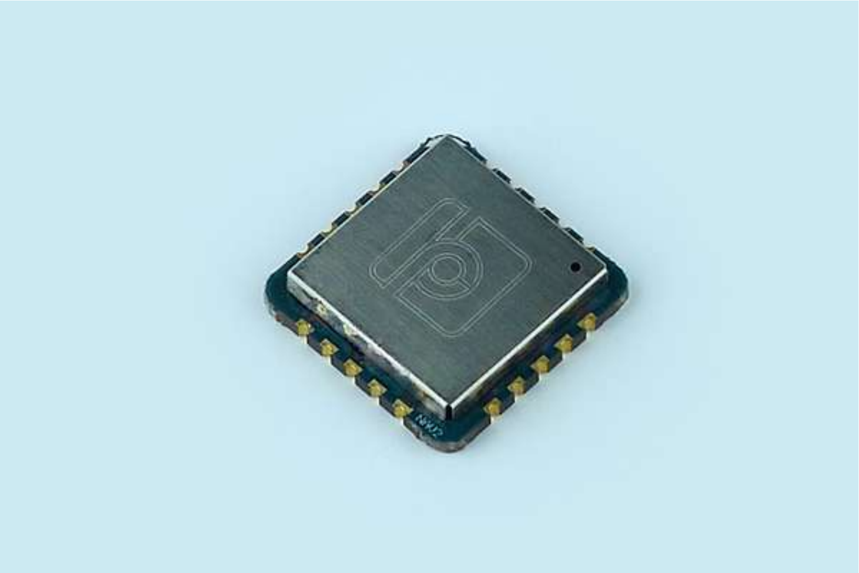
LPMS-NAV2
Manual ver. 1.4



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

Table of Contents

1. Introduction

LPMS-NAV2 is an inertial sensor for navigation application, which is composed of a high accuracy one-axis gyroscope and a 3-axis accelerometer. With the use of our novel fusion algorithm, LPMS-NAV2 can achieve precise heading information with ultra-low drift errors. The output data includes heading angle, angular speed and acceleration via UART interface. The high performance and affordable price of LPMS-NAV2 make it specially suitable for the applications of mobile robot/vehicle navigation.

Main features:

- based on MEMS inertial sensors
- integrating one-axis gyro and 3-axis accelerometer
- novel sensor fusion algorithm
- precise heading data output
- low noise
- high robustness
- interface: UART

Application:

- robotics
- motion capture
- automated guided vehicle (AGV)
- stability control

2. Revisions

Date	Version	Changes
2017-10-25	1.4	- add of Trouble shooting section
2017-10-23	1.3	- update on the specification
2017-10-05	1.2	- change of document layout - update on the LPBUS contents
2017-09-26	1.1	- update the data output rate info
2017-08-17	1.0	- initial release

3. General Information

3.1 Block Diagram

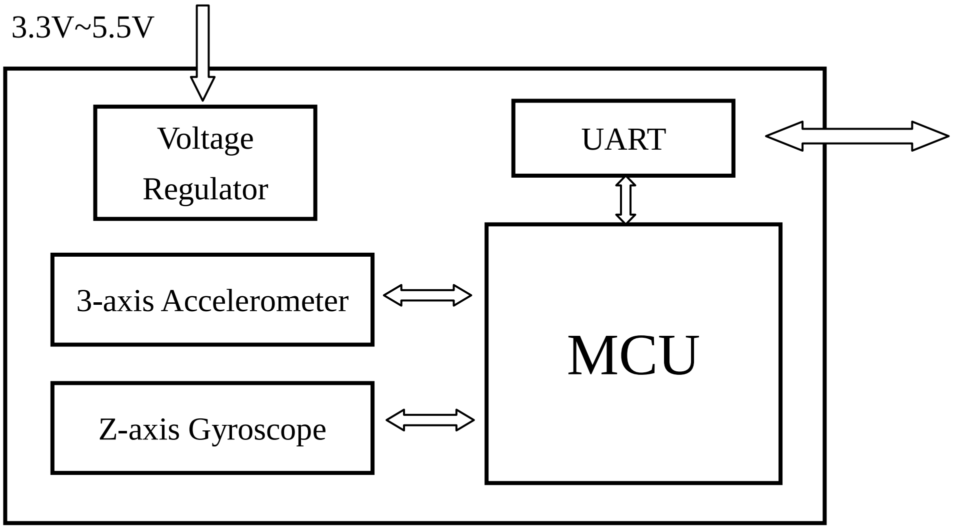


Fig.3.1. Block diagram of LPMS-NAV2

3.2 Pin out

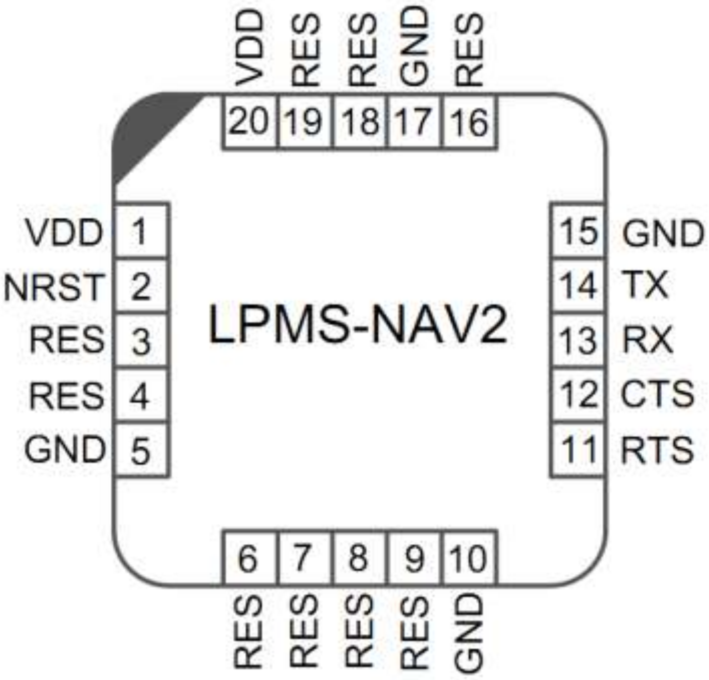


Fig.3.2. Pin out of LPMS-NAV2

Table 3-1 Pin Descriptions

Pin#	Name	Function	Description
1, 20	VDD	Power	Power input (3.3V~5.5V)
2	NRST	Reset pin	Active low. It must be pulled to high or floating for normal operation.
11	RTS	UART_RTS	
12	CTS	UART_CTS	
13	RX	UART_RX(TTL)	
14	TX	UART_TX(TTL)	
5, 10, 15, 17	GND	-	GND
3, 4, 6, 7, 8, 9, 16, 18, 19	RES	-	Reserved

Note:

1. **All reserved pins must be float.**

Table 3-2 Definition of Logic High and Low Level

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

3.3 Typical Application

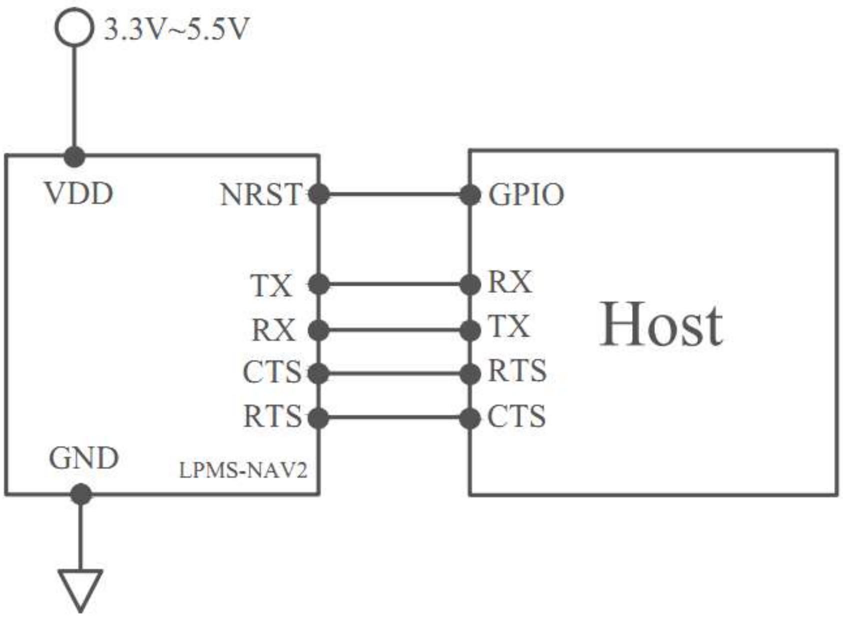


Fig.3.3. LPMS-NAV2 typical application.

3.4 Coordinate System

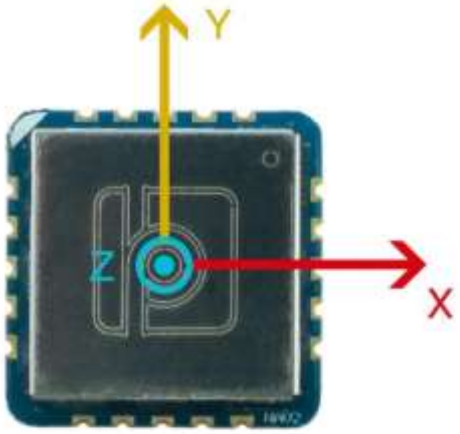


Fig.3.4. Coordinate of LPMS-NAV2

4. Communication Interface

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-NAV2 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 Fig.4.1 and Fig.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.

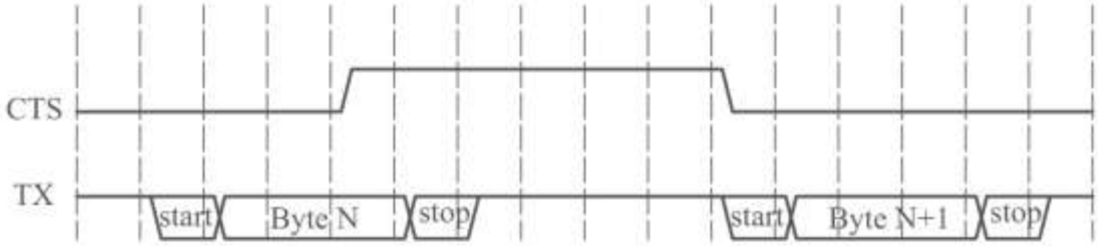


Fig.4.1. Sequence diagram of transmitter with CTS control

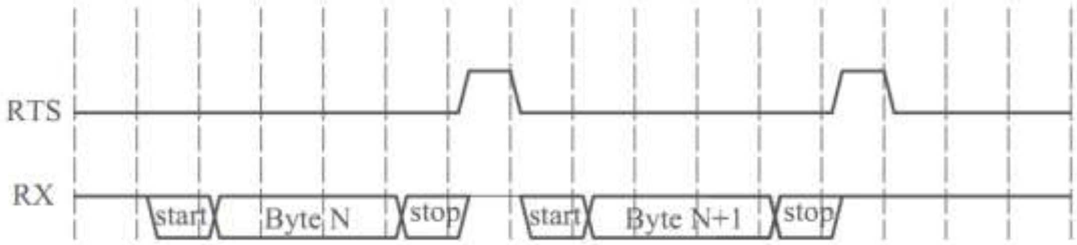


Fig.4.2. Sequence diagram of receiver with RTS control

LPMS-NAV2 supports the following baudrates: 115200, 57600, 38400, 19200, 9600 and 4800. The relation between max data output rate and baudrate is shown in Table 4-1. Please refer to chapter 6 for more detailed information of communication protocol of LPBUS.

Note: The baudrate setting should be correctly set before any change of data output rate based on the information on Table 4-1. Any wrong baudrate setting might lead to unexpected data output.

Table 4-1 Relation between Baudrate and Data Output Rate

Baudrate	115200	57600	38400	19200	9600	4800
Max. Data Output Rate	100Hz	100Hz	100Hz	100Hz	50Hz	25Hz

5. Specifications

Table 5-1 Main Specifications of LPMS-NAV2

Parameter	Value
Name	LPMS-NAV2
Size	20.3x20.3x3.8mm
Weight	2.3g
Angle Resolution	0.01 °(Max.)
Output Angle Range	± 180˚
Data Output Rate	100Hz (10~100Hz selectable)
Angular Speed Range	± 400 dps
Accelerometer Range	± 4 g
UART Baudrate	115200 bps (19200~115200 selectable)
Heading Linear Error	< 0.1 °/m
Angle Random Walk (f=10Hz)	0.18 °/sqrt(h)
Bias Stability (f=10Hz)	< 5 °/h
Power Consumption	~64mW (@3.3V)
Power Supply	3.3~5.5V DC
Work Temperature	-20 ~ 80˚C
Stock Temperature	-40 ~ 85˚C

Table 5-2 Gyroscope Specifications

Parameter	Value	Unit
Measurement Range	±400 (z axis)	dps
Static Bias	±1	dps (0 LSB)
Bias Change/Temperature	±1	dps
Nonlinearity	±0.5	%FS
Sensitivity	±5	%

Table 5-3 Accelerometer Specifications

Parameter	Value	Unit
Measurement Range	±4 (x, y, z axis)	g
Sensitivity	0.122	mg/LSB
Sensitivity Change/Temperature	0.01	%/ ℃

Bias	±30	mg
Bias Change/Temperature	±0.25	mg/ ℃

Table 5-4 Default Setting of LPMS-NAV2

Parameter	Default Value
LED Status	Blinking
Angle Unit	deg
Angle Output Range	±180 deg
Baudrate	115200 bps
Data Output Rate	100 Hz

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 used to set up the parameters and get measurement data of the device. The default communication mode of LPMS-NAV2 is in stream mode which is set at 100Hz data output rate.

Fig. 6.1 shows a flowchart of the commands can be used in each mode. In order to change the sensor setting, the sensor must be set into command mode.

Note: Only four commands can be used under stream mode.

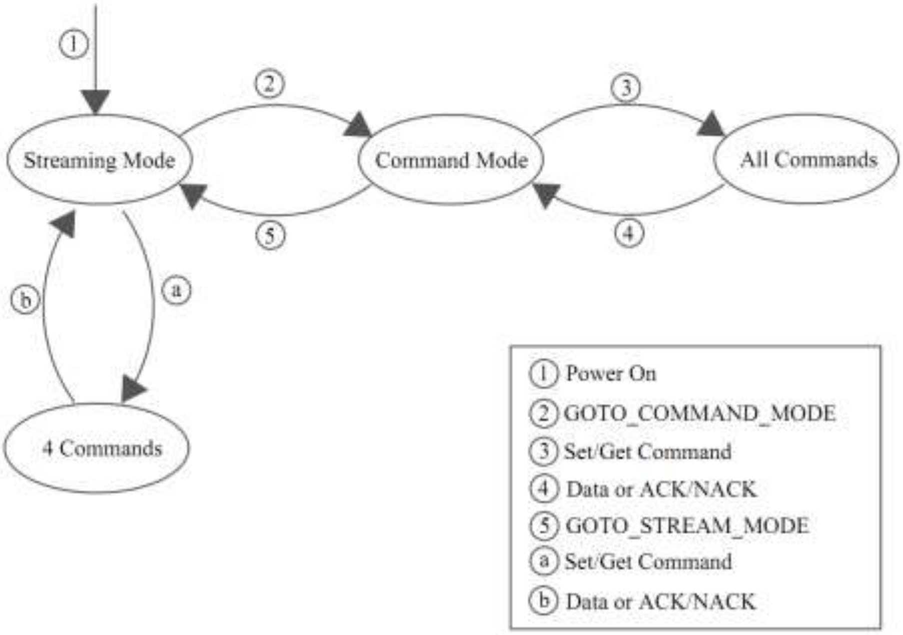


Fig.6.1. Flowchart of sensor parameters setting

6.3 LPBUS Packet Structure

Table 6-1 LPBUS Packet Structure

Byte#	Name	Description
0	Packet start	3Ah
1	Command no.	Command identifiers which can be referred to Appendix.
2	Index	00h~FFh (increased 1 after each packet sent)
3	Packet data length	Length of packet data in bytes.
x	Packet data (n bytes)	If packet data length ***n*** not equal to zero, ***x* = 3+1, 3+2…3+*n***. Otherwise ***x*** = none, the data field is empty.
4+n	LRC	LRC is calculated in the following way: LRC = sum(command no., index, packet data length, packet data)
5+n	Termination byte 1	0Dh
6+n	Termination byte 2	0Ah

Note:

1. **While sending commands from host to sensor, there is no Index in the packet, which means LRC=sum(command no., packet data length, packet data).**
2. **While sending data reply from sensor to host, there is the Index byte in the packet,**

which means LRC= sum(command no., index, packet data length, packet data).

LPMS-NAV2 sensor data are transmitted to the host with pre-scale factor in order to increase precision. Table 6-2 and Table 6-3 show the data format, sensor data type and relative order, pre-scale factor in each data packet.

Table 6-2 Data Format in Each Packet

Chunk#	Data format	Sensor data type	Factor
1	Int16/ Uint16	Heading angle (deg)	100
2	Int16	Angular speed (deg/s)	50
3	Vector3i16	Calibrated acceleration data (g)	1000

Table 6-3 Data Format Identifier Definition

Identifier	Description
Int16	16-bit signed integer value
Uint16	16-bit unsigned integer value
Vector3i16	3 element 16-bit signed integer vector

Note:

1. **When the angle output range is set to -180~180 deg, the heading data format is at Int16. Otherwise it is at Uint16 format in other settings.**
2. **To change the heading output range, please refer to the commands in the Appendix.**
3. **The Packet data is sent in little-endian format, low order byte first, high order byte last.**

6.4 Communication Examples

Go into Command Mode

command sent (host->sensor)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	09h	Command no. (09h = GOTO_COMMAND_MODE)
2	00h	Data length: 0
3	09h	Check sum
4	0Dh	Packet end 1
5	0Ah	Packet end 2

data reply (sensor->host)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	00h	Command no. (00h = REPLY_ACK)
2	01h	Index no.
3	00h	Replied data (replied ACK, no data)
4	01h	Check sum
5	0Dh	Packet end 1
6	0Ah	Packet end 2

Go into Streaming Mode

command sent (host->sensor)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	0Ah	Command no. (0Ah = GOTO_STREAM_MODE)
2	00h	Data length: 0
3	0Ah	Check sum
4	0Dh	Packet end 1
5	0Ah	Packet end 2

data reply (sensor->host)

Packet byte no.	Content	Meaning
0	3Ah	Packet start

1	00h	Command no. (00h = REPLY_ACK)
2	01h	Index no.
3	00h	Replied data (replied ACK, no data)
4	01h	Check sum
5	0Dh	Packet end 1
6	0Ah	Packet end 2

Read Sensor Setting

command sent (host->sensor)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	07h	Command no. (07h = GET_CONFIG)
2	00h	Data length: 0
3	07h	Check sum
4	0Dh	Packet end 1
5	0Ah	Packet end 2

data reply (sensor->host)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	07h	Command no. (07h = GET_CONFIG)
2	01h	Index no.
3	04h	Data length (32 bits integer)
4-7	xxxxxxxxh	Setting data
8	xxh	Check sum
9	0Dh	Packet end 1
10	0Ah	Packet end 2

Note: The replied setting data can be referred to Appendix. xx values depend on the sensor setting.

Read Sensor Data

command sent (host->sensor)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	0Bh	Command no. (11d = GET_SENSOR_DATA)

2	00h	Data length: 0
3	0Bh	Check sum
4	0Dh	Packet end 1
5	0Ah	Packet end 2

data reply (sensor->host)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	0Bh	Command no. (0Bh = GET_SENSOR_DATA)
2	01h	Index no.
3	0Ah	Data length (default: 10 bytes)
4-5	xxxxh	Heading angle (2 bytes)
6-7	xxxxh	Angular speed (2 bytes)
8-9	xxxxh	Acceleration data x axis (2 bytes)
10-11	xxxxh	Acceleration data y axis (2 bytes)
12-13	xxxxh	Acceleration data z axis (2 bytes)
14	xxh	Check sum
15	0Dh	Packet end 1
16	0Ah	Packet end 2

Set Baudrate

command sent (host->sensor)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	15h	Command no. (15h = SET_UART_BAUDRATE)
2	04h	Data length (32 bits integer)
3	80h	To set baudrate to 9600 bps (9600d = 2580h, detailed information referred to command SET_UART_BAUDRATE in Appendix)
4	25h	
5	00h	
6	00h	
7	BEh	Check sum
8	0Dh	Packet end 1
9	0Ah	Packet end 2

data reply (sensor->host)

Packet byte no.	Content	Meaning
0	3Ah	Packet start
1	00h	Command no. (00h = REPLY_ACK)
2	01h	Index no.
3	00h	Data length (replied ACK, no data)
4	01h	Check sum
5	0Dh	Packet end 1
6	0Ah	Packet end 2

Note:

1. **The new baudrate setting will be activated from next power on.**

6.5 Data Parsing Example Code (C Language)

```
struct    _sensorData
{
    float32_t gAngle;
    float32_t gRate;
    float32_t accX;
    float32_t accY;
    float32_t accZ;
} sensorData;

union cArray2intArray
{
    int16_t i[5];
    uint8_t c[10];
}c2i;

bool parse_data(uint8_t *dataBuffer)
{
    uint8_t function;
    uint8_t index;
    uint8_t length;
    int16_t angle;
    int16_t rate;
    int16_t x_acc;
    int16_t y_acc;
    int16_t z_acc;
    uint8_t check_sum;

    // Check header byte
    if (dataBuffer[0] != 0x3A)
    {
        // Error
        return false;
    }

    function = dataBuffer[1];
    index = dataBuffer[2];
    length = dataBuffer[3];
    memcpy(c2i.c, dataBuffer + 4, 10);

    //Verify checksum
    for (int i = 1; i < 14; ++i)
        check_sum +=dataBuffer[i];
    if (check_sum != dataBuffer[14])
    {
        return false;
    }

    //Scale and store data
    sensorData.gAngle = c2i.i[0] / 100.0;// angle / 100.0;
    sensorData.gRate = c2i.i[1] / 50.0;// rate / 50.0;
    sensorData.accX= c2i.i[2] / 1000.0;//x_acc;
    sensorData.accY = c2i.i[3] / 1000.0;//y_acc;
    sensorData.accZ = c2i.i[4] / 1000.0;//z_acc;

    return true;
}
```

7. GUI Software

1) The GUI Software called LpNAV-Control can be downloaded on our website: https://www.lp-research.com/support/
2) Please connect LPMS-NAV2 with a PC via a UART-TO-USB adapter cable.
3) In the device manager window, a COM port of the adapter cable should be recognized. Please note down the COM port number for the communication setup.
4) Start the LpmsNAV-Control software and the GUI will be something like the following image.

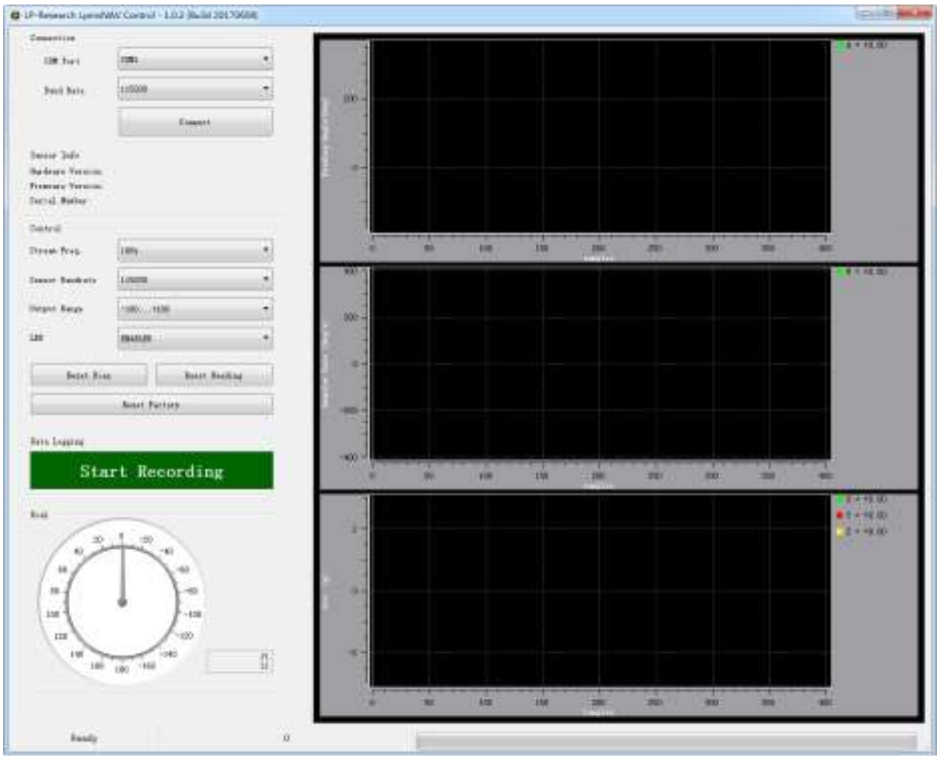


Fig.7.1. LpNAV-Control-V1.0.2 GUI

5) On the COM Port setting, please choose the relative COM port number showed in the device manager, then select sensor baudate and click the "Cnnect" button, as showed in Fig.7.1.

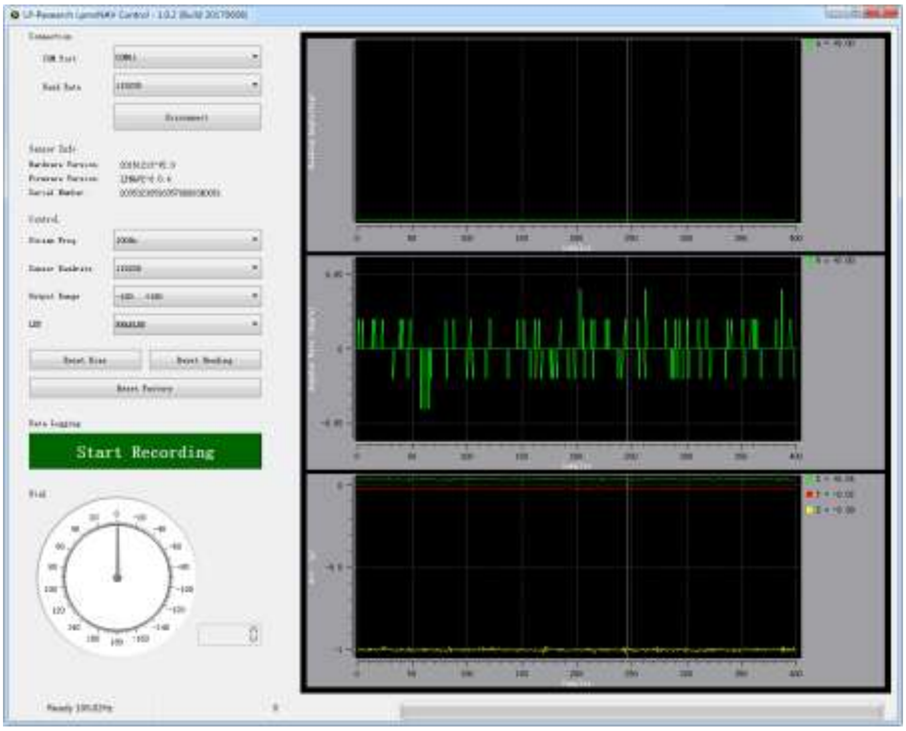
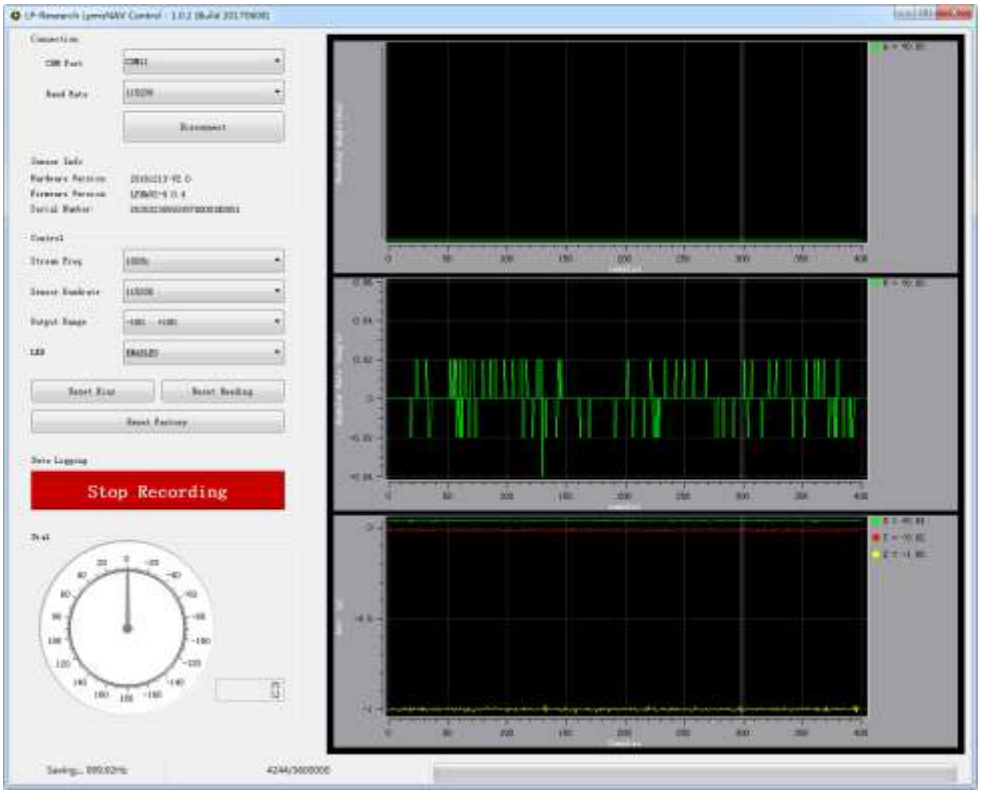
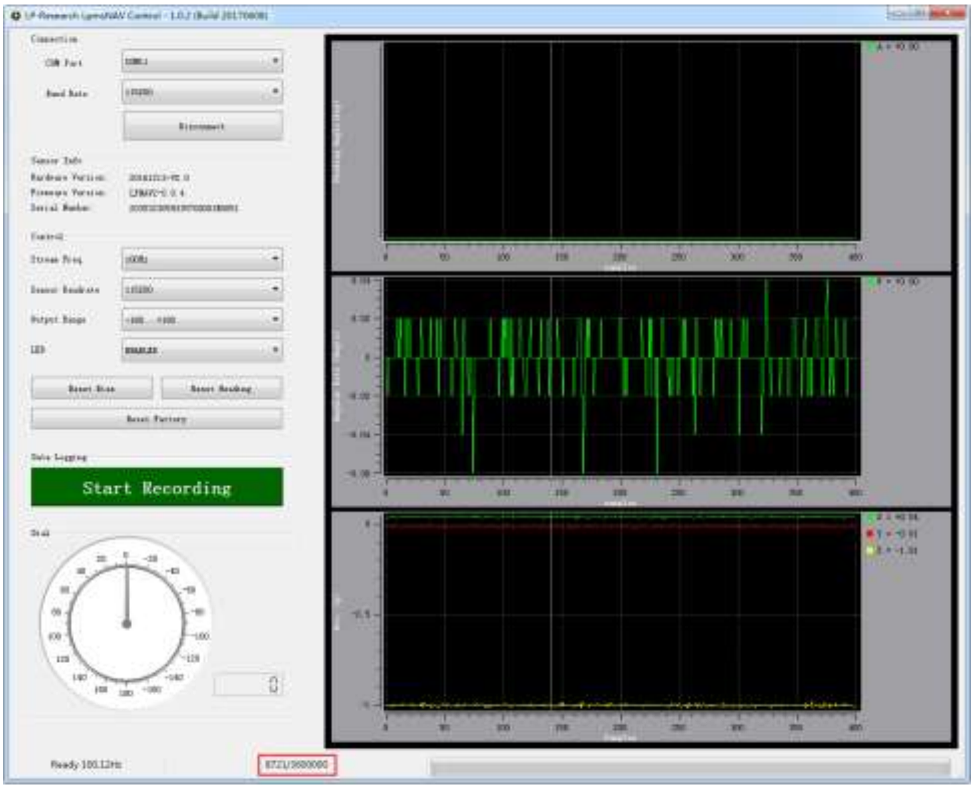
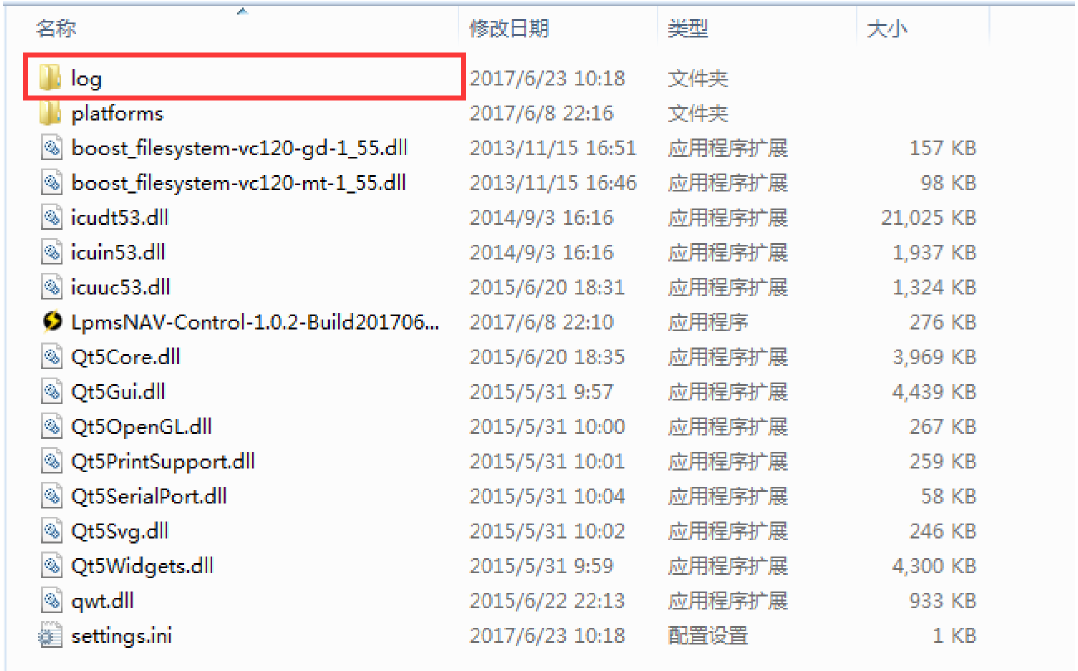


Fig.7.2. LPMS-NAV2 connection setup

6) User can change the sensor setting via the sections of data output rate, baudrate, angle output range, etc.
7) The "Start Recording" is for logging the sensor data. Max data size for one time logging is 3600000 samples. The data will be saved into a folder called "log" under the software root path.







名称	修改日期	类型	大小
log	2017/6/23 10:18	文件夹	
platforms	2017/6/8 22:16	文件夹	
boost_filesystem-vc120-gd-1_55.dll	2013/11/15 16:51	应用程序扩展	157 KB
boost_filesystem-vc120-mt-1_55.dll	2013/11/15 16:46	应用程序扩展	98 KB
icudt53.dll	2014/9/3 16:16	应用程序扩展	21,025 KB
icuin53.dll	2014/9/3 16:16	应用程序扩展	1,937 KB
icuuc53.dll	2015/6/20 18:31	应用程序扩展	1,324 KB
LpmsNAV-Control-1.0.2-Build201706...	2017/6/8 22:10	应用程序	276 KB
Qt5Core.dll	2015/6/20 18:35	应用程序扩展	3,969 KB
Qt5Gui.dll	2015/5/31 9:57	应用程序扩展	4,439 KB
Qt5OpenGL.dll	2015/5/31 10:00	应用程序扩展	267 KB
Qt5PrintSupport.dll	2015/5/31 10:01	应用程序扩展	259 KB
Qt5SerialPort.dll	2015/5/31 10:04	应用程序扩展	58 KB
Qt5Svg.dll	2015/5/31 10:02	应用程序扩展	246 KB
Qt5Widgets.dll	2015/5/31 9:59	应用程序扩展	4,300 KB
qwt.dll	2015/6/22 22:13	应用程序扩展	933 KB
settings.ini	2017/6/23 10:18	配置设置	1 KB

Fig.7.3. Data logging by LpmsNAV-Control

8. Package

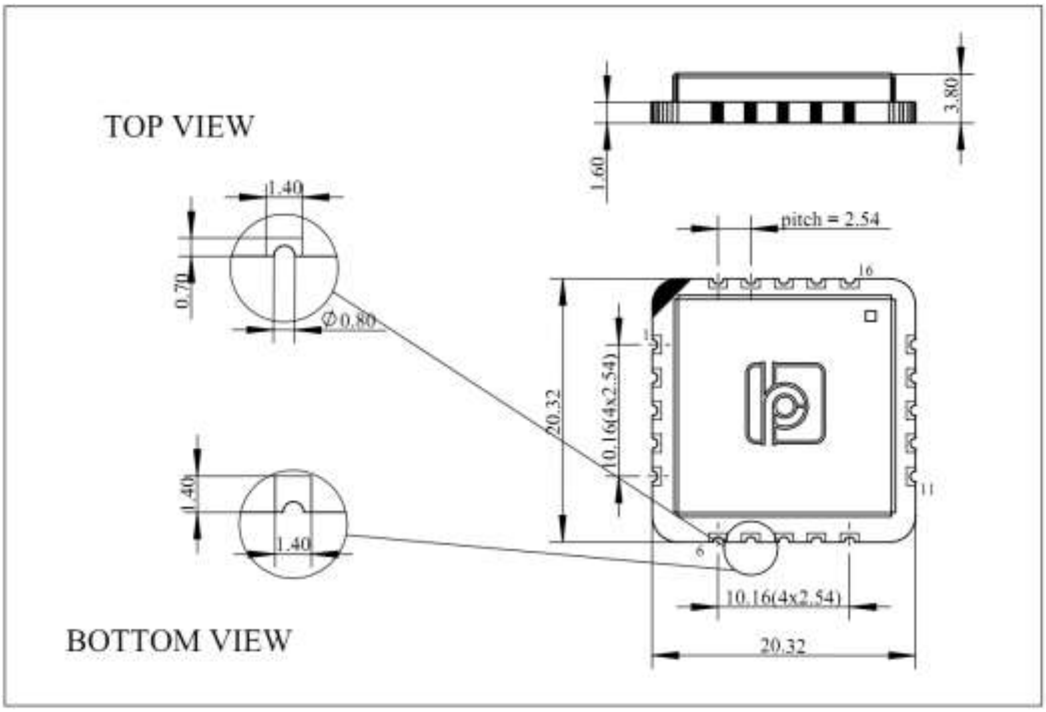


Fig.8.1. Dimension of LPMS-NAV2 (unit: mm)

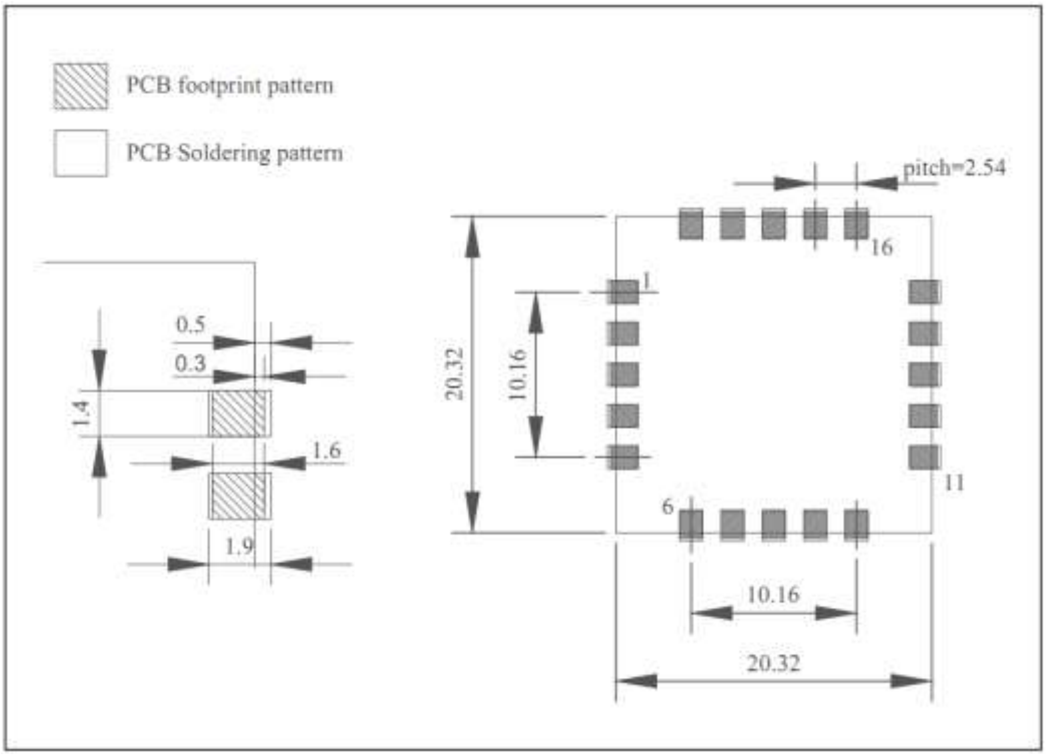


Fig.8.2. Recommended footprint (unit: mm)

9. Trouble Shooting

- Heading angle drifting occurs after installation

  Due to different conditions between factory calibration environment and on-site environment, the sensor heading angle might have drifting problem after initial installation. If this phenomenon occurs, please keep the sensor static and send the "Reset Bias" command to the sensor. After about 1-2s the drifting problem will vanish.

- Heading angle drifting occurs after clicking the "Start Recording" button on host GUI software

  The "Reset Bias" and "Reset Heading" commands will be sent to the sensor after clicking the "Start Recording" function on host GUI software. If the "Start Recording" function is activated during a moving status of sensor, it might lead to the drifting problem. Please keep the sensor static while trying to use the "Reset Bias" function.

10. Appendix

Firmware function / command code list

Applies to LPMS-NAV2 Firmware 0.0.4

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

Sensor Info				
Identifier	Name	Parameter	Response	Default
4 (04h)	GET_FIRMWARE_VERSION	NONE	Char[16]	
5 (05h)	GET_HARDWARE_VERSION	NONE	Char[16]	
6 (06h)	GET_SERIAL_NUMBER	NONE	Char[24]	

Configuration and Status Commands				
Identifier	Name	Parameter	Response	Default
7 (07h)	GET_CONFIG	NONE	Int32	
8 (08h)	GET_STATUS[1]	NONE	Int32	

Mode Switching Commands				
Identifier	Name	Parameter	Response	Default
9 (09h)	GOTO_COMMAND_MODE[1]	NONE	ACK/NACK	
10 (0Ah)	GOTO_STREAM_MODE	NONE	ACK/NACK	

Data Transmission Commands				
Identifier	Name	Parameter	Response	Default
11 (0Bh)	GET_SENSOR_DATA	NONE		
18 (12h)	GET_STREAM_FREQ	NONE	Int32	
19 (13h)	SET_STREAM_FREQ	Int32	ACK/NACK	
20 (14h)	GET_UART_BAUDRATE	NONE	Int32	
21 (15h)	SET_UART_BAUDRATE	Int32	ACK/NACK	

Sensor reset				
Identifier	Name	Parameter	Response	Default
23 (17h)	RESET_BIAS[1]	NONE	ACK/NACK	
24 (18h)	RESET_HEADING[1]	NONE	ACK/NACK	
25 (19h)	RESET_SENSOR	NONE	ACK/NACK	

Set Angle Output Range				
Identifier	Name	Parameter	Response	Default
14 (0Eh)	SET_360_OUTPUT	NONE	ACK/NACK	
15 (0Fh)	SET_180_OUTPUT	NONE	ACK/NACK	

LED Control Commands				
Identifier	Name	Parameter	Response	Default
26 (1Ah)	ENABLE_LED	NONE	ACK/NACK	
27 (1Bh)	DISABLE_LED	NONE	ACK/NACK	

[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.

Sensor Info

Identifier	4 (0x04)
Name	GET_FIRMWARE_VERSION
Description	Get sensor firmware version
Parameter	NONE
Response:	Char[16] Character array of length 16

Identifier	5 (0x05)
Name	GET_HARDWARE_VERSION
Description	Get sensor hardware version
Parameter	NONE
Response:	Char[16] Character array of length 16

Identifier	6 (0x06)
Name	GET_SERIAL_NUMBER
Description	Get sensor serial number
Parameter	NONE
Response:	Char[24] Character array of length 24

Configuration and Status Commands

<table>
<tr><td>Identifier</td><td>7 (0x07)</td></tr>
<tr><td>Name</td><td>GET_CONFIG</td></tr>
<tr><td>Description</td><td>Get the current value of the configuration register of the sensor. The configuration word is read-only.</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int32</td></tr>
<tr><td>Data format</td><td>
<table>
<tr><th>Bit</th><th>Reported State / Parameter</th></tr>
<tr><td><b>0</b></td><td>Start-up data transfer mode, <b>default: 0</b>
(0: streaming mode, 1: Command mode)</td></tr>
<tr><td><b>1</b></td><td>+-180/0~360 degree output angle range, <b>default: 0</b>
(0: +-180, 1: 0~360)</td></tr>
<tr><td><b>2</b></td><td>Output unit: degree, reserved</td></tr>
<tr><td><b>3:4</b></td><td>Data output rate, <b>default: 11</b>
(00: 10Hz, 01: 25Hz, 10: 50Hz, 11: 100Hz)</td></tr>
<tr><td><b>5:7</b></td><td>UART Baudrate, <b>default: 110</b>
(000: 4800bps, 001: 9600bps, 010: 19200bps 011: 28800bps, 100: 38400bps, 101: 57600bps, 110: 115200bps)</td></tr>
<tr><td><b>8:9</b></td><td>reserved</td></tr>
<tr><td><b>10</b></td><td>LED blinking enable, <b>default: 1</b>
(0: disable, 1: enable)</td></tr>
<tr><td><b>11:31</b></td><td>reserved</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>8 (0x08)</td></tr>
<tr><td>Name</td><td>GET_STATUS</td></tr>
<tr><td>Description</td><td>Get the current value of the status register of the sensor. The status word is read-only</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int32</td></tr>
<tr><td>Data format</td><td>
<table>
<tr><th>Bit</th><th>Indicated state</th></tr>
<tr><td>0</td><td>COMMAND mode enabled</td></tr>
<tr><td>1</td><td>STREAM mode enabled</td></tr>
<tr><td>2:31</td><td>Reserved</td></tr>
</table>
</td></tr>
</table>

Mode Switching Commands

Identifier	9 (0x09)
Name	GOTO_COMMAND_MODE
Description	Switch to command mode. In command mode the user can issue commands to the sensor to set parameters
Parameter	NONE
Response:	ACK (success) or NACK (error)

Identifier	10 (0x0A)
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)

Data Transmission Commands

<table>
<tr><td>Identifier</td><td>11 (0x0B)</td></tr>
<tr><td>Name</td><td>GET_SENSOR_DATA</td></tr>
<tr><td>Description</td><td>Retrieves the latest set of sensor data</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>
<table>
<tr><th>Output Data</th><th>Byte</th><th>Reported State / Parameter</th></tr>
<tr><td>Header</td><td>1</td><td>Hex value: 0x3A</td></tr>
<tr><td>Function</td><td>2</td><td>Current function: 0x0B</td></tr>
<tr><td>Index</td><td>3</td><td>Incremental: 0x00 ~ 0xFF</td></tr>
<tr><td>Length</td><td>4</td><td>Length of packet: 0x0A</td></tr>
<tr><td>Z-Axis Angle</td><td>5-6</td><td>Provided in hundredths of deg., normalized to ±180deg.</td></tr>
<tr><td>Rate</td><td>7-8</td><td>Provided in hundredths of deg/sec, scaled by 0.5</td></tr>
<tr><td>X-Axis Acceleration</td><td>9-10</td><td>Provided in 1mg resolution</td></tr>
<tr><td>Y-Axis Acceleration</td><td>11-12</td><td>Provided in 1mg resolution</td></tr>
<tr><td>Z-Axis Acceleration</td><td>13-14</td><td>Provided in 1mg resolution</td></tr>
<tr><td>Checksum</td><td>15</td><td>function + index + z-axis angle(LSB) + z-axis angle(MSB) + rate (LSB) + rate (MSB) + Xacc (LSB) + Xacc (MSB) + Yacc (LSB) + Yacc (MSB) + Zacc (LSB) + Zacc (MSB)</td></tr>
<tr><td>End byte 0</td><td>16</td><td>0x0D</td></tr>
<tr><td>End byte 1</td><td>17</td><td>0x0A</td></tr>
</table>
</td></tr>
</table>

Identifier	18 (0x12)
Name	GET_STREAM_FREQ
Description	Get the current streaming frequency
Parameter	NONE
Response:	Int32 Possible values: 10, 25, 50, 100Hz

<table>
<tr><td>Identifier</td><td>19 (0x13)</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>Value</th></tr>
<tr><td>10</td><td>10</td></tr>
<tr><td>25</td><td>25</td></tr>
<tr><td>50</td><td>50</td></tr>
<tr><td>100</td><td>100</td></tr>
</table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>20 (0x14)</td></tr>
<tr><td>Name</td><td>GET_UART_BAUDRATE</td></tr>
<tr><td>Description</td><td>Get the current UART baudrate</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int32
<table>
<tr><th>Baud rate</th><th>Identifier</th></tr>
<tr><td>4800</td><td>4800</td></tr>
<tr><td>9600</td><td>9600</td></tr>
<tr><td>19200</td><td>19200</td></tr>
<tr><td>38400</td><td>38400</td></tr>
<tr><td>57600</td><td>57600</td></tr>
<tr><td>115200</td><td>115200</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>Identifier</td><td>21 (0x15)</td></tr>
<tr><td>Name</td><td>SET_UART_BAUDRATE</td></tr>
<tr><td>Description</td><td>Set the current UART baudrate</td></tr>
<tr><td>Parameter</td><td>Int32
<table>
<tr><th>Baud rate</th><th>Identifier</th></tr>
<tr><td>4800</td><td>4800</td></tr>
<tr><td>9600</td><td>9600</td></tr>
<tr><td>19200</td><td>19200</td></tr>
<tr><td>38400</td><td>38400</td></tr>
<tr><td>57600</td><td>57600</td></tr>
<tr><td>115200</td><td>115200</td></tr>
</table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

Sensor reset

Identifier	23 (0x17)
Name	RESET_BIAS
Description	Reset gyro static bias
Parameter	NONE
Response:	ACK (success) or NACK (error)

Identifier	24 (0x18)
Name	RESET_HEADING
Description	Set current heading angle to zero
Parameter	NONE
Response:	ACK (success) or NACK (error)

Identifier	25 (0x19)
Name	RESET_SENSOR
Description	Reset sensor to factory defaults
Parameter	NONE
Response:	ACK (success) or NACK (error)

Set Angle Output Range

Identifier	14 (0x0E)
Name	SET_360_OUTPUT
Description	The output range of angle set to be 0~360 deg.
Parameter	NONE
Response:	ACK (success) or NACK (error)

Identifier	15 (0x0F)
Name	SET_180_OUTPUT
Description	The output range of angle set to be -180~+180 deg.
Parameter	NONE
Response:	ACK (success) or NACK (error)

LED Control Commands

Identifier	26 (0x1A)
Name	ENABLE_LED
Description	Enable the LED function.
Parameter	NONE
Response:	ACK (success) or NACK (error)

Identifier	27 (0x1B)
Name	DISABLE_LED
Description	Disable the LED function.
Parameter	NONE
Response:	ACK (success) or NACK (error)