



串口协议

文档版本: 20240613

[查看在线版本](#)

目录

1 串口通信要求	2
2 帧格式	3
3 基础协议	4
3.1 查询产品信息	4
3.2 报告设备联网状态	6
3.3 重置 NB-IoT 模组	8
3.4 实时状态上报	9
3.5 记录型状态上报	11
3.6 模组命令下发	15
3.7 MCU 返回信息	16
3.8 命令或状态数据单元	17
3.9 获取本地时间	19
3.10 获取格林威治时间	21
3.11 查询 NB-IoT 信号强度	22
3.12 查询模组内存	24
3.13 MCU 升级服务	25
3.14 下载 MCU 文件（下载启动）	29
3.15 下载 MCU 文件（文件传输）	32
3.16 查询文件下载百分比	33
3.17 查询绑定状态	35
3.18 设置 NB-IoT 模组睡眠锁	36
3.19 设置 NB-IoT 网络心跳间隔时间	37
3.20 设置 NB-IoT 记录型数据唤醒间隔时间	38
3.21 自动发送心跳包	40
3.22 获取设备联网状态	41
3.23 获取国际移动用户识别码 IMSI	43
3.24 获取 SIM 卡识别码 ICCID	44
3.25 获取拓展信号质量 CESQ	45
3.26 设置活动定时器 T3324	48
3.27 设置定时器 T3412	50
3.28 固件升级电量检测	51
3.29 获取设备 IMEI	52

3.30 报告设备运行状态	54
3.31 获取设备运行状态	55
3.32 设置设备进入睡眠	58
3.33 设置设备进入 PSM	59
3.34 设置设备访问的 APN	60
3.35 查询文件下载进度	62
3.36 强制重启设备	63
3.37 查询活动定时器 T3324 时间	64
3.38 查询定时器 T3412 时间	66
3.39 查询网络心跳间隔时间	67
3.40 设置开机离散信息	68
4 协议版本	70

本文介绍 NB-IoT 通讯方式设备采用 MCU SDK 开发时使用的涂鸦串口协议。

1 串口通信要求

- 波特率：9600/115200
- 数据位：8
- 奇偶校验：无
- 停止位：1
- 数据流控：无
- MCU：控制板控制芯片，与涂鸦模组通过串口对接。在涂鸦的协议设计上，所有包的交互均为全双工通信。

2 帧格式

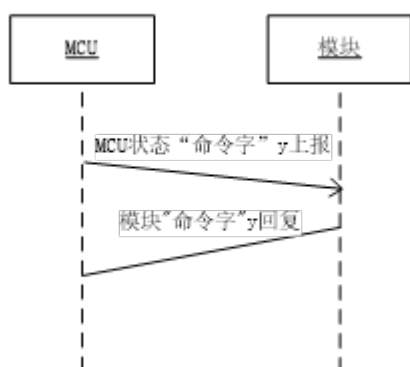
- 大于 1 字节的数据均采用大端模式传输。
- 协议中举例数据为十六进制。
- NB-IoT 模组主动发起的通信发送包超时时长为 1 秒，重发机制会重发三包。
- 通常采用同命令字一发一收同步机制，即一方发出命令，另一方应答。若发送方超时未收到正确的响应包，则视为超时传输。

...info

具体通信方式以[协议详述](#) 章节为准。

...

- MCU 状态上报则采用同步模式，MCU 状态上报 **命令字** 为 **y**，如下所示：



字段	字节数	说明
帧头	2	固定为 0x55aa。
版本	1	升级扩展用。
命令字	1	具体帧类型。
数据长度	2	大端。
数据	N	数据部分包含网络层传下来的数据，例如 IP 数据包等。
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

3 基础协议

3.1 查询产品信息

产品信息由 Product ID、MCU 软件版本等构成。

- Product ID：对应涂鸦开发者平台 PID（产品标识），由涂鸦开发者平台生成，用于云端记录产品相关信息
- MCU 软件版本：版本号格式采用点分十进制形式。例如 `x.x.x`，其中 $0 \leq x \leq 99$ ， x 为十进制数。

模组发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x01
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组发送示例

```
1 55 aa 00 01 00 00 00
```

MCU 返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x01
数据长度	2	N

字段	字节数	说明
数据	N	默认: <code>{"p":"gl9iswyeobu5s93j", "v":"1.0.0", "s":"psm", "c":"isp"}</code> 拓展: <code>{"p":"gl9iswyeobu5s93j", "v":"1.0.0", "s":"psm", "c":"isp", "t":"22092217004703"}</code>
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

数据

参数	状态	说明
p	必选	产品 ID，即在开发者平台所创建的产品的 PID。
v	必选	MCU 版本。
s	必选	设备的功耗模式。 <code>psm</code> ：省电模式 <code>drx</code> ：不连续接收模式 <code>edrx</code> ：扩展不连续接收模式
c	必选	设备连接云的方式。 <code>isp</code> ：通过运营商平台转发到云端 <code>tuya</code> ：直接连接云端

参数	状态	说明
t	可选	MCU 配置 NB-IoT 模组的 RTC 时间，可在需要比较准确时间的场景中增加该配置，例如严格的本地定时的场景，按以下信息格式： 22092217004703。 Data[0-1] 为年份，22 表示 2022 年 Data[2-3] 为月份，从 01 开始到 12 结束 Data[4-5] 为日期，从 01 开始到 31 结束 Data[6-7] 为时钟，从 00 开始到 23 结束 Data[8-9] 为分钟，从 00 开始到 59 结束 Data[10-11] 为秒钟，从 00 开始到 59 结束 Data[12-13] 为星期，从 01 开始到 07 结束，01 代表星期一

MCU 返回示例

```
1 55 AA 00 01 00 38 7B 22 70 22 3A 22 67 6C 39 69 73 77 79 65 6F 62 75
2 35 73 39 33 6A 22 2C 22 76 22 3A 22 31 2E 30 2E 30 22 2C 22 73 22 3
3 A 22 70 73 6D 22 2C 22 63 22 3A 22 69 73 70 22 7D 02
```

3.2 报告设备联网状态

设备联网状态	描述	状态值
状态 1	搜索 NB 网络中。	0x01
状态 2	已搜索 NB 网络。	0x02
状态 3	已连接到电信平台，设备未绑定。	0x03
状态 4	设备已绑定，已连接到云端。	0x04
状态 5	设备附着基站被拒绝。	0x05

模组发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x02
数据长度	2	0x0001
数据	1	指示 NB-IoT 模组工作状态： 0x01：状态 1 0x02：状态 2 0x03：状态 3 0x04：状态 4 0x05：状态 5
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组发送示例

设备已绑定，已连接到云端。

```
1 55 aa 00 02 00 01 04 06
```

MCU 返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x02
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 返回示例

```
1 55 aa 00 02 00 00 01
```

3.3 重置 NB-IoT 模组

设备将会重置一切设置，并刷新模组到出厂状态，然后模组重新向云端请求激活。

MCU 发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x03
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送示例

```
1 55 aa 00 03 00 00 02
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x03
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回示例

```
1 55 aa 00 03 00 00 02
```

3.4 实时状态上报

设备联网状态 必须为 **已连接到云端**，否则实时状态上报命令将执行失败。

MCU 可以通过实时状态上报命令，将数据的实时状态上报到模组。

- 采用同步命令机制，即 MCU 数据上报后需等待模组返回上报结果。在信号良好情况下，通常在 5 秒内可收到云端的上报应答结果。
- 命令支持单个或多个数据单元上报。用户根据需要选择发送组包的方式。具体数据包参见示例。

使用场景

要求实时推送状态的设备，类似报警器等。

MCU 发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x05
数据长度	2	取决于协议版本，状态数据单元 类型以及个数。
数据消息 ID	2	仅支持协议版本 $\geq 0x01$ ，如果协议版本 $= 0x00$ 则不包括该字段。
数据	N	一个或多个组合 状态数据单元 组。
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x05
数据长度	2	当协议版本 $= 0x00$ 时，为 0x0001，当协议版本 $\geq 0x01$ 时为 0x0003。
数据消息 ID	2	仅支持协议版本 $\geq 0x01$ ，如果协议版本 $= 0x00$ 则不包括该字段。
数据	1	0x00 成功 0x01 失败
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

- 单个状态数据单元上报样例（协议版本 $= 0x00$ ）

状态数据：DP 109，布尔型变量，数值为 1。

```
1 55 aa 00 05 00 05 6d 01 00 01 01 79
```

• 多个状态数据单元上报样例（协议版本 =0x00）

- 状态数据 1：DP 109，布尔型变量，数值为 1。
- 状态数据 2：DP 102，字符型变量，数值为 **201804121507**（具体传输对应 ASCII 值）。

```
1 55 aa 00 05 00 15 6d 01 00 01 01 66 03 00 0c 32 30 31 38 30 34
   31 32
2 31 35 30 37 5d
```

• 单个状态数据单元上报样例（协议版本 >=0x01）

- 数据消息 ID：0x00FF。
- 状态数据：DP 109，布尔型变量，数值为 1。

```
1 55 AA 01 05 00 07 00 FF 6D 01 00 01 01 7B
```

• 多个状态数据单元上报样例（协议版本 >=0x01）

- 数据消息 ID：0x0100。
- 状态数据 1：DP 109，布尔型变量，数值为 1。
- 状态数据 2：DP 102，字符型变量，数值为 **201804121507**（具体传输对应 ASCII 值）。

```
1 55 AA 01 05 00 17 01 00 6D 01 00 01 01 66 03 00 0C 32 30 31 38
   30 34
2 31 32 31 35 30 37 61
```

3.5 记录型状态上报

MCU 的记录型数据是由多个 DP 组合的整条数据，因此需要整体上报。上报可通过向 NB-IoT 模组发送下方描述的命令。若数据上报时设备断网，这条数据则会暂时在模组端存储，在下次网络通畅、有数据上报时，模组会再次上传存储的数据。

- 滞留数据：当模组存在滞留的记录时，每上报成功一条内容，模组就会主动发送一条命令字为 08，数据为 01 的回复包，并告知 MCU。
- 设备联网：当网络状态正常时，模组首次上电需要大约 30 秒以连接服务器。此后每当记录产生时，可以先让 MCU 等待[报告设备联网状态](#) 章节所描述的设备联网状态包。但是若等待 6 秒以上还没收到模组连接成功的状态报告，也可选择使用该条指令上报数据，且此后重复等待模组联网成功并返回上报结果。
- 数据长度限制：单次记录上报的数据区域（多个状态数据单元）最大长度为 100，根据实际的 DP 的数据的不同，最后组合实际储存的长度可能会有所变化，因此当模组没有网络时，超过限制长度的数据上报会导致 NB-IoT 模组返回记录发送失败。
- 记录条数限制：模组可以储存历史记录的条数最长限制为 20 条。当超过 20 条后，模组会从最早储存的记录开始循环覆盖。
- 返回状态：NB-IoT 模组接收到一条数据并成功上报后会返回上报成功的状态报告。
 - 00：推送成功
 - * 推送成功且模组无滞留数据
 - * 模组未连接网络，单记录成功存储进闪存
 - 01：推送成功但模组存在滞留数据
 - 02：推送失败。
- 时间标记：在数据上报时添加时间数据，是为了保证记录数据在设备没有网络的情况下，上报的时间和有网络时上报的时间一致。如果以到达服务端的时间为准，则当设备暂时处于没有网络的状态时，产生的记录数据会和在下一次网络恢复正常后上传的记录时间不符合。所以每当设备第一次和 NB-IoT 模组配网的时候，在网络环境可以保证的情况下，MCU 最好可以通过协议获取一次准确的时间数据。之后上传记录的时候带上当前设备的时间数据。服务端的记录时间也以设备提供的时间为准。

使用场景

对于类似门锁等包含多个需要上报到服务端、当作整条信息处理的 DP 数据的类记录型的设备，若设备短暂断网，本命令会要求设备暂存无法上报成功的数据，以满足记录型设备的上报需求。命令上报需要带上本地的时间。

MCU 发送

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x08

字段	字节数	说明
数据长度	2	取决于协议版本，状态数据单元 类型以及个数。
数据消息 ID	2	仅支持协议版本 $\geq 0x01$ ，如果协议版本 $= 0x00$ 则不包括该字段。
数据	7	需要使用 MCU 的时间戳，按以下信息格式： Data[0] 为年份， 0x00 表示 2000 年 Data[1] 为月份， 从 1 开始到 12 结束 Data[2] 为日期， 从 1 开始到 31 结束 Data[3] 为时钟， 从 0 开始到 23 结束 Data[4] 为分钟， 从 0 开始到 59 结束 Data[5] 为秒钟， 从 0 开始到 59 结束 Data[6] 为星期， 从 1 开始到 7 结束， 1 代表星期一 说明： 需要使用模组的时间戳时，只需要将该字段填充 0。模组内部将自动将内部 RTC 时间填充到记录型数据上。

字段	字节数	说明
模组发送	N	一个或多个组合 状态数据单元 组
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送示例

- 单个状态数据单元（协议版本 =0x00）

DP 109，布尔型变量，数值为 1。以模组内部 RTC 时间为准。

```
1 55 aa 00 08 00 0c 00 00 00 00 00 00 00 00 6d 01 00 01 01 d1
```

- 多个状态数据单元（协议版本 =0x00）

- 状态数据 1：DP 109，布尔型变量，数值为 1。
- 状态数据 2：DP 102，字符型变量，**201804121507**（具体传输对应 ASCII 值）。以服务端时间为准。

```
1 55 aa 00 08 00 1c 00 00 00 00 00 00 00 00 6d 01 00 01 01 66 03 00
   0c 32
2 30 31 38 30 34 31 32 31 35 30 37 a7
```

- 单个状态数据单元（协议版本 >=0x01）

- 数据消息 ID：0x00FF。
- DP 109，布尔型变量，数值为 1。以模组内部 RTC 时间为准。

```
1 55 AA 01 08 00 0E 00 FF 00 00 00 00 00 00 00 6D 01 00 01 01 85
```

- 多个状态数据单元（协议版本 >=0x01）

```
1 - 数据消息 ID: 0x0100。
2 - 状态数据 1: DP 109, 布尔型变量, 数值为 1。
3 - 状态数据 2: DP 102, 字符型变量, **201804121507** (具体传输对应
  ASCII 值)。以服务端时间为准。
4   ...
5   55 AA 01 08 00 1E 01 00 00 00 00 00 00 00 00 6D 01 00 01 01 66
6   0
7   3 00 0C 32 30 31 38 30 34 31 32 31 35 30 37 6B
8   ...
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x08
数据长度	2	当协议版本 = 0x00 时为 0x0001, 当协议版本 >= 0x01 时为 0x0003。
数据消息 ID	2	仅支持协议版本 >= 0x01, 如果协议版本 = 0x00 则不包括该字段。
数据	1	0x00: 上报成功。 0x01: 当前记录上报成功, 滞留记录待上报。 0x02: 上报失败。
校验和	1	从帧头开始按字节求和, 得出的结果对 256 求余。

3.6 模组命令下发

命令下发 为异步处理协议, MCU 收到相关控制包, 确认接收到回复下发包后, 完成相关的控制动作。MCU 状态反馈通过 MCU 的**状态上报**实现。

模组发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x09
数据长度	2	取决于 命令数据单元 类型以及个数。
数据	N	命令数据单元 组。
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组发送信息示例

系统开关对应 3 号 DP，使用布尔型变量，开机数值为 1。

```
1 55 aa 00 09 00 05 03 01 00 01 01 13
```

3.7 MCU 返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x09
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 返回信息示例

```
1 55 aa 00 09 00 00 08
```

3.8 命令或状态数据单元

Datapoint 命令/状态数据单元如下所示。

数据字段属性	长度（字节）	说明
dpid	1	设备功能序号
type	1	对应开放平台上设备功能具体的数据类型： raw: raw 型设备功能 (模组输入) 表示值： 0x00 长度（字节）：N bool: 布尔型，取值为 0x00 或 0x01 表示值： 0x01 长度（字节）：1 value: 对应整型，大端表示 表示值： 0x02

数据字段属

性	长度（字节）	说明
---	--------	----

		长度（字节）：4
--	--	----------

		string：对应字符串
--	--	--------------

		表示值： 0x03
--	--	--------------

		长度（字节）：N
--	--	----------

		enum：枚举类型，取值范围 0~255
--	--	----------------------

		表示值： 0x04
--	--	--------------

		长度（字节）：1
--	--	----------

		bitmap： 故障型，长度大于 1 字节时，大端表示
--	--	--------------------------------

		表示值： 0x05
--	--	--------------

		长度（字节）：1、2 或 4
--	--	----------------

len	2	长度对应 value 的字节数（大端）
-----	---	---------------------

value	1/2/4/N	hex 表示，大于 1 字节采用大端传输
-------	---------	----------------------

除 Raw 类型的 Datapoint 状态数据单元外，其他类型均属于 Obj 型。状态数据 可含多个 Datapoint 命令数据单元。

3.9 获取本地时间

- 需要等待设备连接上网络（报告设备联网状态）后，开始发送获取本地时间的数据包。
- 在设备连接上服务器后，在网络情况非常差的情况下，有可能会获取时间数据失败。对于依赖时间的设备（例如门锁），当本地时间没有校准过的情况下获取失败时，需要完善间隔 3 秒的获取时间值的功能，确保时间数据获取成功。

MCU 发送

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x06
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送示例

```
1 55 aa 00 06 00 00 05
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x06
数据长度	2	0x0008

字段	字节数	说明
数据	Data	数据长度为 8 字节： Data[0] 为是否获取时间成功标志，为 0 表示失败，为 1 表示成功 Data[1] 为年份，0x00 表示 2000 年 Data[2] 为月份，从 1 开始到 12 结束 Data[3] 为日期，从 1 开始到 31 结束 Data[4] 为时钟，从 0 开始到 23 结束 Data[5] 为分钟，从 0 开始到 59 结束 Data[6] 为秒钟，从 0 开始到 59 结束 Data[7] 为星期，从 1 开始到 7 结束，1 代表星期一
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回示例

本地时间：18 年 9 月 17 日 16 时 9 分 5 秒星期一。

- 如设备在国内激活使用，则当地时间为北京时间 (东 8 区)。
- 如果设备在国外激活使用，则当地时间为设备所处时区时间。

```
1 55 aa 00 06 00 08 01 12 09 11 10 09 05 01 59
```

3.10 获取格林威治时间

- 格林威治时间不带有时区和夏令时等相关因素，因此支持动态密码功能的设备（门锁等）不需要显示设备所在当地时间，本地只需要运行格林威治时间，在使用支持记录型上报的通道时也只需上报格林威治时间。
- 带动态密码功能的设备（门锁等）如需显示设备所在地的当前时间，在设备本地运行格林威治时间的同时，也需要在定时获取本地时间，并在本地储存格林威治时间和当地时间的时差，每次显示本地时间时，只需在格林威治时间的基础上加上时间差算出当地时间即可

MCU 发送

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x10
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送示例

```
1 55 aa 00 10 00 00 0F
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x10
数据长度	2	0x0008
数据	8	数据长度为 8 字节：

字段	字节数	说明
		Data[0] 为是否获取时间成功标志，为 0 表示失败，为 1 表示成功
		Data[1] 为年份，0x00 表示 2000 年
		Data[2] 为月份，从 1 开始到 12 结束
		Data[3] 为日期，从 1 开始到 31 结束
		Data[4] 为时钟，从 0 开始到 23 结束
		Data[5] 为分钟，从 0 开始到 59 结束
		Data[6] 为秒钟，从 0 开始到 59 结束
		Data[7] 为星期，从 1 开始到 7 结束，1 代表星期一
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回示例

格林威治时间：2018 年 9 月 17 日 8 时 21 分 3 秒 星期一。

```
1 55 aa 00 10 00 08 01 12 09 11 08 15 03 01 65
```

3.11 查询 NB-IoT 信号强度

查询设备 NB-IoT 的信号强度。

MCU 发送

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x0b
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送示例

```
1 55 aa 00 0b 00 00 0a
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x0b
数据长度	2	0x0002
数据	2	Data[0] 取值： 0x00：失败 0x01：成功当 Data[0] 为 0x01 时，Data[1] 表示 RSRP 即： 0：-140 dBm 或以下 1：-140 dBm ≤ RSRP < -139 dBm 2：-139 dBm ≤ RSRP < -138 dBm

字段	字节数	说明
		...
		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: 未知
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回示例

```
1 55 aa 00 0b 00 02 01 50 5D
```

3.12 查询模组内存

查询 NB-IoT 模组的剩余内存。

MCU 发送

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x0f
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送示例

```
1 55 aa 00 0f 00 00 0e
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x0f
数据长度	2	0x0004
数据	2	Data[0-3]：剩余 RAM 内存，单位字节
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回示例

```
1 55 aa 00 0f 00 04 00 01 F4 00 07
```

3.13 MCU 升级服务

- 何时进行 MCU 升级，由您在涂鸦开发者平台的产品页面配置相关升级选项触发。模组仅作为支持 MCU 升级的数据传输通道，不对数据内容做任何解析。
- 目前涂鸦的 MCU 升级支持下面四种升级方式的配置：
 - App 提醒升级：有新版本后，用户每次进入设备控制面板都会收到升级提醒的弹窗，是否确认升级由用户自己在 App 确认
 - App 强制升级：App 端会有升级提醒弹窗，如果用户不确认升级就没法正常使用这个产品的控制面板。
 - App 检测升级：App 端不会有任何升级提醒的弹窗，必须要用户在 App 端自己去单击相关固件版本检测，如果有高版本的固件配置才会显示升级提示信息。
- MCU 升级流程：
 1. NB-IoT 模组发送完所有的升级包，重新发送 01 命令字（查询产品信息）。

2. MCU 需要在一分钟回复产品信息中的 MCU 软件版本号（升级后的版本号）。版本号需要和在涂鸦开发者平台配置升级的版本号保持一致。

升级启动（升级包大小、文件 CRC32 通知）

模组发送

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x0C
数据包长度	2	0x0008
数据	8	Data[0-3]: 文件长度, 0x000013CF Data[4-7]: 文件 CRC32, 0xC20A5FBB
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

示例

```
1 55 AA 00 0C 00 08 000013CF C20A5FBB DB
```

MCU 返回

返回断点地址，可以支持从断点位置下载。

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x0C
数据长度	2	0x0001/0x0005

字段	字节数	说明
数据	5	Data[0]: 0x00: 64 字节每包传输 0x01: 128 字节每包传输 0x02: 256 字节每包传输 Data[1-4]: 断点下载偏移地址 注意: 数据长度为 0x0001 时, 不传断点地址的话, 默认从 0 地址开始完整下载文件。这里为了兼容之前 0x0001 长度协议。
校验和	1	从帧头开始按字节求和, 得出的结果对 256 求余。

示例

1	55 AA 00 0C 00 01 00 0C	// 64 字节
2	55 AA 00 0C 00 01 01 0D	// 128 字节
3	55 AA 00 0C 00 01 02 0E	// 256 字节
4	55 AA 00 0C 00 05 02 00000800 1A	// 从 2048 字节断点地址开始下载

升级包传输

模组发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x0D
数据长度	2	0x0044 (包偏移 + 数据长度)
数据	4	

字段	字节数	说明
		Data[0-3]: OTA 包偏移, 0x00000000
		Data[4 - n]: 实际 OTA 数据
校验和	1	从帧头开始按字节求和, 得出的结果对 256 求余。

示例

升级文件大小 530 Byte, 最后一包也要回复。

- 第一包, 包偏移 0x00000000, 数据包长度 256

```
1 55 AA 00 0D 01 04 00000000 xxx...xxx XX
```

- 第二包, 包偏移 0x00000100, 数据包大小 256

```
1 55 AA 00 0D 01 04 00000100 xxx...xxx XX
```

- 倒数第 2 包, 包偏移 0x00000200, 数据包大小 18

```
1 55 AA 00 0D 01 04 00000200 xxx...xxx XX
```

- 最后一包, 包偏移 0x00000212, 数据包大小 0

```
1 55 AA 00 0D 00 04 00000212 XX
```

MCU 返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x0D

字段	字节数	说明
数据长度	2	0x0000/0x0001
数据	1	Data[0]: 00: MCU CRC32 校验成功。 01: MCU CRC32 校验失败。 注意： 最后一包需要等待 MCU 回复 CRC32 校验结果，兼容之前的协议，旧协议不改动也能正常运行，可参考示例。
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

示例

升级文件大小为 530 Byte，最后一包也要回复。

- 模组最后一包，包偏移 0x00000212，数据包大小 0：

```
1 55 AA 00 0D 00 04 00000212 XX
```

- MCU 回复：

```
1 10s 不回复 // 默认 MCU 校验文件 CRC32 成功
2 55 AA 00 0D 00 00 0C // MCU 校验文件 CRC32 成功
3 55 AA 00 0D 00 01 00 0D // MCU 校验文件 CRC32 成功
4 55 AA 00 0D 00 01 01 0E // MCU 校验文件 CRC32 失败
```

3.14 下载 MCU 文件（下载启动）

区别于 MCU OTA 文件升级服务，这个功能是 MCU 主动请求下载文件。

MCU 发送

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x1e
数据长度	2	N
数据	52	Data[0]: 子命令, 0x00, 请求信息。 Data[1-n]: <code>{"f": "calendar.bin", "p": "gps?in=567", "o": 5678}</code> 参数说明: f: 文件名, 字符串类型。 p: 下载参数, 无参数可传空字符, 字符串类型。 o: 文件偏移地址, 可指定偏移下载, 用于断点下载, 整型。 Data[0]: 子命令, 0x01, 继续下载。 Data[0]: 子命令, 0x02, 停止下载。
校验和	1	从帧头开始按字节求和, 得出的结果对 256 求余。

MCU 发送示例

- 子命令, 0x00, 请求信息:

```
1 55 aa 00 1e 00xx 00 xxxxxxxxxxxxxxxx XX
```

- 子命令, 0x01, 继续下载:

```
1 55 aa 00 1e 0001 01 XX
```

- 子命令, 0x02, 停止下载:

```
1 55 aa 00 1e 0001 02 XX
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x1e
数据长度	2	0x0009
数据	9	
		Data[0]: 0x10, 文件信息获取成功
		Data[1-4]: 文件长度, 0x000013CF
		Data[5-8]: 文件 CRC32, 0xC20A5FBB
		Data[0]: 0x11, 文件信息获取失败
校验和	1	从帧头开始按字节求和, 得出的结果对 256 求余。

模组返回示例

- 子命令, 0x10, 文件信息获取成功:

```
1 55 aa 00 1e 0001 10 XX
```

- 子命令, 0x11, 文件信息获取失败, 停止下载:

```
1 55 aa 00 1e 0001 11 XX
```

3.15 下载 MCU 文件（文件传输）

模组发送

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x1f
数据长度	2	0x0044（包偏移 + 数据长度）
数据	4	Data[0-3]: 数据包偏移，0x00000000 Data[4 - n]: 实际下载数据
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

示例

下载文件大小为 530 Byte，每个数据包固定长度 256。

- 中间包：

```
1 55 AA 00 1f 01 04 00000000 xxx...xxx XX
2 55 AA 00 1f 01 04 00000100 xxx...xxx XX
```

- 最后一包，包偏移 0x00000212，数据包大小 0：

```
1 55 AA 00 1f 00 04 00000212 XX
```

MCU 返回

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x1f
数据长度	2	0x0000/0x0001
数据	1	Data[0]: 00: MCU CRC32 校验成功。 01: MCU CRC32 校验失败。
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

示例

- 中间包回复：

```
1 55 AA 00 1f 0000 0C // 中间接收一包数据成功
```

- 最后一包回复：

```
1 55 AA 00 1f 0001 00 0D // MCU 校验文件 CRC32 成功
2 55 AA 00 1f 0001 01 0E // MCU 校验文件 CRC32 失败
```

3.16 查询文件下载百分比

只要是文件下载，就能查询百分比，包括 Schema 文件、NB 升级文件、MCU 升级文件、MCU 通用文件下载。任何一个运行都有下载百分比。

MCU 发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xc3
数据长度	2	0x0000
数据	0	0
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送示例

```
1 55 aa 00 c3 00 00 c2
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xc3
数据长度	2	0x0002
数据	2	Data[0]: 0x00: 未下载 0x01: 正在下载 Data[1]: 0-100, 下载百分比。
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

示例

```
1 55 aa 00 c3 00 01 01 18 dc
```

3.17 查询绑定状态

查询设备是否已经被用户绑定。

MCU 发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xbb
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送示例

```
1 55 aa 00 bb 00 00 0a
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xbb
数据长度	2	0x0001
数据	1	数据长度为 1 字节。 0x00：未绑定 0x01：绑定

字段	字节数	说明
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息

```
1 55 aa 00 bb 00 01 01 bc
```

3.18 设置 NB-IoT 模组睡眠锁

MCU 将睡眠锁设置为 1 后，模组将禁止进入睡眠。设置为 0 后，模组将自动在 IDLE 进入 PSM 模式低功耗待机。在 PSM 模式下，可以拉低 PSM-INT 引脚，唤醒模组。

MCU 发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xb2
数据长度	2	0x0001
数据	1	数据长度为 1 字节。 0x00：解锁睡眠锁。 0x01：锁住睡眠锁，禁止自动进入睡眠。
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送示例

```
1 55 aa 00 b2 00 01 01 00
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xb2
数据长度	2	0x0000
数据	0	0
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息

```
1 55 aa 00 b2 00 00 b1
```

3.19 设置 NB-IoT 网络心跳间隔时间

用户可以根据自己的业务需求，设置模组的 NB-IoT 网络心跳周期，默认心跳间隔是 8 小时。

MCU 发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xb3
数据长度	2	0x0004
数据	4	uint32_t 类型，单位为秒，大端格式。
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送示例

设置心跳间隔为 1 小时。

```
1 55 aa 00 b3 00 04 00 00 0e 10 da
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xb3
数据长度	2	0x0001
数据	1	0x01：设置成功 0x00：设置失败
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息

```
1 55 aa 00 b3 00 01 01 b4
```

3.20 设置 NB-IoT 记录型数据唤醒间隔时间

- Flash 内记录型数据未成功发出时，使用该时间进行下一次唤醒。记录型数据成功发出后，恢复默认的唤醒心跳间隔。
- 最小唤醒间隔 120 秒。如果设置时间小于 120 秒，按照 120 秒执行。
- 用户可以设置记录型数据唤醒时间功能，进行紧急记录型数据发送的快速唤醒重发。

MCU 发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xc1
数据长度	2	0x0004
数据	4	uint32_t 类型，单位为秒，大端格式。
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送信息示例

设置下一次唤醒间隔为 3 分钟。

```
1 55 aa 00 c1 00 04 00 00 00 00 b4 78
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xc1
数据长度	2	0x0001
数据	1	0x01：设置成功 0x00：设置失败
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息

```
1 55 aa 00 c1 00 01 01 c2
```

3.21 自动发送心跳包

通过设置网络心跳间隔时间，模组将周期性的自动发送心跳包，本命令可以由 MCU 控制主动发送一个心跳包，以用于主动同步下发数据。在通常情况下，可以由**实时状态上报**（例如上报电量）替代本命令的功能。

MCU 发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xb1
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送信息示例

```
1 55 AA 00 B1 00 00 B0
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xb1
数据长度	2	0x0001
数据	1	

字段	字节数	说明
		0x00: 设置失败
		0x01: 设置成功
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息

```
1 55 AA 00 B1 00 01 01 B2
```

3.22 获取设备联网状态

设备联网状态	描述	状态值
状态 1	搜索 NB 网络中	0x01
状态 2	搜索到 NB 网络	0x02
状态 3	已连接到电信平台，设备未绑定	0x03
状态 4	设备已绑定，已连接到云端	0x04

MCU 发送

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x2b
数据长度	2	0x0000
数据	0	无

字段	字节数	说明
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送示例

```
1 55 aa 00 2b 00 00 2c
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x2b
数据长度	2	0x0001
数据	1	指示 NB-IoT 模组工作状态： 0x01：状态 1 0x02：状态 2 0x03：状态 3 0x04：状态 4
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息示例

设备已绑定，已连接到云端。

```
1 55 aa 00 2b 00 01 04 2f
```

3.23 获取国际移动用户识别码 IMSI

本命令可以由 MCU 主动发送 IMSI 获取包，获取模组的国际移动用户识别码。

MCU 发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xb5
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送信息示例

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

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xb5
数据长度	2	0x000F
数据	15	示例：460113012467340。
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息示例

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

3.24 获取 SIM 卡识别码 ICCID

MCU 主动下发指令至模组，获取 ICCID 指令，模组将 ICCID（长度 20 的字串）返回给 MCU。本命令可以由 MCU 控制，主动发送 ICCID 获取包，以用于获取模组的 SIM 卡识别号码。

MCU 发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xb6
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送信息示例

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

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xb6
数据长度	2	0x0014
数据	20	示例：89861118249000363490。
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息示例

```
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
```

3.25 获取拓展信号质量 CESQ

本命令可以由 MCU 控制，主动发送 CESQ 获取包，以获取模组的拓展信号质量参数。

MCU 发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xb7
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送信息示例

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

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xb7
数据长度	2	0x0006

字段	字节数	说明
数据	6	由 ReLev、BER、Rscp、ECNO、Rsrq、Rsrp 组成。
BYTE[0]	1	RXLev: 接收信号强度等级
		0: -100 dBm 或以下
		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: 未知
BYTE[1]	1	BER: 信道位错误百分率
		0~7: RxQual valuses
		99: 未知
BYTE[2]	1	Rscp: 接收信号码功率
		0: -120dBm 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: $-26\text{dBm} \leq \text{rscp} < -25\text{dBm}$

字段	字节数	说明
BYTE[3]	1	255: 未知
		Ecno: Ec/No
		0: -24dBm or less
		1: -24dBm <= Ec/Lo<-23.5dBm
		2: -23.5dBm <= Ec/Lo<-23dBm
		...
		47: -1dBm <= Ec/Lo<-0.5dBm
		48: -0.5dBm <= Ec/Lo<0dBm
		49: 0 dBm <= Ec/Lo
		255: 未知
BYTE[4]	1	Rsrq: 参考信号接收质量
		0: -19.5dB or less
		1: -19.5dB <= rsrq<-19dB
		2: -19dB <= rsrq<-18.5dB
		...
		32: -4dB <= rsrq<-3.5dB
		33: -3.5dB <= rsrq<-3dB
		34: -3dB <= rsrq
		255: 未知
		Rsrp: 参考信号接收功率
BYTE[5]	1	0: -140dBm or less

字段	字节数	说明
		1: -140dBm<= rsrp<-139dBm
		2: -139dBm<= rsrp<-138dBm
		...
		95: -46dBm<= rsrp<-45dBm
		96: -45dBm<= rsrp<-44dBm
		97: -44dBm<= rsrp
		255: 未知
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息示例

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

3.26 设置活动定时器 T3324

本指令需要 SIM 卡支持。

MCU 主动发送一个 T3324 活动定时器指令设置包。本指令可用于设置 T3324 的超时时间，单位为秒。

MCU 发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xb9

字段	字节数	说明
数据长度	2	0x0004
数据	4	活动时间，单位为秒（s）。 例如，活动时间为 120（0x78）秒，即： act_time[0]: 0x00 act_time[1]: 0x00 act_time[2]: 0x00 act_time[3]: 00x78
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送信息示例

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

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xb9
数据长度	2	0x0001
数据	1	0x00: 失败 0x01: 成功
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息示例

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

3.27 设置定时器 T3412

与设置活动定时器 T3324 相同，设置定时器 T3412 指令需要 SIM 卡支持。

MCU 主动发送一个 T3412 活动定时器指令设置包。本指令可用于设置 T3412 的超时时间(单位：秒)。

MCU 发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xba
数据长度	2	0x0004
数据	4	PSM 时间，单位为秒 (s)。 例如，活动时间为 8 (0x08) 秒，即： psm_time[0]: 0x00 psm_time[1]: 0x00 psm_time[2]: 0x00 psm_time[3]: 00x08
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送信息示例

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

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xba
数据长度	2	0x0001
数据	1	0x00: 失败 0x01: 成功
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息示例

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

3.28 固件升级电量检测

对于使用电池供电的设备，在模组进行固件升级时，固件升级过程意外断电将可能会导致模组无法正常工作，因此需要电池剩余足够的电量以完成升级整个过程。

当用户 App 触发升级模组且下载完固件后，App 会向 MCU 询问电池电量是否正常。厂商需要根据硬件设计，确定一个电池电量充足的阈值（通常为大于 50%）来应答模组。

不使用电池供电的设备可以直接应答非低电量状态后开始升级。

模组发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xbc
数据长度	2	0x0000

字段	字节数	说明
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组发送信息示例

```
1 55 AA 00 BC 00 00 BB
```

MCU 返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xbc
数据长度	2	0x0001
数据	1	0x00: 低电量 0x01: 电量正常
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 返回信息示例

```
1 55 AA 00 BC 00 01 01 BD
```

3.29 获取设备 IMEI

MCU 主动下发给模组获取 IMEI 指令，模组将 IMEI（长度 15 的字串）返回给 MCU。本命令可以由 MCU 控制，主动发送 IMEI 获取包，以用于获取设备的 IMEI 码。

MCU 发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xbd
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送信息示例

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

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xbd
数据长度	2	0x000f
数据	15	示例：864237040014733。
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息示例

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

3.30 报告设备运行状态

设备运行状态	描述	状态值
状态 1	设备初始化中	0x00
状态 2	设备识别到 SIM 卡	0x01
状态 3	设备正常运行中	0x02
状态 4	设备被绑定	0x03
状态 5	设备被解绑	0x04
状态 6	设备成功下载 SCHEMA	0x05
状态 7	设备正准备进入睡眠	0x06
状态 8	设备正准备重启	0x07
状态 9	设备正在重置	0x08
状态 10	NB FOTA 下载开始	0x09
状态 11	NB FOTA 下载完成	0x0A
状态 12	NB FOTA 下载失败	0x0B
状态 13	NB FOTA 更新开始	0x0C
状态 14	NB FOTA 更新成功	0x0D
状态 15	NB FOTA 更新失败	0x0E
状态 16	NB schema 下载开始	0x0F
状态 17	NB schema 下载失败	0x10

模组发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xbe
数据长度	2	0x0001
数据	1	指示 NB-IoT 模组运行状态：

字段	字节数	说明
		0x00: 状态 1
		0x01: 状态 2
		0x02: 状态 3
		0x03: 状态 4
		0x04: 状态 5
		0x05: 状态 6
		0x06: 状态 7
		0x07: 状态 8
		0x08: 状态 9
		0x09: 状态 10
		0x0A: 状态 11
		0x0B: 状态 12
		0x0C: 状态 13
		0x0D: 状态 14
		0x0E: 状态 15
		0x0F: 状态 16
		0x10: 状态 17
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组发送信息

设备正常运行中。

```
1 55 aa 00 be 00 01 02 c0
```

3.31 获取设备运行状态

设备运行状态	描述	状态值
状态 1	设备初始化中	0x00
状态 2	设备识别到 SIM 卡	0x01
状态 3	设备正常运行中	0x02
状态 4	设备被绑定	0x03
状态 5	设备被解绑	0x04
状态 6	设备成功下载 Schema	0x05
状态 7	设备正准备进入睡眠	0x06
状态 8	设备正准备重启	0x07
状态 9	设备正在重置	0x08
状态 10	NB FOTA 下载开始	0x09
状态 11	NB FOTA 下载完成	0x0A
状态 12	NB FOTA 下载失败	0x0B
状态 13	NB FOTA 更新开始	0x0C
状态 14	NB FOTA 更新成功	0x0D
状态 15	NB FOTA 更新失败	0x0E
状态 16	NB schema 下载开始	0x0F
状态 17	NB schema 下载失败	0x10

MCU 发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xbf
数据长度	2	0x0000
数据	0	无

字段	字节数	说明
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送信息示例

```
1 55 aa 00 bf 00 00 be
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xbf
数据长度	2	0x0001
数据	1	指示 NB-IoT 模组运行状态：
		0x00：状态 1
		0x01：状态 2
		0x02：状态 3
		0x03：状态 4
		0x04：状态 5
		0x05：状态 6
		0x06：状态 7
		0x07：状态 8
		0x08：状态 9
		0x09：状态 10
		0x0A：状态 11
		0x0B：状态 12
		0x0C：状态 13

字段	字节数	说明
		0x0D: 状态 14
		0x0E: 状态 15
		0x0F: 状态 16
		0x10: 状态 17
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息示例

设备识别到 SIM 卡。

```
1 55 aa 00 bf 00 01 01 c0
```

3.32 设置设备进入睡眠

本命令可以由 MCU 控制主动发送指令包，使模组立刻进入睡眠状态。

MCU 发送

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xc0
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送信息示例

```
1 55 AA 00 CO 00 00 BF
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xc0
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息示例

```
1 55 AA 00 CO 00 00 BF
```

3.33 设置设备进入 PSM

MCU 主动发送，解锁模组并使之允许进入 PSM 状态。

MCU 发送

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xb4
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送信息示例

```
1 55 AA 00 B4 00 00 B3
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xb4
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息示例

```
1 55 AA 00 B4 00 00 B3
```

3.34 设置设备访问的 APN

由于设备可能无法识别部分基站的 APN 接入点，所以可由用户从 MCU 端手动配置设备的 APN 名称。

MCU 发送信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xC2
数据长度	2	无

字段	字节数	说明
数据	取决于具体数据	<code>{"apn":"ctnb","pdp_type":"IP"}</code>
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

数据

参数	说明
apn	设备访问的 APN 名称，根据当前 SIM 接入的运营商而定。如： ctnb：中国电信 cmnbiot：中国移动
pdp_type	分组数据协议的类型，默认使用 IP。 IP：互联网协议（IETF STD 5） IPV6：互联网协议版本 6（IETF RFC 2460） IPV4V6：双栈 Non-IP：将非 IP 数据传输到外部分组网络

MCU 发送信息示例

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

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00

字段	字节数	说明
命令字	1	0xC2
数据长度	2	0x0001
数据	1	0x00: 成功 0x01: 失败
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息示例

```
1 55 AA 00 C2 00 01 00 C2
```

3.35 查询文件下载进度

由 MCU 主动发送，可查询 MCU OTA 下载进度或者 NB FOTA 下载进度等。

MCU 发送

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xC3
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送信息示例

```
1 55 AA 00 C3 00 00 C2
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xC3
数据长度	2	0x0002
数据	2	下载状态与百分比
BYTE[0]	1	
		0x00: 无下载
		0x01: 正在下载
BYTE[1]	1	下载百分比 (0-100)
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息示例

```
1 55 AA 00 C3 00 02 01 05 CA
```

3.36 强制重启设备

由 MCU 主动发送，可立即重启 NB 设备。

MCU 发送

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xC4

字段	字节数	说明
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送信息示例

```
1 55 AA 00 C4 00 00 C3
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xC4
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息示例

```
1 55 AA 00 C4 00 00 C3
```

3.37 查询活动定时器 T3324 时间

由 MCU 主动发送，可查询 NB 卡的 T3324 时间。

MCU 发送

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xC5
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送信息示例

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

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xC5
数据长度	2	0x0004
数据	4	T3324 时间单位为 S
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息示例

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

3.38 查询定时器 T3412 时间

由 MCU 主动发送，可查询定时器 T3412 时间。

MCU 发送

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xC6
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送信息示例

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

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xC6
数据长度	2	0x0004
数据	4	T3412 时间单位为 S
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息示例

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

3.39 查询网络心跳间隔时间

由 MCU 主动发送，可查询 NB 的查询网络心跳间隔时间。

MCU 发送

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xC7
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送信息示例

```
1 55 AA 00 C7 00 00 C6
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xC7
数据长度	2	0x0004
数据	4	LifeTime 心跳时间
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

模组返回信息示例

```
1 55 AA 00 C7 00 04 00 00 70 80 BA
```

3.40 设置开机离散信息

由 MCU 主动发送，可配置 NB-IoT 模组首次启动的离散信息。该功能仅针对首次上电驻网，使设备错峰启动、优化网络阻塞问题，可用于安装比较密集的产品中。

请酌情使用该功能。若设备采用主动断电的场景，禁用该功能。

MCU 发送

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xCb
数据长度	2	N
数据	N	<pre>{"enable": 1, "duration": 360, "step": 1}</pre> <code>enable</code>: 1: 开启该功能, 0: 关闭该功能。 <code>duration</code>: 离散的随机数种子范围 (120-1800), 单位为 s。 <code>step</code>: 1: <code>step</code> 为 60s, 0: <code>step</code> 为 30s。
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余。

MCU 发送信息示例

```
1 55 AA 00 CB 00 27 7B 22 65 6E 61 62 6C 65 22 3A 20 31 2C 22 64 75 72
2 61 74 69 6F 6E 22 3A 20 33 36 30 2C 22 73 74 65 70 22 3A 20 31 7d 9
3 F
```

模组返回信息

字段	字节数	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0xCb
数据长度	2	0x0001
数据	1	0x00：设置成功 0x01：设置失败
校验和	1	从帧头开始按字节求和，得出的结果对 256 求余

模组返回信息示例

```
1 55 AA 00 CB 00 01 00 CB
```

4 协议版本

版本	日期	说明
0.6.19	2023.03.23	增加联网状态 5。通讯协议版本号升级至 0x01，支持用户记录及非记录型数据上报消息 ID。
0.6.18	2022.10.18	增加 MCU 配置 RTC 时间，增加离散功能。
0.6.17	2022.04.20	增加 FOTA 与 schema 下载状态与下载进度查询、T3324 查询、T3412 查询、LifeTime 查询。
0.6.16	2022.01.25	增加文件百分比、查询内存、PSM、T3412 等命令协议。
0.6.15	2022.01.21	更新 MCU 文件下载服务，支持自定义请求下载文件。
0.6.14	2021.11.13	更新 MCU 升级服务，MCU 判断 CRC32 结果，兼容旧协议。
0.6.13	2021.11.02	增加 MCU 升级服务，支持 MCU OTA 断点升级。
0.6.12	2021.04.01	增加波特率 9600/115200 自适应说明。
0.6.11	2020.11.30	增加记录型数据使用 MCU 时间戳。
0.6.10	2020.11.19	新增设置设备 APN 命令。
0.6.9	2020.11.16	更新设置心跳时间间隔的返回状态。
0.6.8	2020.08.17	更新 MCU 获取查询模组 NB-IoT 网络强度中数据呈现形式。

版本	日期	说明
0.6.7	2020.08.13	查询产品信息中，新增功耗模式及云连接方式返回。
0.6.6	2020.07.27	新增记录型数据唤醒时间功能。
0.6.5	2020.05.14	获取/发送设备运行状态信息命令中，新增 SIM 卡识别成功，设备绑定，解绑，Schema 下载完成等状态。
0.6.4	2020.04.28	新增获取/发送设备运行状态信息命令。新增发送强制设备进入睡眠命令。
0.6.3	2020.04.16	新增发送设备 IMEI 信息命令。
0.6.2	2019.11.09	新增发送心跳包命令。
0.6.1	2019.09.05	更新记录型上报协议中的时间字段规则。
0.6.0	2019.06.24	更新 Wi-Fi 相关信息。
0.5.2	2019.03.20	更新睡眠锁说明。
0.5.1	2019.03.08	继承 Wi-Fi 通用门锁。