LPMS-IG1-series
Reference Manual ver. 1.1

Compatible with:
Firmware info: IG1-3.0.3-20190508

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

Version history

Date	Version	Details
2019-05-23	Ver. 1.0	First draft
2019-08-28	Ver. 1.1	English version first draft

Table of Contents

1. Introduction

LPMS-IG1 sensor is a high precision multi-purpose inertial measurement unit specially developed for industrial applications. IG1 sensors embeds a powerful processor, 3 high precision single axis gyroscopes, a general purpose 3 axis gyroscope, accelerometer and magnetometer. The unique dual-gyroscope (Gyro I & II) setup enables accurate dynamics measurement in both low and high-speed applications. Gyro I is suitable for applications where the accuracy requirements are high and the detection range is not critical (<400dps). Gyro II is suitable for general applications where expected measurement range exceeds 400dps.
LPMS-IG1 series include two models, LPMS-IG1 (without GPS) and LPMS-IG1P (with GPS). Both models offer the following communication methods: USB+RS232 or USB+CAN. For details on how to use the communication interface, please refer to the relevant sections below. LPMS-IG1 series comes in a waterproof metal housing with a flat bottom profile and mounting holes for easy installation.

Main features:

- High precision and stability
- Dual 3-axis general purpose gyroscope, 3-axis accelerometer, 3-axis magnetometer
- Gyro I: High precision 3-axis gyroscope
- Gyro II: 3-axis general purpose gyroscope
- Realtime output: Raw and calibrated sensor data, quaternion, Euler angles, temperature, GPS (LPMS-IG1P)
- Communication method：USB+RS232 or USB+CAN

Applications:

- Robot navigation
- Automotive navigation
- Remote control and monitoring for industrial robots
- Automated guided vehicle navigation

Table 1 LPMS-IG1-series model

Model	Communication			GPS Connector
	USB	RS232	CAN	
LPMS-IG1-RS232	✓	✓	×	×
LPMS-IG1-CAN	✓	×	✓	×
LPMS-IG1P-RS232	✓	✓	×	✓
LPMS-IG1P-CAN	✓	×	✓	✓

2. System Overview

2.1 Sensor Architecture

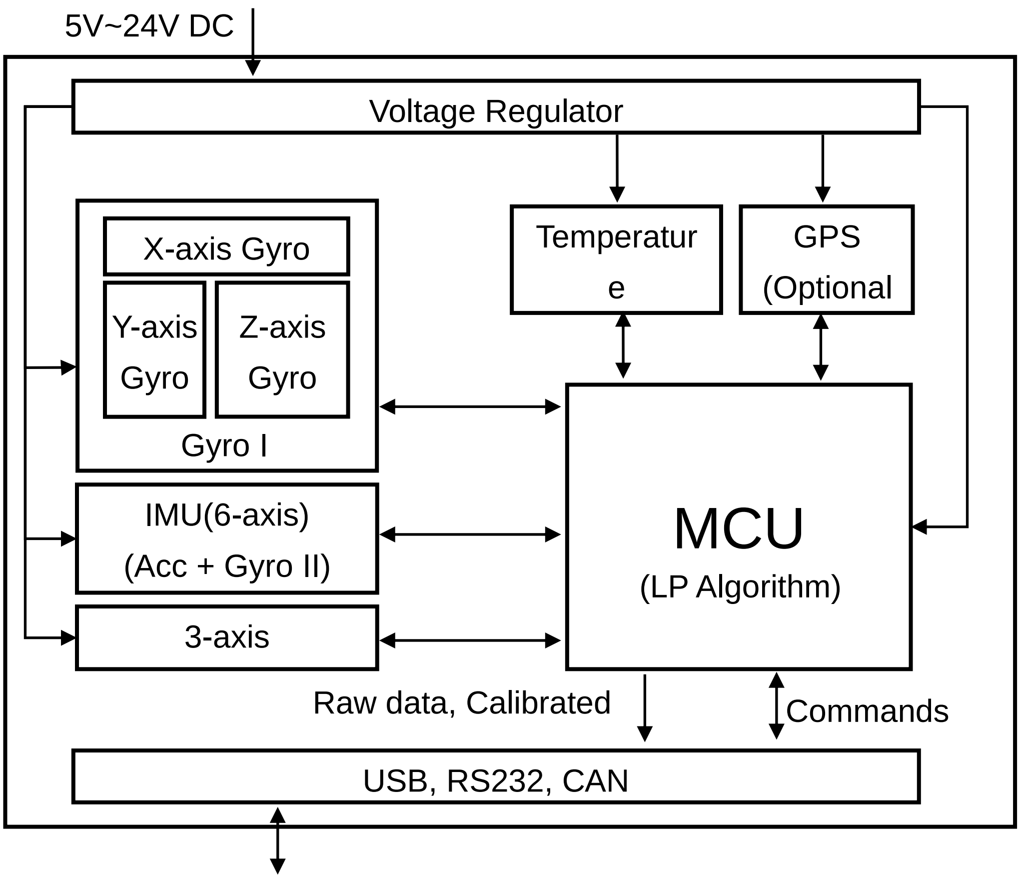

Figure 1 LPMS-IG1-series sensor architecture

2.2 M12 Connector Pinout

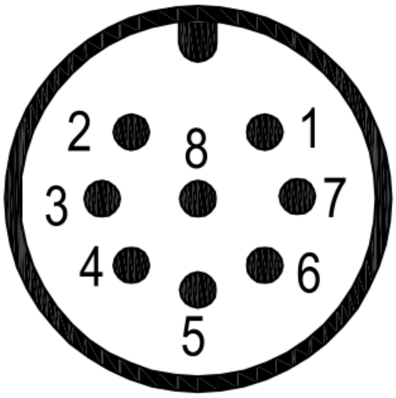

Figure 2 LPMS-IG1-series pin out

Table 2 M12 Connector pin out

Pin	Type	Name	Notes	Color
1	Power	VIN	+5V~+24V DC	White
2		GND	Ground	Brown
3	RS232/CAN	TX/CAN+	-	Green
4		RX/CAN-	-	Yellow
6	USB	D-	-	Pink
7		D+	-	Blue
8	-	EN	Sensor enable/disable Not connected: enable sensor Pull ground: disable sensor	Red
5	-	RESERVED	Not connected	Gray

*Warning: For official cables with USB connector, do not connect USB and VIN power lines at the same time on the provided cable

3. Operation

3.1 Coordinate system

The LPMS sensor calculates the orientation difference between a fixed sensor coordinate system and a global reference coordinate system. The local and the global reference coordinate systems used are defined as right-handed Cartesian coordinate systems with:

- X positive when pointing to the magnetic north
- Y positive when pointing to the magnetic west
- Z positive when pointing up (gravity points vertically down with -1g)

A positive rotation is always right-handed, i.e. defined according to the right-hand rule (corkscrew rule). This means a positive rotation is defined as clockwise in the direction of the axis of rotation.

The definition used for Euler angles in this document is equivalent to roll, pitch, yaw/heading. The Euler angles are of ZYX global type (subsequent rotation around global Z, Y and X axis, also known as aerospace sequence).

Φ : Rotation around global X, defined from -180 °...180 °

θ : Rotation around Y, defined from -90 °...90 °

ω : Rotation around Z, defined from -180 °...180 °

NOTE: Due to the definition of Euler angles there is a mathematical singularity when the sensor-fixed X-axis is pointing up or down in the global reference frame (i.e. pitch approaches+/-90). This singularity is not present in quaternion output.

3.2 Orientation alignment modes

Heading reset

Often it is important that the global Z-axis remains along the vertical (defined by local gravity vector), but the global X-axis has to be pointed in a particular direction. In this case a heading reset may be used. When performing a heading reset, the new global reference frame is chosen such that the global X-axis points in the direction of the sensor while keeping the global Z-axis vertical (along gravity, pointing upwards). In other words: The global Z-axis point upwards along gravity, where the X and Y axis orthogonally form a perpendicular plane.

NOTE: After a heading reset, the yaw may not be exactly zero, this occurs especially when the X-axis is close to the vertical. This is caused by the definition of the yaw when using Euler angles, which becomes unstable when the pitch approaches +/-90 deg.

Alignment reset

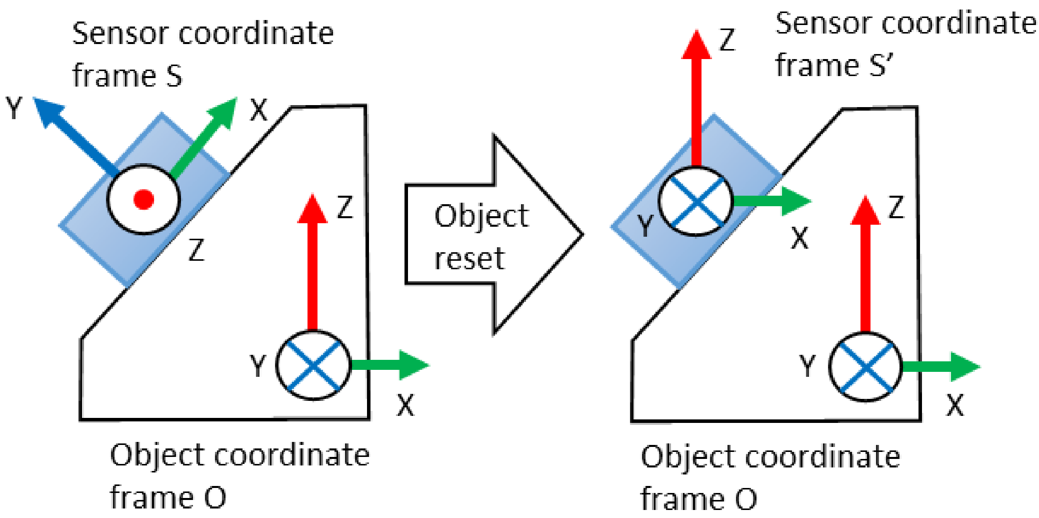

Figure 3 The alignment reset aligns the sensor coordinate system with the object coordinate system.

The alignment reset function aims to facilitate in aligning the LPMS coordinate frame (S) with the coordinate frame of the object to which the sensor is attached (O). After an alignment reset, the S coordinate frame is changed to S' as follows:

The S' Z-axis is the vertical (up) at time of reset
The S' X-axis equals the S X-axis but projected on the new horizontal plane.
The S' Y-axis is chosen as to obtain a right-handed coordinate frame.

NOTE: Once this alignment reset is done, both calibrated data and orientation will be output in the new coordinate frame (S').

The alignment reset aligns the LPMS coordinate frame to that of the object to which it is attached (see Figure 5). The sensor must be attached in such a way that the X-axis is in the XZ-plane of the object coordinate frame, i.e. the LPMS can be used to identify the X-axis of the object. To preserve the global vertical, the object must be oriented such that the object Z-axis is vertical. The alignment reset causes the new S' coordinate frame and the object coordinate frame to be aligned.

NOTE: Since the sensor X-axis is used to describe the direction of the object X-axis, the reset will not work if the sensor X-axis is aligned along the Z-axis of the object.

Object reset

The object reset simply combines alignment reset and the heading reset at a single instant in time. This has the advantage that all coordinate systems can be aligned with a single action. Keep in mind that the new global reference X-axis (heading) is defined by the object X-axis (to which XZ-plane you have aligned the LPMS).

NOTE: Once this object reset is conducted, both calibrated data and orientation will be output with respect to the new S' coordinate frame.

3.5 GPS data

LPMS-IG1P series includes a GPS receiver to output GPS data. GPS data output consists of the following data:

Table 3 LPMS-IG1P GPS data

Name	Description
Timestamp	1Hz update rate
NAV-PVT	Navigation Position Velocity Time Solution
NAV-ATT	Navigation Attitude Solution
ESF-STATUS	External Sensor Fusion Messages

4. Sensor characteristics

4.1 Gyroscope

Table 4 LPMS-IG1-series Gyro I specifications

Parameter	Conditions	Standard			Unit
		Min.	Typ.	Max.	
Sensitivity scale factor	24bit		17920		LSB/(°/s)
Scale factor tolerance	+25°C	-2		+2	%
Sensitivity variation vs temperature	3V, +25°C	-3		+3	%
Bias	+25°C		0		LSB
Bias tolerance	+25°C	-1		+1	°/s
Bias variation over temperature	3V, +25°C	-1		+1	°/s
Rate range		-400		+400	°/s
Nonlinearity	+25°C	-0.5		+0.5	%FS
Cross axis sensitivity	+25°C	-5		+5	%

Table 5 LPMS-IG1-series Gyro II specifications

Parameter	Conditions	Standard			Unit
		Min.	Typ.	Max.	
Sensitivity scale factor	16bit, ±1000dps		32.8		LSB/(°/s)
	16bit, ±2000dps		16.4		
Scale factor tolerance	+25°C	-1		+1	%
Sensitivity variation vs temperature	-40 to 85°C	-2		+2	%
Bias tolerance	+25°C	-1		+1	°/s
Bias variation over temperature	-40 to 85°C	-0.01		+0.01	°/s / °C
Rate range		-1000		+1000	°/s
		-2000		+2000	
Nonlinearity	+25°C	-0.1		+0.1	%
Cross axis sensitivity	+25°C	-1		+1	%
Rate noise spectral	@10Hz		0.004		°/s / $\sqrt{\text{Hz}}$
Total RMS noise	Bandwidth = 100Hz		0.04		kHz

4.2 Accelerometer

Table 6 LPMS-IG1-series Accelerometer specifications

Parameter	Conditions	Standard			Unit
		Min.	Typ.	Max.	
Sensitivity scale factor	16bit, ±2g		16384		LSB/g
	16bit, ±4g		8192		LSB/g
	16bit, ±8g		4096		LSB/g
	16bit, ±16g		2048		LSB/g
Scale factor	Component level	-1		+1	%
Sensitivity variation vs temperature	-40 to 85°C	-1.5		+1.5	%
Zero-G bias tolerance	Component level, all	-25		+25	mg
	Board level, all axes	-40		+40	mg
Zero-G level change vs temperature	X, Y-axis(-40 to 85°C)	-0.5		+0.5	mg / °C
	Z-axis(-40 to 85°C)	-1		+1	mg / °C
Full scale range		-2		+2	g
		-4		+4	g
		-8		+8	g
		-16		+16	g
Nonlinearity	+25°C	-0.3		+0.3	%
Cross axis sensitivity	+25°C	-1		+1	%
Power spectral	@10Hz		100		ug / √Hz
RMS Noise	Bandwidth = 100Hz		1.0		mg-rms

4.3 Magnetometer

Table 7 LPMS-IG1-series Magnetometer specifications

Parameter	Conditions	Standard			Unit
		Min.	Typ.	Max.	
Sensitivity scale factor	16bit, ±2G		12000		LSB/G
	16bit, ±8G		3000		LSB/G
Sensitivity variation vs temperature	-40 to 85°C		100		LSB / °C
Bias tolerance		-10		+10	mG
Range		-2		+2	Gauss
		-8		+8	Gauss
Linearity	±2G		0.1		%FS
Hysteresis	±2G, ±8G		0.3		%FS
Cross axis sensitivity	Cross field = 1G，Happlied = ±2G		0.1		% / G
X-Y-Z Orthogonality			90±1		degree

4.5 GPS

Only applies to LPMS-IG1P (with gps feature)

Table 8 LPMS-IG1P-series GPS specifications

Parameter	Specification
Receiver type	72-channel, GPS L1C/A, SBAS L1C/A, QZSS L1C/A, QZSS L1-SAIF, GLONASS L1OF, BeiDou B1I, Galileo E1B/C
Data update rate	1Hz
Horizontal position accuracy	2.5 m (Autonomous)
Velocity accuracy	0.5 m/s (50% @ 30 m/s)
Heading accuracy	1^{o} (50% @ 30 m/s)
Time-to-first-fix (Cold start)	30 s (GPS)
Sensitivity	-160 dBm (GPS)
Max. altitude	50 km
Max. velocity	500 m/s
Max. dynamics	4g

4.6 LPMS-IG1-series

Table 9 LPMS-IG1-series specifications

Parameter		Conditions	Standard			Unit
			Min.	Typ.	Max.	
Power supply			5	12	24	V
Dimensions				51 x 45 x 24		mm
Weight		LPMS-IG1		74		g
		LPMS-IG1P		76		
Euler angle range		roll	-180		+180	degrees
		pitch	-90		+90	degrees
		yaw	-180		+180	degrees
Angle resolution				0.01		degrees
Power consumption		LPMS-IG1		0.24 (0.02A@12V)		W
		LPMS-IG1P		< 0.4 (0.033A@12V)		W
Data output rate			5	100	500	Hz
Operating temperature			-20	20	80	°C
Gyroscope characteristics	Bias stability	< 400 dps		4		deg / hr
		> 400 dps		6		
	Angle random walk	< 400 dps		0.12		deg / $\sqrt{hr}$
		> 400 dps		0.24		
	Rate noise spectral	< 400 dps		0.002		dps / $\sqrt{Hz}$
		> 400 dps		0.004		
	RMS noise	< 400 dps		0.01		dps
		> 400 dps		0.03		
	Peak to peak noise	< 400 dps		0.05		dps
		> 400 dps		0.15		
	Bandwidth	< 400 dps		10		Hz
		> 400 dps	5	10	92	
Accelerometer characteristics	Bias tolerance			25		ug
	Velocity random walk			0.045		m/s / $\sqrt{hr}$
	RMS Noise	Bandwidth=10Hz		0.4		mg
	Bandwidth		5	10	100	Hz

5. Communication

LPMS-IG1 series offer two types of communication methods:

- USB+RS232
- USB+CAN

Sensor data is streamed to both USB and RS232/CAN terminal simultaneously. Communication protocol for different terminals are summarized as below:

Terminal	Protocol
USB	LPBus Protocol
RS232	LPBus Protocol/ASCII
CAN	CANOpen / CAN sequential

5.1 Communication mode

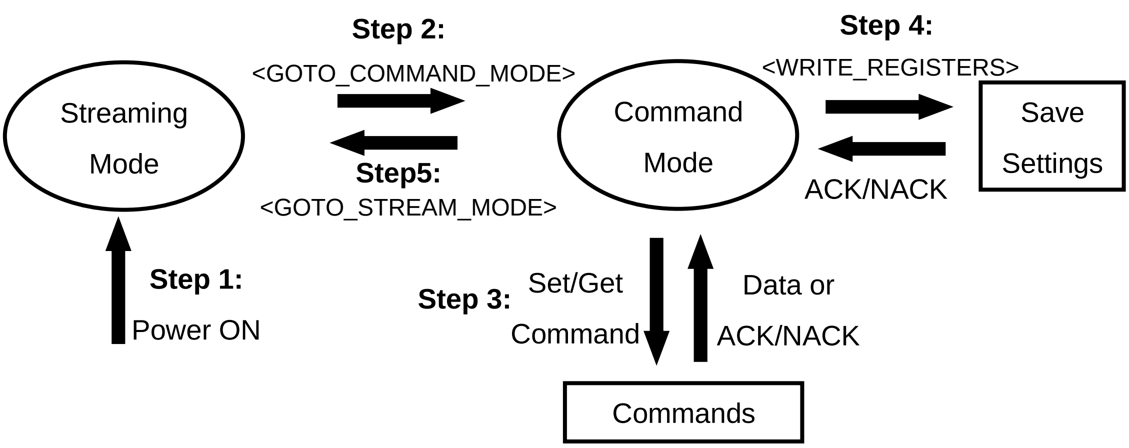

Figure 4 State diagram of sensor communication mode

There are two operational modes, Streaming Mode and Command mode in LPMS-IG1 sensors. By default, the sensor will start in Streaming Mode on power up. In Streaming Mode, the sensor will continuously stream out sensor data via USB and RS232/CAN terminals simultaneously. Streaming frequency for both USB and RS232/CAN terminal is determined by the data streaming rate settings (default at 100Hz). The sensor will stop data streaming in Command Mode. User can change the sensor internal parameters in both Streaming and Command mode, but it is highly recommended to put the sensor in Command Mode before making any parameter changes. The state diagram of the sensor modes is summarized in Figure 4.

NOTE: User must issue a Save Parameters command to retain any parameter changes to the sensor on power cycle.

5.2 LPBUS Protocol

LP-BUS is a communication protocol based on the industry standard MODBUS protocol. It is the default communication format used by LPMS devices. An LP-BUS 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.

5.3 LPBUS packet format

Each packet sent during the communication is based on the following structure:

Table 10 LPBus packet format

Byte#	Name	Description
0	Packet start	Data packet start (3Ah)
1	Sensor ID byte 1	Contains the low byte of the Sensor ID of the sensor to be communicated with. The default value of this ID is 1. The host

		sends out a GET / SET request to a specific LPMS sensor by using this 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 firmware.
2	Sensor ID byte 2	High byte of the Sensor ID of the sensor.
3	Command # byte 1	Contains the low byte of the command to be performed by the data transmission.
4	Command # byte 2	High byte of the command number.
5	Packet data length byte 1	Contains the low byte of the packet data length to be transmitted in the packet data field.
6	Packet data length byte 2	High byte of the data length to be transmitted.
x	Packet data(***n*** bytes)	If data length ***n*** not equal to zero, ***x* = 6+1, 6+2…6+*n***. Otherwise ***x*** = none. This data field contains the packet data to be transferred with the transmission if the data length does not equal to zero, otherwise the data field is empty.
7+*n*	LRC byte 1	The low byte of LRC checksum. To ensure the integrity of the transmitted data the LRC checksum is used. It is calculated in the following way: LRC = sum(Sensor ID, Command, Package data length, and packet data byte no. 1 to no. ***x***) The calculated LRC is usually compared with the LRC transmitted from the remote device. If the two LRCs are not equal, and error is reported.
8+n	LRC byte 2	High byte of LRC check-sum.
9+n	Termination byte 1	0Dh
10+n	Termination byte 2	0Ah

Data Format in a Packet Data Field

Generally, data is sent in little-endian format, low order byte first, high order byte last. Data in the data fields of a packet can be encoded in several ways, depending on the type of information to be transmitted. In the following we list the most common data types. Other command-specific data types are explained in the command reference.

Table 11 Data identifier definition

Identifier	Description
Int32	32-bit signed integer value
UInt32	32-bit unsigned integer value
Int16	16-bit signed integer value
UInt16	16-bit unsigned integer value
Int8	8-bit signed integer value
UInt8	8-bit unsigned 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

Sensor measurement data

IMU data

In streaming mode, LP-BUS transports measurement data in the following form, wrapped into the standard LP-BUS protocol. See the following chapter for examples of transmission packets. The order of the sensor data chunks depends on which sensor data is enabled. The following is the data types in 32-bit float transmission mode.

Table 12 32bit sensor data

Order	Identifier	Description	Unit
1	UInt32	Timestamp)	multiply by factor 0.002 to convert to seconds
2	Vector3f	Raw accelerometer	g
3	Vector3f	Calibrated accelerometer	g
4	Vector3f	Raw Gyro I	dps or rad/s
5	Vector3f	Raw Gyro II	dps or rad/s
6	Vector3f	Static bias calibrated Gyro I	dps or rad/s
7	Vector3f	Static bias calibrated Gyro II	dps or rad/s
8	Vector3f	Alignment calibrated Gyro I	dps or rad/s
9	Vector3f	Alignment calibrated Gyro II	dps or rad/s
10	Vector3f	Raw magnetometer	uT
11	Vector3f	Calibrated magnetometer	uT
12	Vector3f	Reserved	
13	Vector4f	Quaternion	
14	Vector3f	Euler	degree
15	Vector3f	Linear acceleration (g)	g
16	Float32	Reserved	
17	Float32	Reserved	
18	Float32	Temperature	°C

In **16-bit transmission mode** values are transmitted to the host with a multiplication factor applied to increase precision:

Table 13 16bit integer sensor data

Order	Format	Sensor data	Scale factor
1	UInt32	Timestamp (multiply by factor 0.002 to convert to seconds)	-
2	Vector3i16	Raw accelerometer (g)	1000
3	Vector3i16	Calibrated accelerometer (g)	1000
4	Vector3i16	Raw Gyro I (dps or rad/s)	dps: 10 rad/s: 100
5	Vector3i16	Raw Gyro II (dps or rad/s)	10/100
6	Vector3i16	Static bias calibrated Gyro I (dps or rad/s)	10/100

7	Vector3i16	Static bias calibrated Gyro II (dps or rad/s)	10/100
8	Vector3i16	Alignment calibrated Gyro I (dps or rad/s)	10/100
9	Vector3i16	Alignment calibrated Gyro II (dps or rad/s)	dps: 10 rad/s: 100
10	Vector3i16	Raw magnetometer (uT)	100
11	Vector3i16	Calibrated magnetometer (uT)	100
12	Vector3i16	Reserved	
13	Vector4i16	Quaternion	10000
14	Vector3i16	Euler (degree or rad)	dps: 100 rad/s:10000
15	Vector3i16	Linear acceleration (g)	1000
16	Int16	Reserved	
17	Int16	Reserved	
18	Int16	Temperature (°C)	100

GPS Data

In addition to IMU data, LPMS-IG1P (with GPS) will output additional 1Hz GPS data packet with the following format:

Table 14 GPS Data

Order	Format	Sensor data	Scale factor
1	UInt32	Timestamp	
2	UInt32	PVT iTOW (ms) - GPS time of week of the navigation	
3	UInt16	PVT year (UTC)	
4	UInt8	PVT month (UTC)	
5	UInt8	PVT day (UTC)	
6	UInt8	PVT hour (UTC)	
7	UInt8	PVT min (UTC)	
8	UInt8	PVT sec (UTC)	
9	UInt8	PVT valid - Validity flags	
10	UInt32	PVT tAcc - Time accuracy estimate (UTC)	
11	Int32	PVT nano (ns) - Fraction of second (UTC)	
12	UInt8	PVT fixType	
13	UInt8	PVT flags - Fix status flags	
14	UInt8	PVT flags2 - Additional flags	
15	UInt8	PVT numSV - Number of satellites used in Nav Solution	
16	Int32	PVT longitude (deg)	10000000
17	Int32	PVT latitude (deg)	10000000
18	Int32	PVT height (mm) - Height above ellipsoid	
19	Int32	PVT hMSL (mm) - Height above mean sea level	

20	UInt32	PVT hAcc (mm)- Horizontal accuracy estimate	
21	UInt32	PVT vAcc (mm) - Vertical accuracy estimate	
22	Int32	PVT velN (mm/s) - NED north velocity	
23	Int32	PVT velE (mm/s) - NED east velocity	
24	Int32	PVT velD (mm/s) - NED down velocity	
25	Int32	PVT gSpeed (mm/s) - Ground Speed (2-D)	
26	Int32	PVT headMot (deg) - Heading of motion (2-D)	100000
27	UInt32	PVT sAcc (mm/s) - Speed accuracy estimate	
28	UInt32	PVT headAcc (deg) - Heading accuracy estimate	100000
29	UInt16	PVT pDOP - Position DOP	100
30	Int32	PVT headVeh (deg) - Heading of vehicle (2-D)	100000
31	UInt32	ATT iTOW (ms) - GPS time of week of the navigation	
32	UInt8	ATT version - Message version (0 for this version)	
33	Int32	ATT roll (deg) - Vehicle roll	100000
34	Int32	ATT pitch (deg) - Vehicle pitch	100000
35	Int32	ATT heading (deg) - Vehicle heading	100000
36	UInt32	ATT accRoll (deg) - Vehicle roll accuracy	100000
37	UInt32	ATT accPitch (deg) - Vehicle pitch accuracy	100000
38	UInt32	ATT accHeading (deg) - Vehicle heading accuracy	100000
39	UInt32	ESF iTOW (ms) - GPS time of week of the navigation	
40	UInt8	ESF version - Message version (2 for this version)	
41	UInt8	ESF initStatus1	
42	UInt8	ESF initStatus2	
43	UInt8	ESF fusionMode	
44	UInt8	ESF numSens - Number of sensors	
45	UInt32[n]	ESF sensStatus	

5.4 LPBUS example communication

In this section we will show a few practical examples of communication using the LP-BUS protocol.

Goto Command Mode

GET request (HOST -> SENSOR)

Packet byte no.	Content	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB

3	06h	Command no. LSB (06h = GOTO_COMMAND_MODE)
4	00h	Command no. MSB
5	00h	Data length LSB (GET 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	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	00h	Command no. LSB (00h = REPLY_ACK)
4	00h	Command no. MSB
5	00h	Data length LSB (Zero length data for ACK reply)
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

Goto Steaming Mode

GET request (HOST -> SENSOR)

Packet byte no.	Content	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	07h	Command no. LSB (07h = GOTO_STREAMING_MODE)
4	00h	Command no. MSB
5	00h	Data length LSB (GET 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	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	00h	Command no. LSB (00h = REPLY_ACK)
4	00h	Command no. MSB
5	00h	Data length LSB (Zero length data for ACK reply)
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

Request Gyroscope Range

GET request (HOST -> SENSOR)

Packet byte no.	Content	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	3Dh	Command no. LSB (3Dh = GET_GYRO_RANGE)
4	00h	Command no. MSB
5	00h	Data length LSB (GET command = no data)
6	00h	Data length MSB
7	3Eh	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	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	3Dh	Command no. LSB (3Dh = GET_GYRO_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

xx = Value depends on the current sensor configuration.

Set Accelerometer Range

SET request (HOST -> SENSOR)

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

Reply data (SENSOR -> HOST)

Packet byte no.	Content	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	00h	Command no. LSB (00h = REPLY_ACK)
4	00h	Command no. MSB
5	00h	Data length LSB (Zero length data for ACK reply)
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

Save sensor parameters

WRITE_REGISTER request (HOST -> SENSOR)

Packet byte no.	Content	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	04h	Command no. LSB (04h = WRITE_REGISTER)
4	00h	Command no. MSB
5	00h	Data length LSB (WRITE_REGISTER 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	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	00h	Command no. LSB (00h = REPLY_ACK)
4	00h	Command no. MSB
5	00h	Data length LSB (Zero length data for ACK reply)
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: WRITE_REGISTER command involves flash operation, which might result in delay ACK response

Read sensor status

GET request (HOST -> SENSOR)

Packet byte no.	Content	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	08h	Command no. LSB (08h = GET_SENSOR_STATUS)
4	00h	Command no. MSB
5	00h	Data length LSB (GET command = no data)
6	00h	Data length MSB
7	09h	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	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	08h	Command no. LSB (08h = GET_SENSOR_STATUS)
4	00h	Command no. MSB
5	04h	Data length LSB (32-bit integer = 4 bytes)
6	00h	Data length MSB
7-10	xxxxxxxxh	Sensor 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 details of reply data mapping

Set UART / RS232 baudrate

SET request (HOST -> SENSOR)

Packet byte no.	Content	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	82h	Command no. LSB (82h = SET_UART_BAUDRATE)
4	00h	Command no. MSB
5	04h	Data length LSB (32-bit integer = 4 bytes)
6	00h	Data length MSB
7	00h	921600 = 0x000E1000
8	10h	
9	0Eh	
10	00h	
11	A5h	Check sum LSB
12	00h	Check sum MSB
13	0Dh	Packet end 1
14	0Ah	Packet end 2

Reply data (SENSOR -> HOST)

Packet byte no.	Content	Description
0	3Ah	Packet start
1	01h	Sensor ID LSB (ID = 1)
2	00h	Sensor ID MSB
3	00h	Command no. LSB (00h = REPLY_ACK)
4	00h	Command no. MSB
5	00h	Data length LSB (Zero length data for ACK reply)
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: Power cycle is required for baudrate settings to take effect

5.5 CANOpen and Sequential CAN protocol

In CANOpen and sequential CAN transmission mode, two or more output words of measurement data can be assigned to a CAN channel. In sequential CAN mode the channel addressing can be individually controlled. In CANOpen mode, 4 TPDO (Transmission Data Process Object) messages and a heartbeat message are transmitted.

NOTE: In CANOpen mode a heartbeat message is transmitted with a frequency between 0.1 Hz and 2 Hz.

In CANOpen mode, the message base address is calculated in the following way:

CAN ID1 = Base CAN ID = Start ID + IMU ID

CAN ID2 = CAN ID1 + 100h

CAN ID3 = CAN ID2 + 100h

CAN ID4 = CAN ID3 + 100h

CAN ID5 = 700h + IMU ID

Note: In CANOpen mode, Start ID is fixed at 180h

In sequential CAN mode, the message base address is calculated in the following way:

CAN ID1 = Base CAN ID = Start ID + IMU ID

CAN ID2 = CAN ID1 + 1h

CAN ID3 = CAN ID2 + 1h

CAN ID4 = CAN ID3 + 1h

NOTE: In sequential CAN mode, Start ID is set to 514h (1300d) by default. This value can be changed via IG1Control interface

5.6 CAN data output format

Each CAN message can be assigned multiple channels representing the sensor data. The number of assignable sensor data will depend on the data output precision, i.e. 32bit data or 16bit data output. By utilizing 4 CAN message with 16bit sensor data precision, the sensor can output a maximum of 16 different sensor data for a given instance. Table 15 and Table 16 summarizes the channel mapping for 16bit and 32bit data output.

Table 15 CAN channel mapping (16bit data output)

CANOpen ID	Sequential CAN	Output data
181h	515h	channel 1 | channel 2 | channel 3 | channel 4
281h	516h	channel 5 | channel 6 | channel 7 | channel 8
381h	517h	channel 9 | channel 10 | channel 11 | channel 12
481h	518h	channel 13 | channel 14 | channel 15 | channel 16
701h	-	- (heartbeat)

Table 16 CAN channel mapping (32bit data output)

CANOpen ID	Sequential CAN	Output data
181h	515h	channel 1 | channel 2
281h	516h	channel 3 | channel 4
381h	517h	channel 5 | channel 6
481h	518h	channel 7 | channel 8
701h	-	- (heartbeat)

5.7 CAN mapping

Each channel can be assigned different sensor data by changing the CAN mapping via IG1Control. Table 17 summarizes the available sensor output data.

Table 17 CAN data mapping

Mapping index	Data	Unit
0	Not assigned	
1	Raw Accelerometer X	g
2	Raw Accelerometer Y	g
3	Raw Accelerometer Z	g
4	Calibrated Accelerometer X	g
5	Calibrated Accelerometer Y	g
6	Calibrated Accelerometer Z	g
7	Raw Gyro I X	dps or rad/s
8	Raw Gyro I Y	dps or rad/s
9	Raw Gyro I Z	dps or rad/s
10	Raw Gyro II X	dps or rad/s
11	Raw Gyro II Y	dps or rad/s
12	Raw Gyro II Z	dps or rad/s

13	Bias calibrated Gyro I X	dps or rad/s
14	Bias calibrated Gyro I Y	dps or rad/s
15	Bias calibrated Gyro I Z	dps or rad/s
16	Bias calibrated Gyro II X	dps or rad/s
17	Bias calibrated Gyro II Y	dps or rad/s
18	Bias calibrated Gyro II Z	dps or rad/s
19	Alignment calibrated Gyro I X	dps or rad/s
20	Alignment calibrated Gyro I Y	dps or rad/s
21	Alignment calibrated Gyro I Z	dps or rad/s
22	Alignment calibrated Gyro II X	dps or rad/s
23	Alignment calibrated Gyro II Y	dps or rad/s
24	Alignment calibrated Gyro II Z	dps or rad/s
25	Raw magnetometer X	uT
26	Raw magnetometer Y	uT
27	Raw magnetometer Z	uT
28	Calibrated magnetometer X	uT
29	Calibrated magnetometer Y	uT
30	Calibrated magnetometer Z	uT
31	Reserved	
32	Reserved	
33	Reserved	
34	Quaternion W	
35	Quaternion X	
36	Quaternion Y	
37	Quaternion Z	
38	Euler X	deg or rad
39	Euler X	deg or rad
40	Euler Z	deg or rad
41	Reserved	
42	Reserved	
43	Reserved	
44	Reserved	
45	Temperature	°C

5.8 ASCII output

LPMS-IG1 RS232 supports both LPBus protocol and ASCII output via RS232 terminal. The output format can be set via SET_UART_FORMAT command. In ASCII output format, every data packet has a prefix Start character and End character at the end. Default start character is '$' and end character is '\n'. User can define both Start and End character via SET_UART_ASCII_CHARACTER command. Sensor data is streamed out in comma separated format. Sensor data is in 16bit integer format. ASCII data should be scaled according Table 13 to obtain correct data.

Table 18 ASCII output format

Start character	Data 1	,	Data 2	,	......	,	Data n	End character

6. Dimensions

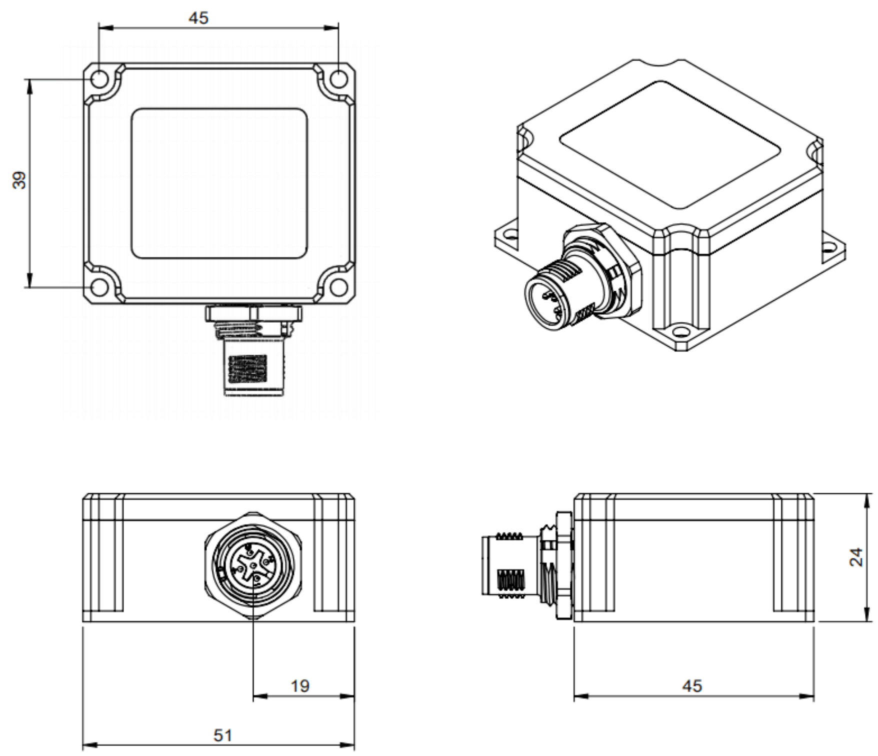

Fig 6.1. LPMS-IG1 Drawings (Unit: mm)

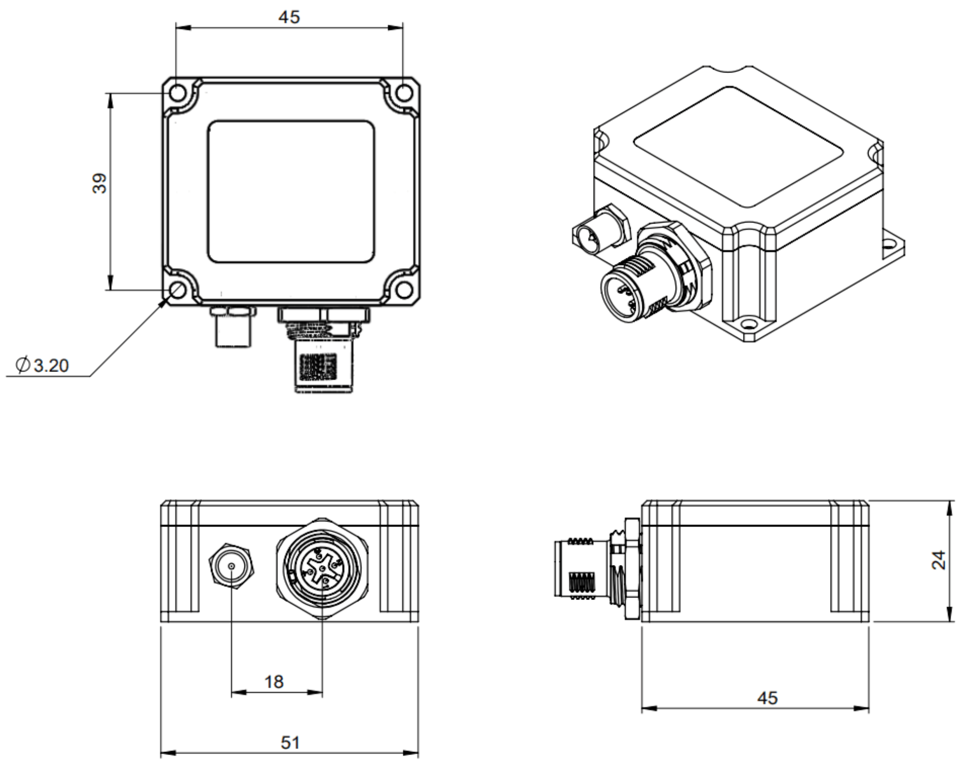

Figure 5 LPMS-IG1P Drawings (Unit: mm)

7. Appendix

7.1 Firmware function / 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		
10 (0Ah)	GET_GPS_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	500dps
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

Magnetometer Settings Commands				
Identifier	**Name**	**Parameter**	**Response**	**Default**
70 (46h)	SET_MAG_RANGE	Int32	ACK/NACK	
71 (47h)	GET_MAG_RANGE	NONE	Int32	8Gauss
84 (54h)	START_MAG_CALIBRATION	NONE	ACK/NACK	
85 (55h)	STOP_MAG_CALIBRATION	NONE	ACK/NACK	
86 (56h)	SET_MAG_CALIBRATION_TIMEOUT	Int32	ACK/NACK	
87 (57h)	GET_MAG_CALIBRATION_TIMEOUT	NONE	Int32	20s

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

Can Settings Command				
Identifier	**Name**	**Parameter**	**Response**	**Default**
110 (6Eh)	SET_CAN_START_ID	Int32	ACK/NACK	
111 (6Fh)	GET_CAN_START_ID	NONE	Int32	0x514
112 (70h)	SET_CAN_BAUDRATE	Int32	ACK/NACK	
113 (71h)	GET_CAN_BAUDRATE	NONE	Int32	500
114 (72h)	SET_CAN_DATA_PRECISION	Int32	ACK/NACK	
115 (73h)	GET_CAN_DATA_PRECISION	NONE	Int32	0
116 (74h)	SET_CAN_MODE	Int32	ACK/NACK	
117 (75h)	GET_CAN_MODE	NONE	Int32	0
118 (76h)	SET_CAN_MAPPING	Int32[16]	ACK/NACK	
119 (77h)	GET_CAN_MAPPING	NONE	Int32[16]	
120 (78h)	SET_CAN_HEARTBEAT	Int32	ACK/NACK	
121 (79h)	GET_CAN_HEARTBEAT	NONE	Int32	1

UART / RS232 Settings Command				
Identifier	**Name**	**Parameter**	**Response**	**Default**
130 (82h)	SET_UART_BAUDRATE	Int32	ACK/NACK	
131 (83h)	GET_UART_BAUDRATE	NONE	Int32	921600
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]	h24 h0D
136 (88h)	SET_LPBUS_DATA_PRECISION	Int32	ACK/NACK	

137 (89h)	GET_LPBUS_DATA_PRECISION	NONE	Int32	1

Sensor Data Timestamp Manipulation				
Identifier	**Name**	**Parameter**	**Response**	**Default**
152 (98h)	SET_TIMESTAMP	Int32	ACK/NACK	
GPS Data Transmission Commands				
Identifier	**Name**	**Parameter**	**Response**	**Default**
160 (A0h)	SET_GPS_TRANSMIT_DATA	Int32[2]	ACK/NACK	
161 (A1h)	GET_GPS_TRANSMIT_DATA	NONE	Int32[2]	
162 (A2h)	SAVE_GPS_STATE	NONE	ACK/NACK	
163 (A3h)	CLEAR_GPS_STATE	NONE	ACK/NACK	

Acknowledged and Not-acknowledged Identifiers

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

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

Register Value Save and Reset Command

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

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

Mode Switching Commands

Identifier	6 (0x06)
Name	GOTO_COMMAND_MODE

Description	Switch to command mode. In command mode the user can issue commands to the firmware to perform calibration, set parameters etc.
Parameter	NONE
Response:	ACK (success) or NACK (error)
Identifier	7 (0x07)
Name	GOTO_STREAM_MODE
Description	Switch to streaming mode. In this mode data is continuously streamed from the sensor, and some commands cannot be performed until the sensor receives the GOTO_COMMAND_MODE command.
Parameter	NONE
Response:	ACK (success) or NACK (error)

Sensor Status Command

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

Get Data Commands

Identifier	9 (0x09)
Name	GET_IMU_DATA
Description	Get the sensor data
Parameter	NONE
Response	Please see Chapter 5.3 for details

Identifier	10 (0x0A)
Name	GET_GPS_DATA
Description	Get the GPS data
Parameter	NONE
Response	Please see Chapter 5.3 for details

Device Info Commands

Identifier	20 (0x14)
Name	GET_SENSOR_MODEL

Description	Get the sensor model information
Parameter	NONE
Response	Char[24]

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

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

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

Data Transmission Commands

Identifier	30 (0x1E)
Name	SET_IMU_TRANSMIT_DATA
Description	Set the current transmitted data of sensor

<table>
<tr><td>Parameter</td><td>Int32
<table>
<tr><th>Bit</th><th>Reported State / Parameter</th></tr>
<tr><td>0</td><td>Accelerometer raw data transmission enabled</td></tr>
<tr><td>1</td><td>Accelerometer calibrated data transmission enabled</td></tr>
<tr><td>2</td><td>Gyro I raw data transmission enabled</td></tr>
<tr><td>3</td><td>Gyro II raw data transmission enabled</td></tr>
<tr><td>4</td><td>Gyro I bias calibrated data transmission enabled</td></tr>
<tr><td>5</td><td>Gyro II bias calibrated data transmission enabled</td></tr>
<tr><td>6</td><td>Gyro I alignment calibrated data transmission enabled</td></tr>
<tr><td>7</td><td>Gyro II alignment calibrated data transmission enabled</td></tr>
<tr><td>8</td><td>Magnetometer raw data transmission enabled</td></tr>
<tr><td>9</td><td>Magnetometer calibrated data transmission enabled</td></tr>
<tr><td>10</td><td>Reserved</td></tr>
<tr><td>11</td><td>Quaternion orientation transmission enabled</td></tr>
<tr><td>12</td><td>Euler angle data transmission enabled</td></tr>
<tr><td>13</td><td>Reserved</td></tr>
<tr><td>14</td><td>Reserved</td></tr>
<tr><td>15</td><td>Reserved</td></tr>
<tr><td>16</td><td>Temperature data transmission enabled</td></tr>
<tr><td>17-31</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>31 (0x1F)</td></tr>
<tr><td>Name</td><td>GET_IMU_TRANSMIT_DATA</td></tr>
<tr><td>Description</td><td>Get the current transmitted data from sensor</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int32
<table>
<tr><th>Bit</th><th>Reported State / Parameter</th></tr>
<tr><td>0</td><td>Accelerometer raw data transmission enabled</td></tr>
<tr><td>1</td><td>Accelerometer calibrated data transmission enabled</td></tr>
<tr><td>2</td><td>Gyro I raw data transmission enabled</td></tr>
<tr><td>3</td><td>Gyro II raw data transmission enabled</td></tr>
<tr><td>4</td><td>Gyro I bias calibrated data transmission enabled</td></tr>
<tr><td>5</td><td>Gyro II bias calibrated data transmission enabled</td></tr>
<tr><td>6</td><td>Gyro I alignment calibrated data transmission enabled</td></tr>
<tr><td>7</td><td>Gyro II alignment calibrated data transmission enabled</td></tr>
<tr><td>8</td><td>Magnetometer raw data transmission enabled</td></tr>
<tr><td>9</td><td>Magnetometer calibrated data transmission enabled</td></tr>
<tr><td>10</td><td>Reserved</td></tr>
<tr><td>11</td><td>Quaternion orientation transmission enabled</td></tr>
<tr><td>12</td><td>Euler angle data transmission enabled</td></tr>
<tr><td>13</td><td>Reserved</td></tr>
<tr><td>14</td><td>Reserved</td></tr>
<tr><td>15</td><td>Reserved</td></tr>
<tr><td>16</td><td>Temperature data transmission enabled</td></tr>
<tr><td>17-31</td><td>Reserved</td></tr>
</table></td></tr>
</table>

IMU ID Setting Command

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

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

Stream Frequency Commands

Identifier	34 (0x22)
Name	SET_STREAM_FREQ
Description	Set the current streaming frequency
Parameter	Int32 <table><tr><th>Frequency (Hz)</th><th>Identifier</th></tr><tr><td>5</td><td>5</td></tr><tr><td>10</td><td>10</td></tr><tr><td>50</td><td>50</td></tr><tr><td>100</td><td>100</td></tr><tr><td>500</td><td>500</td></tr></table>
Response:	ACK (success) or NACK (error)

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

Deg/Rad Output Commands

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

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

Reference Setting and Offset Reset Command

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

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

Accelerometer Settings Command

<table>
<tr><td>Identifier</td><td>50 (0x32)</td></tr>
<tr><td>Name</td><td>SET_ACC_RANGE</td></tr>
<tr><td>Description</td><td>Set the current range of the accelerometer</td></tr>
<tr><td>Parameter</td><td>Int32<table>
<tr><th>Range</th><th>Identifier</th></tr>
<tr><td>2g</td><td>2</td></tr>
<tr><td>4g</td><td>4</td></tr>
<tr><td>8g</td><td>8</td></tr>
<tr><td>16g</td><td>16</td></tr>
</table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>51 (0x33)</td></tr>
<tr><td>Name</td><td>GET_ACC_RANGE</td></tr>
<tr><td>Description</td><td>Get the current range of the accelerometer</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int32<table><tr><th>Range</th><th>Identifier</th></tr><tr><td>2g</td><td>2</td></tr><tr><td>4g</td><td>4</td></tr><tr><td>8g</td><td>8</td></tr><tr><td>16g</td><td>16</td></tr></table></td></tr>
</table>

Gyroscope Settings Command

<table>
<tr><td>Identifier</td><td>60 (0x3C)</td></tr>
<tr><td>Name</td><td>SET_GYR_RANGE</td></tr>
<tr><td>Description</td><td>Set the current range of the gyroscope</td></tr>
<tr><td>Parameter</td><td>Int32<table><tr><th>Range (deg/s)</th><th>Identifier</th></tr><tr><td>400</td><td>400</td></tr><tr><td>1000</td><td>1000</td></tr><tr><td>2000</td><td>2000</td></tr></table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>61 (0x3D)</td></tr>
<tr><td>Name</td><td>GET_GYR_RANGE</td></tr>
<tr><td>Description</td><td>Get current gyroscope range.</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int32<table><tr><th>Range (deg/s)</th><th>Identifier</th></tr><tr><td>400</td><td>400</td></tr><tr><td>1000</td><td>1000</td></tr><tr><td>2000</td><td>2000</td></tr></table></td></tr>
</table>

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

<table>
<tr><td>Identifier</td><td>64 (0x40)</td></tr>
<tr><td>Name</td><td>SET_ENABLE_GYR_AUTOCALIBRATION</td></tr>
<tr><td>Description</td><td>Enable / Disable gyro autocalibration.</td></tr>
<tr><td>Parameter</td><td>Int32<table><tr><th>Function</th><th>Identifier</th></tr><tr><td>ENABLE_GYR_AUTOCAL</td><td>1</td></tr><tr><td>DISABLE_GYR_AUTOCAL</td><td>0</td></tr></table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>65 (0x41)</td></tr>
<tr><td>Name</td><td>GET_ENABLE_GYR_AUTOCALIBRATION</td></tr>
<tr><td>Description</td><td>Get gyro autocalibration status.</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int32<table><tr><th>Function</th><th>Identifier</th></tr><tr><td>ENABLE_GYR_AUTOCAL</td><td>1</td></tr><tr><td>DISABLE_GYR_AUTOCAL</td><td>0</td></tr></table></td></tr>
<tr><td>Identifier</td><td>66 (0x42)</td></tr>
<tr><td>Name</td><td>SET_GYR_THRESHOLD</td></tr>
<tr><td>Description</td><td>Set the gyroscope threshold which can be used to suppress noise or vibrations that might impact the sensor measurements.</td></tr>
<tr><td>Parameter</td><td>Float32</td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

Identifier	66 (0x43)
Name	GET_GYR_THRESHOLD
Description	Get the gyroscope threshold.
Parameter	NONE
Response:	Float32

Magnetometer Settings Command

<table>
<tr><td>Identifier</td><td>70 (0x46)</td></tr>
<tr><td>Name</td><td>SET_MAG_RANGE</td></tr>
<tr><td>Description</td><td>Set the current range of the gyroscope</td></tr>
<tr><td>Parameter</td><td>Int32<table><tr><th>Range</th><th>Identifier</th></tr><tr><td>2 Gauss</td><td>2</td></tr><tr><td>8 Gauss</td><td>8</td></tr></table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

Identifier	71 (0x47)
Name	GET_MAG_RANGE
Description	Get current magnetometer range.
Parameter	NONE
Response:	Int32 <table><tr><th>Range</th><th>Identifier</th></tr><tr><td>2 Gauss</td><td>2</td></tr><tr><td>8 Gauss</td><td>8</td></tr></table>

Identifier	84 (0x54)
Name	START_MAG_CALIBRATION
Description	Start calibration of magnetometer.
Parameter	NONE
Response:	ACK (success) or NACK (error)

Identifier	85 (0x55)
Name	STOP_MAG_CALIBRATION
Description	Stop calibration of magnetometer.
Parameter	NONE
Response:	ACK (success) or NACK (error)
Identifier	86 (0x56)
Name	SET_MAG_CALBRATION_TIMEOUT
Description	Set the time of the magnetometer calibration.
Parameter	Int32
Response:	ACK (success) or NACK (error)

Identifier	87 (0x57)
Name	GET_MAG_CALBRATION_TIMEOUT
Description	Get the time of the magnetometer calibration.
Parameter	NONE
Response:	Int32

Filter Settings Command

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

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

Can Settings Command

Identifier	110 (0x6E)
Name	SET_CAN_START_ID
Description	Set the CAN sequential start ID
Parameter	Int32
Response:	ACK (success) or NACK (error)

Identifier	111 (0x6F)
Name	GET_CAN_START_ID
Description	Get the CAN sequential start ID
Parameter	NONE
Response:	Int32

<table>
<tr><td>Identifier</td><td>112 (0x70)</td></tr>
<tr><td>Name</td><td>SET_CAN_BAUDRATE</td></tr>
<tr><td>Description</td><td>Set the current can baudrate</td></tr>
<tr><td>Parameter</td><td>Int32
<table>
<tr><th>Baud rate</th><th>Identifier</th></tr>
<tr><td>125K</td><td>125</td></tr>
<tr><td>250K</td><td>250</td></tr>
<tr><td>500K</td><td>500</td></tr>
<tr><td>800K</td><td>800</td></tr>
<tr><td>1M</td><td>1000</td></tr>
</table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>113 (0x71)</td></tr>
<tr><td>Name</td><td>GET_CAN_BAUDRATE</td></tr>
<tr><td>Description</td><td>Get the current can 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>125K</td><td>125</td></tr>
<tr><td>250K</td><td>250</td></tr>
<tr><td>500K</td><td>500</td></tr>
<tr><td>800K</td><td>800</td></tr>
<tr><td>1M</td><td>1000</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>Identifier</td><td>114 (0x72)</td></tr>
<tr><td>Name</td><td>SET_CAN_DATA_PRECISION</td></tr>
<tr><td>Description</td><td>Set the CAN output data precision</td></tr>
<tr><td>Parameter</td><td>Int32
<table>
<tr><th>Data Precision</th><th>Identifier</th></tr>
<tr><td>16bit Fixed point</td><td>0</td></tr>
<tr><td>32bit Float point</td><td>1</td></tr>
</table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>115 (0x73)</td></tr>
<tr><td>Name</td><td>GET_CAN_DATA_PRECISION</td></tr>
<tr><td>Description</td><td>Get the CAN output data precision</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int32
<table>
<tr><th>Data Precision</th><th>Identifier</th></tr>
<tr><td>16bit Fixed point</td><td>0</td></tr>
<tr><td>32bit Float point</td><td>1</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>Identifier</td><td>117 (0x75)</td></tr>
<tr><td>Name</td><td>GET_CAN_MODE</td></tr>
<tr><td>Description</td><td>Get the current can mode</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int32
<table>
<tr><th>Mode</th><th>Identifier</th></tr>
<tr><td>CANopen</td><td>0</td></tr>
<tr><td>Sequential can</td><td>1</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>Identifier</td><td>118 (0x76)</td></tr>
<tr><td>Name</td><td>SET_CAN_MAPPING</td></tr>
<tr><td>Description</td><td>Set the current transmitted data of each can channel</td></tr>
<tr><td>Parameter</td><td>Int32[16]
<table>
<tr><th>Int32[]</th><th>CAN Channel</th><th>Identifier</th></tr>
<tr><td>Int32[0]</td><td>Channel1</td><td>Mapping index</td></tr>
<tr><td>Int32[1]</td><td>Channel2</td><td>Mapping index</td></tr>
<tr><td>Int32[2]</td><td>Channel3</td><td>Mapping index</td></tr>
<tr><td>…</td><td>…</td><td>…</td></tr>
<tr><td>Int32[15]</td><td>Channel16</td><td>Mapping index</td></tr>
</table>
Mapping index please refer to Table 5-8</td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>119 (0x77)</td></tr>
<tr><td>Name</td><td>GET_CAN_MAPPING</td></tr>
<tr><td>Description</td><td>Get the current transmitted data of each can channel</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int32[16]
<table>
<tr><th>Int32[]</th><th>CAN Channel</th><th>Identifier</th></tr>
<tr><td>Int32[0]</td><td>Channel1</td><td>Mapping index</td></tr>
<tr><td>Int32[1]</td><td>Channel2</td><td>Mapping index</td></tr>
<tr><td>Int32[2]</td><td>Channel3</td><td>Mapping index</td></tr>
<tr><td>…</td><td>…</td><td>…</td></tr>
<tr><td>Int32[15]</td><td>Channel16</td><td>Mapping index</td></tr>
</table>
Mapping index please refer to Table 5-8</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>120 (0x78)</td></tr>
<tr><td>Name</td><td>SET_CAN_HEARTBEAT</td></tr>
<tr><td>Description</td><td>Set the CANopen heartbeat</td></tr>
<tr><td>Parameter</td><td>Int32
<table>
<tr><th>period</th><th>Identifier</th></tr>
<tr><td>0.5s</td><td>0</td></tr>
<tr><td>1s</td><td>1</td></tr>
<tr><td>2s</td><td>2</td></tr>
<tr><td>5s</td><td>5</td></tr>
<tr><td>10s</td><td>10</td></tr>
</table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>121 (0x79)</td></tr>
<tr><td>Name</td><td>GET_CAN_HEARTBEAT</td></tr>
<tr><td>Description</td><td>Get the CANopen heartbeat</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int32
<table>
<tr><th>period</th><th>Identifier</th></tr>
<tr><td>0.5s</td><td>0</td></tr>
<tr><td>1s</td><td>1</td></tr>
<tr><td>2s</td><td>2</td></tr>
<tr><td>5s</td><td>5</td></tr>
<tr><td>10s</td><td>10</td></tr>
</table></td></tr>
</table>

UART / RS232 Settings Command

<table>
<tr><td>Identifier</td><td>130 (0x82)</td></tr>
<tr><td>Name</td><td>SET_UART_BAUDRATE</td></tr>
<tr><td>Description</td><td>Set the current UART / RS232 baudrate</td></tr>
<tr><td>Parameter</td><td>Int32
<table>
<tr><th>Baud rate</th><th>Identifier</th></tr>
<tr><td>115200</td><td>115200</td></tr>
<tr><td>230400</td><td>230400</td></tr>
<tr><td>256000</td><td>256000</td></tr>
<tr><td>460800</td><td>460800</td></tr>
<tr><td>921600</td><td>921600</td></tr>
</table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>131 (0x83)</td></tr>
<tr><td>Name</td><td>GET_UART_BAUDRATE</td></tr>
<tr><td>Description</td><td>Get the current UART / RS232 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>115200</td><td>115200</td></tr>
<tr><td>230400</td><td>230400</td></tr>
<tr><td>256000</td><td>256000</td></tr>
<tr><td>460800</td><td>460800</td></tr>
<tr><td>921600</td><td>921600</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>Identifier</td><td>132 (0x84)</td></tr>
<tr><td>Name</td><td>SET_UART_FORMAT</td></tr>
<tr><td>Description</td><td>Set the UART / RS232 output format</td></tr>
<tr><td>Parameter</td><td>Int32
<table>
<tr><th>Format</th><th>Identifier</th></tr>
<tr><td>LPBUS</td><td>0</td></tr>
<tr><td>ASCII</td><td>1</td></tr>
</table></td></tr>
<tr><td>Response:</td><td>ACK (success) or NACK (error)</td></tr>
</table>

<table>
<tr><td>Identifier</td><td>133 (0x85)</td></tr>
<tr><td>Name</td><td>GET_UART_FORMAT</td></tr>
<tr><td>Description</td><td>Get the UART / RS232 output format</td></tr>
<tr><td>Parameter</td><td>NONE</td></tr>
<tr><td>Response:</td><td>Int32
<table>
<tr><th>Format</th><th>Identifier</th></tr>
<tr><td>LPBUS</td><td>0</td></tr>
<tr><td>ASCII</td><td>1</td></tr>
</table></td></tr>
</table>

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

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

Identifier	136 (0x88)
Name	SET_LPBUS_DATA_PRECISION
Description	Set the current UART / RS232 output data precision
Parameter	Int32 <table><tr><th>Data Precision</th><th>Identifier</th></tr><tr><td>16bit Fixed point</td><td>0</td></tr><tr><td>32bit Float point</td><td>1</td></tr></table>
Response:	ACK (success) or NACK (error)

Identifier	137 (0x89)
Name	GET_LPBUS_DATA_PRECISION
Description	Get the current UART / RS232 output data precision
Parameter	NONE
Response:	Int32 <table><tr><th>Data Precision</th><th>Identifier</th></tr><tr><td>16bit Fixed point</td><td>0</td></tr><tr><td>32bit Float point</td><td>1</td></tr></table>

Sensor Data Timestamp Manipulation

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

GPS Data Transmission Commands

<table>
<tr><td>Identifier</td><td colspan="3">160 (0xA0)</td></tr>
<tr><td>Name</td><td colspan="3">SET_GPS_TRANSMIT_DATA</td></tr>
<tr><td>Description</td><td colspan="3">Set the current transmitted data of GPS</td></tr>
<tr><td rowspan="50">Parameter</td><td colspan="3">Int32[2]</td></tr>
<tr><td rowspan="31">Int32[0]</td><td>Bit</td><td>Reported State / Parameter</td></tr>
<tr><td>0</td><td>GPS NAV-PVT iTOW transmission enabled</td></tr>
<tr><td>1</td><td>GPS NAV-PVT year transmission enabled</td></tr>
<tr><td>2</td><td>GPS NAV-PVT month transmission enabled</td></tr>
<tr><td>3</td><td>GPS NAV-PVT day transmission enabled</td></tr>
<tr><td>4</td><td>GPS NAV-PVT hour transmission enabled</td></tr>
<tr><td>5</td><td>GPS NAV-PVT min transmission enabled</td></tr>
<tr><td>6</td><td>GPS NAV-PVT sec transmission enabled</td></tr>
<tr><td>7</td><td>GPS NAV-PVT valid transmission enabled</td></tr>
<tr><td>8</td><td>GPS NAV-PVT tAcc transmission enabled</td></tr>
<tr><td>9</td><td>GPS NAV-PVT nano transmission enabled</td></tr>
<tr><td>10</td><td>GPS NAV-PVT fixType transmission enabled</td></tr>
<tr><td>11</td><td>GPS NAV-PVT flags transmission enabled</td></tr>
<tr><td>12</td><td>GPS NAV-PVT flags2 transmission enabled</td></tr>
<tr><td>13</td><td>GPS NAV-PVT numSV transmission enabled</td></tr>
<tr><td>14</td><td>GPS NAV-PVT longitude transmission enabled</td></tr>
<tr><td>15</td><td>GPS NAV-PVT latitude transmission enabled</td></tr>
<tr><td>16</td><td>GPS NAV-PVT height transmission enabled</td></tr>
<tr><td>17</td><td>GPS NAV-PVT hMSL transmission enabled</td></tr>
<tr><td>18</td><td>GPS NAV-PVT hAcc transmission enabled</td></tr>
<tr><td>19</td><td>GPS NAV-PVT vAcc transmission enabled</td></tr>
<tr><td>20</td><td>GPS NAV-PVT velN transmission enabled</td></tr>
<tr><td>21</td><td>GPS NAV-PVT velE transmission enabled</td></tr>
<tr><td>22</td><td>GPS NAV-PVT velD transmission enabled</td></tr>
<tr><td>23</td><td>GPS NAV-PVT gSpeed transmission enabled</td></tr>
<tr><td>24</td><td>GPS NAV-PVT headMot transmission enabled</td></tr>
<tr><td>25</td><td>GPS NAV-PVT sAcc transmission enabled</td></tr>
<tr><td>26</td><td>GPS NAV-PVT headAcc transmission enabled</td></tr>
<tr><td>27</td><td>GPS NAV-PVT pDOP transmission enabled</td></tr>
<tr><td>28</td><td>GPS NAV-PVT headVeh transmission enabled</td></tr>
<tr><td>29-31</td><td>Reserved</td></tr>
<tr><td rowspan="18">Int32[1]</td><td>Bit</td><td>Reported State / Parameter</td></tr>
<tr><td>0</td><td>GPS NAV-ATT iTOW transmission enabled</td></tr>
<tr><td>1</td><td>GPS NAV-ATT version transmission enabled</td></tr>
<tr><td>2</td><td>GPS NAV-ATT roll transmission enabled</td></tr>
<tr><td>3</td><td>GPS NAV-ATT pitch transmission enabled</td></tr>
<tr><td>4</td><td>GPS NAV-ATT heading transmission enabled</td></tr>
<tr><td>5</td><td>GPS NAV-ATT accRoll transmission enabled</td></tr>
<tr><td>6</td><td>GPS NAV-ATT accPitch transmission enabled</td></tr>
<tr><td>7</td><td>GPS NAV-ATT accHeading transmission enabled</td></tr>
<tr><td>8</td><td>GPS ESF-STATUS iTOW transmission enabled</td></tr>
<tr><td>9</td><td>GPS ESF-STATUS version transmission enabled</td></tr>
<tr><td>10</td><td>GPS ESF-STATUS initStatus1 transmission enabled</td></tr>
<tr><td>11</td><td>GPS ESF-STATUS initStatus2 transmission enabled</td></tr>
<tr><td>12</td><td>GPS ESF-STATUS fusionMode transmission enabled</td></tr>
<tr><td>13</td><td>GPS ESF-STATUS numSens transmission enabled</td></tr>
<tr><td>14</td><td>GPS ESF-STATUS sensStatus transmission enabled</td></tr>
<tr><td>15-31</td><td>Reserved</td></tr>
</table>

Response:	ACK (success) or NACK (error)
Identifier	161 (0xA1)
Name	GET_GPS_TRANSMIT_DATA
Description	Get the current transmitted data of GPS
Parameter	NONE

Response:	Int32[2]		
		Bit	**Reported State / Parameter**
	Int32[0]	**0**	GPS NAV-PVT iTOW transmission enabled
		1	GPS NAV-PVT year transmission enabled
		2	GPS NAV-PVT month transmission enabled
		3	GPS NAV-PVT day transmission enabled
		4	GPS NAV-PVT hour transmission enabled
		5	GPS NAV-PVT min transmission enabled
		6	GPS NAV-PVT sec transmission enabled
		7	GPS NAV-PVT valid transmission enabled
		8	GPS NAV-PVT tAcc transmission enabled
		9	GPS NAV-PVT nano transmission enabled
		10	GPS NAV-PVT fixType transmission enabled
		11	GPS NAV-PVT flags transmission enabled
		12	GPS NAV-PVT flags2 transmission enabled
		13	GPS NAV-PVT numSV transmission enabled
		14	GPS NAV-PVT longitude transmission enabled
		15	GPS NAV-PVT latitude transmission enabled
		16	GPS NAV-PVT height transmission enabled
		17	GPS NAV-PVT hMSL transmission enabled
		18	GPS NAV-PVT hAcc transmission enabled
		19	GPS NAV-PVT vAcc transmission enabled
		20	GPS NAV-PVT velN transmission enabled
		21	GPS NAV-PVT velE transmission enabled
		22	GPS NAV-PVT velD transmission enabled
		23	GPS NAV-PVT gSpeed transmission enabled
		24	GPS NAV-PVT headMot transmission enabled
		25	GPS NAV-PVT sAcc transmission enabled
		26	GPS NAV-PVT headAcc transmission enabled
		27	GPS NAV-PVT pDOP transmission enabled
		28	GPS NAV-PVT headVeh transmission enabled
		29-31	Reserved
		Bit	**Reported State / Parameter**
	Int32[1]	**0**	GPS NAV-ATT iTOW transmission enabled
		1	GPS NAV-ATT version transmission enabled
		2	GPS NAV-ATT roll transmission enabled
		3	GPS NAV-ATT pitch transmission enabled
		4	GPS NAV-ATT heading transmission enabled
		5	GPS NAV-ATT accRoll transmission enabled
		6	GPS NAV-ATT accPitch transmission enabled
		7	GPS NAV-ATT accHeading transmission enabled
		8	GPS ESF-STATUS iTOW transmission enabled
		9	GPS ESF-STATUS version transmission enabled
		10	GPS ESF-STATUS initStatus1 transmission enabled
		11	GPS ESF-STATUS initStatus2 transmission enabled
		12	GPS ESF-STATUS fusionMode transmission enabled
		13	GPS ESF-STATUS numSens transmission enabled
		14	GPS ESF-STATUS sensStatus transmission enabled
		15-31	Reserved

Identifier	162 (0xA2)
Name	SAVE_GPS_STATE
Description	Save current GPS state to the flash of GPS module
Parameter	NONE
Response:	ACK (success) or NACK (error)

Identifier	163 (0xA3)
Name	CLEAR_GPS_STATE
Description	Clear flash of GPS module
Parameter	NONE
Response:	ACK (success) or NACK (error)

7.2 Temperature-current-voltage characteristics

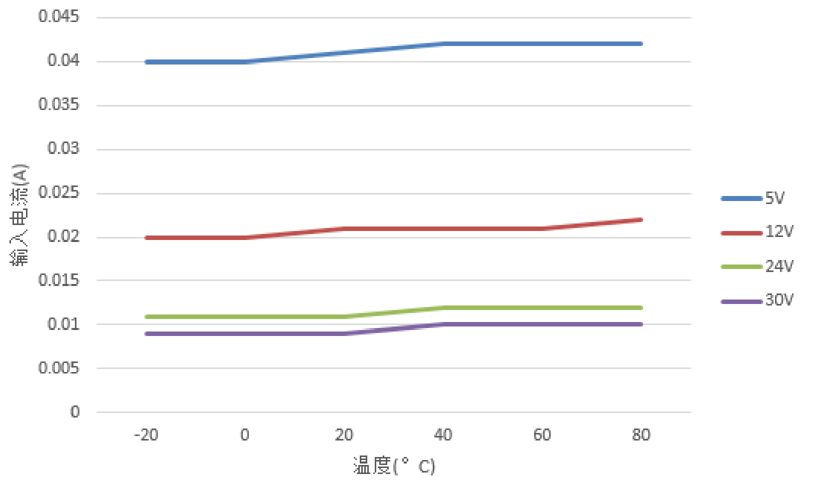

Figure 6 Temperature-current-voltage characteristics

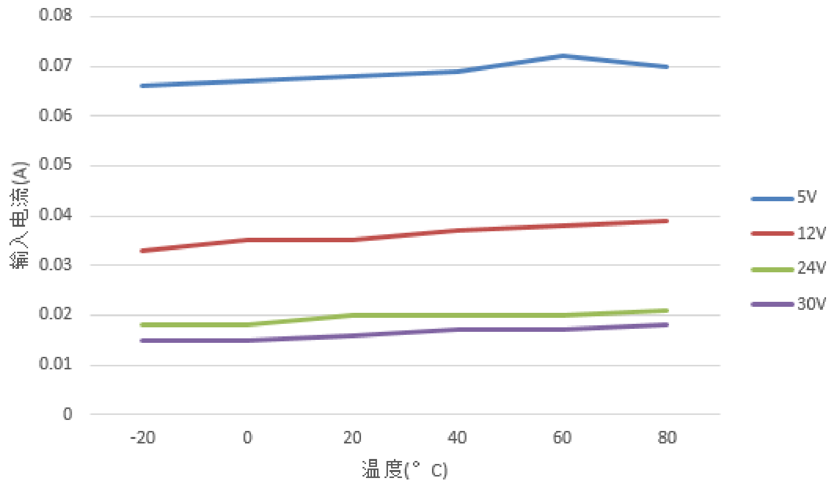

Figure 7 LPMS-IG1P Temperature-current-voltage characteristics

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