LG69T (AP) GNSS Protocol Specification

GNSS Module Series

Rev. LG69T(AP)_GNSS_Protocol_Specification_V1.0

Date: 2020-06-05

Status: Preliminary

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai, 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local office. For more information, please visit:
http://www.quectel.com/support/sales.htm

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm
Or email to: support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

THE INFORMATION CONTAINED HERE IS PROPRIETARY TECHNICAL INFORMATION OF QUECTEL WIRELESS SOLUTIONS CO., LTD. TRANSMITTING, REPRODUCTION, DISSEMINATION AND EDITING OF THIS DOCUMENT AS WELL AS UTILIZATION OF THE CONTENT WITHOUT PERMISSION ARE FORBIDDEN. OFFENDERS WILL BE HELD LIABLE FOR PAYMENT OF DAMAGES. ALL RIGHTS ARE RESERVED IN THE EVENT OF A PATENT GRANT OR REGISTRATION OF A UTILITY MODEL OR DESIGN.

Copyright © Quectel Wireless Solutions Co., Ltd. 2020. All rights reserved.

About the Document

Revision History

Revision	Date	Author	Description
1.0	2020-06-05	Jenn XIANG	Initial

Contents

Table Index

1 Introduction

LG69T(AP) GNSS module supports GPS, Galileo, Beidou and QZSS constellation and it provides fast and accurate acquisition. The simultaneously tracking of GPS L1 C/A, GPS L5, BeiDou B1, BeiDou B2a, Galileo E1, Galileo E5a, QZSS L1 and QZSS L5 makes this module an ideal solution for positioning and navigation in various vertical markets.

This document describes the software protocol used for controlling LG69T (AP) and the output NMEA messages supported by the module. LG69T (AP) can be controlled and configured through NMEA proprietary messages defined by Quectel, and it supports output messages at NMEA 0183 standard format protocol to provide various GNSS information.

NOTE

Please use the commands listed in this document only. Quectel assumes no responsibility for other commands that are not listed within this document.

2 NMEA protocol

2.1. Protocol Structure

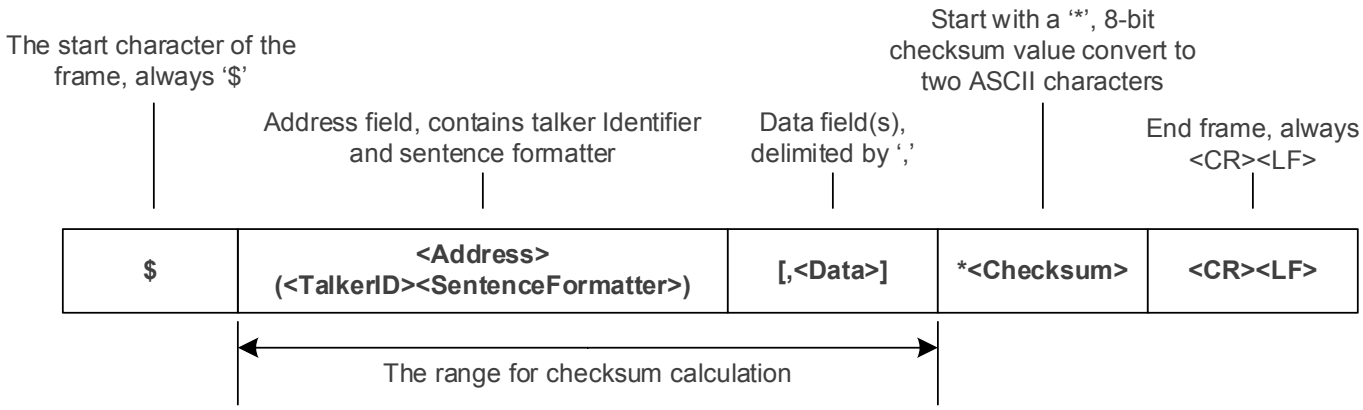

Figure 1: Structure of NMEA protocol message

$	Each NMEA message starts with '$'.
Address	This field contains TalkerID and SentenceFormatter. For the TalkerID, see table below, the SentenceFormatter is used to define the format and the type of data.
Data	These fields in approved sentences follow a "," delimiter with variable length.
Checksum	The checksum is the 8-bit exclusive OR of all characters in the sentence, including "," delimiters, between but not including the "$" and "*".
<CR><LF>	All NMEA message end with <CR><LF> (Hex 0x0D 0x0A) characters.

Table 1: NMEA TalkerID

GNSS Constellation Configuration	TalkerID (NMEA V4.11)
Galileo	GA
BeiDou	GB
GPS	GP
QZSS	GQ

Combination of multiple satellite systems.	GN

2.2. Standard Messages

This chapter introduces the NMEA V4.11 standard messages supported by LG69T (AP).

2.2.1. RMC

Recommended minimum position data (including position, velocity and time).

Type: Output

Format:

```
$<TalkerID>RMC,<Timestamp>,<Status>,<Lat>,<N/S>,<Long>,<E/W>,<SOG>,<COG>,<Date>,<MagVar>,<MagVarDir>,<Mode>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Description
$	Character	Each NMEA message starts with "$"
TalkerID	String, 2 characters	Talker identifier. GA: For Galileo GB: BeiDou GP: GPS GQ: QZSS GN: For multi-constellation mode
Timestamp	hhmmss.sss	UTC Time. hh: Hours (Fixed two digits) mm: Minutes (Fixed two digits) ss: Seconds (Fixed two digits) sss: Decimal fraction of seconds
Status	A or V	Navigation Status. A = Data valid. V=Invalid
Lat	DDMM.MMMMM	Latitude as degrees: DD: Degrees (Fixed two digits) MM: Minutes (Fixed two digits) .MMMMM: Decimal fraction of minutes Please note that this field is empty in case of an

		invalid value.
N/S	N or S	Latitude direction: N=North S=South Please note that this field is empty in case of an invalid value.
Long	DDDMM.MMMMM	Longitude as degrees: DDD: Degrees (Fixed three digits) MM: Minutes (Fixed two digits) MMMMM: Decimal fraction of minutes Please note that this field is empty in case of an invalid value.
E/W	E or W	Longitude direction: E=East W=West Please note that this field is empty in case of an invalid value.
SOG	x.x, variable length field	Speed over ground in knots. Please note that this field is empty in case of an invalid value.
COG	x.x, variable length field	Course over ground. The maximum value is 359.9. Please note that this field is empty in case of an invalid value.
Date	ddmmyy	Date in format ddmmyy
MagVar	-	Not supported.
MagVarDir	-	Not supported.
Mode	D, A, N or E	Positioning system mode indicator: D = Differential mode A = Autonomous mode N = Data not valid E = Estimated (dead reckoning) mode
*	-	End character of data field
Checksum	Hexadecimal	Hexadecimal checksum
<CR><LF>	Character	End of message

Example:

```
$GNRMC,071403.000,A,3149.3044284,N,11706.9136063,E,0.037,,230620,,,A,V*2F
```

2.2.2. GSA

GSA, GNSS DOP and Active Satellites. GNSS receiver operating mode, satellites used in the navigation solution reported by the GGA sentence and DOP values .

Type: Output

Format:

```
$<TalkerID>GSA,<Mode>,<CurrentMode>,<SatPRN1>,...,<SatPRNN>,<PDOP>,<HDOP>,<VDOP>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Description
$	Character	Each NMEA message starts with "$"
TalkerID	String, 2 characters	Talker identifier. GA: For Galileo GB: BeiDou GP: GPS GQ: QZSS GN: For multi-constellation mode
Mode	M or A	M = Manual, forced to operate in 2D or 3D mode A = Automatic, allowed to automatically switch 2D/3D
CurrentMode	Decimal, 1 digit	1 = Fix not available 2 = 2D 3 = 3D
SatPRN(1 to 12)	Decimal, 2 or 3 digits	Satellite ID number used for positioning
PDOP	x.x, variable length field	Position dilution of precision. The maximum value is 99.0.
HDOP	x.x, variable length field	Horizontal dilution of precision. The maximum value is 99.0.
VDOP	x.x, variable length field	Vertical dilution of precision. The maximum value is 99.0.
*	-	End character of data field
Checksum	Hexadecimal	Hexadecimal checksum
<CR><LF>	Character	Each NMEA message ends with "CR" and "LF"

Example:

```
$GNGSA,A,3,2,5,12,,,,,,,,,3.418,2.866,1.864,1*3A
$GNGSA,A,3,105,110,113,122,126,,,,,,,,3.418,2.866,1.864,4*08
```

2.2.3. GSV

GNSS Satellites in View. The GSV sentence provides the number of satellites (SV) in view, satellite ID numbers, elevation, azimuth, and SNR value. The GSV sentence contains four satellites maximum per transmission. The total number of sentences being transmitted and the sentence number being transmitted are indicated in the first two fields.

Type: Output

Format:

```
$<TalkerID>GSV,<GSVAmount>,<GSVNumber>,<TotSats>,<SatxPRN>,<SatxElev>,<SatxAzim>,<SatxCN0>,<SignalID>*<Checksum><CR><LF>
```

Parameter:

Field	Format	Description
$	Character	Each NMEA message starts with "$"
TalkerID	String, 2 characters	Talker identifier. GA: For Galileo GB: BeiDou GP: GPS GQ: QZSS GN: For multi-constellation mode
GSVAmount	Decimal, 1 digit	Total amount of GSV messages. The maximum value is 8.
GSVNumber	Decimal, 1 digit	Message number.
TotSats	Decimal, 2 digits	Total number of satellites in view. The maximum value is 32.
SatxPRN	Decimal, 2 digits	Satellite ID number used for positioning.
SatxElev	Decimal, 2 digits	Elevation of satellite in degrees. Range: 00 – 90.
SatxAzim	Decimal, 3 digits	Azimuth of satellite in degrees, with true north as the reference plane. Range: 000 – 359.

SatxCN0	Decimal, 2 digits	Carrier to noise ratio for satellite x in dB. Range: 00 – 99.
SignalID	Decimal	Signal ID
*	-	End character of data field
Checksum	Hexadecimal	Hexadecimal checksum
<CR><LF>	Character	Each NMEA message ends with "CR" and "LF"

Example:

```
$GPGSV,1,1,3,2,39,296,42,5,22,219,45,12,24,310,43,1*6A
$GPGSV,1,1,3,2,39,296,,5,22,219,,12,24,310,,*5B
$GAGSV,1,1,3,201,30,280,44,221,29,220,44,226,47,317,37,7*44
$GAGSV,1,1,3,201,30,280,23,221,29,220,29,226,47,317,30,1*4F
```

2.3. PQTM Messages

The PQTM(Proprietary Quectel Message) messages are defined by Quectel.

2.3.1. PQTMCOLD

Perform a cold start.

Type: Command

Synopsis:

```
$PQTMCOLD*<checksum><CR><LF>
```

Argument:

None.

Result:

Cold start initialization and GNSS engine restart.

Example:

```
$PQTMCOLD*1C
```

2.3.2. PQTMWARM

Perform a warm start.

Type: Command

Synopsis:

```
$PQTMWARM*<checksum><CR><LF>
```

Argument:

None.

Result:

Warm start initialization and GNSS engine restart.

Example:

```
$PQTMWARM*11
```

2.3.3. PQTMHOT

Perform a hot start.

Type: Command

Synopsis:

```
$PQTMHOT*<checksum><CR><LF>
```

Argument:

None.

Result:

Hot start initialization and GNSS engine restart.

Example:

```
$PQTMHOT*4B
```

2.3.4. PQTMSRR

Execute a system reset. The GNSS firmware will be rebooted.

Type: Command

Synopsis:

```
$PQTMSRR*<checksum><CR><LF>
```

Argument:

None.

Result:

The GNSS firmware will be rebooted and no message will be sent as a reply.

Example:

```
$PQTMSRR*4B
```

2.3.5. PQTMCFGPORT

Configure communication port.

Type: Command

Synopsis:

```
$PQTMCFGPORT,<RW>,<PortType>,<ProtocolType>[,<Par1>,<Par2>,…,<ParN>]*<checksum><CR><LF>
```

Argument:

Field	Format	Description
$	Character	Each NMEA message starts with "$"
RW	Decimal, 1 digit	0 = Read current configuration 1 = Configure

PortType	Decimal, 1 digit	0 = UART1 1 = UART2
ProtocolType	Decimal	Bit 0 = NMEA command input Bit 1 = NMEA message output Bit 2 = RTCM output Bit 3 = RTCM input Bit 4 = Debug output Please note that RTCM Output and Debug cannot be enabled on same port at the same time.
ParN	-	See below.
*	-	End character of data field
Checksum	Hexadecimal	Hexadecimal checksum
<CR><LF>	Character	Each NMEA message ends with "CR" and "LF"

2.3.5.1. UART Config (PortType = 0 or 1)

Synopsis:

```
$PQTMCFGPORT,<RW>,<PortType>,<ProtocolType>[,<BaudRate>]*<checksum><CR><LF>
```

Argument:

Field	Format	Description
BaudRate	Decimal	UART baud rate. Support below: ● 115200bps ● 230400bps ● 460800bps ● 921600bps

Example:

```
$PQTMCFGPORT,0,0*43
$PQTMPORT,0,0,3,460800*38
$PQTMCFGPORT,1,0,1,460800*79
$PQTMCFGPORTOK*47
```

2.3.6. PQTMGNSSSUSPEND

Suspend the GNSS engine.

Type: Command

Synopsis:

```
$PQTMGNSSSUSPEND*<Checksum><CR><LF>
```

Argument:

None.

Result:

The GNSS engine will be suspended.

Example:

```
$PQTMGNSSSUSPEND*5B
$PQTMGNSSSUSPENDOK*5F
```

2.3.7. PQTMGNSSRESUME

Resume the GNSS engine.

Type: Command

Synopsis:

```
$PQTMGNSSRESUME*<Checksum><CR><LF>
```

Argument:

None.

Result:

The GNSS engine will be resumed

Example:

```
$PQTMGNSSRESUME*08
$PQTMGNSSRESUMEOK*0C
```

2.3.8. PQTMGNSSRESTART

Restart the GNSS engine.

Type: Command

Synopsis:

```
$PQTMGNSSRESTART*<Checksum><CR><LF>
```

Argument:

None.

Result:

The GNSS engine will be restarted.

Example:

```
$PQTMGNSSRESTART*46
$PQTMRESTARTOK*4B
```

2.3.9. PQTMSETNMEAMSGMASK

Set the NMEA message mask.

Type: Command

Synopsis:

```
$PQTMSETNMEAMSGMASK,<MsgMask>*<Checksum><CR><LF>
```

Argument:

Field	Format	Description
$	Character	Each NMEA message starts with "$"
MsgMask	Hexadecimal	bit30: Vehicle message bit31: Sensor message
*	-	End character of data field
Checksum	Hexadecimal	Hexadecimal checksum

<CR><LF>	Character	Each NMEA message ends with "CR" and "LF"

Result:

Set the GNSS output messages mask.

Example:

```
$PQTMSETNMEAMSGMASK,0x80000000*7C
$PQTMSETNMEAMSGMASKOK*14
```

2.3.10. PQTMGETNMEAMSGMASK

Get the NMEA message mask.

Type: Command

Synopsis:

```
$PQTMGETNMEAMSGMASK*<Checksum><CR><LF>
```

Argument:

None.

Result:

Get the GNSS output NMEA messages mask.

Example:

```
$PQTMGETNMEAMSGMASK*04
$PQTMNMEAMSGMASK,0x40000000*32
```

2.3.11. PQTMCFGDRMODE

Configure the DR operation mode.

Type: Command

Synopsis:

```
$PQTMCFGDRMODE,<RW>,<SpeedMode>*<Checksum><CR><LF>
```

Argument:

Field	Format	Description
$	Character	Each NMEA message starts with "$"
RW	Decimal, 1 digit	0 = Read Configuration 1 = Write Configuration
SpeedMode	Decimal, 1 digit	0 = No speed input 1 = Get speed from CAN (Default) 2 = Get speed from UART 3 = Get speed from WheelTick
*	-	End character of data field
Checksum	Hexadecimal	Hexadecimal checksum
<CR><LF>	Character	Each NMEA message ends with "CR" and "LF"

Result:

The DR operate mode will be changed.

Example:

```
$PQTMCFGDRMODE,0*53
$PQTMDRMODE,0,1*0C
$PQTMCFGDRMODE,1,2*4C
$PQTMCFGDRMODEOK*4B
```

2.3.12. PQTMCFGFWD

Configure the vehicle forward/backward indicator, the module support gets forward/backward information from GPIO(FWD Pin) or CAN message.

Type: Command

Synopsis:

```
$PQTMCFGFWD,<RW>,<Mode>,<Invert>,<Pull>*<Checksum><CR><LF>
```

Argument:

Field	Format	Description
$	Character	Each NMEA message starts with "$"

RW	Decimal, 1 digit	0 = Read Configuration 1 = Write Configuration
Mode	Decimal, 1 digit	0 = None. 1 = GPIO Mode 2 = CAN Mode
Invert	Decimal, 1 digit	Only take effect in GPIO mode. 0 = Normal, Low level forward, High level backward 1 = Interval, Low level backward, High level forward
Pull	Decimal, 1 digit	Only take effect in GPIO mode. 0 = None pull 1 = Pull up 2 = Pull down
*	-	End character of data field
Checksum	Hexadecimal	Hexadecimal checksum
<CR><LF>	Character	Each NMEA message ends with "CR" and "LF"

Example:

```
$PQTMCFGFWD,0*13
$PQTMFWD,0,2,0,0*4F
$PQTMCFGFWD,1,1,0,0*0F
$PQTMCFGFWDOK*0B
```

2.3.13. PQTMCFGWHEELTICK

Configure the vehicle wheel tick.

Type: Command

Synopsis:

```
$PQTMCFGWHEELTICK,<RW>,<EdgeType>,<Pull>,<MPT>*<Checksum><CR><LF>
```

Argument:

Field	Format	Description
$	Character	Each NMEA message starts with "$"
RW	Decimal, 1 digit	0 = Read Configuration 1 = Write Configuration
EdgeType	Decimal, 1 digit	0 = Edge_Rising

		1 = Edge_Falling 2 = Edge_Rising_Falling
Pull	Decimal, 1 digit	Only take effect in GPIO mode. 0 = None pull 1 = Pull up 2 = Pull down
MPT	x.x, variable length field	Meter per tick. Range: 0 – 50.0. Unit: meter/tick
*	-	End character of data field
Checksum	Hexadecimal	Hexadecimal checksum
<CR><LF>	Character	Each NMEA message ends with "CR" and "LF"

Example:

```
$PQTMCFGWHEELTICK,0*00
$PQTMWHEELTICK,0,0,0,0.036000*75
$PQTMCFGWHEELTICK,1,0,0,0.036*06
$PQTMCFGWHEELTICKOK*18
```

2.3.14. PQTMCFGCAN

Configure the CAN port.

Type: Command

Synopsis:

```
$PQTMCFGCAN,<RW>,<PortID>,<Enable>,<FrameFormat>,<Baudrate>,<DataBaudrate>*<Checksum><CR><LF>
```

Argument:

Field	Format	Description
$	Character	Each NMEA message starts with "$"
RW	Decimal, 1 digit	0 = Read Configuration 1 = Write Configuration
PortID	Decimal, 1 digit	0 = CAN1
Enable	Decimal, 1 digit	0 = Disable 1 = Enable
FrameFormat	Decimal, 1 digit	0 = Classic CAN 1 = CANFD no BRS

		2 = CANFD with BRS
Baudrate	Decimal	Nominal baud rate
DataBaudrate	Decimal	Data baud rate
*	-	End character of data field
Checksum	Hexadecimal	Hexadecimal checksum
<CR><LF>	Character	Each NMEA message ends with "CR" and "LF"

Example:

```
$PQTMCFGCAN,0,0*16
$PQTMCAN,0,0,1,0,500000,0*60
$PQTMCFGCAN,1,0,1,0,500000,0*23
$PQTMCFGCANOK*12
```

2.3.15. PQTMCFGCANFILTER

Configure the CAN filter.

Type: Command

Synopsis:

```
$PQTMCFGCANFILTER,<RW>,<PortID>,<Index>,<Enable>,<FilterType>,<ID_Type>,<ID1>,<ID2>*<Checksum><CR><LF>
```

Argument:

Field	Format	Description
$	Character	Each NMEA message starts with "$"
RW	Decimal, 1 digit	0 = Read Configuration 1 = Write Configuration
PortID	Decimal, 1 digit	0 = CAN1
Index	Decimal, 1 digit	0 = Filter0 1 = Filter1
Enable	Decimal, 1 digit	0 = Disable 1 = Enable
FilterType	Decimal, 1 digit	0 = Range mode 1 = Dual mode 2 = Mask mode

ID_Type	Decimal, 1 digit	0 = Standard ID 1 = Extended ID
ID1	Decimal	CAN ID
ID2	Decimal	CAN ID
*	-	End character of data field
Checksum	Hexadecimal	Hexadecimal checksum
<CR><LF>	Character	Each NMEA message ends with "CR" and "LF"

Example:

```
$PQTMCFGCANFILTER,0,0,0*0A
$PQTMCANFILTER,0,0,0,1,1,0,0x3E9,0x1F5*59
$PQTMCFGCANFILTER,1,0,0,1,1,0,0x3E9,0x1F5*1A
$PQTMCFGCANFILTEROK*12
```

2.3.16. PQTMCFGVEHDBC

Configure the vehicle DBC.

Type: Command

Synopsis:

```
$PQTMCFGVEHDBC,<RW>,<Index>,<MsgID>,<StartBit>,<BitSize>,<ByteOrder>,<ValueType>,<Factor>,<Offset>,<Min>,<Max>*<Checksum><CR><LF>
```

Argument:

Field	Format	Description
$	Character	Each NMEA message starts with "$"
RW	Decimal, 1 digit	0 = Read Configuration 1 = Write Configuration
Index	Decimal, 1 digit	0 = Vehicle speed 1 = Vehicle gear
MsgID	Decimal	CAN message ID
StartBit	Decimal	The start bit of data
BitSize	Decimal	The bit size of data

ByteOrder	Decimal, 1 digit	0 = Motorola 1 = Intel
ValueType	Decimal, 1 digit	Not support, always 0.
Factor	Double	The factor of the value
Offset	Double	The offset of the value
Min	Double	The minimal value.
Max	Double	The maximum value.
*	-	End character of data field
Checksum	Hexadecimal	Hexadecimal checksum
<CR><LF>	Character	Each NMEA message ends with "CR" and "LF"

Example:

```
$PQTMCFGVEHDBC,0,0*44
$PQTMVEHDBC,0,0,0x3E9,8,16,0,0,0.015625,0.000000,0.000000,300.000000*14
$PQTMCFGVEHDBC,1,0,0x3E9,8,16,0,0,0.015625,0.000000,0.000000,300.000000*57
$PQTMCFGVEHDBCOK*40
```

2.3.17. PQTMSETVEHRVAL

Configure the vehicle reverse gear value when get forward/backward information from CAN.

Type: Command

Synopsis:

```
$PQTMSETVEHRVAL,<RVal>*<Checksum><CR><LF>
```

Argument:

Field	Format	Description
$	Character	Each NMEA message starts with "$"
RVal	Decimal	Vehicle reverse gear value. Range: 0 – 255
*	-	End character of data field
Checksum	Hexadecimal	Hexadecimal checksum

<CR><LF>	Character	Each NMEA message ends with "CR" and "LF"

Example:

```
$PQTMSETVEHRVAL,2*16
$PQTMSETVEHRVALOK*0C
```

2.3.18. PQTMGETVEHRVAL

Get the vehicle reverse gear value when get forward/backward information from CAN.

Type: Command

Synopsis:

```
$PQTMGETVEHRVAL*<Checksum><CR><LF>
```

Argument:

None

Example:

```
$PQTMGETVEHRVAL*1C
$PQTMVEHRVAL,2*54
```

2.3.19. PQTMVEHMSG

Output/Input vehicle message.

Type: Input/Output

Synopsis:

```
$PQTMVEHMSG,<MsgID>,<TimeStamp>,<Par1>[,<Par2>]*<Checksum><CR><LF>
```

Argument:

Field	Format	Description
$	Character	Each NMEA message starts with "$"
MsgID	Decimal	1 = Vehicle speed

TimeStamp	Decimal	Always be 0 if input speed information
ParN	Decimal	See below.
*	-	End character of data field
Checksum	Hexadecimal	Hexadecimal checksum
<CR><LF>	Character	Each NMEA message ends with "CR" and "LF"

2.3.19.1. Vehicle Speed Message (MsgID = 1)

Type: Input/Output

Synopsis:

```
$PQTMVEHMSG,1,<TimeStamp>,<VehSpeed>*<Checksum><CR><LF>
```

Argument:

Field	Format	Description
$	Character	Each NMEA message starts with "$"
TimeStamp	Decimal	Timestamp since power on. Always be 0 if input speed information
VehSpeed	Decimal	Speed. Range: -100 ~ 100. Unit: m/s
*	-	End character of data field
Checksum	Hexadecimal	Hexadecimal checksum
<CR><LF>	Character	Each NMEA message ends with "CR" and "LF"

Example:

```
$PQTMVEHMSG,1,0,3.6*1C
$PQTMVEHMSG,1,3748292,3.600000*2D
```

2.3.19.2. Vehicle Wheel Tick Message (MsgID = 2)

Type: Output

Synopsis:

```
$PQTMVEHMSG,2,<TimeStamp>,<WheelTickCount>,<FWD_Ind>*<Checksum><CR><LF>
```

Argument:

Field	Format	Description
$	Character	Each NMEA message starts with "$"
TimeStamp	Decimal	Timestamp since power on.
WheelTickCount	Decimal	Tick count.
FWD_Ind	Decimal	Forward/backward indicator. 0 = Invalid state. 1 = Forward. 2 = Backward.
*	-	End character of data field
Checksum	Hexadecimal	Hexadecimal checksum

Example:

```
$PQTMVEHMSG,2,3422954,0,1*16
```

2.3.20. PQTMSENMSG

Output sensor message.

Type: Output

Synopsis:

```
$PQTMSENMSG,<MsgID>,<TimeStamp>,<Par1>[,<Par2>,…,<ParN>]*<Checksum><CR><LF>
```

Argument:

Field	Format	Description
$	Character	Each NMEA message starts with "$"
MsgID	Decimal	1 = IMU sensor
TimeStamp	Decimal	Timestamp since power on. Always be 0 if input speed information
ParN	Decimal	See below

*	-	End character of data field
Checksum	Hexadecimal	Hexadecimal checksum
<CR><LF>	Character	Each NMEA message ends with "CR" and "LF"

2.3.20.1. IMU sensor message (MsgID = 1)

Synopsis:

```
$PQTMSENMSG,1,<TimeStamp>,<IMU_Temp>,<IMU_GYRO_X>,<IMU_GYRO_Y>,<IMU_GYRO_Z>,
<IMU_ACC_X>,<IMU_ACC_Y>,<IMU_ACC_Z>*<Checksum><CR><LF>
```

Argument:

Field	Format	Description
$	Character	Each NMEA message starts with "$"
TimeStamp	Decimal	Timestamp since power on.
IMU_Temp	Decimal	IMU temperature
IMU_GYRO_X	Decimal	IMU X-axis raw gyro value
IMU_GYRO_Y	Decimal	IMU Y-axis raw gyro value
IMU_GYRO_Z	Decimal	IMU Z-axis raw gyro value
IMU_ACC_X	Decimal	IMU X-axis raw accelerometer value
IMU_ACC_Y	Decimal	IMU Y-axis raw accelerometer value
IMU_ACC_Z	Decimal	IMU Z-axis raw accelerometer value
*	-	End character of data field
Checksum	Hexadecimal	Hexadecimal checksum
<CR><LF>	Character	Each NMEA message ends with "CR" and "LF"

Example:

```
$PQTMSENMSG,1,3895496,3855,100,-100,-119,44,-214,8374*18
$PQTMSENMSG,1,3895506,3842,113,-111,-113,43,-229,8368*1A
$PQTMSENMSG,1,3895516,3842,94,-102,-111,50,-216,8373*21
$PQTMSENMSG,1,3895526,3851,80,-117,-122,44,-216,8376*21
```

2.3.21. PQTMSAVEPAR

Save configurations of GNSS commands into NVM.

Type: Command

Synopsis:

```
$PQTMSAVEPAR*<checksum><CR><LF>
```

Argument:

None.

Result:

- In case of no errors, the returned message will be:

  ```
  $PQTMSAVEPAROK*5E
  ```

- In case of any error, the returned message will be:

  ```
  $PQTMSAVEPARERROR*02
  ```

Example:

```
$PQTMSAVEPAR*5A
$PQTMSAVEPAROK*5E
```

2.3.22. PQTMRESTOREPAR

Restore all configurations to default values.

Type: Command

Synopsis:

```
$PQTMRESTOREPAR*<checksum><CR><LF>
```

Argument:

None.

Result:

- In case of no errors, the returned message will be:

```
$PQTMRESTOREPAROK*17
```

- In case of any error, the returned message will be:

```
$PQTMRESTOREPARERROR*4B
```

Example:

```
$PQTMRESTOREPAR*13
$PQTMRESTOREPAROK*17
```

3 Default Configurations

Table 2: Default Configurations

Item	Default Configuration
NMEA port baud rate	460800bps
Datum	WGS84
Rate of position fixing	1Hz
DGPS mode	OFF
NMEA output messages	RMC, GGA, GSV, GSA
GNSS Configuration	GPS + BeiDou + Galileo + QZSS

4 Appendix A Example

Example for Input Speed via UART:

```
// Set UART mode
$PQTMCFGDRMODE,1,2*4C
// Save the configuration
$PQTMSAVEPAR*5A
// Restart the module

// Input the speed info via UART, $PQTMVEHMSG,1,0,<Speed>
// For example, the speed it 5.6m/s
$PQTMVEHMSG,1,0,5.6*1A
```

Example for Wheel Tick Input:

```
// Set Wheel Tick mode
$PQTMCFGDRMODE,1,3*4D
// Configure the Wheel Tick
// For example: Rising edge, No pull, 0.036 meter/tick
$PQTMCFGWHEELTICK,1,0,0,0.036*06
// Configure forward/backward
// For example: GPIO mode, No pull and normal mode.
$PQTMCFGFWD,1,1,0,0*0F
// Save the configuration
$PQTMSAVEPAR*5A
// Restart the module

// Connect Wheel Tick pin to vehicle.
// If vehicle message is turn on, see below message output:
$PQTMVEHMSG,2,3422954,0,1*16
```

5 Appendix B References

Table 3: Related Document

SN	Document Name	Remark
[1]	Quectel_LG69T_Hardware_Design	LG69T Hardware Design

Table 4: Terms and Abbreviations

Abbreviation	Description
DGPS	Differential Global Positioning System
GGA	Global Positioning System Fix Data
GLONASS	Global Navigation Satellite System (Russian)
GLP	GNSS Low Power
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSA	GNSS DOP and Active Satellites
GSV	GNSS Satellites in View
HDOP	Horizontal Dilution of Precision
NAVIC (IRNSS)	NAVigation with Indian Constellation (Indian Regional Navigation Satellite System)
NMEA	National Marine Electronics Association
NVM	Non-Volatile Memory
PDOP	Position Dilution of Precision
PPS	Pulse Per Second
QZSS	Quasi-Zenith Satellite System
RMC	Recommended Minimum Specific GNSS Data

SBAS	Satellite-Based Augmentation System
UTC	Universal Time Coordinated
VDOP	Vertical Dilution of Precision
WGS84	World Geodetic System 1984