



3GPP Commands

Version: 20240731

Contents

1. Get the IMSI (0xB5)	2
2. Get the ICCID number (0xB6)	4
3. Get extended signal quality (0xB7)	6
4. Get the IMEI (0xBD)	10
5. Set T3324 active timer (0xB9)	12
6. Set timer T3412 (0xBA)	14
7. Request T3324 active timer value (0xC5)	16
8. Request T3412 extended timer value (0xC6)	18
9. Set up APN (0xC2)	20

This topic describes commands related with 3GPP.

1. Get the IMSI (0xB5)

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

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0xb5
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:

```
1 55 AA 00 B5 00 00 B4
```

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0xb5
Data length	2	0x000F
Data	15	For example, 460113012467340 .
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Example:

```
1 55 AA 00 B5 00 0F 34 36 30 31 31 33 30 31 32 34 36 37 33 34 30 BD
```

2. Get the ICCID number (0xB6)

The MCU can send this command to the module to get the 20-digit integrated circuit card identifier (ICCID). 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	0x00
Command	1	0xb6
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:

```
1 55 AA 00 B6 00 00 B5
```

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0xb6
Data length	2	0x0014
Data	20	For example, 89861118249000363490 .
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Example:

```
1 55 AA 00 B6 00 14 38 39 38 36 31 31 31 38 32 34 39 30 30 30 33 36 33
2 34 39 30 DB
```

3. Get extended signal quality (0xB7)

The MCU can send this command to the module to get the extended signal quality of the module.

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0xb7
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:

```
1 55 AA 00 B7 00 00 B6
```

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0xb7
Data length	2	0x0006
Data	6	It consists of RxLev, BER, RSCP, Ec/No, RSRQ, and RSRP.
BYTE[0]	1	RxLev: the received signal level measured in dBm units. 0: -100 dBm or less

Field	Bytes	Description
		1: $-110 \text{ dBm} \leq \text{RSSI} < -109 \text{ dBm}$
		2: $-109 \text{ dBm} \leq \text{RSSI} < -108 \text{ dBm}$
		...
		61: $-50 \text{ dBm} \leq \text{RSSI} < -49 \text{ dBm}$
		62: $-49 \text{ dBm} \leq \text{RSSI} < -48 \text{ dBm}$
		63: $-48 \text{ dBm} \leq \text{RSSI}$
		99: unknown
BYTE[1]	1	BER: the bit error rate.
		0 to 7: RxQual values
		99: unknown
BYTE[2]	1	RSCP: the received signal code power.
		0: -120 dBm or less
		1: $-120 \text{ dBm} \leq \text{RSCP} < -119 \text{ dBm}$
		2: $-119 \text{ dBm} \leq \text{RSCP} < -118 \text{ dBm}$
		...
		94: $-27 \text{ dBm} \leq \text{RSCP} < -26 \text{ dBm}$
		95: $-26 \text{ dBm} \leq \text{RSCP} < -25 \text{ dBm}$
		255: unknown
BYTE[3]	1	Ec/No: the ratio between the received energy from the pilot signal CPICH per chip (Ec) to the noise density (No).
		0: -24 dBm or less

Field	Bytes	Description
		1: $-24 \text{ dBm} \leq E_c/L_o < -23.5 \text{ dBm}$
		2: $-23.5 \text{ dBm} \leq E_c/L_o < -23 \text{ dBm}$
		...
		47: $-1 \text{ dBm} \leq E_c/L_o < -0.5 \text{ dBm}$
		48: $-0.5 \text{ dBm} \leq E_c/L_o < 0 \text{ dBm}$
		49: $0 \text{ dBm} \leq E_c/L_o$
		255: unknown
BYTE[4]	1	RSRQ: reference signal receiving quality.
		0: -19.5 dB or less
		1: $-19.5 \text{ dB} \leq \text{RSRQ} < -19 \text{ dB}$
		2: $-19 \text{ dB} \leq \text{RSRQ} < -18.5 \text{ dB}$
		...
		32: $-4 \text{ dB} \leq \text{RSRQ} < -3.5 \text{ dB}$
		33: $-3.5 \text{ dB} \leq \text{RSRQ} < -3 \text{ dB}$
		34: $-3 \text{ dB} \leq \text{RSRQ}$
		255: unknown
BYTE[5]	1	RSRP: reference signal received power
		0: -140 dBm or less
		1: $-140 \text{ dBm} \leq \text{RSRP} < -139 \text{ dBm}$
		2: $-139 \text{ dBm} \leq \text{RSRP} < -138 \text{ dBm}$
		...

Field	Bytes	Description
		95: $-46 \text{ dBm} \leq \text{RSRP}$ $< -45 \text{ dBm}$
		96: $-45 \text{ dBm} \leq \text{RSRP}$ $< -44 \text{ dBm}$
		97: $-44 \text{ dBm} \leq \text{RSRP}$
		255: unknown
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Example:

```
1 55 AA 00 B7 00 06 28 00 FF FF 22 44 48
```

4. Get the IMEI (0xBD)

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	0x00
Command	1	0xbd
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:

```
1 55 AA 00 BD 00 00 BC
```

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0xbd
Data length	2	0x000f
Data	15	For example, 864237040014733
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Example:

```
1 55 AA 00 BD 00 0F 38 36 34 32 33 37 30 34 30 30 31 34 37 33 33 CF
```

5. Set T3324 active timer (0xB9)



This command requires a SIM card.

The MCU can use this command to set the timeout (in seconds) for the T3324 active timer.

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0xb9
Data length	2	0x0004
Data	4	The active time, in seconds.
		For example, the active time is 120 seconds (0x78).
		act_time[0] : 0x00
		act_time[1] : 0x00
		act_time[2] : 0x00
		act_time[3] : 00x78
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Example:

```
1 55 AA 00 B9 00 04 00 00 00 78 34
```

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa

Field	Bytes	Description
Version	1	0x00
Command	1	0xb9
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.

Example:

```
1 55 AA 00 B9 00 01 01 BA
```

6. Set timer T3412 (0xBA)



The timer T3412 setup depends on SIM card support.

The MCU can use this command to set the timeout (in seconds) for the T3412 active timer.

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0xba
Data length	2	0x0004
Data	4	The PSM time, in seconds.
		For example, the active time is 8 seconds (0x08).
		psm_time[0] : 0x00
		psm_time[1] : 0x00
		psm_time[2] : 0x00
		psm_time[3] : 00x08
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Example:

```
1 55 AA 00 BA 00 04 00 00 00 08 C5
```

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00



Field	Bytes	Description
Command	1	0xba
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.

Example:

```
1 55 AA 00 BA 00 01 01 BB
```

7. Request T3324 active timer value (0xC5)

The MCU can proactively send this command to the NB-IoT module to request the value of the T3324 active timer.

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0xC5
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:

```
1 55 AA 00 C5 00 00 C4
```

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0xC5
Data length	2	0x0004
Data	4	The value of the T3324 timer, in seconds.
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Example:

1 55 AA 00 C5 00 04 00 00 00 00 C8

8. Request T3412 extended timer value (0xC6)

The MCU can proactively send this command to the NB-IoT module to request the value of the T3412 extended timer.

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0xC6
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:

```
1 55 AA 00 C6 00 00 C5
```

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0xC6
Data length	2	0x0004
Data	4	The value of the T3412 timer, in seconds.
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Example:

```
1 55 AA 00 C6 00 04 00 00 70 80 B9
```

9. Set up APN (0xC2)

If the APN cannot be automatically populated, this command allows the user to set up the APN manually.

The MCU sends the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0xC2
Data length	2	None
Data	Depends on data	{ "apn": "ctnb", "pdp_type": "IP" }
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Data

Parameter	Description
apn	The access point name specified by the carrier. For example: ctnb : China Telecom cmnbiot : China Mobile
pdp_type	The packet data protocol, defaulting to IP. IP: Internet Protocol (IETF STD 5) IPv6: Internet Protocol version 6 (IETF RFC 2460) IPv4v6: dual stack Non-IP: transmits the non-IP data to the external packet network.

Example:

```

1 55 AA 00 C2 00 1E 7B 22 61 70 6E 22 3A 22 63 74 6E 62 22 2C 22 70 64
2 70 5F 74 79 70 65 22 3A 22 49 50 22 7D 6B

```

The module returns the following data.

Field	Bytes	Description
Header	2	0x55aa
Version	1	0x00
Command	1	0xC2
Data length	2	0x0001
Data	1	
		0x00: Success.
		0x01: Failure.
Checksum	1	Start from the header, add up all the bytes, and then divide the sum by 256 to get the remainder.

Example:

```

1 55 AA 00 C2 00 01 00 C2

```