验和计算:

01+00+09+00+30+00+D8+49+01+00+F5+FF+F1+FF+16+FC+F6+FF+04+00+1E+FC  
+02+00+07+00+FE+FF+02+00+07+00+FE+FF+02+00+07+00+FE+FF+A5+24+15+F  
F+45+FE+64+0D+4F+00+39+02+56+F0 = 14D8





### 3.5 LPBUS 通信例程

以下将列举几个使用 LPBUS 协议的通信例程。

#### 将传感器进入 Command Mode

发送指令（主机->传感器）

数据包字节号	值	描述
0	3Ah	包头
1	01h	Sensor ID 低位字节（ID = 1）
2	00h	Sensor ID 高位字节
3	06h	指令号低位字节 （06h = GOTO_COMMAND_MODE）
4	00h	指令号高位字节
5	00h	数据长度低位字节（GOTO_COMMAND_MODE 指令数据长度为 0）
6	00h	数据长度高位字节
7	07h	校验和低位字节
8	00h	校验和高位字节
9	0Dh	包尾低位字节
10	0Ah	包尾高位字节

数据返回（传感器->主机）

数据包字节号	值	描述
0	3Ah	包头
1	01h	Sensor ID 低位字节（ID = 1）
2	00h	Sensor ID 高位字节
3	00h	指令号低位字节（00h = REPLY_ACK）
4	00h	指令号高位字节
5	00h	数据长度低位字节（返回 ACK，无数据）
6	00h	数据长度高位字节
7	01h	校验和低位字节
8	00h	校验和高位字节
9	0Dh	包尾低位字节
10	0Ah	包尾高位字节



## 将传感器进入 Streaming Mode

发送指令（主机->传感器）

数据包字节号	值	描述
0	3Ah	包头
1	01h	Sensor ID 低位字节（ID = 1）
2	00h	Sensor ID 高位字节
3	07h	指令号低位字节 （07h = GOTO_STREAMING_MODE）
4	00h	指令号高位字节
5	00h	数据长度低位字节（GOTO_STREAMING_MODE 指令数据长度为 0）
6	00h	数据长度高位字节
7	08h	校验和低位字节
8	00h	校验和高位字节
9	0Dh	包尾低位字节
10	0Ah	包尾高位字节

数据返回（传感器->主机）

数据包字节号	值	描述
0	3Ah	包头
1	01h	Sensor ID 低位字节（ID = 1）
2	00h	Sensor ID 高位字节
3	00h	指令号低位字节（00h = REPLY_ACK）
4	00h	指令号高位字节
5	00h	数据长度低位字节（返回 ACK，无数据）
6	00h	数据长度高位字节
7	01h	校验和低位字节
8	00h	校验和高位字节
9	0Dh	包尾低位字节
10	0Ah	包尾高位字节



## 读取陀螺仪范围

发送指令（主机->传感器）

数据包字节号	值	描述
0	3Ah	包头
1	01h	Sensor ID 低位字节（ID = 1）
2	00h	Sensor ID 高位字节
3	3Dh	指令号低位字节（3Dh = GET_GYR_RANGE）
4	00h	指令号高位字节
5	00h	数据长度低位字节（GET_GYR_RANGE 指令数据长度为 0）
6	00h	数据长度高位字节
7	3Eh	校验和低位字节
8	00h	校验和高位字节
9	0Dh	包尾低位字节
10	0Ah	包尾高位字节

数据返回（传感器->主机）

数据包字节号	值	描述
0	3Ah	包头
1	01h	Sensor ID 低位字节（ID = 1）
2	00h	Sensor ID 高位字节
3	3Dh	指令号低位字节（3Dh = GET_GYR_RANGE）
4	00h	指令号高位字节
5	04h	数据长度低位字节（32 位整型为 4 个字节）
6	00h	数据长度高位字节
7	xxh	配置数据字节 1（最低位）
8	xxh	配置数据字节 2
9	xxh	配置数据字节 3
10	xxh	配置数据字节 4（最高位）
11	xxh	校验和低位字节
12	xxh	校验和高位字节
13	0Dh	包尾低位字节
14	0Ah	包尾高位字节

注：xx 的值取决于当前的传感器配置。



## 设置加速度计范围

发送指令（主机->传感器）

数据包字节号	值	描述
0	3Ah	包头
1	01h	Sensor ID 低位字节（ID = 1）
2	00h	Sensor ID 高位字节
3	32h	指令号低位字节（32h = SET_ACC_RANGE）
4	00h	指令号高位字节
5	04h	数据长度低位字节（32 位整型为 4 个字节）
6	00h	数据长度高位字节
7	08h	范围数据字节 1（范围 8g 为 8d）
8	00h	范围数据字节 2
9	00h	范围数据字节 3
10	00h	范围数据字节 4
11	3Fh	校验和低位字节
12	00h	校验和高位字节
13	0Dh	包尾低位字节
14	0Ah	包尾高位字节

数据返回（传感器->主机）

数据包字节号	值	描述
0	3Ah	包头
1	01h	Sensor ID 低位字节（ID = 1）
2	00h	Sensor ID 高位字节
3	00h	指令号低位字节（00h = REPLY_ACK）
4	00h	指令号高位字节
5	00h	数据长度低位字节（返回 ACK，无数据）
6	00h	数据长度高位字节
7	01h	校验和低位字节
8	00h	校验和高位字节
9	0Dh	包尾低位字节
10	0Ah	包尾高位字节



## 保存参数设置

发送指令（主机->传感器）

数据包字节号	值	描述
0	3Ah	包头
1	01h	Sensor ID 低位字节（ID = 1）
2	00h	Sensor ID 高位字节
3	04h	指令号低位字节（04h = WRITE_REGISTERS）
4	00h	指令号高位字节
5	00h	数据长度低位字节 (WRITE_REGISTERS 指令数据长度为 0)
6	00h	数据长度高位字节
7	05h	校验和低位字节
8	00h	校验和高位字节
9	0Dh	包尾低位字节
10	0Ah	包尾高位字节

数据返回（传感器->主机）

数据包字节号	值	描述
0	3Ah	包头
1	01h	Sensor ID 低位字节（ID = 1）
2	00h	Sensor ID 高位字节
3	00h	指令号低位字节（00h = REPLY_ACK）
4	00h	指令号高位字节
5	00h	数据长度低位字节（返回 ACK，无数据）
6	00h	数据长度高位字节
7	01h	校验和低位字节
8	00h	校验和高位字节
9	0Dh	包尾低位字节
10	0Ah	包尾高位字节

注：保存修改的参数需要一定的时间，该指令发送后不能马上返回数据，需经过 1~2s 才有数据返回。



## 读取传感器状态

发送指令（主机->传感器）

数据包字节号	值	描述
0	3Ah	包头
1	01h	Sensor ID 低位字节（ID = 1）
2	00h	Sensor ID 高位字节
3	08h	指令号低位字节（08h =GET_SENSOR_STATUS）
4	00h	指令号高位字节
5	00h	数据长度低位字节（GET_SENSOR_STATUS 指令数据长度为 0）
6	00h	数据长度高位字节
7	09h	校验和低位字节
8	00h	校验和高位字节
9	0Dh	包尾低位字节
10	0Ah	包尾高位字节

数据返回（传感器->主机）

数据包字节号	值	描述
0	3Ah	包头
1	01h	Sensor ID 低位字节（ID = 1）
2	00h	Sensor ID 高位字节
3	08h	指令号低位字节（08h =GET_SENSOR_STATUS）
4	00h	指令号高位字节
5	04h	数据长度低位字节
6	00h	数据长度高位字节
7-10	xxxxxxxxh	返回的状态值
11	xxh	校验和低位字节
12	xxh	校验和高位字节
13	0Dh	包尾低位字节
14	0Ah	包尾高位字节

注：返回状态值的解读请参考附录。



## 设置 UART 波特率

发送指令（主机->传感器）

数据包字节号	值	描述
0	3Ah	包头
1	01h	Sensor ID 低位字节（ID = 1）
2	00h	Sensor ID 高位字节
3	82h	指令号低位字节（82h = SET_UART_BAUDRATE）
4	00h	指令号高位字节
5	04h	数据长度低位字节
6	00h	数据长度高位字节
7	00h	921600 = 0x000E1000
8	10h	
9	0Eh	
10	00h	
11	A5h	校验和低位字节
12	00h	校验和高位字节
13	0Dh	包尾低位字节
14	0Ah	包尾高位字节

数据返回（传感器->主机）

数据包字节号	值	描述
0	3Ah	包头
1	01h	Sensor ID 低位字节（ID = 1）
2	00h	Sensor ID 高位字节
3	00h	指令号低位字节（0d = REPLY_ACK）
4	00h	指令号高位字节
5	00h	数据长度低位字节（返回 ACK，无数据）
6	00h	数据长度高位字节
7	01h	校验和低位字节
8	00h	校验和高位字节
9	0Dh	包尾低位字节
10	0Ah	包尾高位字节

注：设置波特率后需重启才生效。



### 3.6 ASCII 格式数据输出

LPMS-BE2 的 UART 通讯接口支持 ASCII 输出，可以通过指令 SET\_UART\_FORMAT 设置。在 ASCII 输出格式下，数据将先通过应用一乘法因子(参考表 2-3，其中陀螺仪相关数据的因子为 100/1000)后，转换为整形字符串输出。输出格式如表 2-5 所示。

表 2-5 ASCII 输出格式

开始字符	数据 1	,	数据 2	,	.....	,	数据 n	结束字符
------	------	---	------	---	-------	---	------	------

开始字符默认为“\$”，结束字符默认为“\n”，每个数据中间逗号隔开，其中，开始字符和结束字符可以由用户自定义，通过 SET\_UART\_ASCII\_CHARACTER 指令设置。

例如，默认设置下输出以下字符串数据，解析表 2-6 所示：

\$54210,15,-21,-967,15,-21,-967,24,6,18,3,4,3,3,4,3,9999,-113,-69,3,129,79,-2

表 2-6 ASCII 数据包解析

数据项	字符	值
开始字符	\$	-
时间戳	54210	约 108s
原始的加速度计 X 轴数据	15	0.015g
原始的加速度计 Y 轴数据	-21	-0.021g
原始的加速度计 Z 轴数据	-967	-0.967g
.....	.....	
原始的陀螺仪 X 轴数据	24	0.24dps
原始的陀螺仪 Y 轴数据	6	0.06dps
原始的陀螺仪 Z 轴数据	18	0.18dps
.....	.....	
四元素 w	9999	0.9999
四元素 x	-113	-0.0113
四元素 y	-69	-0.0069
四元素 z	3	-0.0003
欧拉角 Roll	129	1.29°
欧拉角 Pitch	79	0.79°
欧拉角 Yaw	-2	-0.02°
结束字符	\n	“\n”为换行符，若设置为其他如“#”，则字符串以“#”结束

- 注：1. ASCII 输出的数据按 32 位浮点型数据进行换算。  
2. ASCII 输出格式下，仍以 LPBUS 协议格式响应用户指令。





## 4. 附录 Command List

### 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	4g

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

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	

**Acknowledged and Not-acknowledged Identifiers**

<b>Identifier</b>	0(0x00)
<b>Name</b>	REPLY_ACK
<b>Description</b>	Confirms a successful SET command.

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

**Register Value Save and Reset Command**

<b>Identifier</b>	4 (0x04)
<b>Name</b>	WRITE_REGISTERS
<b>Description</b>	Write the currently set parameters to flash memory.
<b>Parameter</b>	NONE
<b>Response:</b>	ACK (success) or NACK (error)

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

**Mode Switching Commands**

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

<b>Identifier</b>	7 (0x07)
<b>Name</b>	GOTO_STREAM_MODE
<b>Description</b>	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.
<b>Parameter</b>	NONE
<b>Response:</b>	ACK (success) or NACK (error)



### Sensor Status Command

<b>Identifier</b>	8 (0x08)	
<b>Name</b>	GET_SENSOR_STATUS	
<b>Description</b>	Get the current sensor status	
<b>Parameter</b>	NONE	
<b>Response:</b>	Int32	
	<b>Sensor status</b>	<b>Identifier</b>
	Command Mode	0
	Streaming Mode	1

### GetData Commands

<b>Identifier</b>	9 (0x09)
<b>Name</b>	GET_IMU_DATA
<b>Description</b>	Get the sensor data
<b>Parameter</b>	NONE
<b>Response</b>	Please see Chapter 3.3 for details

### Device Info Commands

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

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

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

<b>Identifier</b>	23 (0x17)
<b>Name</b>	GET_FILTER_VERSION
<b>Description</b>	Get the internal filter version information
<b>Parameter</b>	NONE
<b>Response</b>	Char[24]

**Data Transmission Commands**

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

<b>Identifier</b>	31 (0x1F)	
<b>Name</b>	GET IMU TRANSMIT DATA	
<b>Description</b>	Get the current transmitted data from sensor	
<b>Parameter</b>	NONE	
<b>Response:</b>	Int32	
	<b>Bit</b>	<b>Reported State / Parameter</b>
	0	Acceleromete raw data transmission enabled
	1	Acceleromete calibrated data transmission enabled
	2	Reserved
	3	Gyro raw data transmission enabled
	4	Reserved
	5	Gyro bias calibrated data transmission enabled
	6	Reserved
	7	Gyro alignment calibrated data transmission enabled
	8-9	Reserved
	10	Angular velocity transmission enabled
	11	Quaternion orientation transmission enabled
	12	Euler angle data transmission enabled
	13	LinAcc data transmission enabled
	14-15	Reserved
	16	Temperature data transmission enabled
	17-31	Reserved

**IMU ID Setting Command**

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

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

**Stream Frequency Commands**

<b>Identifier</b>	34 (0x22)														
<b>Name</b>	SET_STREAM_FREQ														
<b>Description</b>	Set the current streaming frequency														
<b>Parameter</b>	Int32 <table border="1" data-bbox="421 1216 948 1451"><thead><tr><th>Frequency (Hz)</th><th>Identifier</th></tr></thead><tbody><tr><td>5</td><td>5</td></tr><tr><td>10</td><td>10</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></tbody></table>	Frequency (Hz)	Identifier	5	5	10	10	50	50	100	100	250	250	500	500
Frequency (Hz)	Identifier														
5	5														
10	10														
50	50														
100	100														
250	250														
500	500														
<b>Response:</b>	ACK (success) or NACK (error)														

<b>Identifier</b>	35 (0x23)														
<b>Name</b>	GET_STREAM_FREQ														
<b>Description</b>	Get the current streaming frequency														
<b>Parameter</b>	NONE														
<b>Response:</b>	Int32 <table border="1" data-bbox="421 1787 948 2022"><thead><tr><th>Frequency (Hz)</th><th>Identifier</th></tr></thead><tbody><tr><td>5</td><td>5</td></tr><tr><td>10</td><td>10</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></tbody></table>	Frequency (Hz)	Identifier	5	5	10	10	50	50	100	100	250	250	500	500
Frequency (Hz)	Identifier														
5	5														
10	10														
50	50														
100	100														
250	250														
500	500														

**Deg/Rad Output Commands**

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

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

**Reference Setting and Offset Reset Command**

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

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



**Accelerometer Settings Command**

<b>Identifier</b>	50 (0x32)	
<b>Name</b>	SET ACC RANGE	
<b>Description</b>	Set the current range of the accelerometer	
<b>Parameter</b>	Int32	
	<b>Range</b>	<b>Identifier</b>
	2g	2
	4g	4
	8g	8
	16g	16
<b>Response:</b>	ACK (success) or NACK (error)	

<b>Identifier</b>	51 (0x33)	
<b>Name</b>	GET ACC RANGE	
<b>Description</b>	Get the current range of the accelerometer	
<b>Parameter</b>	NONE	
<b>Response:</b>	Int32	
	<b>Range</b>	<b>Identifier</b>
	2g	2
	4g	4
	8g	8
	16g	16

**Gyroscope Settings Command**

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

<b>Identifier</b>	61 (0x3D)	
<b>Name</b>	GET GYR RANGE	
<b>Description</b>	Get current gyroscope range.	
<b>Parameter</b>	NONE	
<b>Response:</b>	Int32	
	<b>Range (deg/s)</b>	<b>Identifier</b>
	125	125
	250	250
	500	500
	1000	1000
	2000	2000



<b>Identifier</b>	62 (0x3E)
<b>Name</b>	START_GYR_CALIBRATION
<b>Description</b>	Start calibrate gyro static bias.
<b>Parameter</b>	NONE
<b>Response:</b>	ACK (success) or NACK (error)

### Filter Settings Command

Identifier	90 (0x5A)		
Name	SET_FILTER_MODE		
Description	Set the sensor filter mode		
Parameter	Int32		
	Mode	Value	
	Gyroscope(Only)	0	
	Accelerometer + gyroscope (Kalman filter)	1	
	Accelerometer + gyroscope (DCM filter)	3	
Response:	ACK (success) or NACK (error)		

Identifier	91 (0x5B)		
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 (DCM filter)	3	

### UART Settings Command

Identifier	130 (0x82)	
Name	SET_UART_BAUDRATE	
Description	Set the current UART baudrate	
Parameter	Int32	
	Baud rate	Identifier
	115200	115200
	230400	230400
	256000	256000
	460800	460800
	921600	921600
Response:	ACK (success) or NACK (error)	



<b>Identifier</b>	131 (0x83)	
<b>Name</b>	GET_UART_BAUDRATE	
<b>Description</b>	Get the current UART baudrate	
<b>Parameter</b>	NONE	
<b>Response:</b>	Int32	
	<b>Baud rate</b>	<b>Identifier</b>
	115200	115200
	230400	230400
	256000	256000
	460800	460800
	921600	921600

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

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

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



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

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

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

### Sensor Data Timestamp Manipulation

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

