DCON Command Set

for GPS-721U-MRTU Series Module

A set of simple and easy-to-use DCON commands are supported on the GPS-721U-MRTU series module for users to obtain GPS data by sending commands. This manual explains the format and the usage of each command.

V1.0.0 • Feb. 2025

Written by Tony Lee

Edited by Sunny Chiu

Table of Contents

1. DCON Command Format

The DCON protocol consists of commands sent by a PC host and responses transmitted by a remote I/O module. Each module has a unique ID number that is used for addressing purposes. The DCON commands start with a delimiter character, followed by a 2-character hexadecimal module ID, the command characters, and end with a CR character.

DCON Command	Delimiter	Module ID	Command	[CHKSUM]	CR
Example	$	01	I	-	[CR]

Only the addressed module will execute the command and respond the result. The format of a response string is similar to a command. It starts with a delimiter character, typically followed by a **2-character hexadecimal module address**, the requested data, and ends with a CR character.

Response	Delimiter	Module ID	Data	[CHKSUM]	CR
Example	!	01	0	-	[CR]

Delimiter: defines the type of command or response. Commonly used characters including “~”, “$”, “@”, “%”, “!”, “?” and “>”

Module ID: the RS-485 address of the I/O module.

Command/Response: varies depending on the definition of each command.

CHKSUM: a 2-character hexadecimal checksum code. If checksum is disabled, there is no need to add the CHKSUM code.

CR: end of command character, carriage return (0x0D)

Checksum Calculation

The letters of checksum are limited to uppercase letters. Checksum is calculated with sum of the ASCII code of all the characters (except [CR]) in a command or response string, and then perform a masking operation on the sum with 0xFFh. It can be said to only take the last two characters.

Example

Command string: $012(CR):

1. Sum of strings = “$” + “0” + “1” + “2” = 24h + 30h + 31h + 32h = B7h
2. The checksum is B7h, that is, [CHKSUM] = “B7”
3. Command string with [CHKSUM] = $012B7(CR)

Response string: !01200600(CR)

1. Sum of the string = “!”+”0”+”1”+”2”+”0”+”0”+”6”+”0”+”0”

   = 21h+30h+31h+32h+30h+30h+36h+30h+30h = 1AAh
2. Therefore the checksum is AAh, and so CHKSUM = “AA”
3. The response string with the checksum = !01200600AA(CR)

Note:
The 2 characters of checksum should be in upper case.

1.1. Soft INIT Function

In applications using the DCON protocol, the configuration parameters including baud rate, parity, checksum and protocol are available to be modified when the Init/Normal switch is placed in the Init position only. Sometimes, the GPS-721U-MRTU may be mounted in a location where manual configuring of the Init/Normal switch on the rear of the module is troublesome.

Therefore, the GPS-721U-MRTU series module provides Soft INIT function for users to remotely set the module to be temporarily in the state where the Init/Normal switch is in the Init position.

The procedure of enabling the Soft INIT function:

Step1. Send command ~AATnn (Section 2.1.8) to set the duration of Soft INIT function

Step2. Send command ~AAI (Section 2.1.7) to enable the Soft INIT function

Step3. Send command %AANNTTCCFF or other setting command in the duration of the Soft INIT function

Commands that need to be set in Init mode to take effect:

Command	Description	Section
%AANNTTCCFF	Sets the module configuration	2.1.1
$AAPN	Sets the protocol of transmission mode 0	2.2.8
~AAMN	Sets the transmission mode	2.2.6
~AAGnn	Sets the satellite system to GPS or GLONASS	2.2.4

2. DCON Command List

Command	Description	Section
%AANNTTCCFF	Sets the module configuration	2.1.1
#**	Synchronized sampling of DO	2.3.7
#AA	Reads GPS information	2.2.1
#AAN	Reads specified GPS information	2.2.2
#AA00(Data)	Sets all DO status	2.3.9
#AA0A(Data)	Sets all DO status	2.3.10
#AA1cDD	Sets one DO status	2.3.11
#AAAcDD	Sets one DO status	2.3.12
$AA2	Reads the module configuration	2.1.2
$AA4	Reads the synchronized DO data	2.3.8
$AA5	Reads the reset status	2.1.3
$AA6	Reads the DO status	2.3.13
$AAC	Clears the latched status of DO	2.3.5
$AAF	Reads the firmware version	2.1.4
$AAI	Reads the status of the Init/Normal switch	2.1.9
$AALS	Reads the latched status of DO	2.3.6
$AAM	Reads the module name	2.1.5
$AAP	Reads the protocol of transmission mode 0	2.2.7
$AAPN	Sets the protocol of transmission mode 0	2.2.8
@AA	Reads the DO status	2.3.1
@AA(Data)	Sets the DO status	2.3.2

Command	Description	Section
~AAD	Reads the definition of DO value	2.3.3
~AADVV	Sets the definition of DO value	2.3.4
~AAG	Reads the satellite system from which the GPS module obtains positioning signals.	2.2.3
~AAGnn	Sets the satellite system to GPS or GLONASS	2.2.4
~AAI	Enables the Soft INIT	2.1.7
~AAM	Reads the transmission mode	2.2.5
~AAMN	Sets the transmission mode	2.2.6
~AAO(Name)	Sets the module name	2.1.6
~AARD	Reads the response delay time	2.2.9
~AARDVV	Sets the response delay time	2.2.10
~AATnn	Sets the duration during which the Soft INIT is activated	2.1.8
~**	Clears host watchdog timeout timer	2.4.2
~AA0	Reads host watchdog status	2.4.3
~AA1	Clears host watchdog timeout status	2.4.4
~AA2	Reads host watchdog enable/disable status and timeout value	2.4.5
~AA3ETT	Enables/disables host watchdog and sets timeout value	2.4.6
~AA4V	Reads power-on value or safe value of DO	2.4.7
~AA5V	Sets power-on value or safe value of DO	2.4.8

2.1. General Commands

General commands include commands to read and write module configuration, module name, reset status, and to enable Soft INIT.

Command	Description	Section
%AANNTTCCFF	Sets the module configuration	2.1.1
$AA2	Reads the module configuration	2.1.2
$AA5	Reads the reset status	2.1.3
$AAF	Reads the firmware version	2.1.4
$AAM	Reads the module name	2.1.5
~AAO(Name)	Sets the module name	2.1.6
~AAI	Enables the Soft INIT	2.1.7
~AATnn	Sets the duration during which the Soft INIT is activated	2.1.8
$AAI	Reads the status of the Init/Normal switch	2.1.9

2.1.1. %AANNTTCCFF

Sets the module configuration

➢ **Syntax:**

%AANNTTCCFF[CHK](CR)

%	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
NN	New address for the module in hexadecimal format
TT	Type code, fixed to 40
CC	Sets baud rate and parity. Baud rate can only be changed when the Init/Normal switch is in the Init position.
FF	Enables/disables checksum. Before modifying the checksum, toggle the Init/Normal switch to the Init position.

➢ **Baud Rate & Parity (CC):**

Bit	7	6	5	4	3	2	1	0
Data	Parity Code		Baud Rate Code					

Parity (Bit 7: 6)

Code	00	01	10	11
Parity	N,8,1	N,8,2	E,8,1	O,8,1

Baud Rate (Bit 5: 0)

Code	03	04	05	06	07	08	09	0A
Baud Rate	1200	2400	4800	9600	19200	38400	57600	115200

➢ **Data Format (FF):**

Bit	7	6	5	4	3	2	1	0
Key	Reserved	**CS**	Reserved					

Key	Description
CS	Checksum Setting 0: Disable 1: Enable

Note:
Reserved bits should be filled with 0.

➢ **Response:**

Valid Command: **!AA[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)

➢ **Example:**

Command	Response
%0109400600	!09
Sets the module address from 01 to 09 Returns the command is valid	
$092	!09400600
Reads the configuration of module with address 09 Returns the configuration is baud rate 9600 bps, checksum disabled	
%0905400940	?09
Changes the baud rate to 57600bps Returns the command is invalid. Check whether the Init/Normal switch is placed in the Init position	
%0909400940	!09
Changes the Baud rate to 57600bps Returns the command is valid	

Note:

The new address is applied immediately after receiving a valid command. However, the new baud rate and new checksum will take effect after the next power cycle.

If the GPS-721U-MRTU is installed in a place where it is not easy to remove to toggle the Init/Normal switch, you can use the Soft INIT function provided to remotely set the module to the Init mode temporarily.
Refer to Chapter 1.1. Soft INIT Function for setting steps.

➢ **Related Commands:**

Section 2.1.2 $AA2, Section 2.1.9 $AAI, Section 2.1.7 ~AAI, Section 2.1.8 ~AATnn

2.1.2. $AA2

Reads the module configuration

- **Syntax:**

$AA2[CHK](CR)

$	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
2	Command to read the module configuration

- **Response:**

Valid Command: **!AATTCCFF[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)
TT	Type code, fixed to 40
CC	Baud rate and parity
FF	Checksum enabled/disabled

- **Baud Rate & Parity (CC):**

Bit	7	6	5	4	3	2	1	0
Data	Parity Code		Baud Rate Code					

Parity (Bit 7: 6)

Code	00	01	10	11
Parity	N,8,1	N,8,2	E,8,1	O,8,1

Baud Rate (Bit 5: 0)

Code	03	04	05	06	07	08	09	0A
Baud Rate	1200	2400	4800	9600	19200	38400	57600	115200

➢ **Data Format (FF):**

Bit	7	6	5	4	3	2	1	0
Key	Reserved	**CS**	Reserved					

Key	Description
CS	Checksum Setting 0: Disabled 1: Enabled

➢ **Example:**

Command	**Response**
$012	!01400A00
Reads the configuration of module with address 01. Returns baud rate is 115200 bps, and checksum is disabled.	
$052	!05400940
Reads the configuration of module with address 05. Returns baud rate is 57600 bps, and checksum is enabled.	

➢ **Related Commands:**

Section 2.1.1 %AANNTTCCFF

2.1.3. $AA5

Reads the module reset status

- **Syntax:**

$AA5[CHK](CR)

$	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
5	Command to read the module reset status

- **Response:**

Valid Command: **!AAS[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)
S	The module reset status 1: it is the first time to read the reset status since the module was powered on 0: It is not the first time to read the reset status since the module was powered on, which denotes that the module has not restarted since the last time the $AA5 command was used.

- **Example:**

Command	**Response**
$035	!031
Reads the reset status of module address 03. Returns that it is the first read of the reset status since the module was powered-on.	
$035	!030
Reads the reset status of module address 03. Returns that it is **NOT** the first read of the reset status since the module was powered-on. The module has not restarted since the last $AA5 command was sent.	

2.1.4. $AAF

Reads the firmware version

- **Syntax:**

$AAF[CHK](CR)

$	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
F	Command to read the firmware version

- **Response:**

Valid Command: **!AA[(Data)CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)
(Data)	Firmware version of the module

- **Example:**

Command	**Response**
$01F	!0101.20
Reads the firmware version of module with address 01. Returns the firmware version is 1.20.	

- **Related Commands:**

None

2.1.5. $AAM

Reads the module name

- **Syntax:**

$AAM[CHK](CR)

$	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
M	Command to read the module name

- **Response:**

Valid Command: **!AA(Data)[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)
(Data)	Module name

- **Example:**

Command	**Response**
$01M	!01GPS721
Reads the name of the module with address 01. Returns the module name is GPS721.	

- **Related Commands:**

Section 2.1.6 ~AAO(Name)

2.1.6. ~AAO(Name)

Sets the name of the module

- **Syntax:**

~AAO(Name)[CHK](CR)

~	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
O	Command to set the module name
(Name)	The new name for the module (up to 6 characters)

- **Response:**

Valid Command: **!AA[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)

- **Example:**

Command	**Response**
$01M	!01GPS721
Read the name of the module with address 01. Returns the command is valid, module name is GPS721.	
~01O87000	!01
Sets the name for the module with address 01 to 87000. Returns the command is valid.	
$01M	!0187000
Read the name of the module with address 01. Returns the command is valid, module name is 87000.	

- **Related Commands:**

Section 2.1.5 $AAM

2.1.7. ~AAI

Enables the “Soft INIT” function to allow the module to be temporarily in the state where the Init/Normal switch is in the Init position. The “~AATnn” command is used to set the duration of the “Soft INIT” mode. During that time, “%AANNTTCCFF” command can change baud rate and checksum of the module without toggling the Init/Normal switch on the rear of the module.

- **Syntax:**

~AAI[CHK](CR)

~	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
I	Command to enable the “Soft INIT” function

- **Response:**

Valid Command: **!AA[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)

- **Example:**

Refer to the example in the next section.

- **Related Commands:**

Section 2.1.8 ~AATnn

2.1.8. ~AATnn

Sets the duration during which the Soft INIT is activated.

It is used before the command “~AAI”. When you change baud rate, checksum, protocol or transmission mode, the Init/Normal switch must be placed in the Init position. “~AAI” and "~AATnn" commands can set the module to be temporarily in the state where the Init/Normal switch is in the Init position. It eliminates the need to toggle the hardware Init/Normal switch.

- **Syntax:**

~AATnn[CHK](CR)

~	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
T	Command to set the Soft INIT duration
nn	A 2-digit hexadecimal value that specifies the Soft INIT duration The maximum duration is 60 seconds Example: a value of 10 = 16 seconds

- **Response:**

Valid Command: **!AA[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)

➢ **Example:**

Command	Response
~01I	!01
Enables the “Soft INIT” function of the module with address 01. Returns the command is valid.	
%0107400740	?01
Sets the module address to 07, baud rate to 19200 bps and enables checksum. Returns the command is invalid. Because the “Soft INIT” duration has not been set using the “~AATnn” command.	
~01T10	!01
Sets the “Soft INIT” duration to 16 (0x10) seconds. Returns the command is valid.	
~01I	!01
Enables the “Soft INIT” function of the module with address 01. Returns the command is valid.	
%0107400740	!07
Sets the module address to 07, baud rate to 19200 bps and enables checksum. Returns the command is valid.	

➢ **Related Commands:**

Section 2.1.7 ~AAI, Section 2.1.1 %AANNTTCCFF

2.1.9. $AAI

Reads the status of the Init/Normal switch on the rear of the module

- **Syntax:**

$AAI[CHK](CR)

$	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
I	Command to read the status of the Init/Normal switch

- **Response:**

Valid Command: **!AA(Data)[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)
(Data)	The status of the Init/Normal switch where: 0: the Init/Normal switch is placed in the Init position 1: the Init/Normal switch is placed in the Normal position

- **Example:**

Command	**Response**
$01I	!010
Reads the status of Init/Normal switch of the module with address 01. Returns the command is valid, the Init/Normal switch is placed in the Init position.	
$01I	!011
Reads the status of Init/Normal switch of the module with address 01. Returns the command is valid, the Init/Normal switch is placed in the Normal position.	

- **Related Commands:**

Section 2.1.1 %AANNTTCCFF, Section 2.1.2 $AA2

2.2. GPS Commands

The GPS commands are used to read GPS information, and set the transmission mode.

Command	Description	Section
#AA	Reads GPS information	2.2.1
#AAN	Reads specified GPS information	2.2.2
~AAG	Reads the satellite system from which the positioning data is being gathered	2.2.3
~AAGnn	Sets the satellite system to GPS or GLONASS	2.2.4
~AAM	Reads the transmission mode	2.2.5
~AAMN	Sets the transmission mode	2.2.6
$AAP	Reads the protocol for transmission mode 0	2.2.7
$AAPN	Sets the protocol for transmission mode 0	2.2.8
~AARD	Reads the response delay time	2.2.9
~AARDVV	Sets the response delay time	2.2.10

2.2.1. #AA

Reads the UTC Time, latitude, longitude, and more GPS information

- **Syntax:**

#AA[CHK](CR)

#	Delimiter character
AA	The module address in hexadecimal format (00 to FF)

- **Response:**

Valid Command: **!AATTTTTT.TTT,LLLL.LLLL,C,NNNNN.NNNN,C,P,S[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)
TTTTTT.TTT	The UTC Time (hhmmss.sss)
LLLL.LLLL	Latitude (ddmm.mmmm)
C	N or S (North or South)
NNNNN.NNNN	Longitude (dddmm.mmmm)
C	E or W (East or West)
P	Position Fix Indicator 0= Invalid GPS fix 1= GPS SPS Mode, fix valid 2= Differential GPS,SPS Mode, fix valid 6= Estimated (Dead Reckoning) fix
S	Number of the satellites in view with a range of 0 to 12

Note:
If there are insufficient GPS satellites in view, the data will be set to 9.

➢ **Example:**

Command	Response
#01	!01999999.999,9999.9999,9,99999.9999,9,0,00
Reads the UTC time, latitude, longitude, mode and number of satellites in view for a module Returns the command is valid. The response shows the GPS data is invalid because there are insufficient GPS satellites in view.	
#01	!01035035.00,2451.7056,N,12100.9803,E,1,9
Reads the UTC time, latitude, longitude, position fix indicator and number of satellites in view for a module Returns the following information The UTC Time: 035035.000 Latitude, Longitude: 2451.7056,N, 12100.9903,E Position Fix Indicator: 1 (GPS SPS Mode, fix valid) Satellites in view: 9	

➢ **Related Commands:**

Section 2.2.2 #AAN

2.2.2. #AAN

Reads the UTC time, date, altitude, coordinates in Google Maps or GPS format, and the number of satellites in view respectively

- **Syntax:**

#AAN[CHK](CR)

#	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
N	1. the UTC Time 2: Latitude and longitude 3: Number of satellites in view 4: Date 5. Latitude and longitude in Google Maps format 6: the HDOP (horizontal dilution of precision) value 7: the altitude in meters 8: the HOG (geoid separation) in meters 9: the speed over ground in 0.1 knot steps or 0.1852 km/h

- **Response:**

Valid Command: **!AAData[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
>	Delimiter to indicate that the #AA5 command was valid
AA	The module address in hexadecimal format (00 to FF)
Data	N=1: TTTTTT.TTT (The UTC Time) N=2: LLLL.LLLL,C,NNNNN.NNNN,C (Latitude and longitude) N=3: SS (The number of satellites in view) N=4: DDMMYY (Date) N=5: >LL.LLLLLLΔC,NNN.NNNNNNΔC

➢ **Data:**

TTTTTT.TTT	The UTC time (hhmmss.sss)
LLLL.LLLL	Latitude (ddmm.mmmm)
C	N or S (North or South)
NNNNN.NNNN	Longitude (dddmm.mmmm)
C	E or W (East or West)
S	The number of satellites in view with range from 0 ~ 12
DDMMYY	Date: DD (day), MM (month), YY (year)
LL.LLLLLL	The Latitude value in decimal degrees in Google Maps format
NNN.NNNNNN	The Longitude value in decimal degrees in Google Maps format
Δ	Space character

Note:
If there are insufficient GPS satellites in view, the data will be set to 9.

➢ **Example:**

Command	**Response**
#011	!01999999.999
Reads the UTC time of module with address 01 Returns the command is valid. The response shows the UTC time is invalid.	
#012	!012451.7057,N,12100.9904,E
Reads the latitude and longitude of module with address 01 Returns the latitude and longitude information is 2451.7057, N,12100.9904, E	
#013	!019
Reads the number of satellites in view for module with address 01 Returns the number of satellites in view is 9	
#014	!01280308
Reads the date from module with address 01 Returns the date is 3/28/2008	
#015	>24.861750 N,121.016200 E
Reads the latitude and longitude in Google Maps format from module with address 01 Returns the that the latitude is 24.861750 N (degrees), and the longitude is 121.016200 E (degrees)	

2.2.3. ~AAG

Reads the satellite system from which the positioning data is being gathered
This command is available in firmware version 2.80 and later.

➢ **Syntax:**

~AAG[CHK](CR)

~	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
G	Command to read the transmission mode.

➢ **Response:**

Valid Command: **!AAS[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)
S	The satellite system used by the module 00: GPS 01: GLONASS

➢ **Example:**

Command	**Response**
~01G	!0100
Reads the satellite system used by the module with address 01. Returns the command is valid, the satellite system is GPS.	
~03G	!0301
Reads the satellite system used by the module with address 03. Returns the command is valid, the satellite system is GLONASS.	

➢ **Related Commands:** Section 2.2.4 ~AAGnn

2.2.4. ~AAGnn

Sets the satellite system to GPS or GLONASS

This command is available in firmware version 2.80 and later.

- **Syntax:**

~AAGN[CHK](CR)

~	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
G	Command to set the transmission mode.
nn	The satellite system 00: GPS 01: GLONASS

- **Response:**

Valid Command: **!AA[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)

- **Example:**

Command	**Response**
~03G00	!03
Sets the satellite system used by module with address 03 to GPS Returns the command is valid.	
~03G01	!03
Sets the satellite system used by module with address 03 to GLONASS Returns the command is valid.	

- **Related Commands:** Section 2.2.3 ~AAG

2.2.5. ~AAM

Reads the transmission mode of the module

➢ **Syntax:**

~AAM[CHK](CR)

~	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
M	Command to read the transmission mode.

➢ **Response:**

Valid Command: **!AASC[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)
S	Transmission modes supported by the module 0: DCON 3: DCON\Modbus RTU\NMEA
C	The current transmission mode in use. 0: DCON or Modbus (RS-232 or RS-485 selectable by wiring) 2: NMEA (RS-232 or RS-485 selectable by wiring)

Note:

1. Use $AAPN command to select DCON or Modbus protocol for GPS-721U-MRTU series module.
2. While NMEA mode is enabled, the NMEA data will be continuously transmitted via RS-485, so the host cannot communicate with other devices connected on the same RS-485 network.

➢ **Example:**

Command	Response
~03M	!0330
Reads the transmission mode of module with address 03. Returns the command is valid, this module supports DCON\Modbus RTU\NMEA. The transmission mode is set to DCON or Modbus RTU via RS-232 or RS-485.	
~03M2	!03
Sets the transmission mode to 2, sending NMEA messages via RS-232 and RS-485. Returns the command is valid. The new setting will take effect after the next power cycle. During this time, the host cannot communicate with other devices connected on the same RS-485 network.	
~03M	!0332
Reads the transmission mode of module with address 03. Returns the command is valid, this module supports DCON\Modbus RTU\NMEA. The transmission mode is 2, sending NMEA messages via RS-232 port and RS-485.	

➢ **Related Commands:**

Section 2.2.6 ~AAMN, Section 2.2.7 $AAP, Section 2.2.8 $AAPN

2.2.6. ~AAMN

Sets the transmission mode of the module

Before changing the transmission mode, the Init/Normal switch must be in the Init position. The new setting will take effect after the module is restarted.

- **Syntax:**

~AAMN[CHK](CR)

~	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
M	Command to set the transmission mode.
N	The transmission mode. 0: DCON or Modbus (RS-232 or RS-485 selectable by wiring) 2: NMEA (RS-232 or RS-485 selectable by wiring)

- **Response:**

Valid Command: **!AA[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)

➢ **Example:**

Command	Response
~03M	!0330
Reads the transmission mode of module with address 03. Returns the command is valid, this module supports DCON\Modbus RTU\NMEA. The transmission mode is set to using DCON or Modbus RTU.	
~03M2	!03
Sets the transmission mode to 2 to send NMEA messages via RS-232 or RS-485 Returns the command is valid. The new setting will take effect after the module is restarted. Keep in mind that the host cannot communicate with other devices connected on the RS-485 interface when the transmission mode 2 is enabled.	
~03M	!0332
Reads the transmission mode of module with address 03. Returns the command is valid, this module supports DCON\Modbus RTU\NMEA. The transmission mode is set to 2, sending NMEA messages via RS-232 or RS-485.	

➢ **Related Commands:**

Section 2.2.5 ~AAM, Section 2.2.7 $AAP, Section 2.2.8 $AAPN

2.2.7. $AAP

Reads the protocol for transmission mode 0

➢ **Syntax:**

$AAP[CHK](CR)

$	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
P	Command to read protocol for transmission mode 0

➢ **Response:**

Valid Command: **!AASC[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)
S	The protocols supported by the module: 0: DCON only 1: DCON and Modbus RTU protocols 3: DCON and Modbus RTU/ASCII protocols
C	The protocol saved in EEPROM, which will be applied after restart 0: DCON 1: Modbus RTU 3: Modbus ASCII

➢ **Example:**

Command	**Response**
$01P	!0130

Reads the protocol for transmission mode 0 of module with address 01.
Returns the command is valid. The module supports DCON and Modbus RTU/ASCII protocols, and the protocol that will be applied after restart is DCON.

➢ **Related Commands:** Section 2.2.8 $AAPN

2.2.8. $AAPN

Sets the protocol for transmission mode 0

Note:

Before changing the protocol, the Init/Normal switch on rear of the module must be placed in the Init position. The new protocol will be saved in the EEPROM and be applied after the next power cycle.

➢ **Syntax:**

$AAPN[CHK](CR)

$	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
P	Command to set the protocol for transmission mode 0
N	The protocol to be used: 0: DCON 1: Modbus RTU 3: Modbus ASCII

➢ **Response:**

Valid Command: **!AA[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)

➢ **Example:**

Command	Response
$01P1	?01
Sets the protocol for transmission mode 0 of module with address 01 to Modbus RTU. Returns the command is invalid, because the Init/Normal module is **not** in the Init position.	
$01P1	!01
Sets the protocol for transmission mode 0 of module with address 01 to Modbus RTU. Returns the command is valid.	

➢ **Related Commands:**

Section 2.2.7 $AAP

2.2.9. ~AARD

Reads the Response Delay time

- **Syntax:**

~AARD[CHK](CR)

~	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
RD	Command to read the response delay time

- **Response:**

Valid Command: **!AA(Data)[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a ignored command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)
(Data)	A 2-digit hexadecimal value to represent the response delay time in a range of 0 to 30 ms

- **Example:**

Command	**Response**
~03RD1E	!03
Sets the response delay time for module with address 03 to 1E (30 ms) Returns the command is valid.	
~03RD	!0300
Reads the response delay time for module with address 03 Returns the command is valid, the response delay time is 0 ms.	

- **Related Commands:**

Section 2.2.10 ~AARDVV

2.2.10. ~AARDVV

Sets the Response Delay time

- **Syntax:**

~AARDVV[CHK](CR)

~	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
RD	Command to set the response delay time
VV	A 2-digit hexadecimal value to represent the response delay time in a range of 0 to 30 ms

- **Response:**

Valid Command: **!AA(Data)[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a ignored command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)

- **Example:**

Command	**Response**
~03RD1E	!03
Sets the response delay time for module with address 03 to 1E (30 ms) Returns the command is valid.	
~03RD	!031E
Reads the response delay time for module with address 03 Returns the command is valid, the response delay time is 30 ms (1E)	
~03RD1F	?03
Sets the response delay time for module with address 03 to 1F (31 ms) Returns the command is invalid because the value 31 exceeds the range of 0 to 30.	

- **Related Commands:** Section 2.2.9 ~AARD

2.3. DO Commands

The GPS-721U-MRTU series module provides a variety of DO commands. In most cases, it is recommended to use the @AA and @AA(Data) commands to read and write DO.

Command	Description	Section
@AA	Reads the DO status	2.3.1
@AA(Data)	Sets the DO status	2.3.2
~AAD	Reads the definition of DO value	2.3.3
~AADVV	Sets the definition of DO value	2.3.4
$AAC	Clears the latched status of DO	2.3.5
$AALS	Reads the latched status of DO	2.3.6
#**	Synchronized Sampling	2.3.7
$AA4	Reads the synchronized DO data	2.3.8
#AA00(Data)	Sets all DO status	2.3.9
#AA0A(Data)	Sets all DO status	2.3.10
#AA1cDD	Sets one DO status	2.3.11
#AAAcDD	Sets one DO status	2.3.12
$AA6	Reads the DO status	2.3.13

2.3.1. @AA

Reads the DO status
By default, a bit value of 1 denotes that the DO channel is on, while 0 denotes off. The value for indicating on/off status can be changed by the "~AADVV" command.

- **Syntax:**

@AA[CHK](CR)

@	Delimiter character
AA	The module address in hexadecimal format (00 to FF)

- **Response:**

Valid Command: **>(Data)[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

>	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)
(Data)	The status of the DO/DI ports is a four-digit hexadecimal value. The higher two bytes represents DO status, while the lower two bytes represents DI status

- **Example:**

Command	**Response**
@01	>0100
Reads the DO/DI port status of the module with address 01. Returns the command is valid, 01 is the DO status.	

- **Related Commands:**

Section 2.3.2 @AA(Data), Section 2.3.3 ~AAD, Section 2.3.4 ~AADVV

2.3.2. @AA(Data)

Sets the DO status

The DO value for indicating on/off status can be changed by the "~AADVV" command.

- **Syntax:**

@AA(Data)[CHK](CR)

@	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
(Data)	A single-digit hexadecimal value which is bitmapped with DO channels. By default, a bit value of 1 denotes that the DO channel is on, while 0 denotes off.

- **Response:**

Valid Command: **>[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Ignored Command: **![CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

>	Delimiter for a valid command
?	Delimiter for an invalid command
!	Delimiter for a ignored command
AA	The module address in hexadecimal format (00 to FF)

➢ **Example:**

Command	Response
@011	>
Sets DO1 of the module with address 01 to on. Returns the command is valid, DO is turned on.	
@010	!
Sets DO1 of the module with address 01 to off. Returns the command is ignored, it may be caused when a host watchdog timeout occurs and the module is set in safe mode	
@019	?01
Sets DO1 and DO4 of the module with address 01 to on. Returns the command is invalid because the module has one DO channel only.	

➢ **Related Commands:**

Section 2.3.1 @AA, Section 2.3.3 ~AAD, Section 2.3.4 ~AADVV

2.3.3. ~AAD

Reads the configuration for DO value of on/off status

- **Syntax:**

~AAD[CHK](CR)

~	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
D	Command to read the configuration for DO value

- **Response:**

Valid Command: **!AAVV[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)
VV	A two-digit hexadecimal configuration value

- **VV:** A 2-digit hexadecimal number representing the value configuration. (00 or 02)

Bit	7	6	5	4	3	2	1	0
Data	Reserved						OAS	

- **OAS: the definition for setting/reading value of DO status**

0: DO value 1 indicates on

DO value 0 indicates off

1: DO value 0 indicates on

DO value 1 indicates off

➢ **Example:**

Command	Response
~01D02	!01
Sets the configuration for DO value of on/off status to 1. (0 is on, 1 is off) Returns the command is valid.	
~01D	!0102
Reads the configuration for DO value. Returns the command is valid. The configuration for DO value (OAS) is 1, it means that 0 is on, while 1 is off.	

➢ **Related Commands:**

Section 2.3.4 ~AADVV

2.3.4. ~AADVV

Sets the configuration for setting/reading value of DO status

- **Syntax:**

~AADVV[CHK](CR)

~	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
D	Command to set the configuration for DO value
VV	A 2-digit hexadecimal number representing the configuration. (00 or 02)

- **VV**

Bit	7	6	5	4	3	2	1	0
Data	Reserved						OAS	

- **OAS: the definition for setting/reading value of DO status**

0: DO value 1 indicates that the channel is on

DO value 0 indicates that the channel is off

1: DO value 0 indicates that the channel is on

DO value 1 indicates that the channel is off

- **Response:**

Valid Command: **!AA[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)

➢ **Example:**

Command	Response
~01D00	!01
Sets the configuration for DO value to 0. (1 is on, 0 is off) Returns the command is valid.	
~01D	!0100
Reads the configuration for DO value. Returns the command is valid. The configuration for DO value is set to 0. (1 is on, while 0 is off)	

➢ **Related Commands:**

Section 2.3.3 ~AAD

2.3.5. $AAC

Clears the latched status of the DO channel

➢ **Syntax:**

$AAC[CHK](CR)

$	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
C	Command to clear the latched status of the DO channel

➢ **Response:**

Valid Command: **!AA[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)

➢ **Example:**

Refer to the example in the next section.

➢ **Related Commands:**

Section 2.3.6 $AALS

2.3.6. $AALS

Reads the high/low latched status of the DO channel

- **Syntax:**

$AALS[CHK](CR)

$	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
L	Command to read the latched status
S	0: Reads the low latched status 1: Reads the high latched status

- **Response:**

Valid Command: **!AA(Data)[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)
(Data)	Latched status of the DO channel

➢ **Example:**

Command	Response
@011	>
Sets DO status of the module with address 01 to 1. Returns the command is valid.	
$01C	!01
Clears the latched status of DO channel on the module with address 01. Returns the command is valid.	
$01L0	!010
Reads the low latched status of DO channel on the module with address 01 Returns the low latched status of DO channel is 0.	
$01L1	!011
Reads the high latched status of DO channel on the module with address 01 Returns the high latched status of DO channel is 1.	

➢ **Related Commands:**

Section 2.3.5 $AAC

2.3.7. #**

Synchronizes all modules to store the DO status for later retrieval

- **Syntax:**

#** [CHK](CR)

#	Delimiter character
**	Synchronous sampling command

- **Response:**

Valid Command: **No response.**

Using $AA4 command to read the synchronized data

- **Example:**

Refer to the example in the next section.

- **Related Commands:**

Section 2.3.8 $AA4

2.3.8. $AA4

Reads the synchronized DO data captured by the last #** command

- **Syntax:**

$AA4[CHK](CR)

$	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
4	Command to read the synchronized DO data

- **Response:**

Valid Command: **!S(Data)[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)
S	The first read flag 1: First read 0: Not the first read
(Data)	Synchronized DO data

➢ **Example:**

Command	Response
#**	No response
Sends the command for synchronizing data to all modules. No response to this command.	
$014	!1010000
Reads the synchronized data of module address 01. Returns the first read flag and synchronized DO status (01). S = 1 indicates that it is the first time to read synchronized data after the last #** command.	
$014	!0010000
Reads the synchronized data of module address 01. Returns the first read flag and synchronized DO status (01). S = 0 indicates that it is **NOT** the first time to read synchronized data after the last #** command.	

➢ **Related Command:**

Section 2.3.7 #**

2.3.9. #AA00(Data)

Sets the digital output status of the module

- **Syntax:**

#AA00(Data)[CHK](CR)

#	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
00	Command to set the DO status
(Data)	A two-digit hexadecimal value. Bit 0 corresponds to DO0, bit 1 to DO1, and so on. A bit value of whether 1 or 0 to indicate the on or off status varies according to the setting of command ~AADVV.

- **Response:**

Valid Command: **> [CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Ignored Command: **![CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

>	Delimiter for a valid command
?	Delimiter for an invalid command
!	Delimiter for an ignored command: If a host watchdog timeout occurs, the module will reset to safe mode and the DO output command will be ignored
AA	The module address in hexadecimal format (00 to FF)

➢ **Example:**

Command	Response
#010001	>
Sets DO0 of module address 01 to 1. Returns the command is valid.	
#010016	?01
Sets DO1, DO2 and DO4 to 1, DO0, DI3, DO5, DO6, DO7 to 0. Returns the command is invalid, because the module has one DO channel only.	
#010001	!
Sets DO0 of module address 01 to 1. Returns the command is ignored. It may be caused when a host watchdog timeout event is occurred.	

➢ **Related Commands:**

Section 2.3.1 @AA, Section 2.3.2 @AA(Data),

Section 2.3.3 ~AAD, Section 2.3.4 ~AADVV

2.3.10. #AA0A(Data)

Sets the digital output status of the module

- **Syntax:**

#AA0A(Data)[CHK](CR)

#	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
0A	Command to set the DO status
(Data)	A two-digit hexadecimal value. Bit 0 corresponds to DO0, bit 1 to DO1, and so on. A bit value of whether 1 or 0 to indicate the on or off status varies according to the setting of command ~AADVV.

- **Response:**

Valid Command: **> [CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Ignored Command: **![CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

>	Delimiter for a valid command
?	Delimiter for an invalid command
!	Delimiter for an ignored command: If a host watchdog timeout occurs, the module will reset to safe mode and the DO output command will be ignored
AA	The module address in hexadecimal format (00 to FF)

➢ **Example:**

Command	Response
#010A01	>
Sets DO0 of module address 01 to 1. Returns the command is valid.	
#010A01	!
Sets DO0 of module address 01 to 1. Returns the command is ignored. It may be caused when a host watchdog timeout event is occurred.	

➢ **Related Commands:**

Section 2.3.1 @AA, Section 2.3.2 @AA(Data),

Section 2.3.3 ~AAD, Section 2.3.4 ~AADVV

2.3.11. #AA1cDD

Sets the status of a single DO channel among the first eight channels of the module

- **Syntax:**

#AA1cDD[CHK](CR)

#	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
1	Command to set the status of a single DO channel
c	Specifies the DO channel number in a range of 0 to 7
DD	00: sets the DO channel to 0 01: sets the DO channel to 1

- **Response:**

Valid Command: **> [CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Ignored Command: **![CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

>	Delimiter for a valid command
?	Delimiter for an invalid command
!	Delimiter for an ignored command: If a host watchdog timeout occurs, the module will reset to safe mode and the DO output command will be ignored
AA	The module address in hexadecimal format (00 to FF)

➢ **Example:**

Command	Response
#011001	>
Sets DO0 of module address 01 to 1. Returns the command is valid.	
#011400	?01
Sets DO4 of module address 01 to 0 Returns the command is invalid, because the module has one DO channel only.	
#011001	!
Sets DO0 of module address 01 to 1. Returns the command is ignored. It may be caused when a host watchdog timeout event is occurred.	

➢ **Related Commands:**

Section 2.3.1 @AA, Section 2.3.2 @AA(Data),

Section 2.3.3 ~AAD, Section 2.3.4 ~AADVV

2.3.12. #AAAcDD

Sets the status of a single DO channel among the first eight channels of the module

- **Syntax:**

#AAAcDD[CHK](CR)

#	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
A	Command to set the status of a single DO channel
c	Specifies the DO channel number in a range of 0 to 7
DD	00: sets the DO channel to 0 01: sets the DO channel to 1

- **Response:**

Valid Command: **> [CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

Ignored Command: **![CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

>	Delimiter for a valid command
?	Delimiter for an invalid command
!	Delimiter for an ignored command: If a Host WatchDog Timeout occurs, the module will reset to safe mode and the DO output command will be ignored
AA	The module address in hexadecimal format (00 to FF)

➢ **Example:**

Command	Response
#01A001	>
Sets DO0 of module address 01 to 1. Returns the command is valid.	
#01A000	>
Sets DO0 of module address 01 to 0. Returns the command invalid.	
#01A001	!
Sets DO0 of module address 01 to 1. Returns the command is ignored. It may be caused when a host watchdog timeout event is occurred.	

➢ **Related Commands:**

Section 2.3.1 @AA, Section 2.3.2 @AA(Data),

Section 2.3.3 ~AAD, Section 2.3.4 ~AADVV

2.3.13. $AA6

Reads the DO status

- **Syntax:**

$AA6[CHK](CR)

$	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
6	Command to read the DO status

- **Response:**

Valid Command: **!AA(Data)[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)
(Data)	The DO status

- **Example:**

Command	**Response**
$016	!011
Reads the DO status of module address 01. Returns the DO value is 1.	
$076	?07
Reads the DO status of module address 07. Returns the command is invalid, it may be caused by that the module at the specified address does not exist.	

- **Related Commands:**

Section 2.3.1 @AA, Section 2.3.2 @AA(Data),
Section 2.3.3 ~AAD, Section 2.3.4 ~AADVV

2.4. Host Watchdog Commands

The GPS-721U-MRTU series module provides dual watchdog to enhance the reliability and stability during operation. The Module Watchdog is a hardware reset circuit that can automatically restart unhealthy modules, easily eliminating unexpected problems during runtime or caused by on-site noise.

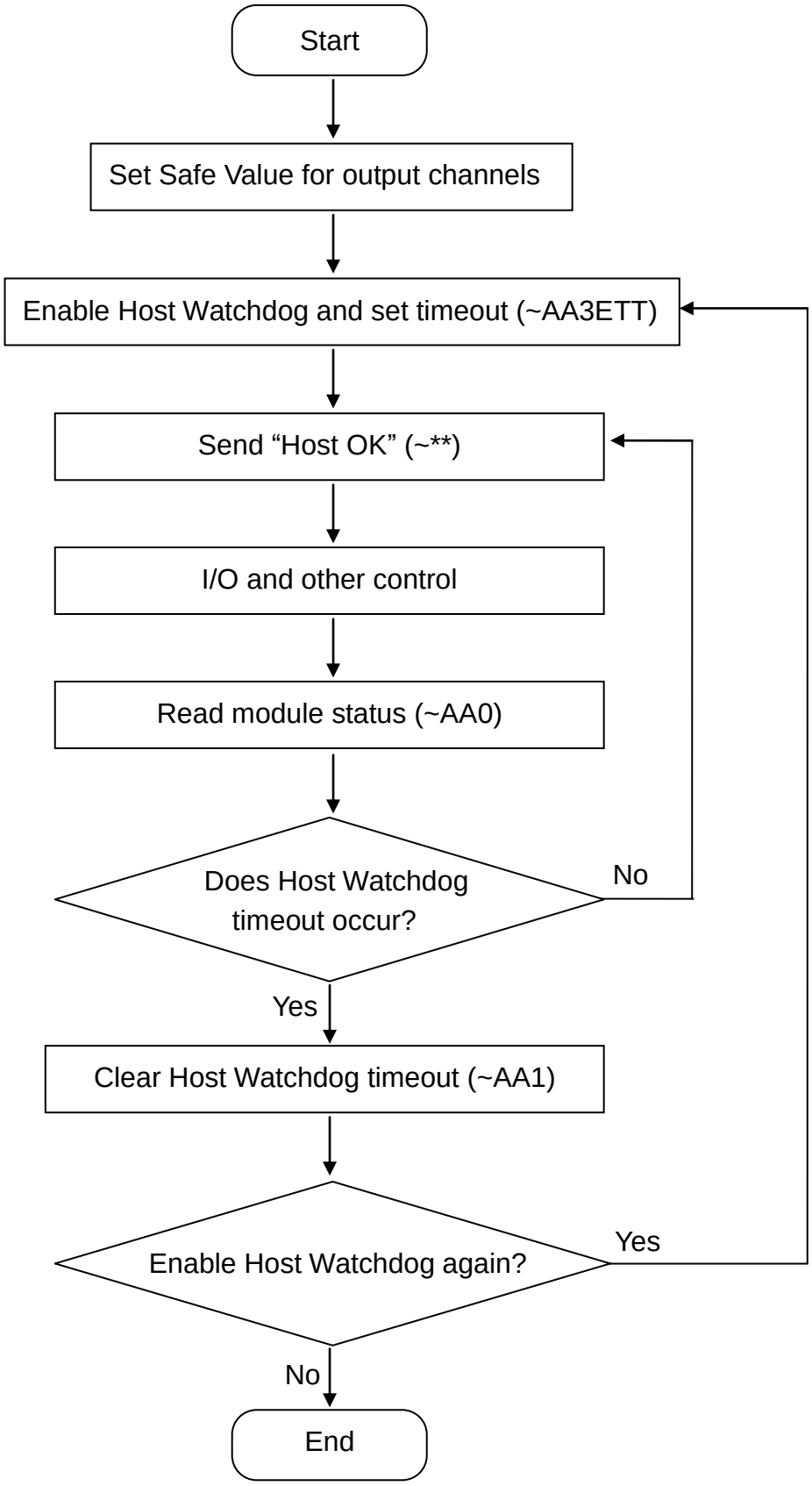
The Host Watchdog is a software function used to monitor the communication status between the host and the GPS module. After the host watchdog is enabled, the host needs to send a broadcast command (~**) to the module within a preset time interval, like a heartbeat signal.

If the module does not receive the "**~****" command within the time interval, a host watchdog timeout event will be triggered. During the timeout period, all output channels will remain at safe values, and cannot be changed by control commands until the timeout status is cleared.

In this section, it introduces the flow chart and the host watchdog commands.

Command	Description	Section
~**	Clears host watchdog timeout timer	2.4.2
~AA0	Reads host watchdog status	2.4.3
~AA1	Clears host watchdog timeout status	2.4.4
~AA2	Reads host watchdog enable/disable status and timeout value	2.4.5
~AA3ETT	Enables/disables host watchdog and sets timeout value	2.4.6
~AA4V	Reads power-on value or safe value of DO	2.4.7
~AA5V	Sets power-on value or safe value of DO	2.4.8

2.4.1. Host Watchdog Program Flowchart



2.4.2. ~**

This is a broadcast command to inform all the modules that have host watchdog enabled. This command will clear host watchdog timer. If a module does not receive this command within the preset time interval, a host watchdog timeout alarm will occur.

- **Syntax:**

~**[CHK](CR)

~	Delimiter character
****	Command for Host OK message

- **Response:**

No response

- **Example:**

Command	Response
~**	No response
Sends a “Host OK” message to all modules on the network.	
~010	!0180
Reads the host watchdog status of module with address 01. Returns the command is valid, host watchdog on the module is enabled and timeout does not occur.	

- **Related Commands:**

Section 2.4.3 ~AA0, Section 2.4.4 ~AA1, Section 2.4.5 ~AA2, Section 2.4.6 ~AA3ETT, Section 2.4.7 ~AA4V, Section 2.4.8 ~AA5V

2.4.3. ~AA0

Reads the host watchdog status

➢ **Syntax:**

~AA0[CHK](CR)

~	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
0	Command to read the host watchdog status

➢ **Response:**

Valid Command: **!AASS[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)
SS	A 2-digit hexadecimal value to represent the host watchdog status

➢ **SS**

Bit	7	6	5	4	3	2	1	0
Data	HWD status					HWD timeout		

- **HWD status: the Host WatchDog enable/disable status**

  0: the host watchdog is disabled
  1: the host watchdog is enabled

- **HWD timeout: the Host WatchDog timeout status**

  0: the host watchdog timeout does not occur
  1: the host watchdog timeout has occurred

Note:
The Host Watchdog status is stored in EEPROM and can only be reset by the "~AA1" command.

➢ **Example:**

Command	Response
~010	!0104
Reads the host watchdog status of module with address 01. Returns the command is valid, a host watchdog timeout has occurred.	
~070	!0700
Reads the host watchdog status of module with address 07. Returns the command is valid, host watchdog on the module is disabled and timeout does not occur.	

➢ **Related Commands:**

Section 2.4.2 ~**, Section 2.4.4 ~AA1, Section 2.4.5 ~AA2,

Section 2.4.6 ~AA3ETT, Section 2.4.7 ~AA4V, Section 2.4.8 ~AA5V

2.4.4. ~AA1

Resets the host watchdog timeout status

- **Syntax:**

~AA1[CHK](CR)

~	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
1	Command to reset the host watchdog timeout status

- **Response:**

Valid Command: **!AA[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)

- **Example:**

Command	**Response**
~070	!0704
Reads the host watchdog status of module with address 07. Returns the command is valid, a host watchdog timeout has occurred.	
~071	!07
Resets the host watchdog status of module with address 07. Returns the command is valid, the host watchdog timeout status is reset.	
~070	!0700
Reads the host watchdog status of module with address 07. Returns the command is valid, the host watchdog timeout event is cleared.	

- **Related Commands:**

Section 2.4.2 ~**, Section 2.4.3 ~AA0, Section 2.4.5 ~AA2,

Section 2.4.6 ~AA3EVV, Section 2.4.7 ~AA4V, Section 2.4.8 ~AA5V

2.4.5. ~AA2

Reads the host watchdog timeout value

➢ **Syntax:**

~AA2[CHK](CR)

~	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
2	Command to read the host watchdog value

➢ **Response:**

Valid Command: **!AAEVV[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)
E	0: the host watchdog is disabled. 1: the host watchdog is enabled.
TT	A 2-digit hexadecimal value (01 to FF) to represent the timeout value in tenths of a second. For example, 01 denotes 0.1 seconds and FF denotes 25.5 seconds.

➢ **Example:**

Command	**Response**
~012	!011FF
Reads the host watchdog timeout value of module with address 01. Returns the command is valid, the host watchdog is enabled, and timeout value is 25.5 seconds.	

➢ **Related Commands:**

Section 2.4.2 ~**, Section 2.4.3 ~AA0, Section 2.4.4 ~AA1,

Section 2.4.6 ~AA3ETT, Section 2.4.7 ~AA4V, Section 2.4.8 ~AA5V

2.4.6. ~AA3ETT

Enables/disables the host watchdog and sets the host watchdog timeout value.

If a host watchdog timeout event has occurred, the watchdog will be disabled.

- **Syntax:**

~AA3ETT[CHK](CR)

~	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
3	Command to set the host watchdog
E	1: enables the host watchdog 0: disables the host watchdog
TT	A 2-digit hexadecimal value (00 to FF) to represent the timeout value in tenths of a second. For example, 01 denotes 0.1 seconds and FF denotes 25.5 seconds.

- **Response:**

Valid Command: **!AA[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)

➢ **Example:**

Command	Response
~013164	!01
Enables the host watchdog of module with address 01, and sets the 10.0 seconds. Returns the command is valid.	
~010	!0104
Reads the host watchdog status of module with address 01. Returns the command is valid, a host watchdog timeout has occurred.	
~010	!0180
Reads the host watchdog status of module with address 01. Returns the command is valid, a host watchdog is enabled.	
~012	!01164
Reads the host watchdog timeout value of module with address 01. Returns the command is valid, the host watchdog is enabled, and timeout value is 10.0 seconds.	

➢ **Related Commands:**

Section 2.4.2 ~**, Section 2.4.3 ~AA0, Section 2.4.4 ~AA1,

Section 2.4.5 ~AA2, Section 2.4.7 ~AA4V, Section 2.4.8 ~AA5V

2.4.7. ~AA4V

Reads the power-on value or safe value of DO

➢ **Syntax:**

~AA4V[CHK](CR)

~	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
4	Command to read the power-on value or safe value of DO
V	P: reads power-on value S: reads safe value

➢ **Response:**

Valid Command: **!AA(Data)[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)
(Data)	Power-on value or safe value of DO

➢ **Example:**

Command	**Response**
~014P	!011
Reads the power-on value of DO on the module with address 01. Returns the command is valid, the power-on value is 1.	
~014S	!010
Reads the safe value of DO on the module with address 01. Returns the command is valid, the safe value is 0.	

➢ **Related Commands:**

Section 2.4.8 ~AA5V

2.4.8. ~AA5V

Sets the power-on value or safe value of DO

- **Syntax:**

~AA5V[CHK](CR)

~	Delimiter character
AA	The module address in hexadecimal format (00 to FF)
5	Command to sets the power-on value or safe value of DO
V	P: sets the current DO status as power-on value S: sets the current DO status as safe value

- **Response:**

Valid Command: **!AA[CHK](cr)**

Invalid Command: **?AA[CHK](cr)**

*There will be no response if a syntax error or a communication error occurs.

!	Delimiter for a valid command
?	Delimiter for an invalid command
AA	The module address in hexadecimal format (00 to FF)

➢ **Example:**

Command	Response
~015P	!01
Sets the current DO status as power-on value on the module with address 01. Returns the command is valid.	
@011	>
Sets the DO status of module with address 01 to 1. Returns the command is valid.	
~015S	!01
Sets the current DO status as safe value on the module with address 01. Returns the command is valid.	
@010	>
Sets the DO status of module with address 01 to 0. Returns the command is valid.	
~014S	!011
Reads the safe value of DO on the module with address 01. Returns the command is valid, the safe value is 1.	

➢ **Related Commands:**

Section 2.4.7 ~AA4V

Revision History

Revision	Date	Description
1.0.0	2024/ 01	First released