



BCXG Serial API V4

Rev 1.0

2016-01-26

Table of Contents

Table of Contents	2
1. Revision History	5
2. Serial Protocol Definition	6
2.1. Configuration	6
2.2. Reliability.....	6
2.3. Message Buffer Size	6
2.4. Host Implementation Tips.....	6
2.5. Message Format.....	7
2.6. Message Encoding	8
Reserved Values	8
CRC Calculation	8
2.7. Applicative Message Format.....	10
Status Byte	10
3. Transport-Layer Messages	11
3.1. Transport (0x00-0x0F).....	11
0x00 - Acknowledge	11
0x01 - Non-acknowledge	11
NAK Message Specialization	12
0x02 - Set Max Message Size	13
4. Applicative-Layer Messages.....	14
4.1. GPIO (0x10 - 0x1F).....	14
0x10 - GPIO Configure Group Mode	15
0x11 - GPIO Set Value	16
0x12 - GPIO Get Value.....	17
0x13 - GPIO Get Group Mode	18
4.2. I2C (0x20-0x2F)	19
0x20 - I2C config.....	19
0x21 - I2C Operation	20
4.3. I2S / Call Handling (0x40 - 0x4F)	22
0x40 - Call Dialout	22
0x41 - Call Answer	23
0x42 - Call Hang Up	24
0x43 - Call Status Change Event.....	26
0x48 - Microphone Mute Get	28

0x49 - Microphone Mute Set	29
0x4A - Microphone Gain Get	30
0x4B - Microphone Gain Set	31
0x4C - Speaker Gain Get	32
0x4D - Speaker Gain Set	33
0x4E - Ring Tone Config Get.....	35
0x4F - Ring Tone Config Set.....	36
4.4. SMS (0x60 - 0x6F).....	37
0x60 - SMS Set Attributes	37
0x61 - SMS Message Append	39
0x62 - SMS Message Complete.....	41
0x63 - SMS Control Reception	43
0x64 - SMS Get Reception Enabled.....	44
4.5. Mail (0x70 - 0x7F).....	45
0x70 - Mail Set SMTP Server	45
0x71 - Mail set SMTP port.....	46
0x72 - Mail set email SMTP authentication user name	47
0x73 - Mail set email SMTP authentication password.....	48
0x74 - Mail set sender email address	49
0x75 - Mail set recipient email address	50
0x76 - Mail set subject	51
0x77 - Mail Append Message Body.....	52
0x78 - Mail Send.....	53
0x79 - Mail send by URL.....	54
4.6. Socket (0x80 - 0x8F)	55
0x80 - Socket Configure APN	55
0x81 - Socket Payload Append.....	57
0x82 - Socket Payload Send	59
0x83 - Socket Operation Complete	61
0x84 - Socket Cancel Send	62
0x85 - Socket Get Configured APN.....	63
4.7. Modem configuration (0x90 - 0x9F)	64



0x90 - Config Set Modem Speed.....	64
0x91 - Config Get Modem Speed	65
4.8. Status / Control (0xA0 - 0xAF).....	66
0xA0 - Get tower connection status	66
0xA1 - Get RSSI.....	67
0xA2 - Get IMEI	68
0xA3 - Get SIM	69
0xA4 - Get Voltage Vin	70
0xA5 - Set device functionality level	71
0xA6 - Get Network Registration Status Information	72
4.9. Version (0xC0 - 0xCF)	73
0xC0 - Version Get Cellular Modem Firmware	73
0xC1 - Version Get Cellular Modem Hardware.....	74
0xC2 - Version Get Cinterion Firmware	75
0xC3 - Version Get Cinterion Hardware.....	76



1. Revision History

Revision	Date	Comments
1.0	2016-01-26	Initial Release

2. Serial Protocol Definition

2.1. Configuration

An 115200 baudrate (8-N-1) is used for UART communication with the Cellular Modem. Hardware Flow Control (CTS/RTS) is used for communication.

2.2. Reliability

The Cellular Modem implementation allows for sending messages in a FIFO order.

Applicative message requests that are not acknowledged (ACK) or non-acknowledged (NAK) by the Host are re-sent every 2100 milliseconds by the Cellular Modem.

Applicative message replies sent by the Cellular Modem need to be acknowledged (or non-acknowledged) by the Host. Even though no retry mechanism is in place for applicative message replies, the Cellular Modem will wait for an ACK/NAK from the Host for 2100 milliseconds before discarding it.

Applicative message requests are sent up to 3 times. After this number of retry has been reached, the message is discarded from the send queue.

By their nature, there are no retry mechanism for transport-layer messages (i.e.: ACK, NAK, SetMaxMessageSize).

2.3. Message Buffer Size

The minimum buffer size allowed is 32 bytes. Although allowed, it is not recommended because several applicative message definitions go over this limit.

Choosing a small buffer size implies forward knowledge of messages to be exchanged between Cellular Modem and Host.

The default buffer size in the Cellular Modem is 512 bytes. This means that the Cellular Modem could theoretically send and receive messages of this size.

The setting of this parameter will only affect applicative messages of varying payload, e.g.: **Socket Payload Append** when sent from the *Cellular Modem* to the *Host*.

You may modify the buffer size in the Cellular Modem by issuing a **Set Max Message Size** message to the Cellular Modem. You may also **NAK** a message that is too big with the error **Message Too Big**, and indicate the buffer size to be used in the supplemental error information section of the message.

2.4. Host Implementation Tips

Never wait indefinitely for a reply: assume messages can be lost.

Implement retry-send mechanism for messages that are not ACKed by the Cellular Modem.

After sending a request to the Cellular Modem, don't expect the next message to be the response you were waiting for. It might be something else. This protocol allows for both endpoints to send messages asynchronously.

2.5. Message Format

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Command Identifier	[0x00 - 0xFF]
2		SEQ	Message sequence number	[0x01 - 0xFF]
...		DAT	Data Payload	[0x00 - 0xFF]
[N-2, N-1]		CRC	CRC of the packet (Big endian)	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

The **Command Identifier** is used to interpret the information found in the data payload.

The **Message Sequence Number** is used to prevent duplicate messages from being processed. It is also used by **ACK** and **NAK** messages to relay transport-layer reception status. This number should be incremented by 1 for each new message sent. It should be initialized to 1. The next message after 0xFF sequence number should have 0x01 sequence number. 0 should not be used for message sequence number.

A message should be re-sent at regular interval until an **ACK** or **NAK** message is received.

2.6. Message Encoding

Reserved Values

Value	Signification
0x7E	Start-of-frame character
0xCE	End-of-frame character
0x9E	Escape character

If any byte between **Start-of-frame** and **End-of-frame** is equal to one of the above reserved values, that byte needs to be escaped for transmission. This escape character must then be discarded at reception and must not be used to compute the CRC.

CRC Calculation

The CRC is calculated on every single byte, excluding Escape character, from command to last bytes of data inclusively. The algorithm used for CRC calculation is a CRC-16-CCITT with a seed value of 0xFFFF, polynomial as 0x1021 and no final xor. Here is the source code in C:

Table 1 CRC16

```

#define CRC16_CCITT_SEED_VALUE          0xFFFF /* Initial CRC value */

static unsigned short crc16_ccitt_table[256] = {
    0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50a5, 0x60c6, 0x70e7,
    0x8108, 0x9129, 0xa14a, 0xb16b, 0xc18c, 0xd1ad, 0xe1ce, 0xf1ef,
    0x1231, 0x0210, 0x3273, 0x2252, 0x52b5, 0x4294, 0x72f7, 0x62d6,
    0x9339, 0x8318, 0xb37b, 0xa35a, 0xd3bd, 0xc39c, 0xf3ff, 0xe3de,
    0x2462, 0x3443, 0x0420, 0x1401, 0x64e6, 0x74c7, 0x44a4, 0x5485,
    0xa56a, 0xb54b, 0x8528, 0x9509, 0xe5ee, 0xf5cf, 0xc5ac, 0xd58d,
    0x3653, 0x2672, 0x1611, 0x0630, 0x76d7, 0x66f6, 0x5695, 0x46b4,
    0xb75b, 0xa77a, 0x9719, 0x8738, 0xf7df, 0xe7fe, 0xd79d, 0xc7bc,
    0x48c4, 0x58e5, 0x6886, 0x78a7, 0x0840, 0x1861, 0x2802, 0x3823,
    0xc9cc, 0xd9ed, 0xe98e, 0xf9af, 0x8948, 0x9969, 0xa90a, 0xb92b,
    0x5af5, 0x4ad4, 0x7ab7, 0x6a96, 0x1a71, 0x0a50, 0x3a33, 0x2a12,
    0xdbfd, 0xcbdc, 0xfbbf, 0xeb9e, 0x9b79, 0x8b58, 0xbb3b, 0xab1a,
    0x6ca6, 0x7c87, 0x4ce4, 0x5cc5, 0x2c22, 0x3c03, 0x0c60, 0x1c41,
    0xedae, 0xfd8f, 0xcdec, 0xddcd, 0xad2a, 0xbd0b, 0x8d68, 0x9d49,
    0x7e97, 0x6eb6, 0x5ed5, 0x4ef4, 0x3e13, 0x2e32, 0x1e51, 0x0e70,
    0xff9f, 0xefbe, 0xdfdd, 0xcffc, 0xbf1b, 0xaf3a, 0x9f59, 0x8f78,
    0x9188, 0x81a9, 0xb1ca, 0xa1eb, 0xd10c, 0xc12d, 0xf14e, 0xe16f,
    0x1080, 0x00a1, 0x30c2, 0x20e3, 0x5004, 0x4025, 0x7046, 0x6067,
    0x83b9, 0x9398, 0xa3fb, 0xb3da, 0xc33d, 0xd31c, 0xe37f, 0xf35e,
    0x02b1, 0x1290, 0x22f3, 0x32d2, 0x4235, 0x5214, 0x6277, 0x7256,
    0xb5ea, 0xa5cb, 0x95a8, 0x8589, 0xf56e, 0xe54f, 0xd52c, 0xc50d,
    0x34e2, 0x24c3, 0x14a0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,
    0xa7db, 0xb7fa, 0x8799, 0x97b8, 0xe75f, 0xf77e, 0xc71d, 0xd73c,
    0x26d3, 0x36f2, 0x0691, 0x16b0, 0x6657, 0x7676, 0x4615, 0x5634,
    0xd94c, 0xc96d, 0xf90e, 0xe92f, 0x99c8, 0x89e9, 0xb98a, 0xa9ab,
    0x5844, 0x4865, 0x7806, 0x6827, 0x18c0, 0x08e1, 0x3882, 0x28a3,
    0xcb7d, 0xdb5c, 0xeb3f, 0xfb1e, 0x8bf9, 0x9bd8, 0xabbb, 0xbb9a,
    0x4a75, 0x5a54, 0x6a37, 0x7a16, 0x0af1, 0x1ad0, 0x2ab3, 0x3a92,
    0xfd2e, 0xed0f, 0xdd6c, 0xcd4d, 0xbdaa, 0xad8b, 0x9de8, 0x8dc9,

```



```
    0x7c26, 0x6c07, 0x5c64, 0x4c45, 0x3ca2, 0x2c83, 0x1ce0, 0x0cc1,  
    0xef1f, 0xff3e, 0xcf5d, 0xdf7c, 0xaf9b, 0xbfba, 0x8fd9, 0x9ff8,  
    0x6e17, 0x7e36, 0x4e55, 0x5e74, 0x2e93, 0x3eb2, 0x0ed1, 0x1ef0  
};  
  
unsigned short crc16_ccitt_update(unsigned short crc, unsigned char data)  
{  
    data = (crc >> 8) ^ data;  
    crc = (crc << 8) ^ crc_table[data];  
  
    return crc;  
}
```

2.7. Applicative Message Format

Status Byte

All applicative messages define a mandatory STATUS byte at byte position 3 in the message. In practice, the message data payload starts then at byte position 4.

This byte conveys two informations:

- bit 7: The message type; REQUEST(0) or REPLY(1)
- bits [0 - 6]: The status code. Only used for reply messages, to qualify how the request was processed.

Not all commands defined below explicitly define a reply message, but **future versions of this protocol may add reply messages for any command.**

CLIENT IMPLEMENTATIONS SHOULD ASSUME A REPLY MESSAGE CAN BE RECEIVED FOR ANY MESSAGE SENT.

3. Transport-Layer Messages

Each applicative message sent between the host and the Cellular Modem needs to be acknowledged (ACK), or non-acknowledged (NAK).

The ACK/NAK message contains the original **Message Sequence Number** they are reporting status for.

Note: ACK and NAK message are used to relay transport-layer status. They are not used to relay applicative statuses.

3.1. Transport (0x00-0x0F)

0x00 - Acknowledge

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		ACK	Acknowledge	0x00
2		SEQ	Sequence number of the message acknowledged	[0x01 - 0xFF]
[3, 4]		CRC	CRC of the packet (Big endian)	[0x0000 - 0xFFFF]
5		EOF	End-of-frame	0xCE

The Acknowledgment message is used to report correct reception of the message with the specified sequence number.

The Cellular Modem is the only one who can handle sending an ACK while waiting for an ACK. This way, the Cellular Modem can unlock the communication if a simultaneous send is done.

0x01 - Non-acknowledge

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		NAK	Non-acknowledge	0x01
2		SEQ	Sequence number of the message in error	[0x01 - 0xFF]
3		ERROR CODE	Reason for non-acknowledge	[0x00 - 0xFF]
4		INFO LEN	Supplemental information - byte size.	[0x00 - 0xFF]
5		INFO[0]	(Optional)Supplemental information	[0x00 - 0xFF]
...		INFO[...]		
[N-2, N-1]		CRC	CRC of the packet (Big endian)	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

The Non-acknowledgment message is used to report faulty reception of the message with the specified sequence number.

Upon reception of a NAK message, the error code should be examined and a correction applied.

NAK Message Specialization

0x01 - NAK: Incorrect CRC

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		NAK	Non-acknowledge	0x01
2		SEQ	Sequence number of the message in error	[0x01 - 0xFF]
3		ERROR CODE	Incorrect CRC	0x00
4		INFO LEN	Supplemental information - byte size. NO SUPPLEMENTAL INFO	0x00
[5, 6]		CRC	CRC of the packet (Big endian)	[0x0000 - 0xFFFF]
7		EOF	End-of-frame	0xCE

This Non-acknowledgment message is used to report a faulty CRC on the message with the specified sequence number

Resolution: re-compute CRC on message with the sequence number and send again.

0x01 - NAK: Payload Too Large

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		NAK	Non-acknowledge	0x01
2		SEQ	Sequence number of the message in error	[0x01 - 0xFF]
3		ERROR CODE	Payload Too Large	0x01
4		INFO LEN	Supplemental information - byte size.	0x01
5		MSG SZ MAX	Message Size Max Number of 4-byte words that can be sent in a single message. <u>Example:</u> 0x10 => messages up to 64 bytes in length can be received by the host. 0x25 => messages up to 148 bytes in length can be received by the host.	[0x08 - 0x80]
[6, 7]		CRC	CRC of the packet (Big endian)	[0x0000 - 0xFFFF]
8		EOF	End-of-frame	0xCE

This Non-acknowledgment message is used by the **host** to report to the **Cellular Modem** that reception of the message with specified sequence number failed because it was too large.

Resolution (Cellular Modem): Decrease size of packets sent to match maximum specified message size.

0x02 - Set Max Message Size

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Set Max Message Size	0x02
2		SEQ	NOT USED	0xFF
3		MSG SZ	Max Message Size Number of 4-byte words that can be sent in a single message. <u>Example:</u> 0x10 => messages up to 64 bytes in length can be received by the host. 0x25 => messages up to 148 bytes in length can be received by the host. Note: This size includes the protocol headers, IDs, etc.	[0x08 - 0x80]
[4, 5]		CRC	CRC of the packet (Big endian)	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

This message is used to configure the maximum size of messages that can be received on the host side. The Cellular Modem takes this information into account, thus preventing messages to be rejected (NAK) by the host because they are too big.

See section [Message Buffer Size](#) for more information on message sizes.

This is mostly useful for small hosts (e.g.: microprocessor) that are restricted in memory.

Note: Although 32 is the minimal size, it is not recommended. Some applicative message definitions go over this limit.

4. Applicative-Layer Messages

4.1. GPIO (0x10 - 0x1F)

GROUP	GPIO#	Modem I/O#	MODE_0 (Boot)	MODE_1
A	20	0	DAI_TX	
	22	1	DAI_TFS	
	23	2	DAI_CLK	
	21	3	DAI_RX	
	24	4	RING#	
	17	5	GPI	
B	16	6	GPI	GPO
	19	7	GPI	GPO
C	7	8	READY	GPO
	3	9	GPO	GPO
D	4	10	GPI	GPO
	18	11	GPI	GPO

0x10 - GPIO Configure Group Mode

Configures the specified IO group in the specified IO mode.

This configuration is volatile and needs to be re-applied after every reboot.

Note: When an IO pin is configured as GPO, its initial value will be 0.

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Configure GPIO group mode	0x10
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		GPIO group	0: Group A 1: Group B 2: Group C 3: Group D	[0 - 3]
5		GPIO group mode	0: MODE_0 1: MODE_1	[0 - 1]
[6, 7]		CRC	CRC	[0x0000 - 0xFFFF]
8		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Configure GPIO group mode	0x10
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Status Code:</u> 0: Success (Mode correctly configured) 1: Error, group number or group mode is not within bounds. 2: Error, invalid group mode for the specified group number. 3: Generic(Unknown) Error.	[0 - 3]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

0x11 - GPIO Set Value

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Set GPO value	0x11
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		GPIO index	Cellular Modem GPIO number	[0 - 11]
5		GPIO value	GPIO value	[0, 1]
[6, 7]		CRC	CRC	[0x0000 - 0xFFFF]
8		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Set GPO value	0x11
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Status Code:</u> 0: Success 1: the GPIO pin is out of bounds. 2: the GPIO pin is not configured as output. 3: an IO error occurred while setting the value. 4: A reboot is required because of previous GPIO configuration changes.	[0 - 3]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

0x12 - GPIO Get Value

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get GPIO value	0x12
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		GPIO index	Cellular Modem GPIO number	[0 - 11]
[5, 6]		CRC	CRC	[0x0000, 0xFFFF]
7		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get GPIO value	0x12
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Status Code:</u> 0: Success 1: the GPIO pin is out of bounds. 2: the GPIO pin is not configured as input. 3: an IO error occurred while getting the value. 4: A reboot is required because of previous GPIO configuration changes.	[0 - 3]
3	7	MSG TYPE	REPLY	1
4		GPIO value	GPIO value	[0, 1]
[5, 6]		CRC	CRC	[0x0000, 0xFFFF]
7		EOF	End-of-frame	0xCE

0x13 - GPIO Get Group Mode

Retrieves the configured IO mode of the specified IO group.

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get GPIO group mode	0x13
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		GPIO group	0: Group A 1: Group B 2: Group C 3: Group D	[0 - 3]
[5, 6]		CRC	CRC	[0x0000 - 0xFFFF]
7		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get GPIO group mode	0x13
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Status Code:</u> 0: Success (Mode correctly retrieved) 1: Error, group number is not within bounds. 2: Generic(Unknown) Error.	[0 - 4]
3	7	MSG TYPE	REPLY	1
4		IO Mode	0: MODE_0 1: MODE_1	[0x00 - 0x01]
[5, 6]		CRC	CRC	[0x0000 - 0xFFFF]
7		EOF	End-of-frame	0xCE

4.2. I2C (0x20-0x2F)

0x20 - I2C config

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	I2C config	0x20
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	
3	7	MSG TYPE	REQUEST	0
4		I2C SPEED	Speed of the I2C bus 0: 100 Kbps 1: 400 Kbps	[0, 1]
[5, 6]		CRC	CRC	[0x0000 - 0xFFFF]
7		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	I2C operation (Response)	0x20
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Status Code:</u> 0: Success 1: Invalid value specified.	[0, 1]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000, 0xFFFF]
6		EOF	End-of-frame	0xCE

0x21 - I2C Operation

Depending if you wish to read or write to the I²C slave device, the fields in the request message below have a slightly different meaning.

For a write operation, the **parameter1 length** field is used to specify the length of the data to write (this data follows in the next field).

For a read operation, the **parameter1 length** field is used to specify the number of bytes to read from the I²C slave. The following field is ignored by the Cellular Modem.

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	I2C operation	0x21
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	
3	7	MSG TYPE	REQUEST	0
4	[0 - 6]	I2C ADDR	I2C slave address (7-bit, right aligned)	[0x00 - 0x7F]
4	7	I2C OP	I2C operation: 0: Read 1: Write	[0, 1]
5		I2C REG	I2C start register to read/write from.	[0x00 - 0xFF]
6		Parameter1 length	<u>Write</u> : Number of I2C bytes to transmit <u>Read</u> : Number of I2C bytes to read.	[0x00 - 0xFF]
7		Parameter1[0]	<u>Write</u> : (Optional) I2C Payload <u>Read</u> : Not used.	[0x00 - 0xFF]
8		Parameter1[..]		
[N-2, N-1]		CRC	CRC	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

Reply

The **parameter1 length** and following field are only used to return data for a read request.

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	I2C operation (Response)	0x21
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status Code: 0: Success 1: No response from I2C subsystem. 2: Transmission Error (I2C NAK) 3: Protocol Error	[0, 1]

3	7	MSG TYPE	REPLY	1
4		Parameter1 length	Number of bytes in the I2C response	[0x00 - 0xFF]
5		Parameter1[0]	(Optional) I2C Response	[0x00 - 0xFF]
6		Parameter1[..]		
[N-2, N-1]		CRC	CRC	[0x0000, 0xFFFF]
N		EOF	End-of-frame	0xCE

4.3. I2S / Call Handling (0x40 - 0x4F)

0x40 - Call Dialout

Allows initiating a mobile originating call.

Refer to the V.250 ITU-T Recommendation for allowable characters in the phone number string.

Note: Only voice calls are supported.

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Call Dialout	0x40
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		CALL TYPE	Call Type: 0: Voice	0
5		LEN	Length of the phone number to dial.	[0x00 - 0xFF]
6		Phone Number[0]	First character of phone number to dial.	0-9, *, #, +, A, B, C, D, p
7		Phone number[..]		
[N-2, N-1]		CRC	CRC	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Call Dialout	0x40
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status Code: 0: Success 1: Invalid call type/phone number parameters. 2: Link not established, not connected. 3: Invalid command or call already active. 4: Generic error.	[0-4]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000, 0xFFFF]
6		EOF	End-of-frame	0xCE

0x41 - Call Answer

Allows answering a mobile terminating call.

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Call Answer	0x41
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Call Answer	0x41
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status Code: 0: Success 1: Internal Error.	[0, 1]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000, 0xFFFF]
6		EOF	End-of-frame	0xCE

0x42 - Call Hang Up

Allows ending a call.

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Call Hang Up	0x42
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		CALL ID	Call identifier to end: 0: Terminate all calls. 1-7: Specific call identifier.	[0-7]
5		REASON	<u>Hang up reason:</u> 1: Send 3GPP TS 24.008 [39] release cause "unassigned (unallocated) number" 16: Send 3GPP TS 24.008 [39] release cause "normal call clearing " 17: Send 3GPP TS 24.008 [39] release cause "user busy " 18: Send 3GPP TS 24.008 [39] release cause "no user responding " 27: Send 3GPP TS 24.008 [39] release cause "destination out of order " 31: Send 3GPP TS 24.008 [39] release cause "normal, unspecified"	[1,16,17,18,27,31]
[6, 7]		CRC	CRC	[0x0000 - 0xFFFF]
8		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Call Hang Up	0x42
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Status Code:</u> 0: Success 1: Invalid reason code/call identifier parameters. 2: Internal Error.	[0, 1, 2]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000, 0xFFFF]



6		EOF	End-of-frame	0xCE
---	--	-----	--------------	------

0x43 - Call Status Change Event

This message is sent asynchronously from the Cellular Modem to the Host whenever a state change occurs for a given call.

This message may be repeated at intervals. For example when a call is incoming, this message is sent approximately every second.

This messages< may be sent in quick succession, with different state and additional info. For example, when a call is incoming, you may get a message with Call Incoming state and unknown call type, immediately followed by a message with Call Incoming state and voice call type. Same thing for Call Disconnected messages.

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Call Status Change Event	0x43
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	
3	7	MSG TYPE	REQUEST	0
4		CALL ID	Identifies the call for which this event applies.	[1-7]
5		STATE	<u>New call state:</u> 0: Call Incoming 1: Call Connected 2: Call Active 3: Call Dialing. 4: Call Alerting. 5: Call Disconnected	[0x00-0x03]
6		ADD. INFO	Additional information concerning the new state. See below for possible values.	[0x00 - 0xFF]
[7, 8]		CRC	CRC	[0x0000 - 0xFFFF]
9		EOF	End-of-frame	0xCE

The following table specifies the nature or reason of the new call state.

Call State	Call State Additional Information
Call Incoming	<u>Incoming Call Type:</u> 0x00: Voice. 0xFF: Call type not yet known. (Occurs initially for each new incoming calls).
Call Connected	Not Used.
Call Active	Not Used.
Call Dialing	Not Used.

Call Alerting	Not Used.
Call Disconnected	<u>Disconnection reason:</u> 0x00: Link not established or disconnected 0x01: No dial tone, dialling impossible, wrong mode 0x02: Remote station busy. 0x03: No answer. 0xFF: Disconnection reason unknown.

0x48 - Microphone Mute Get

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Microphone Mute Get	0x48
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Microphone Mute Get	0x48
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Status Code:</u> 0: Success 1: Internal Error	[0, 1]
3	7	MSG TYPE	REPLY	1
4		MIC MUTE	<u>Microphone Mute state:</u> 0x00 - Not Muted. 0x01 - Muted.	[0, 1]
[5, 6]		CRC	CRC	[0x0000, 0xFFFF]
7		EOF	End-of-frame	0xCE

0x49 - Microphone Mute Set

Note: This property is volatile and is reinitialized at power-up.

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Microphone Mute Set	0x49
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		MIC MUTE	Microphone Mute state: 0x00 - Not Muted. 0x01 - Muted.	[0, 1]
[5, 6]		CRC	CRC	[0x0000 - 0xFFFF]
7		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Microphone Mute Set	0x49
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status Code: 0: Success 1: Invalid value for mute state parameter. 2: Internal Error	[0, 1, 2]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000, 0xFFFF]
6		EOF	End-of-frame	0xCE

0x4A - Microphone Gain Get

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Microphone Gain Get	0x4A
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Microphone Gain Get	0x4A
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status Code: 0: Success 1: Internal Error	[0, 1]
3	7	MSG TYPE	REPLY	1
4		MIC GAIN	Microphone gain adjustment in steps. Each step is equal 0.5db starts from -43.5db to 60db. 0: -96dB 1: -43.5db ... 99: 5.5dB 100: 6.0dB	[0-100]
[5, 6]		CRC	CRC	[0x0000, 0xFFFF]
7		EOF	End-of-frame	0xCE

0x4B - Microphone Gain Set

Note: This property is volatile and is reinitialized at power-up.

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Microphone Gain Set	0x4B
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		MIC GAIN	Microphone gain adjustment in steps. Each step is equal 0.5db starts from -43.5db to 60db. 0: -96dB 1: -43.5db ... 99: 5.5dB 100: 6.0dB	[0-100]
[5, 6]		CRC	CRC	[0x0000 - 0xFFFF]
7		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Microphone Gain Set	0x4B
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Status Code:</u> 0: Success 1: Invalid value for microphone gain. 2: Internal Error	[0, 1, 2]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000, 0xFFFF]
6		EOF	End-of-frame	0xCE

0x4C - Speaker Gain Get

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Speaker Gain Get	0x4C
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Speaker Gain Get	0x4C
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status Code: 0: Success 1: Internal Error	[0, 1]
3	7	MSG TYPE	REPLY	1
4		SPKR GAIN	Speaker gain adjustment in steps. Each step is equal 0.5db, starting from -43.5db to 60db. 0: -96dB 1: -43.5db ... 99: 5.5dB 100: 6.0dB	[0-100]
5		SIDETONE GAIN	Sidetone gain adjustment in steps. Each step is equal 0.5db, starting from -43.5db to 43.5db. 0=-96dB 1=-43.5db ... 174=43.0dB 175=43.5B	[0-175]
[6, 7]		CRC	CRC	[0x0000, 0xFFFF]
8		EOF	End-of-frame	0xCE

0x4D - Speaker Gain Set

Note: This property is volatile and is reinitialized at power-up.

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Speaker Gain Set	0x4D
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		SPKR GAIN	Speaker gain adjustment in steps. Each step is equal 0.5db, starting from -43.5db to 60db. 0: -96dB 1: -43.5db ... 99: 5.5dB 100: 6.0dB	[0-100]
5		SIDETONE GAIN	Sidetone gain adjustment in steps. Each step is equal 0.5db, starting from -43.5db to 43.5db. 0=-96dB 1=-43.5db ... 174=43.0dB 175=43.5B	[0-175]
[6, 7]		CRC	CRC	[0x0000 - 0xFFFF]
8		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Speaker Gain Set	0x4D
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status Code: 0: Success 1: Invalid value for speaker/sidetone gain parameters. 2: Internal Error	[0, 1, 2]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000,



				0xFFFF]
6		EOF	End-of-frame	0xCE

0x4E - Ring Tone Config Get

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Ring Tone Config Get	0x4E
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Ring Tone Config Get	0x4E
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Status Code:</u> 0: Success 1: Internal Error	[0, 1]
3	7	MSG TYPE	REPLY	1
4		MELODY	<u>Ring tone or melody number:</u> There are 8 existing tones/melodies available.	[1-8]
5		VOLUME	<u>Ring tone/melody volume:</u> 0: Muted 1: Very low 2: Low 3: Medium 4: Loud	[0-4]
[6, 7]		CRC	CRC	[0x0000, 0xFFFF]
8		EOF	End-of-frame	0xCE

0x4F - Ring Tone Config Set

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Ring Tone Config Set	0x4F
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		MELODY	<u>Ring tone or melody number:</u> There are 8 existing tones/melodies available.	[1-8]
5		VOLUME	<u>Ring tone/melody volume:</u> 0: Muted 1: Very low 2: Low 3: Medium 4: Loud	[0-4]
[6, 7]		CRC	CRC	[0x0000, 0xFFFF]
8		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Ring Tone Config Set	0x4F
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Status Code:</u> 0: Success 1: Invalid value for melody/volume parameters. 2: Internal Error	[0, 1, 2]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

4.4. SMS (0x60 - 0x6F)

0x60 - SMS Set Attributes

Allows informing the receiving end of the SMS phone number and send date.

The **send date** is only used for SMS reception. This information is not used when the host does an SMS transmission.

Request

Request Message Direction	Description
HOST -> CELLULAR MODEM	(SMS Transmission) Used to configure the recipient (phone number) of the SMS message.
CELLULAR MODEM -> HOST	(SMS Reception) Used to inform the host of the initiator of the received SMS, and the time at which it was sent.

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	SMS Set phone number	0x60
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		Phone number length	Number of digits in the phone number	[0x01 - 0xFF]
[5 - ...]		Phone number	Phone number of SMS	ASCII (digits)
5 + Phone Number Length		Sent Date Length	Number of bytes in the sent date. NOTE: This field must be 0 for SMS transmission.	[0x01 - 0xFF]
[...]		Sent Date	The time at which the SMS was received by the cellular network equipment.	ASCII
[N-2, N-1]		CRC	CRC	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

Reply

Reply Message Direction	Description
CELLULAR MODEM -> HOST	Used to indicate the validation result of the phone number.
HOST -> CELLULAR MODEM	Used to indicate the validation result of the phone number. Note: Although the Cellular Modem expects this response, the status is ignored.

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	SMS Set phone number	0x60
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status code: 0: Indicates success 1: Indicates an invalid phone number was supplied.	[0, 1]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

0x61 - SMS Message Append

This command can be called successively, the payload of the new command must be append to the previous one. Once everything is sent, SMS can be initiated with the command **SMS Message complete**.

Note: Sending this message with a data payload of 0 (Parameter1 length) will flush the aggregated message on the receiving end.

Request

Request Message Direction	Description
HOST -> CELLULAR MODEM	Used to append chunks of message for an eventual SMS Message Complete .
CELLULAR MODEM -> HOST	For SMS reception, the Cellular Modem will send these messages followed ba a SMS Message Complete .

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	SMS Message Append	0x61
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		Parameter1 length	Number of bytes in the SMS message	[0x00 - 0xFF]
5		Parameter1[0]	SMS message	[0x00 - 0xFF]
...		Parameter1[..]		[0x00 - 0xFF]
[N-2, N-1]		CRC	CRC	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

Reply

Reply Message Direction	Description
CELLULAR MODEM -> HOST	(SMS Transmission) Reports back to the host if the message payload to append was successfully aggregated to the whole message.
HOST -> CELLULAR MODEM	(SMS reception) Reports back to the Cellular Modem if the message payload to append was successfully aggregated to the whole message. Note: Although the Cellular Modem expects this response, the status is ignored.

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	SMS Append Message	0x61
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Status Code:</u> 0: Indicates success 1: Indicates aggregated payload is too large.	[0, 1]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

0x62 - SMS Message Complete

This message indicates that the SMS message payload is complete and can be interpreted by the receiving end.

This message is used by both the host (SMS sending) and the Cellular Modem (SMS reception).

Request

Request Message Direction	Description
HOST -> CELLULAR MODEM	Used to indicate the SMS message is complete and can be sent.
CELLULAR MODEM -> HOST	Used to indicate the SMS message is complete and can be interpreted by the host.

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	SMS Message Complete	0x62
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Reply Message Direction	Description
CELLULAR MODEM -> HOST	Used to inform the host of the send status of the SMS.
HOST -> CELLULAR MODEM	Used to indicate the SMS message was received successfully. Note: Although the Cellular Modem expects this response, the status is ignored.

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	SMS Message Complete	0x62
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Operation Status Code: 0: Operation Successful 1: General Error	[0, 1]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

0x63 - SMS Control Reception

This message allows enabling and disabling SMS message reception.

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	SMS Set Control Reception	0x63
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		CTRL	Control SMS Reception: 0: Disable SMS Reception 1: Enable SMS Reception	[0, 1]
[5, 6]		CRC	CRC	[0x0000 - 0xFFFF]
7		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	SMS Set Control Reception	0x63
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status Code: 0: Success 1: Error	[0, 1]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

0x64 - SMS Get Reception Enabled

This message allows to get the status of SMS message reception.

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	SMS Get Control Reception	0x64
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	SMS Get Control Reception	0x64
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status Code: 0: Success 1: Error	[0, 1]
3	7	MSG TYPE	REPLY	1
4		CTRL STATUS	Control SMS Reception Status: 0: SMS Reception Disabled 1: SMS Reception Enabled	[0, 1]
[5, 6]		CRC	CRC	[0x0000 - 0xFFFF]
7		EOF	End-of-frame	0xCE

4.5. Mail (0x70 - 0x7F)

0x70 - Mail Set SMTP Server

This command is used to configure the SMTP server that will be used to send the mail.

Note: This information is persisted until the Cellular Modem is rebooted.

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Mail set SMTP server	0x70
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		Parameter1 length	Number of bytes in the SMTP server address	[0x01 - 0xFF]
5		Parameter1[0]	SMTP server address	ASCII
...		Parameter1[..]		ASCII
[N-2, N-1]		CRC	CRC	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

0x71 - Mail set SMTP port

This command is used to setup the server port that will be used to send the mail.

Note: This information is persisted until the Cellular Modem is rebooted.

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Mail set SMTP port	0x71
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		SMTP	SMTP Port	[0x0000 - 0xFFFF]
[6, 7]		CRC	CRC	[0x0000 - 0xFFFF]
8		EOF	End-of-frame	0xCE

0x72 - Mail set email SMTP authentication user name

This command is used to setup the email account that will be used to login on the remote server to send the mail.

Note: This information is persisted until the Cellular Modem is rebooted.

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Mail set email SMTP authentication user name	0x72
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		Parameter1 length	Number of bytes in the email address	[0x01 - 0xFF]
5		Parameter1[0]	Email address to login on server	ASCII
...		Parameter1[..]		ASCII
[N-2, N-1]		CRC	CRC	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

0x73 - Mail set email SMTP authentication password

This command is used to setup the email account password that will be used to login on the remote server to send the mail.

Note: This information is persisted until the Cellular Modem is rebooted.

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Mail set email SMTP authentication password	0x73
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		Parameter1 length	Number of bytes in the password	[0x01 - 0xFF]
5		Parameter1[0]	Password to login on server	ASCII
...		Parameter1[..]		ASCII
[N-2, N-1]		CRC	CRC	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

0x74 - Mail set sender email address

Allows configuring the sender (origin) e-mail address.

Note: This information is persisted until the Cellular Modem is rebooted.

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Mail set sender email address	0x74
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		Parameter1 length	Size in bytes of the origin e-mail address	[0, 1]
5		Parameter1[0]	Origin e-mail address.	ASCII
...		Parameter1[..]		ASCII
[N-2, N-1]		CRC	CRC	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

0x75 - Mail set recipient email address

This command is used to setup the destination email that will be used to send the mail.

Note: This information is persisted until the Cellular Modem is rebooted.

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Mail set recipient email address	0x75
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		Parameter1 length	Number of bytes in the destination mail	[0x01 - 0xFF]
5		Parameter1[0]	Destination mail	ASCII
...		Parameter1[..]		ASCII
[N-2, N-1]		CRC	CRC	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

0x76 - Mail set subject

This command is used to set the title of the mail.

Note: This information is persisted until the Cellular Modem is rebooted.

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Mail set subject	0x76
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		Parameter1 length	Number of bytes in the mail title	[0x01 - 0xFF]
5		Parameter1[0]	Mail subject	ASCII
...		Parameter1[..]		ASCII
[N-2, N-1]		CRC	CRC	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

0x77 - Mail Append Message Body

This command is used to build to mail body. It can be called successively, the payload of the new command must be append to the previous one. Once everything is sent, request can be initiated with the command "Mail ready to send"

Note: Sending this message with a data payload of 0 (Parameter1 length) will flush the aggregated message on the receiving end.

Note: This information is **not reset** after an e-mail is sent.

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Mail append message body	0x77
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		Parameter1 length	Number of bytes in this chunk	[0x00 - 0xFF]
5		Parameter1[0..]	Mail message chunk	ASCII
...		Parameter1[..]		ASCII
[N-2, N-1]		CRC	CRC	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Mail append message body	0x77
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status code: 0: Operation Successful 1: ERROR, aggregated message is too large.	[0, 1]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

0x78 - Mail Send

This message initiates the sending of the e-mail, with all the information previously set.

Encryption Information: The following encryption mechanisms are available:

- TLS Protocol Version 1.0 as specified in RFC 2246.
- SSL V3 as specified in The SSL Protocol Version 3.0

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Mail Send	0x78
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		SSL	Use SSL connection 0: Not use SSL 1: Use SSL	[0, 1]
[5, 6]		CRC	CRC	[0x0000 - 0xFFFF]
7		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Mail Send	0x78
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Status Code:</u> 0: Mail sent successfully. 1: Error sending mail.	[0, 1]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

0x79 - Mail send by URL

When this command is called, the module will transmit the mail received with following parameters: title and mail body. The URL is used to POST the request by a home made script.

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Mail send by URL	0x79
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		Parameter1 length	Number of bytes in the URL	[0x01 - 0xFF]
5		Parameter1[0]	URL	ASCII
...		Parameter1[..]		ASCII
[N-2, N-1]		CRC	CRC	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Mail Send	0x79
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status Code: 0: Mail sent successfully. 1: Error sending mail.	[0, 1]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

4.6. Socket (0x80 - 0x8F)

0x80 - Socket Configure APN

This message allows configuring the Access Point Name (APN) used to bridge the cellular network with the Internet.

This configuration is REQUIRED and must be done before any **Socket** operation is attempted.

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Socket Configure APN	0x80
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		Parameter1 length	Number of bytes for the APN	[0x01 - 0xFF]
5		parameter1[0]	(Facultative)The access point name.	
...		Parameter1[..]		
5 + parameter 1 length		parameter2 length	Number of bytes in the APN authentication user name.	[0x00-0xFF]
...		parameter2[0]	(Facultative)The user name for authentication to the APN.	
...		Parameter2[..]		
6 + parameter 1 & 2 lengths		parameter3 length	Number of bytes in the APN authentication password.	[0x00-0xFF]
...		parameter3[0]	(Facultative)The password for authentication to the APN.	
...		Parameter3[..]		
[N-2, N-1]		CRC	CRC	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Socket Configure APN	0x80
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Status Code:</u> 0: Indicates success. 1: ERROR, message malformed.	[0, 1]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

0x81 - Socket Payload Append

This message allows data chunks to be exchanged between the host and the Cellular Modem.

This message can be sent successively to exchange large payloads. These chunks are aggregated on the receiving side until the whole payload is transferred.

When the host wants to start a new socket operation, it starts first by transferring the data payload to send. This is done using this message (several times if required).

When the payload is fully transferred, the host sends the **Socket Payload Complete** message to the Cellular Modem.

When the Cellular Modem has sent the payload to the destination, the answer is read and transferred back to the host using this command (several times if required).

When the payload is fully transferred, the Cellular Modem sends the **Socket Operation Complete** message to the Host.

Note: Sending this message with a data payload of 0 (Parameter1 length) will flush the aggregated message on the receiving end.

Request

Request Message Direction	Description
HOST -> CELLULAR MODEM	Used to append chunks of data for an eventual Socket Payload Complete .
CELLULAR MODEM -> HOST	The received bytes (if any) are transferred back to the host, in chunks, using this command, until a Socket Payload Complete is sent.

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Socket payload append	0x81
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		Parameter1 length	Number of bytes in this socket payload chunk	[0x00 - 0xFF]
5		Parameter1[0]	Socket payload chunk	ASCII
...		Parameter1[..]		ASCII
[N-2, N-1]		CRC	CRC	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

Reply

Request Message Direction	Description
CELLULAR MODEM -> HOST	Reports back to the host if the payload to append was successfully aggregated to the whole payload.
HOST -> CELLULAR MODEM	Reports back to the Cellular Modem if the payload to append was successfully aggregated to the whole payload. <u>Note:</u> Although the Cellular Modem expects this response, the status is ignored.

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Socket payload append	0x81
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Status Code:</u> 0: Indicates success. 1: ERROR, APN not set. 2: ERROR, aggregated payload is too large.	[0 - 2]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

0x82 - Socket Payload Send

This message is sent from the Host to the Cellular Modem. It indicates the end of the payload, and that the Cellular Modem may start transmitting the data to the destination and/or receive a reply from the destination.

If the protocol is *TCP*, use the 'RX Timeout' parameter to control how the reply is received (if a reply is expected or not).

If the method is *HTTP GET* or *HTTP POST*, the payload is sent in the variable named '**data**'.

If the method is *HTTP GET*, the payload must be URL-encoded, or the request will fail. The payload is put as-is in the URL string.

Note: This operation may take some time. Concurrent 'send' operations are not allowed.

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Socket Payload Send	0x82
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		Protocol -or- Method	0: TCP 1: HTTP GET 2: HTTP POST	[0 - 2]
[5, 6]		RX Timeout	Time to wait, in milliseconds, for data from the TCP server. 0x0000: Don't read anything from the TCP server. 0xFFFF: Indicates to read until the server closes the socket. 1-65534: Number of milliseconds to wait for data from the TCP server.	[0x0000 - 0xFFFF]
7		Parameter1 length	Number of bytes in the TCP destination or HTTP(S) URL.	[0x01 - 0xFF]
8		Parameter1[0]	If protocol is TCP : the destination in the form "IP:PORT". <u>Examples:</u> 192.168.1.56:1123 somehost.org:22 If method is HTTP GET or HTTP POST : the URL. <u>Examples:</u> http://servername.com/path/to/resource.html https://example.com/page.asp http://70.23.45.67:890/~pjeanson/image.png	ASCII
...		Parameter1[..]		ASCII

[N-2, N-1]		CRC	CRC	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

0x83 - Socket Operation Complete

This message is sent from the Cellular Modem to the Host.

This message indicates the completion of a **Socket Payload Send** message.

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Socket Operation Complete	0x83
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Indicates the overall status of the request-reply operation. 0 - Indicates the request-reply operation was successful. 1: ERROR, APN not set. 2: ERROR, Cannot connect to destination host. 3: ERROR, IO error while sending request payload. 4: ERROR, IO error while receiving response payload. 5: ERROR, a generic IO error occurred. 6: ERROR, another Socket Payload Send operation is in progress. 7: ERROR, the send operation was cancelled (Socket Cancel Send).	[0 - 7]
3	7	MSG TYPE	REQUEST	0
[4, 5]		Bytes Sent	Number of bytes from the request that were successfully transmitted to the host. <u>Note:</u> If the payload sent is greater than 65535, this parameter may not be reliable.	[0x0000 - 0xFFFF]
[6, 7]		CRC	CRC	[0x0000 - 0xFFFF]
8		EOF	End-of-frame	0xCE

0x84 - Socket Cancel Send

Allows cancelling an ongoing **Socket Payload Send** operation of type TCP.

Use this message if a very long send operation prevents the sending of new information.

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Socket Cancel Send	0x84
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Socket Cancel Send	0x84
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Status Code:</u> 0: Indicates the send operation was cancelled. 1: ERROR, no send operation in progress.	[0, 1]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

0x85 - Socket Get Configured APN

This message allows retrieving the configured Access Point Name (APN) used to bridge the cellular network with the Internet.

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Socket Get APN	0x85
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Socket Get APN	0x85
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status Code: 0: Indicates success. 1: ERROR, Unknown.	[0, 1]
3	7	MSG TYPE	REPLY	1
4		Parameter1 length	Number of bytes for the APN	[0x00 - 0xFF]
...		Parameter1[..]	The access point name (APN).	
5 + parameter 1 length		parameter2 length	Number of bytes in the APN authentication user name.	[0x00-0xFF]
...		Parameter2[..]	The user name for authentication to the APN.	
6+parameter 1 & 2 lengths		parameter3 length	Number of bytes in the APN authentication password.	[0x00-0xFF]
...		Parameter3[..]	The password for authentication to the APN.	
[N-2, N-1]		CRC	CRC	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

4.7. Modem configuration (0x90 - 0x9F)

0x90 - Config Set Modem Speed

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Set modem speed	0x90
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		Mode	0: Auto 1: 2G 2: 3G	[0 - 2]
[5, 6]		CRC	CRC	[0x0000 - 0xFFFF]
7		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Set modem speed	0x90
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Status Code:</u> 0: Success 1: Error	[0, 1]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

0x91 - Config Get Modem Speed

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get modem speed	0x91
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get modem speed	0x91
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status Code: 0: Success 1: Error	[0, 1]
3	7	MSG TYPE	REPLY	1
4		Mode	1: 2G 2: 3G	[1 - 2]
[5, 6]		CRC	CRC	[0x0000 - 0xFFFF]
7		EOF	End-of-frame	0xCE

4.8. Status / Control (0xA0 - 0xAF)

0xA0 - Get tower connection status

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get tower connection status	0xA0
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get tower connection status	0xA0
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Tower Connection Status:</u> 0: Not connected 1: Connected	[0, 1]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

0xA1 - Get RSSI

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get RSSI	0xA1
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get RSSI	0xA1
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status Code: 0: Success 1: Error	[0, 1]
3	7	MSG TYPE	REPLY	1
4		RSSI	Signal quality (dBm) 0: -113 dBm or less 1: -111 dBm 2-30: -109... -53 dBm 31: - 51 dBm or greater 99: not known or not detectable	[0 - 31, 99]
[5, 6]		CRC	CRC	[0x0000 - 0xFFFF]
7		EOF	End-of-frame	0xCE

0xA2 - Get IMEI

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get IMEI	0xA2
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get IMEI	0xA2
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status Code: 0: Success 1: Error	[0, 1]
3	7	MSG TYPE	REPLY	1
[4 - 19]		IMEI	IMEI	ASCII
[20, 21]		CRC	CRC	[0x0000 - 0xFFFF]
22		EOF	End-of-frame	0xCE

0xA3 - Get SIM

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get SIM	0xA3
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get SIM	0xA3
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status Code: 0: Success 1: Error	[0, 1]
3	7	MSG TYPE	REPLY	1
[4 - 23]		SIM	SIM	ASCII
[24, 25]		CRC	CRC	[0x0000 - 0xFFFF]
26		EOF	End-of-frame	0xCE

0xA4 - Get Voltage Vin

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get voltage Vin	0xA4
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get voltage Vin	0xA4
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status Code: 0: Success 1: Error	[0, 1]
3	7	MSG TYPE	REPLY	1
[4, 5]		VOLTAGE	Vin voltage 0 to 1200mV	[0 - 1200]
[6, 7]		CRC	CRC	[0x0000 - 0xFFFF]
8		EOF	End-of-frame	0xCE

0xA5 - Set device functionality level

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Set device functionality level	0xA5
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
4		MODE	<u>Functionality level:</u> 0: Power Off 1: Airplane mode enable 2: Airplane mode disable 3: Reboot	[0 - 3]
[5, 6]		CRC	CRC	[0x0000 - 0xFFFF]
7		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Set device functionality level	0xA5
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Status Code:</u> 0: Success 1: Error	[0, 1]
3	7	MSG TYPE	REPLY	1
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

0xA6 - Get Network Registration Status Information

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get Network Reg. Status	0xA6
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get tower connection status	0xA6
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Status Code:</u> 0: Success 1: Error	[0, 1]
3	7	MSG TYPE	REPLY	1
4		NET CONNECTED	<u>Network Connected:</u> 0: Not Connected 1: Connected	[0, 1]
5		Radio Access Technology	<u>Radio Access Technology:</u> 0: GSM 2: UTRAN 3: GSM w/EGPRS 4: UTRAN w/HSDPA 5: UTRAN w/HSUPA 6: UTRAN w/HSDPA and w/HSUPA 7: Information not available.	[0 - 7]
[6, 7]		CRC	CRC	[0x0000 - 0xFFFF]
8		EOF	End-of-frame	0xCE

4.9. Version (0xC0 - 0xCF)

0xC0 - Version Get Cellular Modem Firmware

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get Cellular Modem Firmware	0xC0
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get Cellular Modem Firmware	0xC0
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	<u>Status Code:</u> 0: Success 1: Error	[0, 1]
3	7	MSG TYPE	REPLY	1
4		VER LENGTH	Length of the firmware version string	[0x00 - 0xFF]
[5 - N-3]		BRIOW SW REV	Cellular Modem Firmware Version string. Each portion of the version number is allowed a maximum of two decimal digits (i.e., 0-99) <u>Format:</u> <MAJOR>.<MINOR>.<BUILD>	ASCII
[N-2, N-1]		CRC	CRC	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

0xC1 - Version Get Cellular Modem Hardware

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get Cellular Modem Hardware	0xC1
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get Cellular Modem Hardware	0xC1
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status Code: 0: Success 1: Error	[0, 1]
3	7	MSG TYPE	REPLY	1
4		VER LENGTH	Length of the hardware version string	[0x01 - 0xFF]
[5 - N-3]		BRIOWIRELESS HW REV	Cellular Modem Hardware version Format: <YYY>-<LOT> YYY: Board (Ex: 100, 110, 200) LOT: Production lot (Ex: BC1, BC1A)	ASCII
[N-2, N-1]		CRC	CRC	[0x0000 - 0xFFFF]
N		EOF	End-of-frame	0xCE

0xC2 - Version Get Cinterion Firmware

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get Cinterion firmware version	0xC2
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get Cinterion firmware version	0xC2
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status Code: 0: Success 1: Error	[0, 1]
3	7	MSG TYPE	REPLY	1
[4 - 9]		CINT FW VER	Cinterion firmware version format is XX.YYYY	ASCII
[10, 11]		CRC	CRC	[0x0000 - 0xFFFF]
12		EOF	End-of-frame	0xCE

0xC3 - Version Get Cinterion Hardware

Request

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get Cinterion hardware version	0xC3
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Unused	0
3	7	MSG TYPE	REQUEST	0
[4, 5]		CRC	CRC	[0x0000 - 0xFFFF]
6		EOF	End-of-frame	0xCE

Reply

Byte #	Bit #	Name	Description	Value
0		SOF	Start-of-frame	0x7E
1		CMD	Get Cinterion hardware version	0xC3
2		ID	Packet ID	[0x01 - 0xFF]
3	[0 - 6]	STATUS	Status Code: 0: Success 1: Error	[0, 1]
3	7	MSG TYPE	REPLY	1
[4 - 10]		CINT HW REV	Cinterion hardware version format is XXXX-YY XXXX: Name (Ex: EHS5) YY: Type (Ex: US)	ASCII
[11, 12]		CRC	CRC	[0x0000 - 0xFFFF]
13		EOF	End-of-frame	0xCE