



Cellular Service

Version: 20240604

Contents

1	Get system time in GMT (0x0c)	2
2	Test module functionality (0x0e)	4
3	Get the local time (0x1c)	6
4	Get module's memory (0x0f)	9
5	Get Unix timestamp (with time zone) (0x1b)	11
6	(Optional) Get signal strength (0x24)	13
7	(Optional) Notify the module to disable heartbeat (0x25)	15
8	Get network status (0x2b)	17
9	Set the cellular mode (0x05)	19
10	Get the cellular mode (0x71 01)	21
11	Get the IMSI (0x71 02)	23
12	Get the ICCID number (0x71 03)	25
13	Get the IMEI (0x71 04)	27
14	Get cellular network type (0x71 43)	29
15	Query the current PLMN (0x71 48)	31
16	(Optional) Set or get radio resource control (RRC) (0x71 4A)	33

This topic describes the commands for the cellular service.

1 Get system time in GMT (0x0c)

- As the international standard of civil time, GMT is independent of the time zone and DST.
- After connecting to the network, the module returns success and valid time data after the local timestamp is synced.

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x03
Command	1	0x0c
Data length	2	0x0000
Data	0	None
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Example: 0x55aa 03 0c 0000 0e

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0x0c
Data length	2	0x0007
Data	7	The data length is seven bytes.

Field	Bytes	Description
		Data[0] indicates whether the time is obtained successfully. 0: Failure. 1: Success.
		Data[1] indicates the year. 0x00 indicates the year 2000.
		Data[2] indicates the month, ranging from 1 to 12.
		Data[3] indicates the day, ranging from 1 to 31.
		Data[4] indicates the hour, ranging from 0 to 23.
		Data[5] indicates the minute, ranging from 0 to 59.
		Data[6] indicates the second, ranging from 0 to 59.
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

For example, 0x55aa 00 0c 0007 01 10 04 13 05 06 07 4c indicates the time 05:06:07 on April 19, 2016 in GMT.

2 Test module functionality (0x0e)

- Insert the SIM card into the module to test whether the card can be detected.
- Test whether the module is authorized.
- Test whether the RF is calibrated.
- Get the 4G signal strength. A SIM card must be inserted. Signal strength ranges from 0 to 31. You can set a quality standard as needed.

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x03
Command	1	0x0e
Data length	2	0x0000
Data	0	None
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0x0e
Data length	2	0x0004
Data	4	The data length is four bytes.

Field	Bytes	Description
		Data[0]: 0 indicates the SIM card is not detected. 1 indicates the SIM card is detected.
		Data[1]: 0 indicates the module is not authorized. 1 indicates the module has been authorized.
		Data[2]: 0 indicates RF is not calibrated. 1 indicates RF has been calibrated.
		Data[3]: Signal strength, ranging from 0 to 31.
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

3 Get the local time (0x1c)

- The local time is calculated by adding the time zone offset and DST to the GMT. The time zone is where the device is activated.
- After connecting to the network, the module returns success and valid time data after the local timestamp is synced.

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x03
Command	1	0x1c
Data length	2	0x0000
Data	0	None
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0x1c
Data length	2	0x0008
Data	8	

Data[0] indicates whether the time is obtained successfully. 0: Failure. 1: Success.

Field	Bytes	Description
		Data[1] indicates the year. 0x00 indicates the year 2000.
		Data[2] indicates the month, ranging from 1 to 12.
		Data[3] indicates the day, ranging from 1 to 31.
		Data[4] indicates the hour, ranging from 0 to 23.
		Data[5] indicates the minute, ranging from 0 to 59.
		Data[6] indicates the second, ranging from 0 to 59.
		Data[7] indicates the day of the week, ranging from 1 to 7, with Monday as 1.
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Example:

- If the device is activated in mainland China, the local time is Beijing time (GMT+08:00). For example, the module returns 0x55aa 00 1c 0008 01 10 04 13 05 06 07 02 5f to indicate Beijing time at 05:06:07 on April 19, 2016.
- If the device is activated in other countries or regions, the local time is the time

zone in which the device is located.

4 Get module' s memory (0x0f)

This command can be used to get the module' s remaining memory. If the module returns -1, getting the module' s remaining memory is not supported.

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x03
Command	1	0x0f
Data length	2	0x0000
Data	0	None
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0x0f
Data length	2	0x0004
Data	4	The value is represented in big-endian format. For example, 0x00 0x00 0x28 0x00 represents 10240 bytes remaining memory.

Field	Bytes	Description
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

5 Get Unix timestamp (with time zone) (0x1b)

- The MCU can get a Unix timestamp, time zone, and daylight saving time (DST).
- This command only applies to door locks with the keep-alive feature.

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x03
Command	1	0x1B
Data length	2	0x0000
Data	0	None
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0x1B
Data length	2	N
Data	N	Data[0]: indicates whether the time is obtained successfully. 1: Success. 0: Failure.

Field	Bytes	Description
		Data[1] to Data[4]: Unix timestamp.
		Data[5]: indicates whether the time zone is obtained successfully.
		1: Success.
		0: Failure.
		Data[6]: 0 indicates local time ahead of GMT. 1 indicates local time behind GMT.
		Data[7]: time zone.
		Data[8]: indicates whether DST is in use in a time zone. 1: in use. 0: not in use.
		Data[9] to Data[12]: indicates timestamp when DST starts, in big-endian format.
		Data[13] to Data[16]: indicates timestamp when DST ends, in big-endian format.
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Currently, getting the DST is not supported.

6 (Optional) Get signal strength (0x24)

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x03
Command	1	0x24
Data length	2	0x0000
Data	0	None
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0x24
Data length	2	0x0001
Data	1	The RSSI value ranges from 0 to 31. The larger the value, the better the signal quality. 0 indicates -113 dBm or less.

Field	Bytes	Description
		31 indicates -51 dBm or above.If the return value is not within the range, the value is invalid and the operation fails.
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

7 (Optional) Notify the module to disable heartbeat (0x25)

- Version 03:

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x03
Comman	1	0x25
Data length	2	0x0000
Data	0	None
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Comman	1	0x25
Data length	2	0x0000
Data	0	None
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

- Version 04:

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x04
Comman	1	0x25
Data length	2	0x0001
Data	1	0: Disable. 1: Enable.
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Comman	1	0x25
Data length	2	0x0001
Data	1	0x00: Failure. 0x01: Success.
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Before the module goes to sleep for reducing power consumption, the MCU can send this command to notify the module to disable the heartbeat. Because heartbeats are required for building communication between the module and the MCU, this command must not be sent when the device is just powered on.

8 Get network status (0x2b)

Network status	Description	Status value
Status 1	The SIM card is not connected.	0x00
Status 2	The module searches for cellular networks.	0x01
Status 3	The module is registered with the cellular network but not connected to the network.	0x02
Status 4	The module is connected to the network and gets an IP address.	0x03
Status 5	The module has been connected to the cloud.	0x04
Status 6	SIM card registration is denied, which might be because it is not subscribed to a cellular service provider.	0x05
Status 7	The device is ready for pairing.	0x06
Status 8	Unknown status	0xFF

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x03
Command	1	0x2b
Data length	2	0x0000

Field	Bytes	Description
Data	0	None
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0x2b
Data length	2	0x0001
Data	1	See the description of network status .
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

9 Set the cellular mode (0x05)

The module can work in fully operational mode or airplane mode.

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x03
Command	1	0x05
Data length	2	0x0001
Data	1	1: Fully-operational mode. 4: Airplane mode.
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Example: 0x55aa 03 04 0000 06

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0x05
Data length	2	0x0001
Data	1	0: Success. 1: Failure.

Field	Bytes	Description
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Example: 0x55aa 00 04 0000 03

10 Get the cellular mode (0x71 01)

The module can work in fully operational mode or airplane mode.

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x03
Command	1	0x71
Data length	2	0x0001
Data	1	Subcommand: 0x01
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Example: 55 AA 03 71 00 01 01 75

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0x71
Data length	2	0x0002
Data	2	Subcommand: 0x01
		Data:
		1: Fully-operational mode.
		4: Airplane mode.

Field	Bytes	Description
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

11 Get the IMSI (0x71 02)

The MCU can send this command to the module to get the international mobile subscriber identity (IMSI) of the module.

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x03
Command	1	0x71
Data length	2	0x0001
Data	1	Subcommand: 0x02
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0x71
Data length	2	0x0010
Data	16	Subcommand: 0x02 For example, a 15-digit IMSI code is 460113012467340.

Field	Bytes	Description
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Example: 55 AA 00 71 00 10 02 34 36 30 31 31 33 30 31 32 34 36 37 33 34 30 7C

12 Get the ICCID number (0x71 03)

- The MCU can send this command to the module to get the integrated circuit card identifier (ICCID). The module returns a 20-digit ICCID number to the MCU.
- The MCU can send this command to the module to get the ICCID of the module.

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x03
Command	1	0x71
Data length	2	0x0001
Data	1	Subcommand: 0x03
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0x71
Data length	2	0x0015
Data	21	Subcommand: 0x03 For example, a 20-digit ICCID code is 89861118249000363490.

Field	Bytes	Description
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Example: 55 AA 00 71 00 15 03 38 39 38 36 31 31 31 38 32 34 39 30 30 30 33 36 33 34 39 30 9A

13 Get the IMEI (0x71 04)

The MCU can send this command to request the 15-digit international mobile equipment identity (IMEI) of the module. The MCU can send this command to the module to get the IMEI of the module.

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x03
Command	1	0x71
Data length	2	0x0001
Data	1	Subcommand: 0x04
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0x71
Data length	2	0x0010
Data	16	

Subcommand: 0x04

For example, a 15-digit IMEI number is
864237040014733.

Field	Bytes	Description
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Example: 55 AA 00 71 00 10 04 38 36 34 32 33 37 30 34 30 30 31 34 37 33 33 88

14 Get cellular network type (0x71 43)

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x03
Command	1	0x71
Data length	2	0x0001
Data	1	Subcommand: 0x43
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0x71
Data length	2	0x0002
Data	2	Subcommand: 0x43
		Type:
		1: GPRS
		2: LTE Cat.1
		4: CATM
		5: NB-IoT
		Error code

Field	Bytes	Description
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

15 Query the current PLMN (0x71 48)

- This command applies to the MA510 module only and is available when COPS is specified by the MCU in the product information.
- The module can return the PLMN only after it successfully registers to the network.

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x03
Command	1	0x71
Data length	2	0x0001
Data	1	Data[0]: Subcommand 0x48
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0x71
Data length	2	0x0002 + N × 4
Data	0x0002 + N × 4	Subcommand: 0x48 Data[1]: The result. 1: Success. 0: Failure.

Field	Bytes	Description
		PLMN (4 bytes): The PLMN of the country where the device is currently registered. An array of PLMN in big endian is returned.
		Current PLMN + PLMN1 ... + PLMNn
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Example: 55 AA 00 71 00 12 48 01 00 04 BA FC 00 04 BA FC 00 04 BC 8A 00 04 BB F4 3C

16 (Optional) Set or get radio resource control (RRC) (0x71 4A)

The MCU integration protocol must be v1.1.1 or later.

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x03
Command	1	0x71
Data length	2	0x0002/0x0003
Data	0x0002/ 0x0003	Subcommand: 0x4A
		Command (1 byte):
		0: Get
		1: Set
		For the set command, the 1-byte RRC value ranges from 0 to 20, corresponding to 0 to 10 seconds.
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0x71

Field	Bytes	Description
Data length	2	0x0003/0x0004
Data	0x0003/	Subcommand: 0x4A Command (1 byte): 0: Get 1: Set For the set command, the 1-byte RRC value ranges from 0 to 20, corresponding to 0 to 10 seconds. For the get command: The 1-byte RRC value ranges from 0 to 20, corresponding to 0 to 10 seconds.
	0x0004	
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.