Commands List

Suitable for the following GNSS Series Product

Product Type	Model Name
GNSS Module	MC-1010-V2a、MC-1010-V3a MC-1612-V2b、MC-1612-V3b MC-1612a-V2b、MC-1612a-V3b
GNSS Smart Antenna	LC20030-V2、LC20031-V2、LC20032-V2 LC20030-V3、LC20031-V3、LC20032-V3
GNSS Mouse	LU23030-V2、LU23032-V2、LU23036-V2 LU23030-V3、LU23032-V3、LU23036-V3

Version：1.0

2024/5/27

LOCOSYS Technology Inc.
20F.-13, No.79, Sec. 1, Xintai 5th Rd.,
Xizhi District, New Taipei City 221, Taiwan
☏ 886-2-8698-3698
🖷 886-2-8698-3699
www.locosystech.com

LOCOSYS Technology Inc.
20F.-13, No.79, Sec. 1, Xintai 5th Rd.,
Xizhi Dist., New Taipei City 221, Taiwan
☏ 886-2-8698-3698
886-2-8698-3699
www.locosystech.com

PAIR Commands

PAIR command is an AIROHA proprietary GNSS data transfer protocol. This protocol is used to configure the GNSS module's parameters, aiding information and to receive notifications from the GNSS module. The PAIR aligns with the NMEA sentence format to process data more conveniently.

Command Format

The PAIR packet format with number of bytes allocated for each field is show in the table.

1	4	3	variable	1	1	1	1	1
Preamble	Tracker ID	PktType	DataField	*	CHK1	CHK2	CR	LF

The details of each field in the packet are provided in the table.

Parameter	**Description**
Preamble	One-byte character. - Default: '$'
TalkerID	Four-byte character string. - Default: PAIR. Customer can change this string for customization commands.
PktType	Three-byte character string. - From 000 to 999. An identifier used to tell the decoder how to decode the packet.
DataField	Has a variable length depending on the packet type. - A comma symbol ',' must be inserted before each data field to help the decoder process the DataField.
*	One-byte character. - The asterisk symbol is used to mark the end of the DataField.
CHK1, CHK2	One-byte character for each. - CHK1 and CHK2 are the checksum of the data between Preamble and '*'. Please use gnss_app_get_command_checksum or reference **Appendix - How to get checksum**.
CR, LF	One byte ASCII data for each. - The two bytes are used to identify the end of a packet.

LOCOSYS

LOCOSYS Technology Inc.
20F.-13, No.79, Sec. 1, Xintai 5th Rd.,
Xizhi Dist., New Taipei City 221, Taiwan
☏ 886-2-8698-3698
886-2-8698-3699
www.locosystech.com

Command List

The PAIR command list for each packet type is given in the table.

Note:

1. All Commands will respond with PAIR001 (**PAIR_ACK**) as the default result. You can check the command result from PAIR001 (**PAIR_ACK**) parameter.
2. All command strings shown do not include spaces.

Packet Type	Command	Description
001	PAIR_ACK	Acknowledge of PAIR command.
002	PAIR_GNSS_SUBSYS_POWER_ON	Power on the GNSS system. Include DSP/RF/Clock and other GNSS modules.
003	PAIR_GNSS_SUBSYS_POWER_OFF	Power off GNSS system. Include DSP/RF/Clock and other GNSS modules.
004	PAIR_GNSS_SUBSYS_HOT_START	Hot Start. Use the available data in the NVRAM.
005	PAIR_GNSS_SUBSYS_WARM_START	Warm Start. Not using Ephemeris data at the start.
006	PAIR_GNSS_SUBSYS_COLD_START	Cold Start. Not using the Position, Almanac or Ephemeris data at the start.
008	PAIR_GNSS_SUBSYS_GET_STATUS	Get GNSS subsys status.
010	PAIR_REQUEST_AIDING	Request GNSS system reference data.
011	PAIR_INDICATION_SYSTEM_MESSAGE	GNSS System message indication.
020	PAIR_GET_VERSION	Query the firmware release information.
021	PAIR_GET_SETTING_INFO	Query the customer related setting, such as the firmware release information, DCB values, HW interface, ULP enable and NVRAM auto saving.
023	PAIR_SYSTEM_REBOOT	Reboot GNSS whole chip, including the GNSS submodule and other all CM4 modules.
030	PAIR_COMMON_GET_POS_XYZ	The WGS84 ECEF XYZ Cartesian Position vector (in meters) with an estimated 1-sigma accuracy.
031	PAIR_COMMON_GET_VEL_XYZ	The WGS84 ECEF XYZ Cartesian velocity vector (m/s) with an estimated 1-sigma accuracy.
043	PAIR_COMMON_GET_TOW_WN	Get TOW (Time of week) and WN (Week number) information.
050	PAIR_COMMON_SET_FIX_RATE	Set Position Fix Interval.

051	PAIR_COMMON_GET_FIX_RATE	Get Position Fix Interval.
058	PAIR_COMMON_SET_MIN_SNR	Set the minimum SNR of used satellites.
059	PAIR_COMMON_GET_MIN_SNR	Query the minimum SNR of used satellites.
060	PAIR_COMMON_SET_ESTIMATED_NUM	Set the number of estimated fixes when entering the tunnel.
061	PAIR_COMMON_GET_ESTIMATED_NUM	Query the number of estimated fixes when entering the tunnel.
062	PAIR_COMMON_SET_NMEA_OUTPUT_RATE	Set the NMEA sentence output interval of corresponding NMEA type.
063	PAIR_COMMON_GET_NMEA_OUTPUT_RATE	Get the NMEA sentence output interval of corresponding NMEA type.
064	PAIR_COMMON_SET_HACC_LIMIT	Set horizontal accuracy mask. Range from 30m to 200m or 0. GPS only gets the fix when hacc value < mask.
065	PAIR_COMMON_GET_HACC_LIMIT	Query horizontal accuracy mask.
066	PAIR_COMMON_SET_GNSS_SEARCH_MODE	Configure the receiver to start searching for satellites. The setting is available when the NVRAM data is valid.
067	PAIR_COMMON_GET_GNSS_SEARCH_MODE	This command is to get GPS, GLONASS, Galileo, BeiDou, QZSS and NavIC search settings.
068	PAIR_COMMON_SET_HDOP_THRESHOLD	This command is for setting the HDOP threshold.
069	PAIR_COMMON_GET_HDOP_THRESHOLD	This command is to get the HDOP threshold.
070	PAIR_COMMON_SET_STATIC_THRESHOLD	Set the speed threshold for static navigation.
071	PAIR_COMMON_GET_STATIC_THRESHOLD	Query the static navigation speed threshold.
072	PAIR_COMMON_SET_ELEV_MASK	Set satellite elevation mask.
073	PAIR_COMMON_GET_ELEV_MASK	Get satellite elevation mask.
074	PAIR_COMMON_SET_AIC_ENABLE	Enable or disable active interference cancellation function.
075	PAIR_COMMON_GET_AIC_STATUS	Get the status of active interference cancellation function.
076	PAIR_COMMON_SET_DATUM	Set default datum.
077	PAIR_COMMON_GET_DATUM	Get default datum.
080	PAIR_COMMON_SET_NAVIGATION_MODE	Set navigation mode.
081	PAIR_COMMON_GET_NAVIGATION_MODE	Get navigation mode.
086	PAIR_COMMON_SET_DEBUGLOG_OUTPUT	This command is to set enable/disable debug log output in binary format

087	PAIR_COMMON_GET_DEBUGLOG_OUTPUT	Query setting of debug log output.
092	PAIR_COMMON_SET_STATIC_MODE	Enable static mode
093	PAIR_COMMON_GET_STATIC_MODE	Query if current mode is static mode
098	PAIR_COMMON_SET_NMEA_POS_DECIMAL_PRECISION	This command is for setting the digits shown in the NMEA position.
099	PAIR_COMMON_GET_NMEA_POS_DECIMAL_PRECISION	This command is to get NMEA position shown digit mode
100	PAIR_COMMON_SET_NMEA_OUTPUT_MODE	This command is to set NMEA output mode
101	PAIR_COMMON_GET_NMEA_OUTPUT_MODE	This command is to get NMEA output mode
104	PAIR_COMMON_SET_DUAL_BAND	This command is to set Dual Band state when GNSS service is powered off.
105	PAIR_COMMON_GET_DUAL_BAND	Query whether Dual Band is enabled or disabled.
130	PAIR_COMMON_SET_SV_BLACKLIST	Set sv blacklist for the selected constellation, which excludes specific PRNs and does not search for them.
131	PAIR_COMMON_GET_SV_BLACKLIST	Get sv blacklist for the selected constellation, which excludes specific PRNs and does not search for them.
156	PAIR_COMMON_SET_VERTICAL_CONSTRAINT	Enable vertical constraint for smoother height estimate. (Fitness default is enable)
157	PAIR_COMMON_GET_VERTICAL_CONSTRAINT	Query if vertical constraint is enabled.
160	PAIR_COMMON_SET_IMMEDIATE_SPEED_MODE	Enable immediate speed mode.
161	PAIR_COMMON_GET_IMMEDIATE_SPEED_MODE	Query if immediate speed mode is enabled.
162	PAIR_COMMON_SET_2D_FIX_ENABLE	Enable 2D fix mode.
163	PAIR_COMMON_GET_2D_FIX_ENABLE	Query if 2D fix mode is enabled.
164	PAIR_COMMON_SET_ODOMETER_ENABLE	Enable the odometer function.
165	PAIR_COMMON_GET_ODOMETER_STATUS	Query the status of the odometer function and the traveling distance value if the function is enabled.
400	PAIR_DGPS_SET_MODE	DGPS correction data source mode.
401	PAIR_DGPS_GET_MODE	Query the DGPS data source mode.
410	PAIR_SBAS_ENABLE	Enable searching a SBAS satellite or not.
411	PAIR_SBAS_GET_STATUS	Query the status of SBAS to whether it is enabled.
412	PAIR_SBAS_GET_SAT_INFO	Get information about the SBAS satellites, such as

		SVid, SNR, azimuth, and elevation.
416	PAIR_SBAS_SET_RAW_DATA_OUTPUT	Enable output SBAS raw data in the way of binary proprietary message.
417	PAIR_SBAS_GET_RAW_DATA_OUTPUT	Query the status of SBAS raw data output to check whether it is enabled.
420	PAIR_SLAS_ENABLE	Enable the QZSS SLAS (Sub-meter Level Augmentation Service) operation.
421	PAIR_SLAS_GET_STATUS	Query the status of SLAS to check whether it is enabled.
430	PAIR_RTCM_SET_INPUT_VERSION	Set RTCM input version.
431	PAIR_RTCM_GET_INPUT_VERSION	Get RTCM input version.
432	PAIR_RTCM_SET_OUTPUT_MODE	Set RTCM output mode.
433	PAIR_RTCM_GET_OUTPUT_MODE	Get RTCM output mode.
434	PAIR_RTCM_SET_OUTPUT_ANT_PNT	This command is to set enable/disable stationary antenna reference point for RTCM output.
435	PAIR_RTCM_GET_OUTPUT_ANT_PNT	Query setting of stationary antenna reference point for RTCM output.
436	PAIR_RTCM_SET_OUTPUT_EPHEMERIS	This command is to set enable/disable RTCM output with satellite ephemeris.
437	PAIR_RTCM_GET_OUTPUT_EPHEMERIS	Query setting of RTCM satellite ephemeris output.
496	PAIR_EPOC_ENABLE	Enable or Disable EPOC
507	PAIR_EPOC_CLEAR_DATA	Clear All EPOC data in NVRAM
508	PAIR_EPOC_GET_STATUS	Get EPOC status
513	PAIR_NVRAM_SAVE_SETTING	Save the current configuration from RTC RAM to flash.
514	PAIR_NVRAM_RESTORE_DEFAULT_SETTING	Clear the current configuration and restore the default settings.
596	PAIR_TIME_GET_CURRENT_TOW	Get current time of week in milliseconds.
597	PAIR_TIME_GET_GNSS_TOW	Get the last GNSS epoch's time of week in milliseconds.
650	PAIR_LOW_POWER_ENTRY_RTC_MODE	Shutdown all systems, including GNSS and other CM4 modules.
680	PAIR_GLP_ENABLE	This command is to activate low-power GLP mode.
681	PAIR_GLP_GET_STATUS	This command is to get low-power GLP mode setting
690	PAIR_PERIODIC_SET_MODE	This command is used to set Periodic Power Saving Mode Settings.

691	PAIR_PERIODIC_GET_MODE	This command is used to get Periodic Power Saving Mode Settings.
700	PAIR_ULP_ENABLE	This command is to set to enable Ultra Low Power.
701	PAIR_ULP_GET_STATUS	Query whether ULP is enabled or disabled.
710	PAIR_ADP_L5_ENABLE	Enable or disable adaptive L5 when GNSS system is powered off
711	PAIR_ADP_L5_GET_STATUS	Query the status of adaptive L5 to check whether it is enabled
720	PAIR_DTM_ENABLE	The GNSS Duty tracking mode is a duty cycle concept to ensure good performance
721	PAIR_DTM_GET_STATUS	This command is to get Duty tracking mode configuration setting.
730	PAIR_FLP_ENABLE	The fitness low power mode is a concept to ensure good performance
731	PAIR_FLP_GET_STATUS	This command is to get fitness low-power mode configuration setting.
732	PAIR_ALP_ENABLE	The adaptive low power mode is a concept to ensure good performance
733	PAIR_ALP_GET_STATUS	This command is to get adaptive low-power mode configuration setting.
752	PAIR_PPS_SET_CONFIG_CMD	Configure the PPS settings.
830	PAIR_RAW_ENABLE	Set enable/disable output binary raw measurement
831	PAIR_RAW_GET_STATUS	Get enable/disable output binary raw measurement
864	PAIR_IO_SET_BAUDRATE	Set port baud rate configuration.
865	PAIR_IO_GET_BAUDRATE	Get port baud rate configuration.
866	PAIR_IO_SET_FLOW_CONTROL	Set port flow control configuration.
867	PAIR_IO_GET_FLOW_CONTROL	Get port SW flow control configuration.
890	PAIR_GEOFENCE_SET_CONFIG	This command is used to set Geofencing configuration.
891	PAIR_GEOFENCE_GET_CONFIG	This command is used to get Geofencing configuration.
892	PAIR_GEOFENCE_SET_GPIO_POLARITY	This command is used to set GPIO polarity for geofencing combined state.
900	PAIR_LOCUS_ENABLE	Enable or disable LOCUS save data.
901	PAIR_LOCUS_GET_STATUS	Get LOCUS status.
902	PAIR_LOCUS_SET_MODE	Set LOCUS saving mode.

903	PAIR_LOCUS_GET_MODE	Get LOCUS saving mode.
904	PAIR_LOCUS_SET_THRESHOLD	Set LOCUS mode threshold.
905	PAIR_LOCUS_GET_THRESHOLD	Get LOCUS mode threshold.
906	PAIR_LOCUS_CLEAR	Clear LOCUS Data.
907	PAIR_LOCUS_LOG_NOW	Save current location data.
908	PAIR_LOCUS_GET_DATA	Get all record data.
909	PAIR_LOCUS_GET_RECORD_NUM	Get total record number.

PAIR_ACK	**[Packet Type]** 001 **[Command]** PAIR_ACK **[Description]** Acknowledge of PAIR command. **[Return]** **$PAIR001,Command_ID,Result*CS<CR><LF>** Command_ID: The command / packet type the acknowledge responds. Result: The result of the command. 0: The command was successfully sent. 1: The command is processing. You must wait for the result. 2: Sending the command failed. 3: This command ID is not supported. 4: Command parameter error. Out of range / some parameters were lost / checksum error. 5: MNL service is busy. You can try again soon. **[Example]** Send: $PAIR666*3C\r\n Response: $PAIR001,666,0*3D\r\n ==> Success **[Note]** This item is the response of commands. The GNSS system automatically sends this command. Do not directly send it to the GNSS system.
PAIR_GNSS_SUBSYS_POWER_ON	**[Packet Type]** 002 **[Command]** PAIR_GNSS_SUBSYS_POWER_ON **[Description]** Power on the GNSS system. Include DSP/RF/Clock and other GNSS modules. **[Data Field]** **$PAIR002*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR002*38\r\n Response:

	$PAIR001,002,1*38\r\n ==> The power on process is running. Please wait a moment. $PAIR001,002,0*39\r\n ==> Power on was successful. **[Note]** Please send this command before using any location service.
PAIR_GNSS_SUBSYS_POWER_OFF	**[Packet Type]** 003 **[Command]** PAIR_GNSS_SUBSYS_POWER_OFF **[Description]** Power off GNSS system. Include DSP/RF/Clock and other GNSS modules. CM4 also can receive commands (Include the AT command / the race Command / the part of PAIR command which is not dependent on DSP.) after sending this command. **[Data Field]** **$PAIR003*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR003*39\r\n Response: $PAIR001,003,1*39\r\n ==> The power off process is running. Please wait a moment. $PAIR001,003,0*38\r\n ==> Power off was successful. **[Note]** The location service is not available after this command is executed. The system can still receive configuration PAIR commands. The application is running if necessary. CM4 will go to sleep if the application is not working at this time. The system can be awoken by the GNSS_DATA_IN_EINT pin after going to sleep.
PAIR_GNSS_SUBSYS_HOT_START	**[Packet Type]** 004 **[Command]** PAIR_GNSS_SUBSYS_HOT_START **[Note]** This command will get a failed response if GNSS-SubSys does not power on. You can get the GNSS-SubSys status by **PAIR_GNSS_SUBSYS_GET_STATUS**. **[Description]** Hot Start. Use the available data in the NVRAM. **[Data Field]** **$PAIR004*CS<CR><LF>**

	[Return] 1. **PAIR_ACK** for send result.**[Example]** Send: $PAIR004*3E\r\n Response: $PAIR001,004,0*3F\r\n ==> Success
PAIR_GNSS_SUBSYS_WARM_START	**[Packet Type]** 005 **[Command]** PAIR_GNSS_SUBSYS_WARM_START **[Note]** This command will get a failed response if GNSS-SubSys does not power on. You can get the GNSS-SubSys status by **PAIR_GNSS_SUBSYS_GET_STATUS**. **[Description]** Warm Start. Not using Ephemeris data at the start. **[Data Field]** **$PAIR005*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR005*3F\r\n Response: $PAIR001,005,0*3E\r\n ==> Success
PAIR_GNSS_SUBSYS_COLD_START	**[Packet Type]** 006 **[Command]** PAIR_GNSS_SUBSYS_COLD_START **[Note]** This command will get a failed response if GNSS-SubSys does not power on. You can get the GNSS-SubSys status by **PAIR_GNSS_SUBSYS_GET_STATUS**. **[Description]** Cold Start. Not using the Position, Almanac or Ephemeris data at the start. **[Data Field]** **$PAIR006*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result.

	[Example] Send: $PAIR006*3C\r\n Response: $PAIR001,006,0*3D\r\n ==> Success
PAIR_GNSS_SUBSYS_GET_STATUS	**[Packet Type]** 008 **[Command]** PAIR_GNSS_SUBSYS_GET_STATUS **[Description]** Get GNSS subsys status. **[Data Field]** **$PAIR008*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR008,<status>*CS<CR><LF>** status: the status of GNSS subsys. 0: The GNSS subsys has powered on. 1: The GNSS subsys is powered off. **[Example]** Send: $PAIR008*32\r\n Response: $PAIR001,008,0*33\r\n ==> Success $PAIR008,1,0*33\r\n ==> GNSS subsys has powered on.
PAIR_REQUEST_AIDING	**[Packet Type]** 010 **[Command]** PAIR_REQUEST_AIDING **[Description]** Request GNSS system reference data. **[Return]** **$PAIR010,<Type>,<GNSS_System>,<Week_Number>,<Time_of_Week>*CS<CR><LF>** Type: The data type. 0: Need to update EPO data. 1: Need to update the time. 2: Need to update the location.

	GNSS_System: The GNSS system type is needed. 0: Need GPS data. 1: Need GLONASS data. 2: Need GALILEO data. 3: Need BEIDOU data. 4: Need QZSS data. Week_Number: The current GNSS week number. Time_of_Week: The current GNSS time of week. **[Example]** Response: $PAIR010,0,0,2044,369413*33\r\n ==> Please send GPS EPO data when this command is received. $PAIR010,1,-1*16\r\n ==> Please send reference time when this command is received. $PAIR010,2,-1*15\r\n ==> Please send reference location when this command is received. **[Note]** The GNSS system automatically sends this command. Please do not actively send it to the GNSS system.
PAIR_INDICATION_SYSTEM_MESSAGE	**[Packet Type]** 011 **[Command]** PAIR_INDICATION_SYSTEM_MESSAGE **[Description]** GNSS System message indication. **[Data Field]** **$PAIR011,<Type>*CS<CR><LF>** Type: "1", Notification for GNSS system startup **[Return]** NONE **[Example]** $PAIR011,001*27 **[Note]** The GNSS system automatically sends this command. Please do not actively send it to the GNSS system.
PAIR_GET_VERSION	**[Packet Type]** 020 **[Command]** PAIR_GET_VERSION **[Description]**

Query the firmware release information.

[Data Field]

$PAIR020*CS<CR><LF>

[Return]

1. **PAIR_ACK** for send result.
2. **$PAIR020,<Project Version>,<Frequency>,<SW package>,<Service version>,<Service build time>,<DSP L1 rom version>,<DSP L1 ram version>,<DSP L5 rom version>,<DSP L5 ram version>,<Kernel version>,<Kernel build time>,<KF version>,<KF build time>, <RTK version>,<RTK build time>,<Fusion version>,<Fusion build time>*CS<CR><LF>**

Project Version:

<Project_board>_<SDK version>_<SDK Build time>

<Project_board> AG3335A / AG3335M / AG3335S / AG3353

<SDK version> VX.Y.Z - X:Major Y:Minor Z. Bug fix

<SDK build time> YYYYMMDD

Note: Max length of this filed is 100 Bytes

Ex:

AG3335A_V1.0.0_20190729

Frequency:

S: single

D: dual

SW package:

N: normal

W: raw

R: RTK

I: NavIC

F: Fusion

Service version:

mnl_service version in 7 characters

Ex:

xxxxxxx

Service build time:

mnl_service library build time

Ex:

yyMMDDhhmm

DSP L1 rom version:

Null before first power on

Ex:

xx

DSP L1 ram version:

Null before first power on

	Ex: xxx DSP L5 rom version: Null for L1 only project Null before first power on Ex: xx DSP L5 ram version: Null for L1 only project Null before first power on Ex: xxx Kernel version: mnl_kernel version in 7 characters Ex: xxxxxxx Kernel build time: mnl_kernel library build time Ex: yyMMDDhhmm KF version: mnl_kf version in 7 characters Ex: xxxxxxx KF build time: mnl_kf library build time Ex: yyMMDDhhmm RTK version: RTK version in 7 characters anything other than the RTK project Ex: xxxxxxx RTK build time: RTK library build time Null for not RTK project Ex: yyMMDDhhmm Fusion version: Fusion version in 7 characters

	Ex: xxxxxxx Fusion build time: Fusion library build time Null for not fusion project Ex: yyMMDDhhmm **[Example]** Send: $PAIR020*38\r\n Response: $PAIR001,020,0*39\r\n ==> Success $PAIR020,AG3335A_V1.0.0_YYYYMMDD,D,N,xxxxxxx,yyMMDDhhmm,xx,xxx,xx,xxx,xxx xxxx,yyMMDDhhmm,xxxxxxx,yyMMDDhhmm,,*40\r\n
PAIR_GET_SETTING_INFO	**[Packet Type]** 021 **[Command]** PAIR_GET_SETTING_INFO **[Description]** Query the customer related setting, such as the firmware release information, DCB values, HW interface, ULP enable and NVRAM auto saving. **[Data Field]** **$PAIR021*CS<CR><LF>[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR021,<Project Version>,<Frequency>,<SW package>,<Service version>,<Service build time>,<DSP L1 rom version>,<DSP L1 ram version>,<DSP L5 rom version>,<DSP L5 ram version>,<Kernel version>,<Kernel build time>,<KF version>,<KF build time>,<RTK version>,<RTK build time>,<Fusion version>,<Fusion build time>,<GPS DCB>,<GAL DCB>,<BDS DCB>,<QZS DCB>,<TCXO Freq Error>,<Gain>,<SWPRT Info>,<ULP enable>,<NVRAM Auto Saving>*CS<CR><LF>** Refer to PAIR020 (Project Version,Frequency,...,RTK build time) GPS DCB : The Differential Code Biases value for GPS. GAL DCB : The Differential Code Biases value for GAL. BDS DCB : The Differential Code Biases value for BDS. QZS DCB : The Differential Code Biases value for QZS. TCXO Frequency error : '0' 0.5ppm '1' 1.0ppm

	'2' 1.5ppm '3' 2.0ppm Gain : '0' High gain '1' Low gain SWPRT Check : ##: No Check #P: Pass #F: Fail ULP enable : '0' disable '1' enable NVRAM Auto Saving : '0' Disable '1' Enable **[Example]** Send: $PAIR021*39\r\n Response: $PAIR001,021,0*38\r\n $PAIR021,AG3335A_V1.0.0_YYYYMMDD,D,N,xxxxxxx,yyMMDDhhmm,xx,xxx,xx,xxx,xxxxxxx,yyMMDDhhmm,xxxxxxx,yyMMDDhhmm,,-9.25,-9.66,-8.08,-9.61,1,1,#P,0,0*3B\r\n
PAIR_SYSTEM_REBOOT	**[Packet Type]** 023 **[Command]** PAIR_SYSTEM_REBOOT **[Description]** Reboot GNSS whole chip, including the GNSS submodule and other all CM4 modules. **[Data Field]** **$PAIR023*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR023*3B\r\n Response: Reboot directly. Without Response.
PAIR_COMMON_GE	**[Packet Type]**

T_POS_XYZ	030 **[Command]** PAIR_COMMON_GET_POS_XYZ **[Note]** This command will get a failed response if GNSS-SubSys does not power on. You can get the GNSS-SubSys status by **PAIR_GNSS_SUBSYS_GET_STATUS**. **[Description]** The WGS84 ECEF XYZ Cartesian Position vector (in meters) with an estimated 1-sigma accuracy. **[Data Field]** **$PAIR030*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR030,<X>,<Y>,<Z>,<Acc>*CS<CR><LF>** X: WGS84 ECEF X Cartesian position (in meters). Y: WGS84 ECEF Y Cartesian position (in meters). Z: WGS84 ECEF Z Cartesian position (in meters). Acc: 3-dimensional position space 1-sigma accuracy estimate (in meters). **[Example]** Send: $PAIR030*39\r\n Response: $PAIR001,030,0*38\r\n ==> Success $PAIR030,-2984524.0,4966958.3,2656485.3,3.0*14\r\n ==> The WGS84 ECEF XYZ Cartesian Position.
PAIR_COMMON_GET_VEL_XYZ	**[Packet Type]** 031 **[Command]** PAIR_COMMON_GET_VEL_XYZ **[Note]** This command will get a failed response if GNSS-SubSys does not power on. You can get the GNSS-SubSys status by **PAIR_GNSS_SUBSYS_GET_STATUS**. **[Description]** The WGS84 ECEF XYZ Cartesian velocity vector (m/s) with an estimated 1-sigma accuracy. **[Data Field]** **$PAIR031*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR031,<VX>,<VY>,<VZ>,<Acc>*CS<CR><LF>** VX: WGS84 ECEF X Cartesian velocity vector (m/s).

	VY: WGS84 ECEF Y Cartesian velocity vector (m/s). VZ: WGS84 ECEF Z Cartesian velocity vector (m/s). Acc: 3-dimensional speed 1-sigma accuracy (m/s). **[Example]** Send: $PAIR031*38\r\n Response: $PAIR001,031,0*39\r\n ==> Success $PAIR031,0.19,-0.07,-0.11,0.49*3A\r\n ==> The WGS84 ECEF XYZ Cartesian Velocity.
PAIR_COMMON_GET_TOW_WN	**[Packet Type]** 043 **[Command]** PAIR_COMMON_GET_TOW_WN **[Description]** Get TOW (Time of week) and WN (Week number) information. **[Data Field]** **$PAIR043*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR043,<TOW>,<WN>*CS<CR><LF>** TOW: GNSS Time of week WN: GNSS Week number **[Example]** Send: $PAIR043*3D\r\n Response: $PAIR001,043,0*3C\r\n $PAIR043,2065,394925.000*22\r\n
PAIR_COMMON_SET_FIX_RATE	**[Packet Type]** 050 **[Command]** PAIR_COMMON_SET_FIX_RATE **[Description]** Set Position Fix Interval. If set less than 1000 ms, ASCII NMEA will automatically increase the update interval in order to decrease IO throughput. It will return false if the operating voltage setting is not correct.

	(Any fix interval change between 1Hz <-> multihz causes GNSS to power on/off) (ULP mode only support 1Hz) **[Data Field]** **$PAIR050,<Fix_Interval>*CS<CR><LF>** Fix_Interval: Position fix interval in milliseconds (ms). [Range: 100 ~ 1000] **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR050,1000*12\r\n Response: $PAIR001,050,0*3E\r\n ==> Success
PAIR_COMMON_GET_FIX_RATE	**[Packet Type]** 051 **[Command]** PAIR_COMMON_GET_FIX_RATE **[Description]** Get Position Fix Interval. **[Data Field]** **$PAIR051*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR050,<Fix_Interval>*CS<CR><LF>** Fix_Interval: Position fix interval in milliseconds (ms). [Range: 100 ~ 1000] **[Example]** Send: $PAIR051*3E\r\n Response: $PAIR001,051,0*3F\r\n ==> Success $PAIR051,1000*13\r\n
PAIR_COMMON_SET_MIN_SNR	**[Packet Type]** 058 **[Command]** PAIR_COMMON_SET_MIN_SNR **[Description]** Set the minimum SNR of used satellites. **[Data Field]**

	$PAIR058,<MIN_SNR>*CS<CR><LF> MIN_SNR: Minimum SNR threshold of used satellites. (Valid range: 9~37, default value: 9) **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR058,15*1F\r\n ==> Set the minimum SNR threshold to 15, the chip would not use the satellite with an SNR that is smaller than 15. Response: $PAIR001,058,0*36\r\n ==> Success
PAIR_COMMON_GET_MIN_SNR	**[Packet Type]** 059 **[Command]** PAIR_COMMON_GET_MIN_SNR **[Description]** Query the minimum SNR of used satellites. **[Data Field]** **$PAIR059*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR059,<MIN_SNR>*CS<CR><LF>** MIN_SNR: Minimum SNR threshold of used satellites. (Valid range: 9~37, default value: 9) **[Example]** Send: $PAIR059*36\r\n Response: $PAIR001,059,0*37\r\n ==> Success $PAIR059,15*1E\r\n
PAIR_COMMON_SET_ESTIMATED_NUM	**[Packet Type]** 060 **[Command]** PAIR_COMMON_SET_ESTIMATED_NUM **[Description]** Set the number of estimated fixes when entering the tunnel. **[Data Field]** **$PAIR060,<NUM_ESTIMATED>*CS<CR><LF>**

	NUM_ESTIMATED: Number of estimated fixes. (Valid range: 0~500, default value: 0) **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR060,0*20\r\n ==> Disable the estimated fix when entering the tunnel. Response: $PAIR001,060,0*3D\r\n ==> Success Send: $PAIR060,3*23\r\n ==> Keep outputting three fixes when entering the tunnel. Response: $PAIR001,060,0*3D\r\n ==> Success
PAIR_COMMON_GET_ESTIMATED_NUM	**[Packet Type]** 061 **[Command]** PAIR_COMMON_GET_ESTIMATED_NUM **[Description]** Query the number of estimated fixes when entering the tunnel. **[Data Field]** **$PAIR061*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR061,<NUM_ESTIMATED>*CS<CR><LF>** NUM_ESTIMATED: Number of estimated fixes. (Valid range: 0~500, default value: 0) **[Example]** Send: $PAIR061*3D\r\n Response: $PAIR001,061,0*3C\r\n ==> Success $PAIR061,0*21\r\n ==> The user disabled the estimated fix
PAIR_COMMON_SET_NMEA_OUTPUT_RATE	**[Packet Type]** 062 **[Command]** PAIR_COMMON_SET_NMEA_OUTPUT_RATE **[Description]** Set the NMEA sentence output interval of corresponding NMEA type. **[Data Field]**

	$PAIR062,<Type>,<Output_Rate>*CS<CR><LF> Type: NMEA Type -1 Reset all sentence to default value. 0 NMEA_SEN_GGA, // GGA interval - GPS Fix Data 1 NMEA_SEN_GLL, // GLL interval - Geographic Position - Latitude longitude 2 NMEA_SEN_GSA, // GSA interval - GNSS DOPS and Active Satellites 3 NMEA_SEN_GSV, // GSV interval - GNSS Satellites in View 4 NMEA_SEN_RMC, // RMC interval - Recommended Minimum Specific GNSS Sentence 5 NMEA_SEN_VTG, // VTG interval - Course Over Ground and Ground Speed 6 NMEA_SEN_ZDA, // ZDA interval - Time & Date 7 NMEA_SEN_GRS, // GRS interval - GNSS Range Residuals 8 NMEA_SEN_GST, // GST Interval - GNSS Pseudorange Error Statistics 9 NMEA_SEN_GNS, // GNS Interval - Fix data for single or combined satellite navigation systems 10 NMEA_SEN_DTM, // DTM Interval - Local geodetic datum and datum offsets from a reference datum Output_Rate: Output interval setting (Valid range: 0~20, default value: 1) 0 - Disabled or not supported sentence 1 - Output once every one position fix 2 - Output once every two position fixes 3 - Output once every three position fixes 4 - Output once every four position fixes 5 - Output once every five position fixes **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR062,0,3*3D\r\n Response: $PAIR001,062,0*3F\r\n ==> Success
PAIR_COMMON_GET_NMEA_OUTPUT_RATE	**[Packet Type]** 063 **[Command]** PAIR_COMMON_GET_NMEA_OUTPUT_RATE **[Description]** Get the NMEA sentence output interval of corresponding NMEA type. **[Data Field]** **$PAIR063,<Type>*CS<CR><LF>** Type: NMEA Type

-1 return all sentence configuration.

0 NMEA_SEN_GGA, // GGA interval - GPS Fix Data

1 NMEA_SEN_GLL, // GLL interval - Geographic Position - Latitude longitude

2 NMEA_SEN_GSA, // GSA interval - GNSS DOPS and Active Satellites

3 NMEA_SEN_GSV, // GSV interval - GNSS Satellites in View

4 NMEA_SEN_RMC, // RMC interval - Recommended Minimum Specific GNSS Sentence

5 NMEA_SEN_VTG, // VTG interval - Course Over Ground and Ground Speed

6 NMEA_SEN_ZDA, // ZDA interval - Time & Date

7 NMEA_SEN_GRS, // GRS interval - GNSS Range Residuals

8 NMEA_SEN_GST, // GST Interval - GNSS Pseudorange Error Statistics

9 NMEA_SEN_GNS, // GNS Interval - Fix data for single or combined satellite navigation systems

10 NMEA_SEN_DTM, // DTM Interval - Local geodetic datum and datum offsets from a reference datum

[Return]

1. **PAIR_ACK** for send result.
2. **$PAIR063,<Type>,<Output_Rate>*CS<CR><LF>**

Type: NMEA Type

0 NMEA_SEN_GGA, // GGA interval - GPS Fix Data

1 NMEA_SEN_GLL, // GLL interval - Geographic Position - Latitude longitude

2 NMEA_SEN_GSA, // GSA interval - GNSS DOPS and Active Satellites

3 NMEA_SEN_GSV, // GSV interval - GNSS Satellites in View

4 NMEA_SEN_RMC, // RMC interval - Recommended Minimum Specific GNSS Sentence

5 NMEA_SEN_VTG, // VTG interval - Course Over Ground and Ground Speed

6 NMEA_SEN_ZDA, // ZDA interval - Time & Date

7 NMEA_SEN_GRS, // GRS interval - GNSS Range Residuals

8 NMEA_SEN_GST, // GST Interval - GNSS Pseudorange Error Statistics

9 NMEA_SEN_GNS, // GNS Interval - Fix data for single or combined satellite navigation systems

10 NMEA_SEN_DTM, // DTM Interval - Local geodetic datum and datum offsets from a reference datum

Output_Rate: Output interval setting (Valid range: 0~20, default value: 1)

0 - Disabled or not supported sentence

1 - Output once every one position fix

2 - Output once every two position fixes

3 - Output once every three position fixes

4 - Output once every four position fixes

5 - Output once every five position fixes

[Example]

Send:

	$PAIR063,0*23\r\n Response: $PAIR001,063,0*3E\r\n ==> Success $PAIR063,0,3*3C\r\n
PAIR_COMMON_SET_HACC_LIMIT	**[Packet Type]** 064 **[Command]** PAIR_COMMON_SET_HACC_LIMIT **[Description]** Set horizontal accuracy mask. Range from 30m to 200m or 0. GPS only gets the fix when hacc value < mask. **[Data Field]** **$PAIR064,<HaccMask>*CS<CR><LF>** HaccMask: 30~200: enable hacc mask feature. (Units: meter) 0 [Default Value]: disable hacc mask feature. **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR064,50*11\r\n Response: $PAIR001,064,0*39\r\n ==> Success **[Note]** If horizontal accuracy > HaccMask is in use. The GNSS system will not output NMEA sentences.
PAIR_COMMON_GET_HACC_LIMIT	**[Packet Type]** 065 **[Command]** PAIR_COMMON_GET_HACC_LIMIT **[Description]** Query horizontal accuracy mask. **[Data Field]** **$PAIR065*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR065,<HaccMask>*CS<CR><LF>** HaccMask: 30~200: enable hacc mask feature. (Units: meter)

	0 [Default Value]: disable hacc mask feature. **[Example]** Send: $PAIR065*39\r\n Response: $PAIR001,065,0*38\r\n ==> Success $PAIR065,50*10\r\n
PAIR_COMMON_SET_GNSS_SEARCH_MODE	**[Packet Type]** 066 **[Command]** PAIR_COMMON_SET_GNSS_SEARCH_MODE **[Description]** Configure the receiver to start searching for satellites. The setting is available when the NVRAM data is valid. The device restarts when it receives this command. Abbreviation: (GPS: "G", GLONASS: "R", Galileo: "E", BeiDou: "B", NavIC, "I") Support constellation in L1 package: G/ GR/ GE/ GB/ GREB Support constellation in L1 + L5 package: GREB / GEB Support constellation in L1 + NavIC package G/ I/ GEI/ GREB/ GREBI QZSS is always switchable. **[Data Field]** **$PAIR066,<GPS_Enabled>,<GLONASS_Enabled>,<Galileo_Enabled>,<BeiDou_Enabled>,<QZSS_Enabled>,<NavIC_Enabled>*CS<CR><LF>** GPS_Enabled: "0", disable (DO NOT search GPS satellites). "1", search GPS satellites. GLONASS_Enabled: "0", disable (DO NOT search GLONASS satellites). "1", search GLONASS satellites. Galileo_Enabled: "0", disable (DO NOT search Galileo satellites). "1", search Galileo satellites. BeiDou_Enabled: "0", disable (DO NOT search BeiDou satellites). "1", search BeiDou satellites. QZSS_Enabled: "0", disable (DO NOT search QZSS satellites). "1", search QZSS satellites. NavIC_Enabled:

	"0", disable (DO NOT search NavIC satellites). "1", search NavIC satellites. **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR066,1,0,0,0,0,0*3B\r\n ==> Search GPS satellites only. Response: $PAIR001,066,0*3B\r\n ==> Success Send: $PAIR066,1,1,1,1,1,0*3B\r\n ==> Search GPS, GLONASS, Galileo, BeiDou, QZSS satellites. Response: $PAIR001,066,0*3B\r\n ==> Success Send: $PAIR066,1,1,0,0,0,0*3A\r\n ==> Search GPS and GLONASS satellites. Response: $PAIR001,066,0*3B\r\n ==> Success
PAIR_COMMON_GET_GNSS_SEARCH_MODE	**[Packet Type]** 067 **[Command]** PAIR_COMMON_GET_GNSS_SEARCH_MODE **[Description]** This command is to get GPS, GLONASS, Galileo, BeiDou, QZSS and NavIC search settings. **[Data Field]** **$PAIR067*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR067,<GPS_Enabled>,<GLONASS_Enabled>,<Galileo_Enabled>,<BeiDou_Enabled>,<QZSS_Enabled>,<NavIC_Enabled>*CS<CR><LF>** GPS_Enabled: "0", disable (DO NOT search GPS satellites). "1", search GPS satellites. GLONASS_Enabled: "0", disable (DO NOT search GLONASS satellites). "1", search GLONASS satellites. Galileo_Enabled: "0", disable (DO NOT search Galileo satellites). "1", search Galileo satellites. BeiDou_Enabled:

	"0", disable (DO NOT search BeiDou satellites). "1", search BeiDou satellites. QZSS_Enabled: "0", disable (DO NOT search QZSS satellites). "1", search QZSS satellites. NavIC_Enabled: "0", disable (DO NOT search NavIC satellites). "1", search NavIC satellites. **[Example]** Send: $PAIR067*3B\r\n Response: $PAIR001,067,0*3A\r\n ==> Success $PAIR067,1,0,0,0,0,0*3A\r\n ==> Search GPS satellites only.
PAIR_COMMON_SET_HDOP_THRESHOLD	**[Packet Type]** 068 **[Command]** PAIR_COMMON_SET_HDOP_THRESHOLD **[Description]** This command is for setting the HDOP threshold. If the HDOP value is larger than this threshold value, the position will not be fixed. **[Data Field]** **$PAIR068,<HDOPThreshold>*CS<CR><LF>** HDOPThreshold: "0": Disable this function. Other value: Enable setting the HDOP threshold [Range:] **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR068,0.8*3E\r\n Response: $PAIR001,068,0*35\r\n ==> Success
PAIR_COMMON_GET_HDOP_THRESHOLD	**[Packet Type]** 069 **[Command]** PAIR_COMMON_GET_HDOP_THRESHOLD

	[Description] This command is to get the HDOP threshold. **[Data Field]** **$PAIR069*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR069,<HDOPThreshold>*CS<CR><LF>** HDOPThreshold: "0": Disable this function. Other value: Enable setting the HDOP threshold [Range:] **[Example]** Send: $PAIR069*35\r\n Response: $PAIR001,069,0*34\r\n ==> Success $PAIR069,0.8*3F\r\n
PAIR_COMMON_SET_STATIC_THRESHOLD	**[Packet Type]** 070 **[Command]** PAIR_COMMON_SET_STATIC_THRESHOLD **[Description]** Set the speed threshold for static navigation. If the actual speed is less than the threshold, the output position remains the same and the output speed will be zero. If the threshold value is set to 0, this function is disabled. **[Data Field]** **$PAIR070,<Speed_threshold>*CS<CR><LF>** Speed_threshold. 0~20 dm/s. Default value is 0 dm/s. The minimum is 1 dm/s, the maximum is 20 dm/s. 1 dm/s = 0.1m/s **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR070,4*25\r\n Response: $PAIR001,070,0*3C\r\n ==> Success

PAIR_COMMON_GET_STATIC_THRESHOLD	**[Packet Type]** 071 **[Command]** PAIR_COMMON_GET_STATIC_THRESHOLD **[Description]** Query the static navigation speed threshold. **[Data Field]** **$PAIR071*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR071,<Speed_threshold>*CS<CR><LF>** Speed_threshold. 0~2 m/s. Default value is 0 m/s. The minimum is 0.1 m/s, the maximum is 2.0 m/s. **[Example]** Send: $PAIR071*3C\r\n Response: $PAIR001,071,0*3D\r\n ==> Success $PAIR071,0.4*3A\r\n
PAIR_COMMON_SET_ELEV_MASK	**[Packet Type]** 072 **[Command]** PAIR_COMMON_SET_ELEV_MASK **[Description]** Set satellite elevation mask. Satellites below the elevation mask are not used. **[Data Field]** **$PAIR072,<Degree>*CS<CR><LF>** Degree: Satellite elevation-mask. (Valid range: -90 ~ 90, default value: 5) **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR072,5*26\r\n Response: $PAIR001,072,0*3E\r\n ==> Success

PAIR_COMMON_GET_ELEV_MASK	**[Packet Type]** 073 **[Command]** PAIR_COMMON_GET_ELEV_MASK **[Description]** Get satellite elevation mask. **[Data Field]** **$PAIR073*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR073,<Degree>*CS<CR><LF>** Degree: Satellite elevation-mask. (Valid range: -90 ~ 90, default value: 5) **[Example]** Send: $PAIR073*3E\r\n Response: $PAIR001,073,0*3F\r\n ==> Success $PAIR073,5*27\r\n
PAIR_COMMON_SET_AIC_ENABLE	**[Packet Type]** 074 **[Command]** PAIR_COMMON_SET_AIC_ENABLE **[Note]** This command will get a failed response if GNSS-SubSys does not power on. You can get the GNSS-SubSys status by **PAIR_GNSS_SUBSYS_GET_STATUS**. **[Description]** Enable or disable active interference cancellation function. **[Data Field]** **$PAIR074,<Enabled>*CS<CR><LF>** Enabled. Enable or disable "0", Disable. "1", Enable. **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR074,1*24\r\n Response:

	$PAIR001,074,0*38\r\n ==> Success
PAIR_COMMON_GET_AIC_STATUS	**[Packet Type]** 075 **[Command]** PAIR_COMMON_GET_AIC_STATUS **[Description]** Get the status of active interference cancellation function. **[Data Field]** **$PAIR075*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR075,<Enabled>*CS<CR><LF>** Enabled: Enable or disable "0", Disable. "1", Enable. **[Example]** Send: $PAIR075*38\r\n Response: $PAIR001,075,0*39\r\n ==> Success $PAIR075,0*24\r\n ==> AIC is disabled.
PAIR_COMMON_SET_DATUM	**[Packet Type]** 076 **[Command]** PAIR_COMMON_SET_DATUM **[Description]** Set default datum. **[Data Field]** **$PAIR076,<Datum>*CS<CR><LF>** Datum: 0: WGS84 1: TOKYO-M 2: TOKYO-A ... **[Return]** 1. **PAIR_ACK** for send result. **[Example]**

	Send: $PAIR076,0*27\r\n Response: $PAIR001,076,0*3A\r\n ==> Success **[Note]** The total datums list in the **Appendix - Datum List**.
PAIR_COMMON_GET_DATUM	**[Packet Type]** 077 **[Command]** PAIR_COMMON_GET_DATUM **[Description]** Get default datum. **[Data Field]** **$PAIR077*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR077,<Datum>*CS<CR><LF>** Datum: 0: WGS84 1: TOKYO-M 2: TOKYO-A The total datums list in the **Appendix - Datum List**. **[Example]** Send: $PAIR077*3A\r\n Response: $PAIR001,077,0*3B\r\n ==> Success $PAIR077,0*26\r\n
PAIR_COMMON_SET_NAVIGATION_MODE	**[Packet Type]** 080 **[Command]** PAIR_COMMON_SET_NAVIGATION_MODE **[Description]** Set navigation mode. The command fails in the following situations: if ULP is enabled, it fails to set Non Fitness mode. all detailed failure information will be displayed in system log. **[Data Field]**

	$PAIR080,<CmdType>*CS<CR><LF> CmdType: '0' Normal mode: For general purpose '1' [Default Value] Fitness mode: For running and walking activities so that the low-speed (< 5 m/s) movement will have more of an effect on the position calculation. '2' Reserved '3' Balloon mode: For high-altitude balloon purpose that the vertical movement will have more effect on the position calculation. '4' Stationary mode: For stationary applications where a zero dynamic assumed. '5' Drone mode: Used for drone applications with equivalent dynamics range and vertical acceleration on different flight phase. (Ex. hovering, cruising, etc.). (Note: The NMEA decimal precision will change to lat/lon in 7 digits, alt in 3 digit automaully) '6' Reserved '7' Swimming mode: For swimming purpose so that it smooths the trajectory and improves the accuracy of distance calculation. '9' Bike mode: For sharing bike application. **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR080,1*2F\r\n ==> Enter fitness mode. Response: $PAIR001,080,0*33\r\n ==> Success **[Note]** Each mode has its altitude limitation. Please choose the appropriate mode based on the following table. If your test scenario exceeds the limitation, the position calculation will be incorrect. Mode — Altitude Limitation Normal mode — 10000 m Fitness mode — 10000 m Balloon mode — 80000 m Stationary mode — 10000 m Drone mode — 10000 m Swimming mode — 10000 m
PAIR_COMMON_GET_NAVIGATION_MODE	**[Packet Type]** 081 **[Command]** PAIR_COMMON_GET_NAVIGATION_MODE **[Description]** Get navigation mode.

	[Data Field] **$PAIR081*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR081,<CmdType>*CS<CR><LF>** CmdType: '0' Normal mode: For general purpose '1' [Default Value] Fitness mode: For running and walking activities so that the low-speed (< 5 m/s) movement will have more of an effect on the position calculation. '2' Reserved '3' Balloon mode: For high-altitude balloon purposes so that the vertical movement will have more effect on the position calculation. '4' Stationary mode: For stationary applications where a zero dynamic assumed. '5' Drone mode: Used for drone applications with equivalent dynamics range and vertical acceleration on different flight phase. (Ex. hovering, cruising, etc.) (Note: The NMEA decimal precision will automatically change to lat/lon in 7 digits, alt in 3 digits) '6' Reserved '7' Swimming mode: For swimming purpose so that it smooths the trajectory and improves the accuracy of distance calculation. '9' Bike mode: For sharing bike application. **[Example]** Send: $PAIR081*33\r\n Response: $PAIR001,081,0*32\r\n ==> Success $PAIR081,1*2E\r\n ==> Current is fitness mode.
PAIR_COMMON_SET_DEBUGLOG_OUTPUT	**[Packet Type]** 086 **[Command]** PAIR_COMMON_SET_DEBUGLOG_OUTPUT **[Description]** This command is to set enable/disable debug log output in binary format **[Data Field]** **$PAIR086,<Status>*CS<CR><LF>** Status: 0: Disable 1: Enable with full debuglog output 2: Enable with lite debuglog output **[Return]**

	1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR086,1*29\r\n Response: $PAIR001,086,0*35\r\n ==> Success
PAIR_COMMON_GET_DEBUGLOG_OUTPUT	**[Packet Type]** 087 **[Command]** PAIR_COMMON_GET_DEBUGLOG_OUTPUT **[Description]** Query setting of debug log output. **[Data Field]** **$PAIR087*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR087,<Status>*CS<CR><LF>** Status: 0: Disable 1: Enable with full debuglog output 2: Enable with lite debuglog output **[Example]** Send: $PAIR087*35\r\n Response: $PAIR001,087,0*34\r\n ==> Success $PAIR087,1*28 ==> Enable with full debuglog output
PAIR_COMMON_SET_STATIC_MODE	**[Packet Type]** 092 **[Command]** PAIR_COMMON_SET_STATIC_MODE **[Description]** Enable static mode **[Data Field]** **$PAIR092,<Enable>*CS<CR><LF>** Enable: 0: Disable static mode

	1: Enable static mode **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR092,1*2C\r\n Response: $PAIR001,092,0*30\r\n ==> Success
PAIR_COMMON_GET_STATIC_MODE	**[Packet Type]** 093 **[Command]** PAIR_COMMON_GET_STATIC_MODE **[Description]** Query if current mode is static mode **[Data Field]** **$PAIR093*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR093,<Enable>*CS<CR><LF>** Enable: 0: Not in static mode 1: In static mode **[Example]** Send: $PAIR093*30\r\n Response: $PAIR001,093,0*31\r\n ==> Success $PAIR093,1*2D\r\n ==> In static mode
PAIR_COMMON_SET_NMEA_POS_DECIMAL_PRECISION	**[Packet Type]** 098 **[Command]** PAIR_COMMON_SET_NMEA_POS_DECIMAL_PRECISION **[Description]** This command is for setting the digits shown in the NMEA position. **[Data Field]** **$PAIR098,<MODE>*CS<CR><LF>** MODE:

	0: Latitude, Longitude in 4 digits, Altitude in 1 digit 1: Latitude, Longitude in 5 digits, Altitude in 2 digit 2: Latitude, Longitude in 6 digits, Altitude in 3 digit 3: Latitude, Longitude in 7 digits, Altitude in 3 digit **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR098,0*27\r\n ==> Set the Lat/Lon digit 4 digit, and Alt in 1 digit (GGA/GLL/RMC) Response: $PAIR001,098,0*3A\r\n ==> Success
PAIR_COMMON_GET_NMEA_POS_DECIMAL_PRECISION	**[Packet Type]** 099 **[Command]** PAIR_COMMON_GET_NMEA_POS_DECIMAL_PRECISION **[Description]** This command is to get NMEA position shown digit mode **[Data Field]** **$PAIR099*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR099,<MODE>*CS<CR><LF>** MODE: 0: Latitude, Longitude in 4 digits, Altitude in 1 digit 1: Latitude, Longitude in 5 digits, Altitude in 2 digit 2: Latitude, Longitude in 6 digits, Altitude in 3 digit 3: Latitude, Longitude in 7 digits, Altitude in 3 digit **[Example]** Send: $PAIR099*3A\r\n Response: $PAIR001,099,0*3B\r\n ==> Success $PAIR099,0*26\r\n ==> Latitude, Longitude in 4 digits, Altitude in 1 digit
PAIR_COMMON_SET_NMEA_OUTPUT_MODE	**[Packet Type]** 100 **[Command]**

	PAIR_COMMON_SET_NMEA_OUTPUT_MODE **[Description]** This command is to set NMEA output mode **[Data Field]** **$PAIR100,<NMEA_MODE>,<PROPRIETARY_MODE>*CS<CR><LF>** NMEA_MODE: 0: Disable NMEA 1: ASCII NMEA v4.10(Default) 2: ASCII NMEA v3.0 3: ASCII NMEA v4.11 PROPRIETARY_MODE: 0: Disable extra proprietary sentence (Default) 1: Enable proprietary sentence **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR100,1,0*3A\r\n ==> ASCII NMEA v4.10, Disable extra proprietary sentence Response: $PAIR001,100,0*3A\r\n ==> Success Send: $PAIR100,0,1*3A\r\n ==> No ASCII NMEA output, Enable proprietary sentence Response: $PAIR001,100,0*3A\r\n ==> Success
PAIR_COMMON_GET_NMEA_OUTPUT_MODE	**[Packet Type]** 101 **[Command]** PAIR_COMMON_GET_NMEA_OUTPUT_MODE **[Description]** This command is to get NMEA output mode **[Data Field]** **$PAIR101*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR101,<NMEA_MODE>,<PROPRIETARY_MODE>*CS<CR><LF>** NMEA_MODE: 0: Disable NMEA 1: ASCII NMEA v4.10(Default) 2: ASCII NMEA v3.0

	3: ASCII NMEA v4.11 PROPRIETARY_MODE: 0: Disable extra proprietary sentence (Default) 1: Enable proprietary sentence 2: Enable proprietary sentence and output before NMEA **[Example]** Send: $PAIR101*3A\r\n Response: $PAIR001,101,0*3B\r\n ==> Success $PAIR101,0,1*3B\r\n ==> Disable NMEA output, enable proprietary sentence
PAIR_COMMON_SET_DUAL_BAND	**[Packet Type]** 104 **[Command]** PAIR_COMMON_SET_DUAL_BAND **[Description]** This command is to set Dual Band state when GNSS service is powered off. The command fails in the following cases: If GNSS Service is powered on, refer to PAIR002 and PAIR003. If firmware is a single band. If GNSS search mode cannot be supported in the next Dual Band state, for example: If the current GNSS Search mode is G+GLO, "PAIR104,1" will fail, because Dual Band doesn't support G+GLO. All detailed information related to failure will be displayed in system log. **[Data Field]** **$PAIR104,<DUAL_BAND_Enabled>*CS<CR><LF>** DUAL_BAND_Enabled: "0", disable (DO NOT open Dual Band). "1", enable. **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR104,0*23\r\n Response: $PAIR001,104,0*3E\r\n ==> Success
PAIR_COMMON_GE	**[Packet Type]**

T_DUAL_BAND	105 **[Command]** PAIR_COMMON_GET_DUAL_BAND **[Description]** Query whether Dual Band is enabled or disabled. **[Data Field]** **$PAIR105*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR105,<Enable>*CS<CR><LF>** Enable: Enable or disable '0': Disable '1': Enable **[Example]** Send: $PAIR105*3E\r\n Response: $PAIR001,105,0*3F\r\n ==> Success $PAIR105,1*23\r\n
PAIR_COMMON_SET_SV_BLACKLIST	**[Packet Type]** 130 **[Command]** PAIR_COMMON_SET_SV_BLACKLIST **[Note]** This configuration will not keep in the flash or RTC RAM. Please send this command every time after the GNSS subsystem or main power reboots. This command will get a failed response if GNSS-SubSys does not power on. You can get the GNSS-SubSys status by **PAIR_GNSS_SUBSYS_GET_STATUS**. **[Description]** Set sv blacklist for the selected constellation, which excludes specific PRNs and does not search for them. **[Data Field]** **$PAIR130,<System_ID>,<Blacklist>*CS<CR><LF>** System_ID: The GNSS constellation 0: GPS L1/L5 1: GLONASS L1 2: Galileo E1/E5a 3: BeiDou B1/B2a 4: QZSS L1/L5

	5: NavIC L5 Blacklist: Bitwise format in HEX (LSB for QZSS NMEA PRN: 193, GLONASS NMEA PRN: 65, others PRN: 1) **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR130,0,8000C001*72\r\n ==> Disable tracking GPS PRN 1,15,16,32 Response: $PAIR001,130,0*39\r\n ==> Success
PAIR_COMMON_GET_SV_BLACKLIST	**[Packet Type]** 131 **[Command]** PAIR_COMMON_GET_SV_BLACKLIST **[Note]** This command will get a failed response if GNSS-SubSys does not power on. You can get the GNSS-SubSys status by **PAIR_GNSS_SUBSYS_GET_STATUS**. **[Description]** Get sv blacklist for the selected constellation, which excludes specific PRNs and does not search for them. **[Data Field]** **$PAIR131,<System_ID>*CS<CR><LF>** System_ID: The GNSS constellation 0: GPS L1/L5 1: GLONASS L1 2: Galileo E1/E5a 3: BeiDou B1/B2a 4: QZSS L1/L5 5: NavIC L5 **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR131,<System_ID>,<Blacklist>*CS<CR><LF>** System_ID: The GNSS constellation 0: GPS L1/L5 1: GLONASS L1 2: Galileo E1/E5a 3: BeiDou B1/B2a 4: QZSS L1/L5 5: NavIC L5

	Blacklist: Bitwise format in HEX (LSB for QZSS NMEA PRN: 193, GLONASS NMEA PRN: 65, others PRN: 1) **[Example]** Send: $PAIR131,0*25\r\n Response: $PAIR001,131,0*38\r\n ==> Success $PAIR131,0,8000C001*73\r\n ==> Disable tracking GPS PRN 1,15,16,32
PAIR_COMMON_SET_VERTICAL_CONSTRAINT	**[Packet Type]** 156 **[Command]** PAIR_COMMON_SET_VERTICAL_CONSTRAINT **[Description]** Enable vertical constraint for smoother height estimate. (Fitness default is enable) **[Data Field]** **$PAIR156,<Enable>*CS<CR><LF>** Enable: 0: Disable 1: Enable **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR156,0*24\r\n ==> Disable vertical constraint Response: $PAIR001,156,0*39\r\n ==> Success
PAIR_COMMON_GET_VERTICAL_CONSTRAINT	**[Packet Type]** 157 **[Command]** PAIR_COMMON_GET_VERTICAL_CONSTRAINT **[Description]** Query if vertical constraint is enabled. **[Data Field]** **$PAIR157*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR157,<Enable>*CS<CR><LF>**

	Enable: 0: Disable 1: Enable **[Example]** Send: $PAIR157*39\r\n Response: $PAIR001,157,0*38\r\n ==> Success $PAIR157,0*25\r\n ==> vertical constraint is disabled
PAIR_COMMON_SET_IMMEDIATE_SPEED_MODE	**[Packet Type]** 160 **[Command]** PAIR_COMMON_SET_IMMEDIATE_SPEED_MODE **[Description]** Enable immediate speed mode. This mode will output more prompts and with a less smoothing speed under normal mode. This mode is currently only supported for normal mode. The default setting is disabled. **[Data Field]** **$PAIR160,<Enable>*CS<CR><LF>** Enable: 0: Disable [Default] 1: Enable **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR160,1*20\r\n ==> Enabl immediate speed mode Response: $PAIR001,160,0*3C\r\n ==> Success
PAIR_COMMON_GET_IMMEDIATE_SPEED_MODE	**[Packet Type]** 161 **[Command]** PAIR_COMMON_GET_IMMEDIATE_SPEED_MODE **[Description]** Query if immediate speed mode is enabled. **[Data Field]**

	$PAIR161*CS<CR><LF> **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR161,<Enable>*CS<CR><LF>** Enable: 0: Disable 1: Enable **[Example]** Send: $PAIR161*3C\r\n Response: $PAIR001,161,0*3D\r\n ==> Success $PAIR161,0*20\r\n ==> immediate speed mode is disabled
PAIR_COMMON_SET_2D_FIX_ENABLE	**[Packet Type]** 162 **[Command]** PAIR_COMMON_SET_2D_FIX_ENABLE **[Description]** Enable 2D fix mode. **[Data Field]** **$PAIR162,<Enable>*CS<CR><LF>** Enable: 0: Disable [Default] 1: Enable **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR162,1*22\r\n ==> Enable immediate speed mode Response: $PAIR001,162,0*3E\r\n ==> Success
PAIR_COMMON_GET_2D_FIX_ENABLE	**[Packet Type]** 163 **[Command]** PAIR_COMMON_GET_2D_FIX_ENABLE **[Description]** Query if 2D fix mode is enabled.

	[Data Field] **$PAIR163*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for sending result. 2. **$PAIR163,<Enable>*CS<CR><LF>** Enable: 0: Disable 1: Enable **[Example]** Send: $PAIR163*3E\r\n Response: $PAIR001,163,0*3F\r\n ==> Success $PAIR163,0*22\r\n ==> 2D fix mode is disabled
PAIR_COMMON_SET_ODOMETER_ENABLE	**[Packet Type]** 164 **[Command]** PAIR_COMMON_SET_ODOMETER_ENABLE **[Description]** Enable the odometer function. STATIONARY mode does not support this command. **[Data Field]** **$PAIR164,<Enable>*CS<CR><LF>** Enable: 0: Disable [Default] 1: Enable **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR164,1*24\r\n ==> Enable the odometer function. Response: $PAIR001,164,0*38\r\n ==> Success
PAIR_COMMON_GET_ODOMETER_STATUS	**[Packet Type]** 165 **[Command]** PAIR_COMMON_GET_ODOMETER_STATUS

	[Description] Query the status of the odometer function and the traveling distance value if the function is enabled. **[Data Field]** **$PAIR165*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR165,<Enable>,<distance>*CS<CR><LF>** Enable: 0: Disable 1: Enable distance: This is the traveling distance, which is ranged from 0.0m to 268,435,455.0m, since the epoch when the function is enabled. If the distance exceeds the max. value, the distance will be rolled over to 0.0m and then accumulated again. When the odometer function is disabled and then enabled again, the distance will be restarted from 0.0m. **[Example]** Send: $PAIR165*38\r\n ==> Enable the odometer function. Response: $PAIR001,165,0*39\r\n ==> Success $PAIR165,0,0.0*26\r\n
PAIR_DGPS_SET_MODE	**[Packet Type]** 400 **[Command]** PAIR_DGPS_SET_MODE **[Description]** DGPS correction data source mode. **[Data Field]$PMTK400,<Mode> *CS<CR><LF>** Mode: DGPS data source mode. '0': No DGPS source '1': RTCM '2': SBAS(Include WAAS/EGNOS/GAGAN/MSAS) '3': SLAS **[Return]** 1. **PAIR_ACK** for send result. **[Example]**

	Send: $PAIR400,2*20\r\n ==> Set SBAS Mode Response: $PAIR001,400,0*3F\r\n ==> Success
PAIR_DGPS_GET_MODE	**[Packet Type]** 401 **[Command]** PAIR_DGPS_GET_MODE **[Description]** Query the DGPS data source mode. **[Data Field]** **$PAIR401*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR401,<Mode>*CS<CR><LF>** Mode: DGPS data source mode. '0': No DGPS source '1': RTCM '2': SBAS(Include WAAS/EGNOS/GAGAN/MSAS) '3': SLAS **[Example]** Send: $PAIR401*3F\r\n Response: $PAIR001,401,0*3E\r\n ==> Success $PAIR401,2*21\r\n ==> SBAS Mode
PAIR_SBAS_ENABLE	**[Packet Type]** 410 **[Command]** PAIR_SBAS_ENABLE **[Description]** Enable searching a SBAS satellite or not. When navigation mode is Fitness or Swimming mode, SBAS is not supported. **[Data Field]** **$PAIR410,<Enabled>*CS<CR><LF>** Enabled: Enable or disable '0' = Disable

	'1' = Enable **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR410,1*22\r\n ==> Enable SBAS Response: $PAIR001,410,0*3E\r\n ==> Success
PAIR_SBAS_GET_STATUS	**[Packet Type]** 411 **[Command]** PAIR_SBAS_GET_STATUS **[Description]** Query the status of SBAS to whether it is enabled. **[Data Field]** **$PAIR411*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR411,<Enabled>*CS<CR><LF>** Enabled: Enable or disable '0' = Disable '1' = Enable **[Example]** Send: $PAIR411*3E\r\n Response: $PAIR001,411,0*3F\r\n ==> Success $PAIR411,1*23\r\n ==> Enable SBAS
PAIR_SBAS_GET_SAT_INFO	**[Packet Type]** 412 **[Command]** PAIR_SBAS_GET_SAT_INFO **[Description]** Get information about the SBAS satellites, such as SVid, SNR, azimuth, and elevation. **[Data Field]** **$PAIR412*CS<CR><LF>** **[Return]**

	1. **PAIR_ACK** for send result. 2. **$PAIR412,<SVid>,<SNR>,<Azim>,<Elev>*CS<CR><LF>** **[Example]** Send: $PAIR412*3D\r\n Response: $PAIR001,412,0*3C\r\n ==> Success $PAIR412,50,42,134,50*0D\r\n
PAIR_SBAS_SET_RAW_DATA_OUTPUT	**[Packet Type]** 416 **[Command]** PAIR_SBAS_SET_RAW_DATA_OUTPUT **[Description]** Enable output SBAS raw data in the way of binary proprietary message. For more detailed information, please refer to the Proprietary message chapter of binary proprietary message section in the Airoha_IoT_SDK_for_GNSS_Developers_Guide under the doc folder in IoT_SDK_for_Location package. **[Data Field]** **$PAIR416,<Enabled>*CS<CR><LF>** Enabled: Enable or disable '0' = Disable '1' = Enable **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR416,1*24\r\n Response: $PAIR001,416,0*38\r\n ==> Success
PAIR_SBAS_GET_RAW_DATA_OUTPUT	**[Packet Type]** 417 **[Command]** PAIR_SBAS_GET_RAW_DATA_OUTPUT **[Description]** Query the status of SBAS raw data output to check whether it is enabled. **[Data Field]**

	$PAIR417*CS<CR><LF> **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR417,<Enabled>*CS<CR><LF>** Enabled: Enable or disable '0' = Disable '1' = Enable **[Example]** Send: $PAIR417*38\r\n Response: $PAIR001,417,0*39\r\n ==> Success $PAIR417,1*25\r\n ==> Enable SBAS raw data output
PAIR_SLAS_ENABLE	**[Packet Type]** 420 **[Command]** PAIR_SLAS_ENABLE **[Description]** Enable the QZSS SLAS (Sub-meter Level Augmentation Service) operation. **[Data Field]** **$PAIR420,<Enabled>*CS<CR><LF>** Enabled: Enable or disable '0' = Disable '1' = Enable **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR420,1*21\0d\0a ==> Enable QZSS SLAS Response: $PAIR001,420,0*3D\0d\0a ==> Success
PAIR_SLAS_GET_STATUS	**[Packet Type]** 421 **[Command]** PAIR_SLAS_GET_STATUS **[Description]** Query the status of SLAS to check whether it is enabled.

	[Data Field] **$PAIR421*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR421,<Enabled>*CS<CR><LF>** Enabled: Enable or disable '0' = Disable '1' = Enable **[Example]** Send: $PAIR421*3D\0d\0a Response: $PAIR001,421,0*3C\0d\0a ==> Success $PAIR421,1*20\0d\0a ==> The status of QZSS SLAS is enabled
PAIR_RTCM_SET_INPUT_VERSION	**[Packet Type]** 430 **[Command]** PAIR_RTCM_SET_INPUT_VERSION **[Description]** Set RTCM input version. **[Data Field]** **$PAIR430,<Version>*CS<CR><LF>** Version: Input version 0: RTCM v2.x 1: RTCM v3.x **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR430,0*21\r\n ==> set RTCM v2.x input Response: $PAIR001,430,0*3C\r\n ==> Success
PAIR_RTCM_GET_INPUT_VERSION	**[Packet Type]** 431 **[Command]** PAIR_RTCM_GET_INPUT_VERSION **[Description]**

	Get RTCM input version. **[Data Field]** **$PAIR431*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR431,<Version>*CS<CR><LF>** Version: Input version 0: RTCM v2.x 1: RTCM v3.x **[Example]** Send: $PAIR431*3C\r\n Response: $PAIR001,431,0*3D\r\n ==> Success $PAIR431,0*20\r\n ==> RTCM v2.x input
PAIR_RTCM_SET_OUTPUT_MODE	**[Packet Type]** 432 **[Command]** PAIR_RTCM_SET_OUTPUT_MODE **[Description]** Set RTCM output mode. **[Data Field]** **$PAIR432,<MODE>*CS<CR><LF>** MODE: Measurement output mode (MSM4/MSM7) -1: Output None (Default) 0: Output RTCM3.x with message type MSM4 1: Output RTCM3.x with message type MSM7 **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR432,1*22\r\n ==> set RTCM3.x output with type MSM7 Response: $PAIR001,432,0*3E\r\n ==> Success
PAIR_RTCM_GET_OUTPUT_MODE	**[Packet Type]** 433 **[Command]**

	PAIR_RTCM_GET_OUTPUT_MODE **[Description]** Get RTCM output mode. **[Data Field]** **$PAIR433*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR433,<MODE>*CS<CR><LF>** MODE: Measurement output mode (MSM4/MSM7) -1: Output None (Default) 0: Output RTCM3.x with message type MSM4 1: Output RTCM3.x with message type MSM7 **[Example]** Send: $PAIR433*3E\r\n Response: $PAIR001,433,0*3F\r\n ==> Success $PAIR433,0*22\r\n ==> RTCM3.x output with type MSM4
PAIR_RTCM_SET_OUTPUT_ANT_PNT	**[Packet Type]** 434 **[Command]** PAIR_RTCM_SET_OUTPUT_ANT_PNT **[Description]** This command is to set enable/disable stationary antenna reference point for RTCM output. **[Data Field]** **$PAIR434,<ENABLE>*CS<CR><LF>** ENABLE: Stationary antenna reference point (Message type 1005) 0: Disable 1: Enable **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR434,1*24\r\n ==> set RTCM3.x output with message type 1005 Response: $PAIR001,434,0*38\r\n ==> Success
PAIR_RTCM_GET_O	**[Packet Type]**

UTPUT_ANT_PNT	435 **[Command]** PAIR_RTCM_GET_OUTPUT_ANT_PNT **[Description]** Query setting of stationary antenna reference point for RTCM output. **[Data Field]** **$PAIR435*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR435,<ENABLE>*CS<CR><LF>** ENABLE: Stationary antenna reference point (Message type 1005) 0: Disable 1: Enable **[Example]** Send: $PAIR435*38\r\n Response: $PAIR001,435,0*39\r\n ==> Success $PAIR435,1*25\r\n ==> RTCM3.x output with message type 1005
PAIR_RTCM_SET_OUTPUT_EPHEMERIS	**[Packet Type]** 436 **[Command]** PAIR_RTCM_SET_OUTPUT_EPHEMERIS **[Description]** This command is to set enable/disable RTCM output with satellite ephemeris. **[Data Field]** **$PAIR436,<ENABLE>*CS<CR><LF>** ENABLE: 0: Disable 1: Enable **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR436,1*26\r\n ==> set RTCM3.x output with satellite ephemeris Response: $PAIR001,436,0*3A\r\n ==> Success

PAIR_RTCM_GET_OUTPUT_EPHEMERIS	**[Packet Type]** 437 **[Command]** PAIR_RTCM_GET_OUTPUT_EPHEMERIS **[Description]** Query setting of RTCM satellite ephemeris output. **[Data Field]** **$PAIR437*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR437,<ENABLE>*CS<CR><LF>** ENABLE: 0: Disable 1: Enable **[Example]** Send: $PAIR437*3A\r\n Response: $PAIR001,437,0*3B\r\n ==> Success $PAIR437,1*27\r\n ==> RTCM3.x output with satellite ephemeris
PAIR_EPOC_ENABLE	**[Packet Type]** 496 **[Command]** PAIR_EPOC_ENABLE **[Description]** Enable or Disable EPOC **[Data Field]** **$PAIR496,<Enable>*CS<CR><LF>** Enable: Enable or disable 0: Disable 1: Enable **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR496,1*2C\r\n Response: $PAIR001,496,0*30\r\n ==> Success

PAIR_EPOC_CLEAR_DATA	**[Packet Type]** 507 **[Command]** PAIR_EPOC_CLEAR_DATA **[Description]** Clear All EPOC data in NVRAM **[Data Field]** **$PAIR507*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR507*38\r\n Response: $PAIR001,507,0*39\r\n
PAIR_EPOC_GET_STATUS	**[Packet Type]** 508 **[Command]** PAIR_EPOC_GET_STATUS **[Description]** Get EPOC status **[Data Field]** **$PAIR508*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR508,<Enabled>,<Mode>,<Constellation>*CS<CR><LF>** Enabled: 1: EPOC Enabled, 2: EPOC Disabled Mode: 1: normal mode, 2: normal test mode, 3: independent test mode Constellation: EPOC constellation in hexadecimal, bit 0: GPS, bit 1: GAL, bit 2: BDS, bit 3: GLO, bit 4: QZS **[Example]** Send: $PAIR508*37\r\n Response: $PAIR001,508,0*36\r\n $PAIR508,0,1,1*2B\r\n
PAIR_NVRAM_SAVE	**[Packet Type]**

_SETTING	513 **[Command]** PAIR_NVRAM_SAVE_SETTING **[Description]** Save the current configuration from RTC RAM to flash. **[Data Field]** **$PAIR513*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR513*3D\r\n Response: $PAIR001,513,0*3C\r\n **[Note]** You need to send this command every time after modifying any parameters, if the HW not keep RTC power. Otherwise, the changes will be lost after system reboot and the GNSS module must be reconfigured again. If HW will keep RTC power, no need to use this command. The change of configuration will keep in the RTC RAM. In multi-Hz, this command can only be set when the GNSS system is powered off, while 1Hz does not have this limitation.
PAIR_NVRAM_RESTORE_DEFAULT_SETTING	**[Packet Type]** 514 **[Command]** PAIR_NVRAM_RESTORE_DEFAULT_SETTING **[Description]** Clear the current configuration and restore the default settings. This function does not support run time restore when GNSS is power on. Please send **PAIR_GNSS_SUBSYS_POWER_OFF** to power off GNSS before use this command. **[Data Field]** **$PAIR514*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR514*3A\r\n Response:

	$PAIR001,514,0*3B\r\n
PAIR_TIME_GET_CURRENT_TOW	**[Packet Type]** 596 **[Command]** PAIR_TIME_GET_CURRENT_TOW **[Description]** Get current time of week in milliseconds. **[Data Field]** **$PAIR596*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR596,<Validity>,<TOW>*CS<CR><LF>** Validity: Validity of TOW. 0: Invalid TOW. 1: Valid TOW. TOW: Current time of week in milliseconds. **[Example]** Send: $PAIR596*30\r\n Response: $PAIR001,596,0*31\r\n ==> Success $PAIR596,1,96939680*03\r\n ==> Valid current TOW
PAIR_TIME_GET_GNSS_TOW	**[Packet Type]** 597 **[Command]** PAIR_TIME_GET_GNSS_TOW **[Description]** Get the last GNSS epoch's time of week in milliseconds. **[Data Field]** **$PAIR597*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR597,<Validity>,<TOW>*CS<CR><LF>** Validity: Validity of TOW. 0: Invalid TOW. 1: Valid TOW. TOW: The last GNSS epoch's time of week in milliseconds.

	[Example] Send: $PAIR597*31\r\n Response: $PAIR001,597,0*30\r\n ==> Success $PAIR597,1,96710000*09\r\n ==> Valid TOW of the last GNSS epoch
PAIR_LOW_POWER_ENTRY_RTC_MODE	**[Packet Type]** 650 **[Command]** PAIR_LOW_POWER_ENTRY_RTC_MODE **[Description]** Shutdown all systems, including GNSS and other CM4 modules. CM4 will go into RTC-Mode after sending this command and cannot receive any commands. CM4 can be awoken by the timer or the RTC_EINT pin. All system resource will re-initialize after wake up. **[Data Field]** **$PAIR650,<Second>*CS<CR><LF>** Second: the timer to leave RTC-Mode [Valid range: 0 and 10 ~ 62208000 (2 years)] '0' enter RTC-Mode without any timer **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR650,1*24\r\n Response: $PAIR001,650,4*3C\r\n ==> Parameter error Send: $PAIR650,10*14\r\n Response: Enter RTC-Mode without any response and wake up after 10 seconds **[Note]** S/SD EVK must require timer to enter RTC-Mode.
PAIR_GLP_ENABLE	**[Packet Type]** 680 **[Command]** PAIR_GLP_ENABLE **[Note]** This configuration will not keep in the flash or RTC RAM. Please send this command every time after the GNSS subsystem or main power reboots.

	This command will get a failed response if GNSS-SubSys does not power on. You can get the GNSS-SubSys status by **PAIR_GNSS_SUBSYS_GET_STATUS**. **[Description]** This command is to activate low-power GLP mode. GLP mode supports 1-Hz PVT, GPS L1 only, and Fitness mode. **[Data Field]** **$PAIR680,<Enable>*CS<CR><LF>** Enable: 0: Disable GLP 1: Enable GLP **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR680,1*29\r\n Response: $PAIR001,680,0*35\r\n ==> Success
PAIR_GLP_GET_STATUS	**[Packet Type]** 681 **[Command]** PAIR_GLP_GET_STATUS **[Note]** This command will get a failed response if GNSS-SubSys does not power on. You can get the GNSS-SubSys status by **PAIR_GNSS_SUBSYS_GET_STATUS**. **[Description]** This command is to get low-power GLP mode setting **[Data Field]** **$PAIR681*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR681,<Enable>*CS<CR><LF>** Enable: 0: Disable GLP 1: Enable GLP **[Example]** Send: $PAIR681*35\r\n Response:

	$PAIR001,681,0*34\r\n ==> Success $PAIR681,1*28\r\n
PAIR_PERIODIC_SET_MODE	**[Packet Type]** 690 **[Command]** PAIR_PERIODIC_SET_MODE **[Note]** This configuration will not keep in the flash or RTC RAM. Please send this command every time after the GNSS subsystem or main power reboots. This command will get a failed response if GNSS-SubSys does not power on. You can get the GNSS-SubSys status by **PAIR_GNSS_SUBSYS_GET_STATUS**. **[Description]** This command is used to set Periodic Power Saving Mode Settings. There are two stages in periodic power saving mode (Run stage and Sleep stage), and it will change periodically according to the setting. Run stage: the GNSS module measures and calculates the position. Sleep stage: the GNSS module may enter power saving modes. <Note> Sleep will be interrupted by any DSP corresponding PAIR command. Any restart will force it to go back to normal mode. For more detailed information, please refer to the Power Saving Mode chapter of Periodic Mode section in the Airoha_IoT_SDK_for_GNSS_Developers_Guide under the doc folder in IoT_SDK_for_Location package. **[Data Field]** **$PAIR690,<Mode>,<FirstRun>,<FirstSleep>,<SecondRun>,<SecondSleep>*CS<C R><LF>** Mode: 0: Disable periodic mode. 1: Smart periodic mode. In this mode, GNSS system dynamically increases run time in order to collect more navigation data. 2: Strict periodic mode. In this mode, GNSS system periodically forces entry into low-power mode. If <Mode> is 1 or 2, it needs the following parameter for low-power periodic mode. FirstRun: Interval in seconds to exit the minimum power sleep mode and get a new position fix. [Range: 3~518400 s] FirstSleep: Duration in seconds to get a fix (or attempt to get a fix) before switching from running mode back to a minimum power sleep mode. [Range: 3~518400 s] SecondRun: GNSS system will use "second run time" instead of "run time" setting when there is no signal. [Range: 0 or 3~518400 s] The second run time duration can be "0" only when the second sleep time is "0". SecondSleep: GNSS system will use "second sleep time" instead of "sleep time" setting when there

	is no signal. [Range: 0 or 3~518400 s] The second sleep time duration can be "0" only when the second run time is "0". **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR690,1,21,39,48,72*28\r\n Response: $PAIR001,690,0*34\r\n ==> Success Send: $PAIR690,0*29\r\n ==> Normal mode Response: $PAIR001,690,0*34\r\n ==> Success
PAIR_PERIODIC_GET_MODE	**[Packet Type]** 691 **[Command]** PAIR_PERIODIC_GET_MODE **[Note]** This command will get a failed response if GNSS-SubSys does not power on. You can get the GNSS-SubSys status by **PAIR_GNSS_SUBSYS_GET_STATUS**. **[Description]** This command is used to get Periodic Power Saving Mode Settings. For more detailed information, please refer to the Power Saving Mode chapter of Periodic Mode section in the Airoha_IoT_SDK_for_GNSS_Developers_Guide under the doc folder in IoT_SDK_for_Location package. **[Data Field]** **$PAIR691*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR691,<Mode>,<FirstRun>,<FirstSleep>,<SecondRun>,<SecondSleep>*CS<CR><LF>** Mode: 0: Disable periodic mode. 1: Smart periodic mode. In this mode, GNSS system dynamically increases run time in order to collect more navigation data. 2: Strict periodic mode. In this mode, GNSS system periodically forces entry into low-power mode. FirstRun: Interval in seconds to exit the minimum power sleep mode and get a new position fix. [Range: 3~518400 s]

	FirstSleep: Duration in seconds to get a fix (or attempt to get a fix) before switching from running mode back to a minimum power sleep mode. [Range: 3~518400 s] SecondRun: GNSS system will use "second run time" instead of "run time" setting when there is no signal. [Range: 0 or 3~518400 s] SecondSleep: GNSS system will use "second sleep time" instead of "sleep time" setting when there is no signal. [Range: 0 or 3~518400 s] **[Example]** Send: $PAIR691*34\r\n Response: $PAIR001,691,1*34\r\n $PAIR691,1,21,39,48,72*29\r\n Send: $PAIR691*34\r\n Response: $PAIR001,691,1*34\r\n $PAIR691,0,21,39,48,72*28\r\n ==> Normal mode
PAIR_ULP_ENABLE	**[Packet Type]** 700 **[Command]** PAIR_ULP_ENABLE **[Description]** This command is to set to enable Ultra Low Power. The command fails in the following situations: if Periodic is on. refer to PAIR690 if the current navigation mode is not Fitness mode. all detailed failure information will be displayed in system log. **[Data Field]** **$PAIR700,<ULP_Enabled>*CS<CR><LF>** ULP_Enabled: "0", disable (DO NOT open ULP). "1", enable. **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR700,1*20\r\n Response:

	$PAIR001,700,0*3C\r\n ==> Success
PAIR_ULP_GET_STATUS	**[Packet Type]** 701 **[Command]** PAIR_ULP_GET_STATUS **[Description]** Query whether ULP is enabled or disabled. **[Data Field]** **$PAIR701*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR701,<Enable>*CS<CR><LF>** Enable: Enable or disable '0': Disable '1': Enable **[Example]** Send: $PAIR701*3C\r\n Response: $PAIR001,701,0*3D\r\n ==> Success $PAIR701,1*21\r\n
PAIR_ADP_L5_ENABLE	**[Packet Type]** 710 **[Command]** PAIR_ADP_L5_ENABLE **[Description]** Enable or disable adaptive L5 when GNSS system is powered off In the following cases, the command will return fail: 1)The GNSS system is powered on 2)The firmware is single band 3)The navigation mode is not fitness mode 4)The fix rate is not 1 Hz 5)The periodic mode is on **[Data Field]** **$PAIR710,<Enable>*CS<CR><LF>** Enable: Enable, disable, or other statuses '0': Disable

	'1': Enable **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR710,1*21\r\n Response: $PAIR001,710,0*3D\r\n ==> Success
PAIR_ADP_L5_GET_STATUS	**[Packet Type]** 711 **[Command]** PAIR_ADP_L5_GET_STATUS **[Description]** Query the status of adaptive L5 to check whether it is enabled **[Data Field]** **$PAIR711*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR711,<Enable>*CS<CR><LF>** Enable: Enable, disable, or other statuses '0': Disable '1': Enable **[Example]** Send: $PAIR711*3D\r\n Response: $PAIR001,711,0*3C\r\n ==> Success $PAIR711,1*20\r\n
PAIR_DTM_ENABLE	**[Packet Type]** 720 **[Command]** PAIR_DTM_ENABLE **[Note]** This configuration will not keep in the flash or RTC RAM. Please send this command every time after the GNSS subsystem or main power reboots. This command will get a failed response if GNSS-SubSys does not power on. You can get the GNSS-SubSys status by **PAIR_GNSS_SUBSYS_GET_STATUS**.

	[Description] The GNSS Duty tracking mode is a duty cycle concept to ensure good performance and low power consumption in different environments. This command is to activate Duty tracking mode. For more detailed information, please refer to the Power Saving Mode chapter of Duty tracking mode section in the Airoha_IoT_SDK_for_GNSS_Developers_Guide under the doc folder in IoT_SDK_for_Location package.**[** **Data Field]** **$PAIR720,<Enable>,<Update Period>*CS<CR><LF>** Enable: 0: Disable Duty tracking mode 1: Enable Duty tracking mode Update Period: Position update period Range: 2,5,10 (s) **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Enable: Send: $PAIR720,1,2*3C\r\n Response: $PAIR001,720,0*3E\r\n ==> Success Disable: Send: $PAIR720,0*23\r\n Response: $PAIR001,720,0*3E\r\n ==> Success
PAIR_DTM_GET_STATUS	**[Packet Type]** 721 **[Command]** PAIR_DTM_GET_STATUS **[Description]** This command is to get Duty tracking mode configuration setting. **[Data Field]** **$PAIR721*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR721,<Enable>,<Update Period>*CS<CR><LF>**

	Enable: 0: Duty tracking mode is disabled. 1: Duty tracking mode is enabled. Update Period: Position update period Range: 2,5,10 (s) If Duty tracking mode is disabled, the value will be 1. **[Example]** Send: $PAIR721*3E\r\n Response: $PAIR001,721,0*3F\r\n ==> Success $PAIR721,1,2*3D\r\n
PAIR_FLP_ENABLE	**[Packet Type]** 730 **[Command]** PAIR_FLP_ENABLE **[Note]** This configuration will not keep in the flash or RTC RAM. Please send this command every time after the GNSS subsystem or main power reboots. This command will get a failed response if GNSS-SubSys does not power on. You can get the GNSS-SubSys status by **PAIR_GNSS_SUBSYS_GET_STATUS**. **[Description]** The fitness low power mode is a concept to ensure good performance and low power consumption in different environments. This command is to activate fitness low power mode. For more detailed information, please refer to the Power Saving Mode chapter of fitness low power mode section in the Airoha_IoT_SDK_for_GNSS_Developers_Guide under the doc folder in IoT_SDK_for_Location package. This command is only supported under FITNESS navigation mode. **[Data Field]** **$PAIR730,<Mode>*CS<CR><LF>** Mode: 0: Disable both fitness low-power power saving mode and fitness low-power performance mode 1: Enable fitness low-power power saving mode which might sacrifice GNSS performance in favor of a reduced power consumption. 2: Enable fitness low-power performance mode which achieves relatively good performance while keeping the power consumption low; the support constellation for this mode is only GPS + GLONASS + Galileo + BeiDou for both

	L1/L1 + L5 package. **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Enable power saving mode : Send: $PAIR730,1*23\r\n Response: $PAIR001,730,0*3F\r\n ==> Success Enable performance mode : Send: $PAIR730,2*20\r\n Response: $PAIR001,730,0*3F\r\n ==> Success Disable: Send: $PAIR730,0*22\r\n Response: $PAIR001,730,0*3F\r\n ==> Success
PAIR_FLP_GET_STATUS	**[Packet Type]** 731 **[Command]** PAIR_FLP_GET_STATUS **[Description]** This command is to get fitness low-power mode configuration setting. This command is only supported under FITNESS navigation mode. **[Data Field]** **$PAIR731*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR731,<Mode>*CS<CR><LF>** Mode: 0: Disable both fitness low-power power saving mode and fitness low-power performance mode 1: Fitness low-power power saving mode is enabled. 2: Fitness low-power performance mode is enabled. **[Example]** Send: $PAIR731*3F\r\n Response:

	$PAIR001,731,0*3E\r\n ==> Success $PAIR731,1*22\r\n
PAIR_ALP_ENABLE	**[Packet Type]** 732 **[Command]** PAIR_ALP_ENABLE **[Note]** This configuration will not keep in the flash or RTC RAM. Please send this command every time after the GNSS subsystem or main power reboots. This command will get a failed response if GNSS-SubSys does not power on. You can get the GNSS-SubSys status by **PAIR_GNSS_SUBSYS_GET_STATUS**. **[Description]** The adaptive low power mode is a concept to ensure good performance and low power consumption in different environments. This command is to activate adaptive low power mode. For more detailed information, please refer to the Power Saving Mode chapter of adaptive low power mode section in the Airoha_IoT_SDK_for_GNSS_Developers_Guide under the doc folder in IoT_SDK_for_Location package. This command is only supported under NORMAL and FITNESS navigation modes. **[Data Field]** **$PAIR732,<Mode>*CS<CR><LF>** Mode: 0: Disable both adaptive low-power power saving mode and adaptive low-power performance mode 1: Enable adaptive low-power power saving mode which might sacrifice GNSS performance in favor of a reduced power consumption (L1 only). 2: Enable adaptive low-power performance mode which achieves relatively good performance while keeping the power consumption low; **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Enable: Send: $PAIR732,1*21\r\n Response: $PAIR001,732,0*3D\r\n ==> Success Disable: Send:

	$PAIR732,0*20\r\n Response: $PAIR001,732,0*3D\r\n ==> Success
PAIR_ALP_GET_STATUS	**[Packet Type]** 733 **[Command]** PAIR_ALP_GET_STATUS **[Description]** This command is to get adaptive low-power mode configuration setting. This command is only supported under NORMAL and FITNESS navigation modes. **[Data Field]** **$PAIR733*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR733,<Mode>*CS<CR><LF>** Mode: 0: Disable both adaptive low-power power saving mode and adaptive low-power performance mode 1: adaptive low-power power saving mode is enabled. 2: adaptive low-power performance mode is enabled. **[Example]** Send: $PAIR733*3D\r\n Response: $PAIR001,733,0*3C\r\n ==> Success $PAIR733,1*20\r\n
PAIR_PPS_SET_CONFIG_CMD	**[Packet Type]** 752 **[Command]** PAIR_PPS_SET_CONFIG_CMD **[Description]** Configure the PPS settings. **[Data Field]** **$PAIR752,<PPSType>,<PPSPulseWidth>*CS<CR><LF>** PPSType: Availability "0", Disable "1", After the first fix

	"2", 3D fix only "3", 2D/3D fix only "4", Always PPSPulseWidth. PPS Pulse Width (unit in ms). [Range: 1 ~ 999] **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR752,2,100*39\r\n Response: $PAIR001,752,0*3B\r\n ==> Success
PAIR_RAW_ENABLE	**[Packet Type]** 830 **[Command]** PAIR_RAW_ENABLE **[Description]** Set enable/disable output binary raw measurement **[Data Field]** **$PAIR830,<Enable>*CS<CR><LF>** Enable: 0: disable 1: raw meas 2: raw meas + sv info + pvt (including time offset data between GPS and GLO/GAL/BDS) **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR830,1*2C\r\n Response: $PAIR001,830,0*30\r\n ==> Success
PAIR_RAW_GET_STATUS	**[Packet Type]** 831 **[Command]** PAIR_RAW_GET_STATUS **[Description]** Get enable/disable output binary raw measurement **[Data Field]**

	$PAIR831*CS<CR><LF> **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR831,<Enable>*CS<CR><LF>** Enable: 0: disable 1: raw meas 2: raw meas + sv info + pvt **[Example]** Send: $PAIR831*30\r\n Response: $PAIR001,831,0*31\r\n ==> Success $PAIR831,1*2D\r\n
PAIR_IO_SET_BAUDRATE	**[Packet Type]** 864 **[Command]** PAIR_IO_SET_BAUDRATE **[Description]** Set port baud rate configuration. **[Data Field]** **$PAIR864,<Port_Type>,<Port_Index>,<Baudrate>*CS<CR><LF>** Port_Type: HW Port Type. 0: UART Port_Index: HW Port Index UART - 0: UART0, 1: UART1, 2: UART2 Baudrate: the baud rate need config. Support 115200, 230400, 460800, 921600, 3000000 **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR864,0,0,115200*1B\r\n Response: $PAIR001,864,0*31\r\n ==> Success **[Note]** Must reboot the device after changing the port baud rate. The change will valid after reboot.

PAIR_IO_GET_BAUDRATE	**[Packet Type]** 865 **[Command]** PAIR_IO_GET_BAUDRATE **[Description]** Get port baud rate configuration. **[Data Field]** **$PAIR865,<Port_Type>,<Port_Index>*CS<CR><LF>** Port_Type: HW Port Type. 0: UART Port_Index: HW Port Index UART - 0: UART0, 1: UART1, 2: UART2 **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR865,<Baudrate>*CS<CR><LF>** Baudrate: the baud rate need config. Support 115200, 230400, 460800, 921600, 3000000 **[Example]** Send: $PAIR865,0,0*31\r\n Response: $PAIR001,865,0*30\r\n ==> Success $PAIR865,115200*1A\r\n ==> Get UART0 baud rate is 115200. **[Note]** Must reboot the device after changing the port baud rate.
PAIR_IO_SET_FLOW_CONTROL	**[Packet Type]** 866 **[Command]** PAIR_IO_SET_FLOW_CONTROL **[Description]** Set port flow control configuration. **[Data Field]** **$PAIR866,<Port_Type>,<Port_Index>,<Flow_control>*CS<CR><LF>** Port_Type: HW Port Type. 0: UART Port_Index: HW Port Index UART - 0: UART0, 1: UART1, 2: UART2 Flow_control: 0, disable flow control. 1, enable SW flow control. 2, enable HW flow control. **[Return]**

	1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR866,0,2,1*2D\r\n ==> Set UART2 SW Flow Control ON Response: $PAIR001,866,0*33\r\n ==> Success **[Note]** Must reboot the device after changing the flow control type. The change will valid after reboot.
PAIR_IO_GET_FLOW_CONTROL	**[Packet Type]** 867 **[Command]** PAIR_IO_GET_FLOW_CONTROL **[Description]** Get port SW flow control configuration. **[Data Field]** **$PAIR867,<Port_Type>,<Port_Index>*CS<CR><LF>** Port_Type: HW Port Type. 0: UART Port_Index: HW Port Index UART - 0: UART0, 1: UART1, 2: UART2 **[Return]** 1. **PAIR_ACK** for send result. 2. $PAIR867,<sw_flow_control>*CS<CR><LF> Flow_control: 0, disable flow control. 1, enable SW flow control. 2, enable HW flow control. **[Example]** Send: $PAIR867,0,2*31\r\n Response: $PAIR001,867,0*32\r\n $PAIR867,0*2F\r\n ==> Get UART2 Flow Control OFF
PAIR_GEOFENCE_SET_CONFIG	**[Packet Type]** 890 **[Command]** PAIR_GEOFENCE_SET_CONFIG **[Description]** This command is used to set Geofencing configuration. **[Data Field]**

	$PAIR890,<FenceNum>,<ConfLvl>,<Lat1>,<Lon1>,<Rad1>,...,<RadN>*CS<CR> <LF> FenceNum(N): Number of geofences, the value is limited to 4. When the value is set to 0, the geofencing function is disabled. ConfLvl: The confidence level for state evaluation. '0' No requirement '1' 1-Sigma (68%) '2' 2-Sigma (95%) '3' 3-Sigma (99.7%) '4' 4-Sigma (99.9999%) '5' 5-Sigma (99.999999%) Lat: Latitude of the geofence circle center (deg) Lon: Longitude of the geofence circle center (deg) Rad: Radius of the geofence circle (m) **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Enable the geofencing function: Send: $PAIR890,1,1,25.0567,121.5743,30*20\r\n Response: $PAIR001,890,0*3A\r\n ==> Success Disable the geofencing function: Send: $PAIR890,0*27\r\n Response: $PAIR001,890,0*3A\r\n ==> Success
PAIR_GEOFENCE_GET_CONFIG	**[Packet Type]** 891 **[Command]** PAIR_GEOFENCE_GET_CONFIG **[Description]** This command is used to get Geofencing configuration. **[Data Field]** **$PAIR891*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR891,<FenceNum>,<ConfLvl>,<Lat1>,<Lon1>,<Rad1>,...,<RadN>*CS<CR> <LF>** FenceNum(N): Number of geofences, the value is limited to 4. ConfLvl: The confidence level for state evaluation.

	'0' No requirement '1' 1-Sigma (68%) '2' 2-Sigma (95%) '3' 3-Sigma (99.7%) '4' 4-Sigma (99.9999%) '5' 5-Sigma (99.999999%) Lat: Latitude of the geofence circle center (deg) Lon: Longitude of the geofence circle center (deg) Rad: Radius of the geofence circle (m) **[Example]** Send: $PAIR891*3A\r\n Response: $PAIR001,891,0*3B\r\n ==> Success $PAIR891,1,1,25.0567,121.5743,30*21\r\n
PAIR_GEOFENCE_SET_GPIO_POLARITY	**[Packet Type]** 892 **[Command]** PAIR_GEOFENCE_SET_GPIO_POLARITY **[Description]** This command is used to set GPIO polarity for geofencing combined state. **[Data Field]** **$PAIR892,<GPIOPolarity>*CS<CR><LF>** GPIOPolarity: Pin polarity. '0' Low means outside '1' Low means inside Unknown state is always High. **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR892,1*24\r\n Response: $PAIR001,892,0*38\r\n ==> Success
PAIR_LOCUS_ENABLE	**[Packet Type]** 900 **[Command]**

	PAIR_LOCUS_ENABLE **[Description]** Enable or disable LOCUS save data. **[Data Field]** **$PAIR900,<Enable>*CS<CR><LF>** Enable: Enable or disable '0': Disable '1': Enable **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR900,1*2E\r\n ==> Enable LOCUS Response: $PAIR001,900,0*32\r\n ==> Enable Success
PAIR_LOCUS_GET_STATUS	**[Packet Type]** 901 **[Command]** PAIR_LOCUS_GET_STATUS **[Description]** Get LOCUS status. **[Data Field]** **$PAIR901*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR901,<Enable>*CS<CR><LF>** Enable: Enable or disable '0': Disable '1': Enable **[Example]** Send: $PAIR901*32\r\n Response: $PAIR001,901,0*33\r\n $PAIR901,0*2E\r\n ==> LOCUS is disable
PAIR_LOCUS_SET_MODE	**[Packet Type]** 902 **[Command]**

	PAIR_LOCUS_SET_MODE **[Description]** Set LOCUS saving mode. **[Data Field]** **$PAIR902,<Mode>,<Check_3D_Fix>*CS<CR><LF>** Mode: Saving Mode Normal, (1 << 0). Record per fix. Out of time, (1 << 1). Record every N s. N is customer configuration (**PAIR_LOCUS_SET_THRESHOLD**). Out of speed, (1 << 2). Record after speed more than N m/s. N is customer configuration (**PAIR_LOCUS_SET_THRESHOLD**). Out of distance, (1 << 3). Record after distance more than N m. N is customer configuration (**PAIR_LOCUS_SET_THRESHOLD**). Before entry sleep, (1 << 4). Record before entry sleep. User control, (1 << 5). Record after user send **PAIR_LOCUS_LOG_NOW**. Check_3D_Fix: Need check 3D fix or not. 0: not check. 1: need check. If set this type as 1, system will not save the location without 3D fixed. **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR902,6,1*36\r\n ==> Set mode as out of time & out of speed mode. Need check 3D fix. Response: $PAIR001,902,0*30\r\n ==> Set success. **[Note]** Must disable LOCUS saving before send this command.
PAIR_LOCUS_GET_MODE	**[Packet Type]** 903 **[Command]** PAIR_LOCUS_GET_MODE **[Description]** Get LOCUS saving mode. **[Data Field]** **$PAIR903*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR903,<Mode>,<Check_3D_Fix>*CS<CR><LF>** Mode: Saving Mode Normal, (1 << 0). Record per fix.

	Out of time, (1 << 1). Record every N s. N is customer configuration (**PAIR_LOCUS_SET_THRESHOLD**). Out of speed, (1 << 2). Record after speed more than N m/s. N is customer configuration (**PAIR_LOCUS_SET_THRESHOLD**). Out of distance, (1 << 3). Record after distance more than N m. N is customer configuration (**PAIR_LOCUS_SET_THRESHOLD**). Before entry sleep, (1 << 4). Record before going to sleep. User control, (1 << 5). Record after user send **PAIR_LOCUS_LOG_NOW**. Check_3D_Fix: Need check 3D fix or not. 0: not check. 1: need check. If set this type as 1, system will not save the location without 3D fixed. **[Example]** Send: $PAIR903*30\r\n Response: $PAIR001,903,0*31\r\n $PAIR903,6,1*37\r\n ==> LOCUS saving mode is out of time & out of speed mode. Need check 3D fix.
PAIR_LOCUS_SET_THRESHOLD	**[Packet Type]** 904 **[Command]** PAIR_LOCUS_SET_THRESHOLD **[Description]** Set LOCUS mode threshold. **[Data Field]** **$PAIR904,<Mode>,<Threshold>*CS<CR><LF>** Mode: Saving Mode 0: Out of time mode. 1: Out of speed mode. 2: Out of distance mode. Threshold: The threshold of saving mode. If mode == 0, out of time mode, the time threshold is 1s ~ 12hours. Unit is second. Default is 15s. If mode == 1, out of speed mode, the speed threshold is 1m/s ~ 100m/s. Unit is meters/second. Default is 1m/s. If mode == 2, out of distance mode, the distance threshold is 1m ~ 50000m. Unit is meter. Default is 1m. **[Return]** 1. **PAIR_ACK** for send result.

	[Example] end: $PAIR904,1,5*33\r\n ==> Set out of time mode threshold is 5s. Response: $PAIR001,904,0*36\r\n ==> Set success. LOCUS will save record every 5s. **[Note]** 1. Must disable LOCUS saving before send this command. 2. If the threshold out of range, it responds with a parameter error ("$PAIR001,904,4*32\r\n").
PAIR_LOCUS_GET_THRESHOLD	**[Packet Type]** 905**[Command]** PAIR_LOCUS_GET_THRESHOLD**[Description]** Get LOCUS mode threshold.**[Data Field]** **$PAIR905,<Mode>*CS<CR><LF>** Mode: Saving Mode 0: Out of time mode. 1: Out of speed mode. 2: Out of distance mode.**[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR905,<Threshold>*CS<CR><LF>** Threshold: The threshold of saving mode. If mode == 0, out of time mode, the time threshold is 1s ~ 12hours. Unit is second. Default is 15s. If mode == 1, out of speed mode, the speed threshold is 1m/s ~ 100m/s. Unit is meters/second. Default is 1m/s. If mode == 2, out of distance mode, the distance threshold is 1m ~ 50000m. Unit is meter. Default is 1m.**[Example]** Send: $PAIR905,0*2A\r\n ==> Get time threshold Response: $PAIR001,905,0*37\r\n $PAIR905,15*1E\r\n ==> Time threshold is 15s.
PAIR_LOCUS_CLEAR	**[Packet Type]** 906 **[Command]** PAIR_LOCUS_CLEAR **[Description]** Clear LOCUS Data. **[Data Field]**

	$PAIR906,<Type>*CS<CR><LF> Type: Clear Type 0: Clear record data and restore to default setting (configuration in gnss_config.bin). 1: Clear record data only. 2: Clear user setting. Restore to default setting. **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR906,0*29\r\n Response: $PAIR001,906,0*34\r\n **[Note]** Must disable LOCUS saving before send this command.
PAIR_LOCUS_LOG_NOW	**[Packet Type]** 907 **[Command]** PAIR_LOCUS_LOG_NOW **[Description]** Save current location data. **[Data Field]** **$PAIR907*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. **[Example]** Send: $PAIR907*34\r\n Response: $PAIR001,907,0*35\r\n **[Note]** Must keep user control (1 << 5) in saving mode if need use this command.
PAIR_LOCUS_GET_DATA	**[Packet Type]** 908 **[Command]** PAIR_LOCUS_GET_DATA **[Description]** Get all record data. **[Data Field]** **$PAIR908,<Type>*CS<CR><LF>** Type: Response type.

0: Response as NMEA.

1: Response as PAIR command.

[Return]

1. **PAIR_ACK** for send result.
2. **$PAIR908,0*CS<CR><LF>**

   LOCUS read begin
3. **$PAIR908,1,<Record_Num>,<Record_Size>*CS<CR><LF>**

   LOCUS read information.

   Record_Num: the total record numbers.

   Record_Size: the size of data per record.
4. LOGGA + LORMC.

   If type is 0, system will response LOGGA + GPGGA. The format is same as GPGGA + GPRMC.
5. **$PAIR908,2,<UTC>,<Fix_Type>,<Lat>,<Lon>,<Heighing>,<Speed>,<Heading>,<HDOP>,<SatNo>*CS<CR><LF>**

   If type is 1, system will response PAIR908,2,xxxx list for every record.

   None saved data will show 0.
6. **$PAIR908,3*CS<CR><LF>**

   LOCUS read end.**[Example]**

Send:

$PAIR908,0*27\r\n

Response:

$PAIR001,908,0*3A\r\n

$PAIR908,0*27\r\n

$PAIR908,1,2,16*13\r\n

$LOGGA,080931.000,011772.4267,N,0016183.7702,E,1,0,0.0,0.53,M,,M,,*59\r\n

$LORMC,080931.000,A,011772.4267,N,0016183.7702,E,260320,,,A,V*C\r\n

$LOGGA,080932.000,011772.4267,N,0016183.7702,E,1,0,0.0,0.53,M,,M,,*5A\r\n

$LORMC,080932.000,A,011772.4267,N,0016183.7702,E,260320,,,A,V*F\r\n

$PAIR908,3*24\r\n

Send:

$PAIR908,1*26\r\n

Response:

$PAIR001,908,0*3A\r\n

$PAIR908,0*27\r\n

$PAIR908,2,5EA541BB,01,12341A1C,3E06BA8C,0210,0000,0000,0000,00*04\r\n

$PAIR908,2,5EA541BC,01,12341A1B,3E06BA8A,0210,0000,0000,0000,00*06\r\n

$PAIR908,1,2,16*13\r\n

$PAIR908,3*24\r\n

[Note]

PAIR_LOCUS_GET_RECORD_NUM	Must disable LOCUS saving before send this command. **[Packet Type]** 909 **[Command]** PAIR_LOCUS_GET_RECORD_NUM **[Description]** Get total record number. **[Data Field]** **$PAIR909*CS<CR><LF>** **[Return]** 1. **PAIR_ACK** for send result. 2. **$PAIR909,<Record_Num>*CS<CR><LF>** Record_Num: total record number. **[Example]** Send: $PAIR909*3A\r\n Response: $PAIR001,909,0*3B\r\n $PAIR909,15*12\r\n ==> LOCUS has save 15 records.

LOCOSYS

LOCOSYS Technology Inc.
20F.-13, No.79, Sec. 1, Xintai 5th Rd.,
Xizhi Dist., New Taipei City 221, Taiwan
☎ 886-2-8698-3698
886-2-8698-3699
www.locosystech.com

Appendix - How to get checksum

Checksum value are the XOR checksum of the data between Preamble and '*'.
Example: $PMTK666,666*2E\0d\0a ==> Checksum value "2E" are the XOR checksum of "PMTK666,666".
Without "$" and without "*".
Calculation formula:

- checksum ^= *buffer
- CHK1 = (checksum >> 4) & 0x0F
- CHK2 = checksum & 0x0F

This is example code to get checksum value.

Example

```
void get_full_pair_command(int8_t* buf, int32_t buf_len, int8_t* temp_buf, int32_t temp_buf_len)
{
	const int32_t wait_ticket = 0xFFFFFFFF;
	int32_t ret_len = 0;
	int8_t* ind;
	uint8_t checkSumL = 0, checkSumR;

	if (buf_len + 6 > temp_buf_len) {
		return;
	}

	ind = buf;

	while(ind - buf < buf_len) {
		checkSumL ^= *ind;
		ind++;
	}

	temp_buf[0] = '$';
	memcpy(temp_buf + 1, buf, buf_len);
	temp_buf[buf_len + 1] = '*';
	checkSumR = checkSumL & 0x0F;
	checkSumL = (checkSumL >> 4) & 0x0F;
	temp_buf[buf_len + 2] = checkSumL >= 10 ? checkSumL + 'A' - 10 : checkSumL + '0';
	temp_buf[buf_len + 3] = checkSumR >= 10 ? checkSumR + 'A' - 10 : checkSumR + '0';
```

```
        temp_buf[buf_len + 4] = '\\';
        temp_buf[buf_len + 5] = '0';
        temp_buf[buf_len + 6] = 'd';
        temp_buf[buf_len + 7] = '\\';
        temp_buf[buf_len + 8] = '0';
        temp_buf[buf_len + 9] = 'a';
        temp_buf[buf_len + 10] = '\0';
        buf_len += 10;
}

void main()
{
    char command[1024] = {0};
    char command_checksum[1024] = {0};
    printf("Press command:\n");
    scanf("%s",command);
    get_full_pair_command(command, strlen(command), command_checksum, 1024);
    printf("%s\n",command_checksum);
    system("pause");
}
```

Input : PAIR666,666
Output : $PAIR666,666*26\0d\0a

LOCOSYS Technology Inc.
20F.-13, No.79, Sec. 1, Xintai 5th Rd.,
Xizhi Dist., New Taipei City 221, Taiwan
☏ 886-2-8698-3698
886-2-8698-3699
www.locosystech.com

Appendix - Datum List

All the datum type supported are shown in this table.

No	Datum	Region
0	WGS1984	International
1	Tokyo	Japan
2	Tokyo	Mean For Japan, South Korea, Okinawa
3	User Setting	User Setting
4	Adindan	Burkina Faso
5	Adindan	Cameroon
6	Adindan	Ethiopia
7	Adindan	Mali
8	Adindan	Mean For Ethiopia, Sudan
9	Adindan	Senegal
10	Adindan	Sudan
11	Afgooye	Somalia
12	Ain El Abd1970	Bahrain
13	Ain El Abd1970	Saudi Arabia
14	American Samoa1962	American Samoa Islands
15	Anna 1 Astro1965	Cocos Island
16	Antigua Island Astro1943	Antigua(Leeward Islands)
17	Arc1950	Botswana
18	Arc1950	Burundi
19	Arc1950	Lesotho
20	Arc1950	Malawi
21	Arc1950	Mean For Botswana, Lesotho, Malawi, Swaziland, Zaire, Zambia, Zimbabwe
22	Arc1950	Swaziland
23	Arc1950	Zaire

24	Arc1950	Zambia
25	Arc1950	Zimbabwe
26	Arc1960	Mean For Kenya Tanzania
27	Arc1960	Kenya
28	Arc1960	Tamzamia
29	Ascension Island1958	Ascension Island
30	Astro Beacon E 1945	Iwo Jima
31	Astro Dos 71/4	St Helena Island
32	Astro Tern Island (FRIG) 1961	Tern Island
33	Astronomical Station 1952	Marcus Island
34	Australian Geodetic 1966	Australia, Tasmania
35	Australian Geodetic 1984	Australia, Tasmania
36	Ayabelle Lighthouse	Djibouti
37	Bellevue (IGN)	Efate and Erromango Islands
38	Bermuda 1957	Bermuda
39	Bissau	Guuinea-Bissau
40	Bogota Observatory	Colombia
41	Bukit Rimpah	Indonesia(Bangka and Belitung Ids)
42	Camp Area Astro	Antarctica(McMurdi Camp Area)
43	Campo Inchauspe	Argentina
44	Canton Astro1966	Phoenix Island
45	Cape	South Africa
46	Cape Canaveral	Bahamas, Florida
47	Carthage	Tunisia
48	Chatham Island Astro1971	New Zealand(Chatham Island)
49	Chua Astro	Paraguay

50	Corrego Alegre	Brazil
51	Dabola	Guinea
52	Deception Island	Deception Island, Antarctia
53	Djakarta (Batavia)	Indonesia(Sumatra)
54	Dos 1968	New Georgia Islands (Gizo Island)
55	Easter Island 1967	Easter Island
56	Estonia Coordinate System1937	Estonia
57	European 1950	Cyprus
58	European 1950	Egypt
59	European 1950	England, Channel Islands, Scotland, Shetland Islands
60	European 1950	England, Ireland, Scotland, Shetland Islands
61	European 1950	Finland, Norway
62	European 1950	Greece
63	European 1950	Iran
64	European 1950	Italy (Sardinia)
65	European 1950	Italy (Slcily)
66	European 1950	Malta
67	European 1950	Mean For Austria, Belgium,Denmark, Finland, France, W Germany, Gibraltar, Greece, Italy, Luxembourg, Netherlands, Norway, Portuga,l Spain, Sweden, Switzerland
68	European 1950	Mean For Austria, Debnmark,France, W Germany, Netherland ,Switzerland
69	European 1950	Mean For Irag, Israel, Jordan, Lebanon, Kuwait, Saudi Arabia, Syria
70	European 1950	Portugal, Spain
71	European 1950	Tunisia,
72	European 1979	Mean For Austria, Finland ,Netherlands ,Norway, Spain, Sweden, Switzerland
73	Fort Thomas 1955	Nevis St Kitts (Leeward Islands)
74	Gan 1970	Republic Of Maldives
75	Geodetic Dataum 1970	New Zealand
76	Graciosa Base SW1948	Azores (Faial, Graciosa, Pico, Sao, Jorge, Terceria)
77	Guam1963	Guam

78	Gunung Segara	Indonesia (Kalimantan)
79	Gux l Astro	Guadalcanal Island
80	Herat North	Afghanistan
81	Hermannskogel Datum	Croatia-Serbia, Bosnia-Herzegoivna
82	Hjorsey 1955	Iceland
83	Hongkong 1963	Hongkong
84	Hu Tzu Shan	Taiwan
85	Indian	Bangladesh
86	Indian	India,Nepal
87	Indian	Pakistan
88	Indian 1954	Thailand
89	Indian 1960	Vietnam (Con Son Island)
90	Indian 1960	Vietnam (Near 16 deg N)
91	Indian 1975	Thailand
92	Indonesian 1974	Indonesian
93	Ireland 1965	Ireland
94	ISTS 061 Astro 1968	South Georgia Islands
95	ISTS 073 Astro 1969	Diego Garcia
96	Johnston Island 1961	Johnston Island
97	Kandawala	Sri Lanka
98	Kerguelen Island 1949	Kerguelen Island
99	Kertau 1948	West Malaysia and Singapore
100	Kusaie Astro 1951	Caroline Islands
101	Korean Geodetic System	South Korea
102	LC5 Astro 1961	Cayman Brac Island
103	Leigon	Ghana
104	Liberia 1964	Liberia
105	Luzon	Philippines (Excluding Mindanao)
106	Luzon	Philippines (Mindanao)

107	M'Poraloko	Gabon
108	Mahe 1971	Mahe Island
109	Massawa	Ethiopia (Eritrea)
110	Merchich	Morocco
111	Midway Astro 1961	Midway Islands
112	Minna	Cameroon
113	Minna	Nigeria
114	Montserrat Island Astro 1958	Montserrat (Leeward Island)
115	Nahrwan	Oman (Masirah Island)
116	Nahrwan	Saudi Arabia
117	Nahrwan	United Arab Emirates
118	Naparima BWI	Trinidad and Tobago
119	North American 1927	Alaska (Excluding Aleutian Ids)
120	North American 1927	Alaska (Aleutian Ids East of 180 degW)
121	North American 1927	Alaska (Aleutian Ids West of 180 degW)
122	North American 1927	Bahamas (Except San Salvador Islands)
123	North American 1927	Bahamas (San Salvador Islands)
124	North American 1927	Canada (Alberta, British Columbia)
125	North American 1927	Canada (Manitoba, Ontario)
126	North American 1927	Canada (New Brunswick, Newfoundland, Nova Scotia, Qubec)
127	North American 1927	Canada (Northwest Territories, Saskatchewan)
128	North American 1927	Canada (Yukon)
129	North American 1927	Canal Zone
130	North American 1927	Cuba
131	North American 1927	Greenland (Hayes Peninsula)
132	North American 1927	Mean For Antigua, Barbados, Barbuda, Caicos Islands, Cuba, Dominican, Grand Cayman, Jamaica, Turks Islands
133	North American 1927	Mean For Belize, Costa Rica, El Salvador, Guatemala, Honduras, Nicaragua
134	North American 1927	Mean For Canada

135	North American 1927	Mean For Conus
136	North American 1927	Mean For Conus (East of Mississippi, River Including Louisiana,Missouri, Minnesota)
137	North American 1927	Mean For Conus (West of Mississippi, Rive Excluding Louisiana, Minnesota, Missouri)
138	North American 1927	Mexico
139	North American 1983	Alaska (Excluding Aleutian Ids)
140	North American 1983	Aleutian Ids
141	North American 1983	Canada
142	North American 1983	Conus
143	North American 1983	Hahawii
144	North American 1983	Mexico, Central America
145	North Sahara 1959	Algeria
146	Observatorio Meteorologico 1939	Azores (Corvo and Flores Islands)
147	Old Egyptian 1907	Egypt
148	Old Hawaiian	Hawaii
149	Old Hawaiian	Kauai
150	Old Hawaiian	Maui
151	Old Hawaiian	Mean For Hawaii, Kauai, Maui, Oahu
152	Old Hawaiian	Oahu
153	Oman	Oman
154	Ordnance Survey Great Britian 1936	England
155	Ordnance Survey Great Britian 1936	England, Isle of Man, Wales
156	Ordnance Survey Great Britian 1936	Mean For England ,Isle of Man, Scotland, Shetland Island, Wales
157	Ordnance Survey Great Britian 1936	Scotland, Shetland Islands
158	Ordnance Survey Great Britian 1936	Wales
159	Pico de las Nieves	Canary Islands

160	Pitcairn Astro 1967	Pitcairn Island
161	Point 58	Mean For Burkina Faso and Niger
162	Pointe Noire 1948	Congo
163	Porto Santo 1936	Porto Santo, Maderia Islands
164	Provisional South American 1956	Bolovia
165	Provisional South American 1956	Chile (Northern Near 19 deg S)
166	Provisional South American 1956	Chile (Southern Near 43 deg S)
167	Provisional South American 1956	Colombia
168	Provisional South American 1956	Ecuador
169	Provisional South American 1956	Guyana
170	Provisional South American 1956	Mean For Bolivia Chile,Colombia, Ecuador, Guyana, Peru, Venezuela
171	Provisional South American 1956	Peru
172	Provisional South American 1956	Venezuela
173	Provisional South Chilean 1963	Chile (Near 53 deg S) (Hito XVIII)
174	Puerto Rico	Puerto Rico, Virgin Islands
175	Pulkovo 1942	Russia
176	Qatar National	Qatar
177	Qornoq	Greenland (South)
178	Reunion	Mascarene Island
179	Rome 1940	Italy (Sardinia)
180	S-42 (Pulkovo 1942)	Hungary
181	S-42 (Pulkovo 1942)	Poland
182	S-42 (Pulkovo 1942)	Czechoslavakia

183	S-42 (Pulkovo 1942)	Lativa
184	S-42 (Pulkovo 1942)	Kazakhstan
185	S-42 (Pulkovo 1942)	Albania
186	S-42 (Pulkovo 1942)	Romania
187	S-JTSK	Czechoslavakia (Prior 1 Jan1993)
188	Santo (Dos) 1965	Espirito Santo Island
189	Sao Braz	Azores (Sao Miguel, Santa Maria Ids)
190	Sapper Hill 1943	East Falkland Island
191	Schwarzeck	Namibia
192	Selvagem Grande 1938	Salvage Islands
193	Sierra Leone 1960	Sierra Leone
194	South American 1969	Argentina
195	South American 1969	Bolivia
196	South American 1969	Brazial
197	South American 1969	Chile
198	South American 1969	Colombia
199	South American 1969	Ecuador
200	South American 1969	Ecuador (Baltra, Galapagos)
201	South American 1969	Guyana
202	South American 1969	Mean For Argentina, Bolivia, Brazil,Chile, Colombia, Ecuador, Guyana, Paraguay, Peru, Trinidad and Tobago, Venezuela
203	South American 1969	Paraguay
204	South American 1969	Peru
205	South American 1969	Trinidad and Tobago
206	South American 1969	Venezuela
207	South Asia	Singapore
208	Tananarive Observatory 1925	Madagascar
209	Timbalai 1948	Brunei, E Malaysia (Sabah Sarawak)
210	Tokyo	Japan

LOCOSYS Technology Inc.
20F.-13, No.79, Sec. 1, Xintai 5th Rd.,
Xizhi Dist., New Taipei City 221, Taiwan
☏ 886-2-8698-3698
886-2-8698-3699
www.locosystech.com

211	Tokyo	Mean For Japan, South Korea, Okinawa
212	Tokyo	Okinawa
213	Tokyo	South Korea
214	Tristan Astro 1968	Tristam Da Cunha
215	Viti Levu 1916	Fiji (Viti Levu Island)
216	Voirol 1960	Algeria
217	Wake Island Astro 1952	Wake Atoll
218	Wake-Eniwetok 1960	Marshall Islands
219	WGS 1972	Global Definition
220	WGS 1984	Global Definition
221	Yacare	Uruguay
222	Zanderij	Suriname
223	PZ-90 v11	GLONASS

LOCOSYS Technology Inc.
20F.-13, No.79, Sec. 1, Xintai 5th Rd.,
Xizhi Dist., New Taipei City 221, Taiwan
☎ 886-2-8698-3698
886-2-8698-3699
www.locosystech.com

Document change list

Revision 1.0

- Initial version released on May 27, 2024.