Q/ENN

新奥能源控股有限公司企业标准

Q/ENN XB XXX-2021

物联网平台设备接入标准-工商业燃气报警器

Access standards for internet of things platform equipment- Industrial Gas alarm equipment

2021—X—XX 发布　　2021—X—XX 实施

新奥能源控股有限公司　　发布

前　言

根据新奥能源物联网平台燃气设备的接入要求，认真总结实践经验，参考国家行业标准，并在广泛征求意见的基础上，制定本标准。

本标准的主要内容是：1 总则；2 术语；3 协议特性；4 数据链路层；5 数据传输层；6 数据安全性；7 业务流程；8 协议解析。

本标准由新奥能源技术与创新产品中心负责管理，由新奥能源物联网项目组负责具体条文内容的解释。

本标准在执行过程中，希望使用方结合实践，注意总结经验，积累资料，如发现本标准需要修改和补充，请将意见和有关资料反馈至新奥能源物联网项目组（地址：河北省廊坊市开发区新源东道新奥科技园 A 楼，邮政编码：065001），以便今后修订时参考。

本标准主编单位、主要起草人和主要审查人：

主编单位：　新奥能源控股有限公司

主要起草人：

主要审查人：

目　录

1 总 则

1.0.1 本标准适用于新奥能源及合作企业接入新奥能源物联网平台新增的燃气设备。

1.0.2 本标准制定的燃气设备的接入标准，具体描述了新奥能源物联网平台和接入设备的通讯接口约定，同时给出涉及到的关键技术流程。

1.0.3 本标准将帮助相关开发人员、业务人员了解其内容，指导并规范各自后续的详细设计及测试等工作。

2 术 语

2.0.1 窄带物联网（NB-IoT）Narrow Band Internet of Things

基于 3GPP 演进的通用陆地无线接入(E-UTRA）技术，使用 180kHz 的载波传输带宽，支持低功耗设备在广域网的蜂窝数据连接。

2.0.2 集成电路卡识别码（ICCID）Integrate circuit card identity

ICCID 为 IC 卡的唯一识别号码，由 20 位数字和字母组成。

2.0.3 加密 encipherment/encryption

对数据进行密码变换以产生密文的过程。

2.0.4 解密 decipherment/decryption

加密过程对应的逆过程。

2.0.5 密钥 key

控制密码变换中，为增加安全性或使密码设备同步而引入的用于数据变换的起始数据。

2.0.6 短连接 short connnection

短连接是相对于长连接而言的概念，指的是在数据传送过程中，只在需要发送数据时，才去建立一个连接，数据发送完成后，则断开此连接，即每次连接只完成一项业务的发送。

2.0.7 长连接 long connnection

长连接，指在一个连接上可以连续发送多个数据包，在连接保持期间，如果没有数据包发送，需要双方发链路检测包。

2.0.8 HEX 码 HEX

十六进制数据。

3 协议特性

3.1 可扩展性

3.1.1“命令码”为1字节(8bit)HEX码,从1~254,最大可支持254条命令码(即 0x01~0xFE,00000001~11111110。0和255不使用）。

3.1.2 协议支持一个远传终端下同时进行多个设备终端的数据传输能力。

3.1.3 协议中定义多个设备终端按照不同的数据块进行数据传输，一次上报数据的报文中包含多个设备的数据信息。

3.1.4 数据块包括表示段和数据段，一次连接即可同时满足多台设备的数据信息传输工作。

3.2 兼容性

3.2.1“协议版本”信息协议制定后，可能会修正、变更，继而推出后续版本，会导致多个不同版本的协议存在，不同的客户端可以实现的不同版本的协议。服务端同时支持客户端多个版本的协议，能够正确解析不同版本协议里面的数据内容。不同版本协议定义了不同的数据封包格式，服务端按照不同的版本使用不同的方法来解析数据。客户端上报的数据包里面应该有标明是属于相应版本协议的字段。

当协议变更时，可通过不同的版本号来判别，具有协议兼容性。版本号的格式为X.Y.Z(又称 Major.Minor.Patch)，递增的规则为：

X 表示主版本号，当协议兼容性变化时，X 需递增。从1开始，加1递增；

Y 表示次版本号,当在协议中增加功能、或者微调时(不影响协议的兼容性),Y需递增；

Z 表示修订号，当作 Bug 修复时(不影响协议的兼容性)，Z 需递增。

版本号占用1字节（注意保证字节对齐），分配如表3.2.1

表3.2.1 版本号字节分配表

0	1	2	3	4	5	6	7
X		Y				Z	

注：X,Y,Z的取值范围分别为：X：1～3　Y：0～15　Z：0～3 版本号

例：1.0.0，1.0.2，1.1.2，1.10.0，2.1.1

版本号	1.0.0	1.0.2	1.1.2	1.10.0	2.1.1
对应值	01000000	01000010	01000110	01101000	10000101
HEX码	0x40	0x42	0x46	0x68	0x85

关于版本号描述：

1）X, Y, Z 必须为非负整数（0，1~n），且不得包含前导零，必须按数值递增，如 1.1.0 -> 1.1.1 -> 1.2.0；

2）版本号的排序规则为依次比较主版本号、次版本号和修订号的数值，如 1.0.0 < 1.0.1 < 1.1.1 < 2.0.0；

3）版本一经发布，不得修改其内容，任何修改必须在新版本发布。

3.2.2 “设备类型”信息，通过不同数值代表不同的设备类型，可兼容后续开发的其它类型的设备。协议中包含标明“设备类型”的字段，根据字段值区分不同的设备类型，同一版本的协议可以解析多种设备上报的数据，而无须为不同的设备制定多种不同的协议。

4　数据链路层

4.1　无线传输

适用以下传输技术：　GPRS、　NB-IoT、　4G、　5G。

4.2　网络传输

4.2.1 采用 NB-IoT 通讯技术的终端，如图 4.2.2，应满足：

1）采用 CoAP、LwM2M 协议，接入运营商平台后，再接入新奥能源物联网平台。

2）采用 TCP 协议，直接与新奥能源物联网平台连接。

4.2.2 采用 GPRS、4G、5G 通讯技术的终端，如图 4.2.2，应满足：

采用 TCP 协议，直接与新奥能源物联网平台连接。

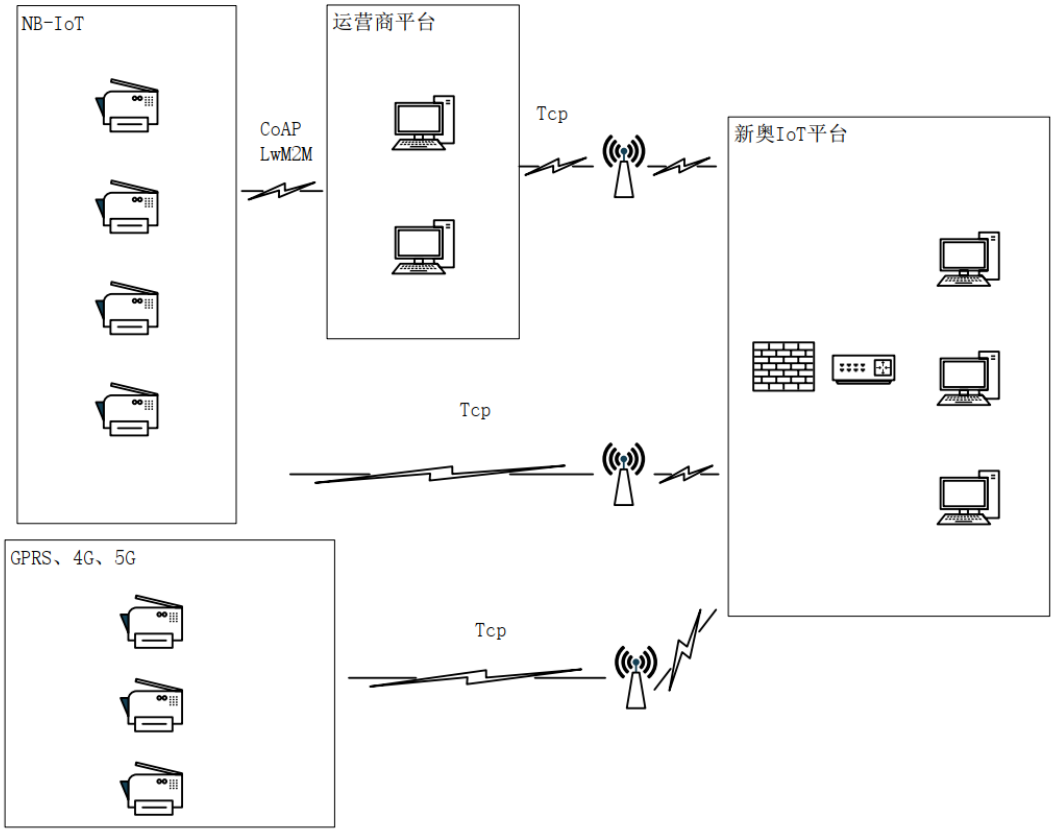


图 4.2.2 网络传输示意图

4.3 通讯响应要求

在通讯过程中可能会因为某种原因访问堵塞，导致终端和系统间通讯故障，为了节省终端电量，避免服务端资源浪费，网络消耗，客户端应该在一定时间（30S）后，结束单次通讯。服务端也同样会在一定时间后，没有接收到终端的响应后断开此次连接。

5　数据传输层

5.1　通讯方式

通讯方式采用半双工。为统一和服务于网络传输和业务服务，统一使用小端模式。

5.2　数据帧结构

5.2.1 主发包格式是指系统向终端发送的数据包格式，详见表 5.2.1。

表 5.2.1 主发包格式表

主发	系统>>>终端									
序号	1	2	3	4-11	12	13	14-15	16… 16*(N+1)-1	16*(N+1)-16*(N+1)+1	16*(N+1)+2
名称	包头	协议版本	设备类型	设备编号	秘钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0~DL-1	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	16*N	2	1
数值	83H				0-9H		16*N			EDH

注：指令说明：

Head：

包头，1 字节。系统到终端的下发指令，其包头为固定值 83H，HEX 码。

Ver：

协议版本号，从 40H 开始，HEX 码，1 字节。

Type：

设备类型，HEX 码，1 字节，由01H 开始编号。新增终端类型编码由新奥能源物联网平台统一制定。

15H：工商业气体报警器。

A0-A7：

设备编号,规则说明如下：

1）16位编号，不足16位长度，前补0。

2）第1位至第4位为厂家编码，第5位至第6位为设备类型ID，由新奥能源物联网平台统一备案；

3）第7位至第8位为设备型号ID， 1X代表LoRaWAN通讯设备，20为标准协议直连通讯方式CH4探测类型设备，21为标准协议非直连通讯方式CH4探测类型设备；

4）第9位至第16位为硬件序列号，由各厂家自行定义。

注：1X中X代表0-9，此协议文档为标准协议。

第9位至第16位为硬件序列号，由各厂家自行定义。

Key：

密钥代码，密钥号0-9，HEX码，1字节。每个非零密钥号代表16 字节密钥。0号密钥：代表无密钥，数据明文传送。

1号密钥：代表测试密钥，即远传终端在生产、检测中及在下发密钥前使用的密钥。

2-9 号密钥：代表运营密钥，即远传终端出厂后，抄表管理系统下发给终端的密钥，用于远传终端在运营中使用的密钥。

说明：终端已下发运营密钥后，对于非0密钥协议，不能使用1号测试密钥，只能使用2-9号运营密钥。

CMD：

命令码，用于识别命令内容。如 09H 为表“读 ICCID”命令码。HEX 码，1 字节。

L：

数据域长度，数据域长度为16×N（N 为自然数，包括0），2 个字节，HEX 码，低字节在前，高字节在后。

例：L=10H 00H，则代表数据域的长度为16 字节。

D0....DL-1：

数据域内容。数据域内容由“数据信息+补位”组成。其中“数据信息”为具体数据/指令内容；“补位”统一为00H。

例如，“数据信息”为26字节，则“数据域长度”需定为大于26的16的最小整倍数32字节（20H），那么26字节数据比32字节数据长度少6字节，则“补位”为6字节，需在数据域内补齐“00H00H00H00H00H00H”到32 字节。

CRC：

校验码，校验方式为CRC16，从协议版本字节开始到校验字节之前的CRC16校验。低字

节在前，高字节在后。HEX 码，2字节。CRC 校验方式详见附录A。

End：

包尾，系统到终端的下发指令，包尾固定值为 EDH，HEX 码，1 字节。

5.2.2 从发包格式定义了系统向终端发送的数据包格式，详见下表 5.2.2

表 5.2.2 从发包格式表

从发	终端>>>系统									
序号	1	2	3	4-11	12	13	14-15	16… 16*(N+1)-1	16*(N+1)-16*(N+1)+1	16*(N+1)+2
名称	包头	协议版本	设备类型	设备编号	秘钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0~DL-1	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	16*N	2	1
数值	67H				0-9H		16*N			EDH

注：指令说明：

Head：

包头，1 字节。系统到终端的下发指令，其包头为固定值 67H，HEX 码。

其他指令说明同主发包。

6　数据安全性

6.1　数据加密

数据加密采用 AES 加密方式。具体说明如下：

1 远传终端出厂时，密钥为 1，即测试密钥，测试密钥为系统给定。

2 远传终端在系统内开户后，如远传终端以测试密钥进行上报，则系统给远传终端下发 8 组运营密钥，每组密钥 16 字节。

3 远传终端存储 8 组运营密钥后，当需要上传数据时，对于需要加密的数据，在 8 组密钥内随机选择一组密钥，对数据域进行加密。

4 加密运算时，数据域内的数据分成 16 个字节为单位的数据块，最后的数据块可能为 1～16 个字节。当最后的数据块长度是 16 个字节时，正常对每 16 字节数据进行加密；当最后的数据块长度不足 16 个字节时，则在其后每个字节加上 00H，直到长度达到 16 个字节。

5 对数据域内的每一个 16 字节数据块按照设定密钥分别进行加密，所有加密后的数据块按照原有顺序连接在一起，加密后的密文字节数亦为 16 的整数倍。

6 对于“0”密钥，如数据域有数据，则按 16 的倍数关系发送数据，不足 16 字节，则在其后上 00H，补齐 16 字节，所有数据域内容不加密；如数据域为空，则数据长度为 0。

7 数据采用先 AES 加密，再 CRC 校验的方式。

8 加密算法详见附录 A。

6.2　数据解密

当远传终端接收到系统采用加密方式下发的数据包时，远传终端读取密钥号，根据密钥号提取表中对应的密钥进行解密，并执行解密后的命令。数据采用先 CRC 校验，后 AES 解密的方式，解密算法详见附录 A。

6.3　密钥管理

密钥由抄表管理系统生成 8 组随机密钥，每组密钥 16 字节。

7 业务流程

7.1 主业务流程

7.1.1 NB-IoT 短连接通讯流程图

经过运营商平台连接新奥能源物联网平台的短连接 NB-IoT 通讯流程见下图 7.1.1

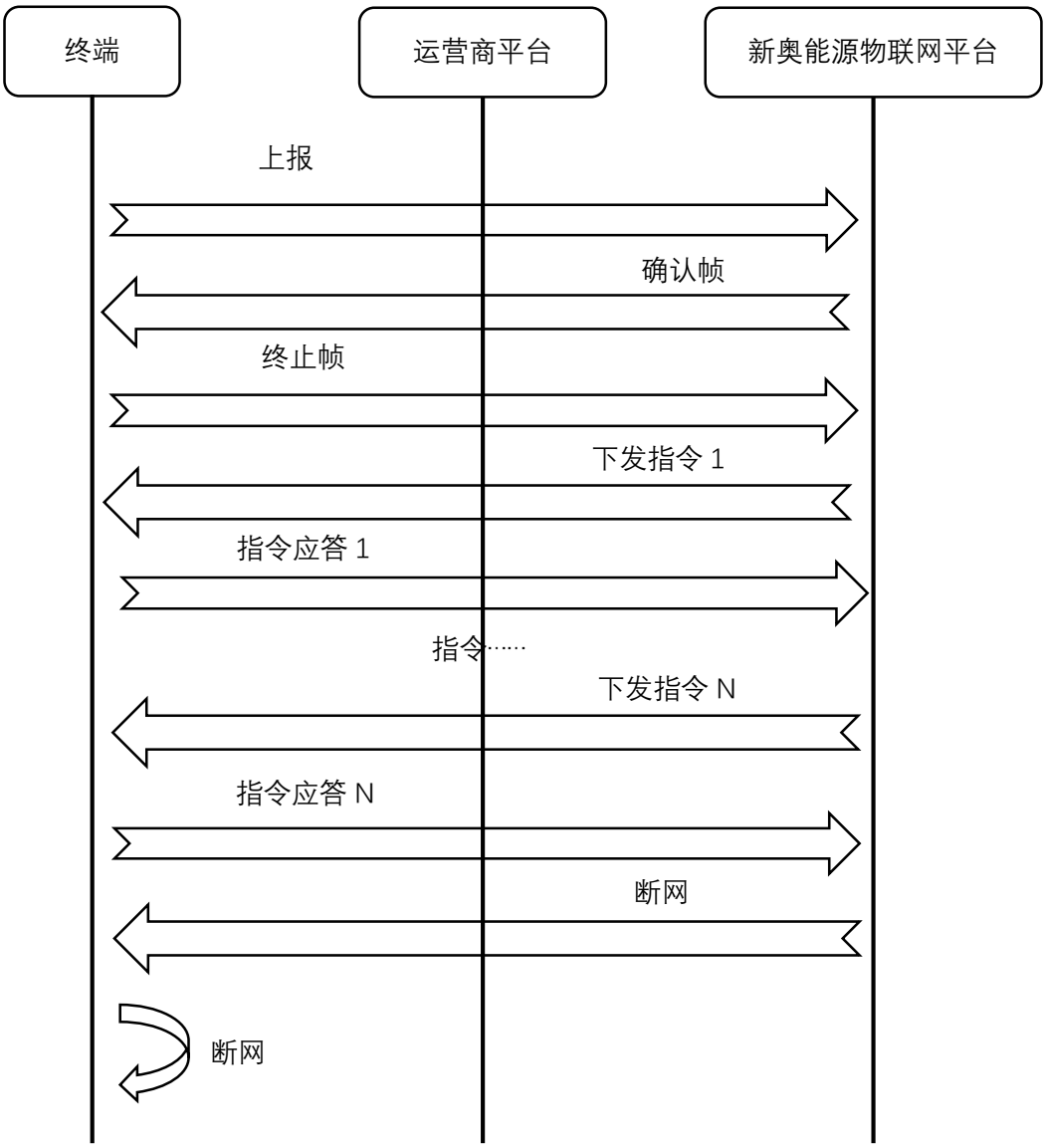


图 7.1.1 NB-IoT 短连接通讯流程图

7.1.2　GPRS/4G/5G 及直连 NB-IoT 短连接通讯流程图

GPRS/4G/5G 通讯流程和不经过运营商平台连接新奥能源物联网平台的短连接 NB-IoT 通讯流程见下图 7.1.2。

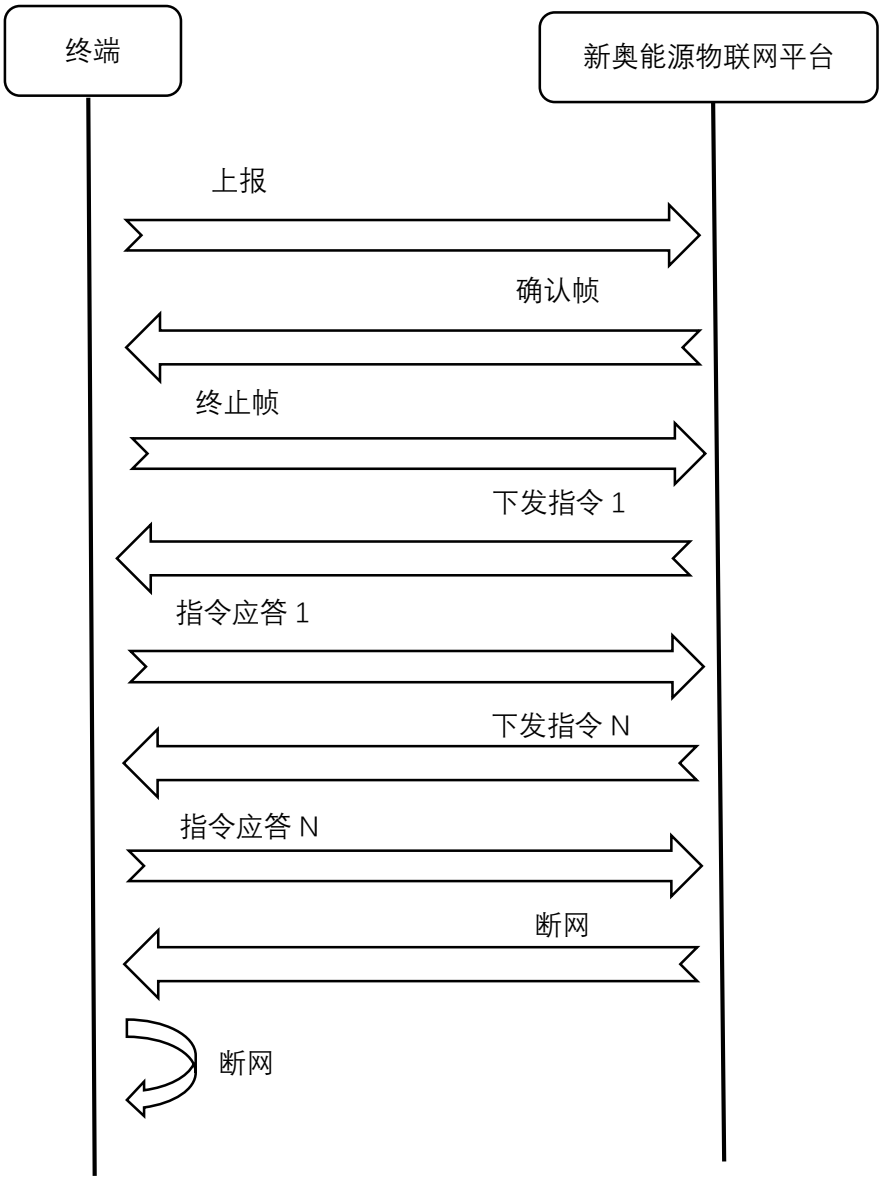


图 7.1.2　GPRS/4G/5G 及直连 NB-IoT 短连接通讯流程图

7.1.3 NB-IoT 长连接通讯流程图

经过运营商平台连接新奥能源物联网平台的长连接 NB-IoT 通讯流程见下图 7.1.3。

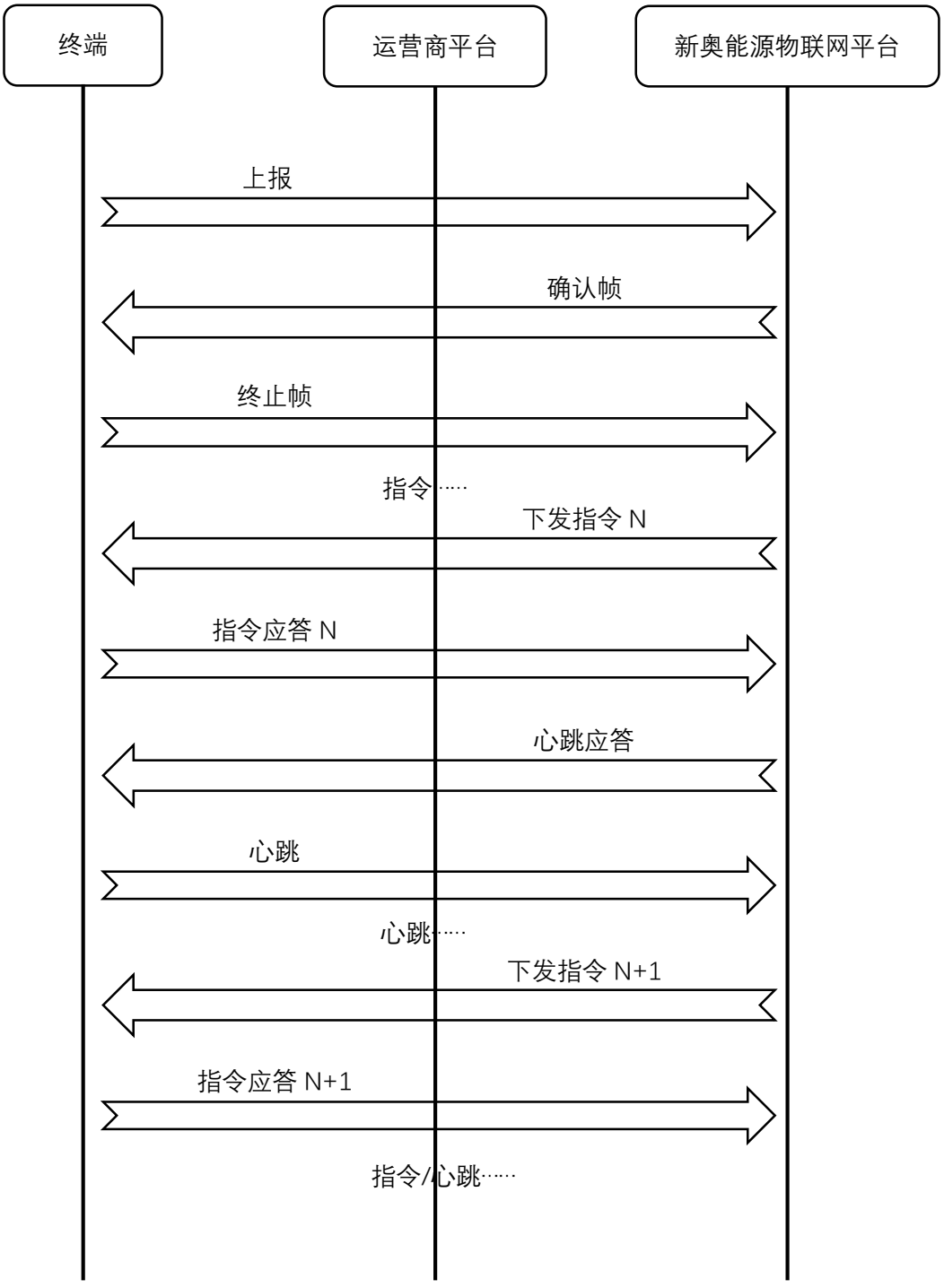


图 7.1.3 NB-IoT 长连接通讯流程图

7.1.4 GPRS/4G/5G 及直连 NB-IoT 长连接通讯流程图

GPRS/4G/5G 通讯流程和不经过运营商平台连接新奥能源物联网平台的长连接 NB-IoT 通讯流程见下图 7.1.4。

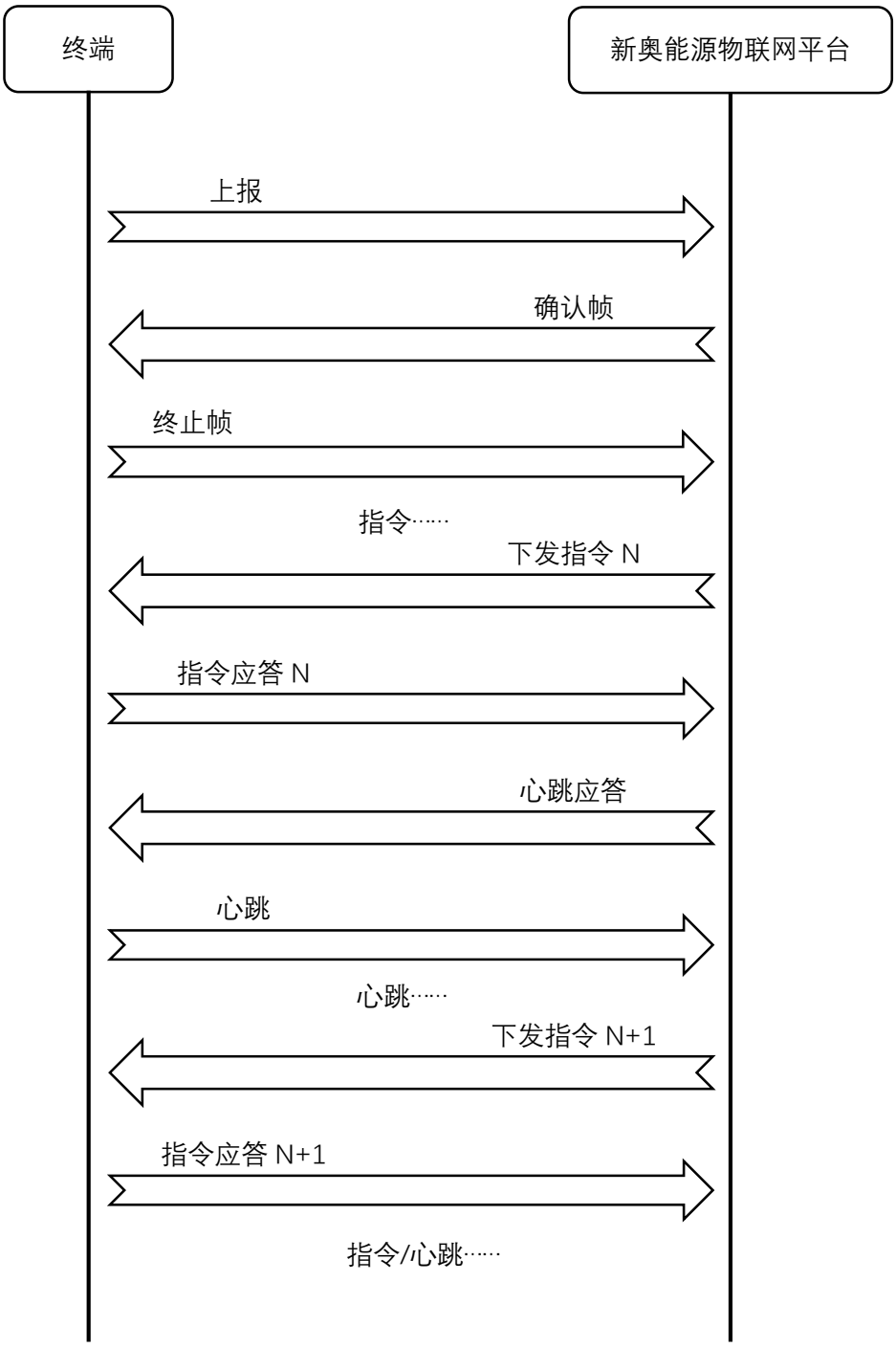


图 7.1.4 GPRS/4G/5G 及直连 NB-IoT 长连接通讯流程图

7.2 通讯流程说明

通讯流程分为短期在线和长期在线两种模式。

7.2.1 短期在线模式

“短期在线模式”为终端主动连接系统，与系统进行信息交互。信息交互完成后，系统断开连接。

（1）远传终端上电或满足上报条件（定时上报、按键触发上报、上电上报、报警上报或重报等），终端发起与系统的连接。

（2）远传终端上报终端信息，系统接收后解析数据。解析完成后系统给终端进行指令下发操作。

（3）系统对终端的操作完成，发送断网命令（附带时钟校验数据），断开与终端的连接。终端接收断网指令及时钟校时信息后，断网并更新远传终端内时钟，不应答。

（4）在任一通讯过程中，不论系统或终端，在收不到对方应答的时长超设定时长时，系统或终端会主动断开连接。由终端主动断开的连接，则认为上报失败，需根据终端重发设置进行重报操作。

7.2.2 长期在线模式

“长期在线模式”为终端与系统连接后，连接状态长期保持，终端和系统端随时可以向对方发送信息。长连接通过终端向系统端发送心跳包以保持连接状态。

（1） 远传终端上电或满足上报条件（定时上报、按键触发上报、上电上报、异常上报、重报或用气上报等），终端发起与系统的连接。

（2） 远传终端上报终端信息，系统接收后解析数据。解析完成后系统给终端进行指令下发操作。

（3） 若无指令可下发，系统向终端下发心跳应答以保持两者之间的通讯连接。

（4） 终端长期在线期间，如系统和终端间无数据交互，则终端按照设定间隔，向系统发送心跳包，以保持连接状态；系统接收到心跳包后进行应答。

8 协议解析

8.1 业务指令

8.1.1 状态信息上报（02H）

终端上电、掉电上报，若上报失败则在上报条件恢复后重新进行上报。系统收到终端上报数据后进行解析，解析完成后，回复确认收到帧。上报帧格式，详见表 8.1.1-1

表 8.1.1-1 状态信息上报格式

从发	终端>>>系统									
序号	1	2	3	4-11	12	13	14-15	16… 16*(N+1)-1	16*(N+1)- 16*(N+1)+1	16*(N+1)+2
名称	包头	协议版本	设备类型	设备编号	秘钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0~DL-1	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	16*N	2	1
数值	67H				0-9H	02H	20H 00H			EDH

指令说明：

终端将实时状态信息上报。包含采集时间、电池电压、信号强度、ICCID、电池状态等信息。

D0~DL-1：

同时上传设备所有报警信息，每个报警信息占一位，终端支持报警内容配置，各数据位代表报警信息与新奥能源物联网平台指令模型内容一致。

上报报警数据域示例说明详见下表 8.1.1-2。

表 8.1.1-2 状态信息上报数据域

数据域	D0~DL-1				
符号	D0	D1~D6	D7	D8~D27	D28~D31
名称	上报类型	设备状态			
		时间	供电状态	ICCID 号	补位
类型	HEX	HEX	HEX	ASCII	HEX
字节	1	6	1	20	4
数值					00H00H00H00H.

D0：

上报的类型。不同的字节数值代表不同的上报类型。1 字节，HEX 码。

01H：上电上报：终端上电后，自动发起连接上报；

02H：断电上报：终端断电后，主动发起上报。

D1~D6：

时间为采集：6 字节（年月日时分秒）。BCD 码。

例：20160704152250，年份 2016 只使用后两位 16。

D1：16H-代表 16 年；

D2：07H-代表 7 月；

D3：04H-代表 4 日；

D4：15H-代表 15 时；

D5：22H-代表 22 分；

D6：50H-代表 50 秒；

D7：

供电状态：

01H：正常供电；

02H：断电。

D8~D27：

20 个字符的 ICCID 号，用 ASCII 码表示。

例：ICCID 字符串"00000000000000000001"

D13-D32：30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 31H。

D28~D31：

数据补位：00H00H00H00H。

8.1.2 采集数据上报（22H）

终端在运行时根据设置频率主动上报采集数据，超过限制的数据，进行分包上传，上传完成后，终端上传终止帧通知系统所有包数据上传完毕。系统收终端历史数据后进行解析，解析完成后，回复确认收到帧。所有数据传输完成后，终端发送终止帧，系统收到终止帧后，下发配置指令，若无配置指令，立即下发断网指令，长连接模式下，不下发断网指令。重试三次不成功，设备休眠，休眠时间厂家自行定义。帧格式详见表 8.1.2-1。

表 8.1.2-1 采集数据上报格式

从发	终端>>>系统									
序号	1	2	3	4-11	12	13	14-15	16… 16*(N+1)-1	16*(N+1)-16*(N+1)+1	16*(N+1)+2
名称	包头	协议版本	设备类型	设备编号	秘钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0~DL-1	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	16*N	2	1
数值	67H				0-9H	22H	16*N			EDH

指令说明：

终端将历史采集数据信息进行上报。

D0~DL-1：

燃气设备终端信息上报数据域示例说明详见下表 8.1.2-2。

表 8.1.2-2 上报数据域

数据域	D0~DL-1															
符号	D0	D1	D2	D3~DL-1												
名称	本包条数	上报类型	数据类型	数据 1											…	数据 N
				时间	电池电压	信号强度	探测器 1 数据		…	探测器 n 状态		输出模块 1 状态	…	输出模块 n 状态	…	
							浓度	状态		浓度	状态					
类型	HEX	HEX	HEX	HEX	HEX	HEX	HEX	HEX	…	HEX	HEX	HEX		HEX	…	
字节	1	1	1	6	2	1	2	1	…	2	1	1		1	…	
数值																

D0:

当前本包数据条数。1 字节，HEX 码。

D1：

上报的类型。不同的字节数值代表不同的上报类型。1 字节，HEX 码。

03H：定时上报：终端内设定的上报时间到达时，终端主动发起上报；

04H：触发上报：对终端进行现场触发，实现终端发起上报；

05H：重报：当终端发起的上报失败时，终端根据重报次数和间隔设置，进行重报；

06H：间隔上报：终端内设定的间隔上报时间到达时，终端主动发起上报；

07H: 报警上报：当终端有异常发生时，终端进行异常上报；

08H：报警恢复：异常报警解除后，终端报警状态回复，进行恢复上报。

D2：

数据类型，不同的字节数值代表不同的数据种类，1 字节，HEX 码

01H：甲烷 CH4 传感器；

02H：一氧化碳 CO 传感器；

03H：丙烷 C3H8 传感器；

04H：氢气 H2 传感器；

05H：温度传感器；

06H：湿度传感器；

07H：PM2.5 传感器；

08H：预留；

D3~D8：

时间：6 字节（年月日时分秒）。BCD 码。

例：20160704152250，年份 2016 只使用后两位 16。

D1：16H-代表 16 年；

D2：07H-代表 7 月；

D3：04H-代表 4 日；

D4：15H-代表 15 时；

D5：22H-代表 22 分；

D6：50H-代表 50 秒；

D9~D10：

电池电压，数据不体现小数，但数值默认缩小 100 倍，即具有 2 位小数。

例：电池电压 3.65V， 数据为"6D01H"

D8D9：6D01H。

市电供电方式使用 FF 填充。

D11：

信号强度 RSSI，数据不体现负数，不设置用 FF 填充。

物联网模块返回信号等级 0-31 对应信号强度为-113dBm~-53dBm;

信号强度=-113 dBm+(信号强度等级×2)

例：信号强度-85，数据为"55H"

D11：55H。

一条数据包含固定的 10 路探测器状态和浓度以及 10 路输出模块状态。

探测器数据 3 字节，其中浓度数据 2 字节，数据不体现小数，但数值默认缩小 100 倍，即具有 2 位小数；

传感器状态 1 字节，状态示例见详见下表 8.1.2-3。

表 8.1.2-3 传感器状态

序号	代码	状态类型
0	0x00	探头短路
1	0x01	探头断开
2	0x02	探头老化
3	0x03	其它故障
4	0x04	未标定
5	0x05	零点变化
6	0x06	检测周期过期
7	0x07	无响应
8	0x08	使用寿命过期
9	0x09	探头自检
10	0x0a	正常
11	0x0D	低限报警
12	0x0E	高限报警
13	0xFF	空(无探测器)

输出模块状态 1 字节，状态示例见详见下表 8.1.2-4。

表 8.1.2-4 输出模块状态

序号	代码	状态类型
0	0x00	关闭输出
1	0x01	开启输出
2	0x02	输出故障
3	0xFF	空（无输出）

注：数据域长度为 16 倍数，不足时候通过 00H 补足。

8.1.3 终止帧（06H）

表 8.1.3-1 终止帧帧格式

从发	终端>>>系统									
序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	秘钥号	命令码	长度	数据域	校验	包尾

符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1
数值	67H				0-9H	06H	10H 00H			EDH

指令说明：

多包数据进行上传时，终端将数据全部上报完成后，向系统发送终止帧，帧格式如表8.1.3-1。

表8.1.3-2 终止帧数据域

数据域	D0-D15	
符号	D0	D1-D15
名称	终止	补位
类型	HEX	HEX
字节	1	15
数值	01H	00H00H…00H

D0：

固定值：01H。

D1-D15:

固定值：00H00H…00H（16字节补位）

8.1.4 确认帧（07H）

表 8.1.4-1 确认帧帧格式

主发	系统>>>终端									
序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	秘钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1

数值	83H				0-9H	07H	10H 00H			EDH

指令说明：

上报历史数据和历史数据补调时，每一包发完后需要等系统回复确认帧才能上传下一帧。若终端没有收到系统发送的确认帧命令，该包数据需重新进行上传。

表 8.1.4-2 确认帧数据域

数据域	D0	D1-D15
符号	D0	D1-D15
名称	确认	补位
类型	HEX	HEX
字节	1	15
数值	01H	00H00H…00H

D0：

固定值：01H。

D1-D15:

固定值：00H00H…00H（16 字节补位）

8.1.5 历史数据补调（23H）

系统可以主动向终端要求补发未接收到的数据，超过限制的数据，进行分包上传，上传完成后，终端上传终止帧通知系统所有包数据上传完毕。对于短连接模式，系统收到终端历史数据后进行解析，解析完成后，回复确认收到帧。收到终止帧后，系统下发配置指令，若无配置指令，立即下发断网指令，长连接模式下，不下发断网指令。

表 8.1.5-1 设备历史数据补调命令

主发	系统>>>终端									
序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	密钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0-D15	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX		HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1
数值	83H					23H	10H00H			EDH

指令说明：

读取终端设备历史数据信息。

表 8.1.5-2 设备历史数据补调命令

数据域	D0~D15			
符号	D0~D5	D6-D11	D12	D13-D15
名称	补调数据的起始时间	补调数据的截止时间	数据类型	补位
类型	BCD	BCD	HEX	HEX
字节	6	6	1	3
数值				00H00H00H

D0~D5：

需要终端上传的数据起始时间。

时间：6 字节（年月日时分秒）。BCD 码。

例：20160704152250，年份 2016 只使用后两位 16。

D0：16H-代表 16 年；

D1：07H-代表 7 月；

D2：04H-代表 4 日；

D3：15H-代表 15 时；

D4：22H-代表 22 分；

D5：50H-代表 50 秒；

D6~D11：

需要终端上传的数据截止时间。

时间：6 字节（年月日时分秒）。BCD 码。

例：20160704152250，年份 2016 只使用后两位 16。

D6：16H-代表 16 年；

D7：07H-代表 7 月；

D8：04H-代表 4 日；

D9：15H-代表 15 时；

D10：22H-代表 22 分；

D11：50H-代表 50 秒；

D12：

数据类型，不同的字节数值代表不同的数据种类，1 字节，HEX 码

01H：甲烷 CH4 传感器；

02H：一氧化碳 CO 传感器；

03H：丙烷 C3H8 传感器；

04H：氢气 H2 传感器；

05H：温度传感器；

06H：湿度传感器；

07H：PM2.5 传感器；

08H：预留；

D13~D15：

固定值：00H00H00H

终端根据系统下发的补调时间，将数据上，数据上报格式详见表 8.1.5-3。

表 8.1.5-3 历史数据补调应答帧格式

从发	**终端>>>系统**									
序号	1	2	3	4-11	12	13	14-15	16… 16*(N+1)-1	16*(N+1)-16*(N+1)+1	16*(N+1)+2
名称	包头	协议版本	设备类型	设备编号	秘钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0~DL-1	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	16*N	2	1
数值	67H				1-9H	13H	16*N			EDH

指令说明：

终端将需要补调的历史数据信息进行上报。

D0~DL-1：

数据域内容要求同8.1.2 采集数据上报指令（22H）数据域内容。

8.1.6 密钥下发（08H）

系统和终端的通讯通过 8 组随机密钥进行加解密。

表 8.1.6-1 密钥下发帧格式

主发	系统>>>终端									
序号	1	2	3	4-11	12	13	14-15	16-143	144-145	146
名称	包头	协议版本	设备类型	设备编号	密钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0....D127	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX		HEX	HEX
字节	1	1	1	8	1	1	2	128	2	1
数值	83H				1-9H	08H	80H00H			EDH

指令说明：

系统给终端下发的 8 组密钥，每组密钥 16 字节，共 128 字节，密钥号为 2-9 号。

D0~D127：密钥下发命令数据域说明详见下表。

表 8.1.6-2 秘钥下发数据域

数据域	D0~D127			
符号	D0~D15	D16~D31	……	D112~D127
名称	2 号密钥	3 号密钥	……	9 号密钥
类型	HEX	HEX	……	HEX
字节	16	16	……	16
数值	02H	03H	……	09H

D0~D15：2 号密钥；

D16~D31：3 号密钥；

…… ……

D112~D127：9 号密钥。

表 8.1.6-3 密钥下发应答帧格式

从发	终端>>>系统									
序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	密钥号	命令码	长度	数据域	校验	包尾

符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0-D15	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX		HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1
数值	67H				1-9H	08H	10H00H			EDH

指令说明：

远传终端应答“密钥下发”命令。

表 8.1.6-4 密钥下发应答数据域

数据域	D0	D1-D15
符号	D0	D1-D15
名称	确认	补位
类型	HEX	HEX
字节	1	15
数值	01H	00H00H…00H

D0：

应答标识，数值为 00H：接收密钥成功；01H：接收密钥失败。HEX 码。

D1-D15：

固定值：00H00H...00H（16 字节补位）

8.1.7 断网（04H）

短连接模式下，若系统内无配置命令给终端下发，立即下发断网命令，终端收到该命令后，进入休眠状态。

表 8.1.7-1 断网帧格式

主发	系统>>>终端									
序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	密钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0-D15	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX		HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1
数值	83H				1-9H	04H	10H00H			EDH

指令说明：

系统给终端下达断网命令，并附带下发时钟信息，远传终端与系统的连接被断开。远传终端根据收到的时钟信息校对表内时钟，此命令不需终端的应答。

此指令对长期在线模式不适用。

D0~D6：断网命令数据域说明详见下表 8.1.7-2。

表 8.1.7-2 断网数据域

数据域	**D0~D15**						
符号	D0-D6						D7-D15
	D0	D1	D2	D3	D4	D5	D6-D15
名称	系统当前时钟						补位
	年	月	日	时	分	秒	
类型	BCD						HEX
字节	6						10
数值							00H00H…00H

D0~D5：

用于校对终端的系统当前时钟。年月日时分秒，年在高位，秒在低位，6 字节 BCD 码。

例：20210604164250

D0：21H-代表16年；

D1：06H-代表7月；

D2：04H-代表4日；

D3：16H-代表15时；

D4：42H-代表22分；

D5：50H-代表 50 秒。

D6~D15:

固定值：00H00H…00H

8.1.8 心跳包（CCH）

长期在线模式下，若无命令交互，终端通过心跳包与系统保持连接。

表 8.1.8-1　心跳包帧格式

从发	终端>>>系统		
序号	1	2	3
名称	包头	命令码	包尾
符号	Head	CMD	End
类型	HEX	HEX	HEX
字节	1	1	1
数值	67H	CCH	EDH

指令说明：

“长期在线模式”下，终端与系统间的通讯信息交互完成后，终端发送心跳包给系统，以保持与服务器的连接状态，以及让长时间空闲(很长时间内不向服务器发送数据)的终端检测连接状态是否有效。当连接异常时，终端无法收到系统的心跳包应答，发送失败次数大于 3 次时，模块认为连接异常，将尝试重新接入服务器。心跳包是在网络没有数据的时候才会发送，如果数据交互小于心跳时间，则不会发送心跳包。心跳包发送间隔为 0s-255s（一般 10s-180s），根据网络状况确定具体时间。此指令无数据域，不需加密。

表 8.1.8-2 心跳包应答帧格式

主发	系统>>>终端		
序号	1	2	3
名称	包头	命令码	包尾
符号	Head	CMD	End
类型	HEX	HEX	HEX
字节	1	1	1
数值	83H	CCH	EDH

指令说明：

“长期在线模式”下，终端上报后若当前无指令可下发，系统下发心跳包应答，以保持两者之间的连接状态；当系统接收到终端的心跳包后，系统下发应答指令。

8.2 配置指令

8.2.1 读 SIM 卡 ICCID 号（09H）

读取远传终端 SIM 卡的 ICCID 号。

表 8.2.1-1 读 SIM 卡 ICCID 号帧格式

主发	系统>>>终端									
序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	密钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0-D15	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX		HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1
数值	83H				1-9H	09H	10H00H			EDH

指令说明：

读取远传终端 SIM 卡的 ICCID 号。

D0：

固定值：01H，HEX 码。

D1-D15：

固定值：00H00H…00H，HEX 码

表 8.2.1-2 读 SIM 卡 ICCID 号应答帧格式

从发	终端>>>系统									
序号	1	2	3	4-11	12	13	14-15	16-47	48-49	50
名称	包头	协议版本	设备类型	设备编号	密钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0~D31	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX		HEX	HEX
字节	1	1	1	8	1	1	2	32	2	1
数值	67H				1-9H	09H	20H00H			EDH

指令说明：

远传终端应答“读 SIM 卡 ICCID 号”命令，返回 SIM 卡 ICCID 号。

D0~D31：读 SIM 卡 ICCID 号应答命令数据域说明详见下表 8.2.1-3。

表 8.2,1-3　读 SIM 卡 ICCID 号应答数据域

数据域	D0~D19	D20-D31
符号	D0-D19	D20-D31
名称	ICCID 号	补位
类型	ASCII	HEX
字节	20	12
数值		00H00H…00H

D0~D19：

20 个字符的 ICCID 号，用 ASCII 码表示。

例：ICCID 字符串"00000000000000000001"

D0-D19：30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 30H 31H。

D20~D31：

固定值：00H00H…00H

8.2.2　错误状态码应答（0EH）

表 8.2.2-1 错误状态码帧格式

从发	终端>>>系统									
序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	密钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0-D15	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX		HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1
数值	67H				1-9H	0EH	10H00H			EDH

指令说明：

远传终端收到不正确的协议格式等错误时，根据具体错误内容进行“错误状态码应答”。指令内容见表。

D0~D15：错误状态码命令数据域说明详见下表 8.2.2-2。

表 8.2.2-2 错误状态码数据域

数据域	D0	D1-D15
符号	D0	D1-D15
名称	错误码	补位
类型	HEX	HEX
字节	1	15
数值		00H00H…00H

D0：

错误码。HEX 码。

00H：未知错误；

01H：帧头格式错误；

02H：帧尾格式错误；

03H：密钥号错误；

04H：协议中数据“长度”数值超范围；

05H：CRC 校验错；

06H：协议版本号错；

07H：设备类型错误；

08H：设备编号错误；

09H：无效命令码。

D1-D15：

固定值：00H00H…00H

8.2.3 设置网络 IP 参数（56H）

表 8.2.3-1 设置网络 IP 参数帧格式

主发	系统>>>终端									
序号	1	2	3	4-11	12	13	14-15	16-47	48-49	50
名称	包头	协议版本	设备类型	设备编号	密钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0-D31	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	32	2	1
数值	83H				1-9H	56H	20H00H			EDH

指令说明：

系统将 IP 参数设置给终端，并由“IP 类别”区分所设置的类别。

每个终端可设置 0-2 个 IP 地址，IP 地址的启用情况由“IP 标识”确定。当有多个 IP 地址启用时，终端按照从 IP 小号位置到大号位置的顺序依次进行连接，即先使用 IP1 连接，再使用 IP2 连接，依此类推。可将两个 IP 设置为同一种类别，可同时启用两个 IP。

D0~D31：设置网络 IP 参数命令数据域说明详见表 8.2.3-2。

表 8.2.3-2　设置网络 IP 参数数据域

数据域	D0~D31						
符号	D0	D1	D2~D3	D4	D5~D8	D9~D16	D17-D31
名称	IP 标识	IP 设置 1				IP 设置 2	补位
		IP 类别	端口号	网络	IP1	……	
类型	HEX	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	2	1	4	8	15
数值							00H00H…00H

D0：

IP 标识为启用标识，区分测试 IP、运营 IP。00H，测试 IP 地址；01H，运营 IP 地址。

D1~D8：

IP 设置 1：

D1：IP 类别：00H 为测试 IP；

D2~D3：端口号：低字节在前，高字节在后。例如，9013H，D2=13H，D3=90H，代表端口 5008。

D4：网络：网络选择，不同的字节数值代表不同的网络类型选择。00H，未指定，表自动选择；01H，移动卡；02H，联通卡；03H：电信卡。

D5~D8：IP 地址 1：包含 4 字节 IP 地址。高字节在前，低字节在后。例如，IP 地址为：10.36.128.15，则 D5=0AH，D6=24H，D7=80H，D8=0FH。

D9~D16：

IP 设置 2：

D9：IP 类别：01 为运营 IP。

D10~D15：同 IP 设置 2。

D17~D31：

固定值：00H00H…00H

表 8.2.3-3　设置网络 IP 参数应答帧格式

从发	终端>>>系统									
序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	密钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0-D15	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1
数值	67H				1-9H	56H	10H00H			EDH

指令说明：

终端对系统设置 IP 参数命令进行应答。

D0~D15：设置 IP 参数命令应答数据域说明详见下表 8.2.3-4。

表 8.2.3-4 设置网络 IP 参数应答帧数据域

数据域	D0~D15	
符号	D0	D1-D15
名称	应答标识	补位
类型	HEX	HEX
字节	1	15
数值		00H00H…00H

D0：

应答标识，数值为 00H：设置网络 IP 成功；01H：设置网络 IP 失败。HEX 码。

D1-D15:

固定值：00H00H…00H

8.2.4　设置网络域名参数（57H）

表 8.2.4-1 设置网络域名参数帧格式

主发	系统>>>终端									
序号	1	2	3	4-11	12	13	14-15	16-95	96-97	98
名称	包头	协议版本	设备类型	设备编号	密钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0~D79	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	80	2	1
数值	83H				1-9H	57H	50H00H			EDH

指令说明：

系统将域名设置给终端，并根据“域名标识”区分所设置的类别。

每个终端可设置 2 个域名，由“域名标识”确定域名的启用情况。每个“域名设置”占用 35 字节，其中“域名”固定占用 30 字节，但域名具体长度值由相应位置号的“域名长度”确定。当有多个域名启用时，终端按照从域名小号位置到大号位置的顺序依次进行连接，即先使用域名 1 连接，再使用域名 2 连接，依此类推。可将两个域名设置为同一种类别，可同时启用两个域名。

D0~D79：设置网络域名参数命令数据域说明详见下表 8.2.4-2。

表 8.2.4-2　设置网络域名参数数据域

数据域	D0~D79											
符号	D0	D1	D2-D3	D4	D5	D6-D35	D36	D37-D38	D39	D40	D41-D70	D71-D79
名称	域名标识	域名设置 1					域名设置 2					补位
		域名类别	端口号	网络	长度	域名 1	域名类别	端口号	网络	长度	域名 2	
类型	HEX	HEX	HEX	HEX	HEX	HEX	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	2	1	1	30	35					9
数值		00H					01H					00H00H…00H

D0：

域名标识，00H 代表测试域名被启用，01H 代表运营域名被启用。1 字节，HEX 码。

D1：

00H，测试域名地址。

D2-D35：

域名设置 1：

D1-D2：端口号：低字节在前，高字节在后。例如，901FH，D1=1FH，D2=90H，代表端 8080。

D3：网络：网络选择，不同的字节数值代表不同的网络类型选择。00H，未指定表自动选择 01H，移动卡；02H，联通卡；03H：电信卡。

D4：域名长度 1：表示“域名 1”内域名实际占用字节的长度。

D5-D34：域名 1：固定长度 30 字节。但域名占用字节长度由“域名长度 1”确定，不足 30 字节部分补 00H。

例如，域名为：www.xinao.com，则“域名长度 1”=0DH，则 D6=‘w’ D7=‘w’ D8=‘w’ D9=‘.’ D10=‘x’ D11=‘i’ D12=‘n’ D13=‘a’ D14=‘o’D15=‘.’D16=‘c’D17=‘o’D18=‘m’ D19-D35：00H 00H 00H 00H 00H 00H 00H 00H 00H 00H 00H 00H 00H 00H 00H 00H 00H。

D36：

01H，运营域名地址。

D37-D70：

域名设置 2，同域名设置 1。

D71-D79：

固定值：00H00H…00H

表 8.2.4-3 设置网络域名参数应答帧格式

从发	终端>>>系统									
序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	密钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0~D15	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX		HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1
数值	67H				1-9H	57H	10H00H			EDH

指令说明：

设置终端网络域名参数命令应答。

D0~D15：设置网络域名参数应答命令数据域说明详见下表 8.2.4-4。

表 8.2.4-4　设置网络域名参数应答数据域

数据域	D0~D15	
符号	D0	D1-D15
名称	应答标识	补位
类型	HEX	HEX
字节	1	15
数值		00H00H…00H

D0：

应答标识，数值为 00H：设置网络域名成功；01H：设置网络域名失败。HEX 码。

D1-D15:

固定值：00H00H···00H

8.2.5　设置采集上报周期（49H）

表8.2.5-1　设置采集上报周期帧格式

主发	系统>>>终端									
序号	1	2	3	4-11	12	13	14-15	16-47	48-49	50
名称	包头	协议版本	设备类型	设备编号	秘钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0~D31	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	32	2	1
数值	83H				0-9	49H	20H00H			EDH

指令说明：

设置远传终端采集上报周期参数。

D0~D31：设置采集上报周期数据域详见下表8.2.5-2。

表8.2.5-2　设置采集上报周期数据域

数据域	D0~D31							
符号	D0	D1	D2~D25					D30~D31
			D2~D3	……	D24~D25	D26~D27	D28~D29	
名称	上报方式	定时上报日次数	定时上报时间			间隔上报的间隔时间	间隔上报首次上报时间	采集的间隔时间
			定时上报	……	定时上报			

			时间 1		时间 12			
类型	HEX	HEX	BCD	……	BCD	HEX	BCD	HEX
字节	1	1	2	……	2	2	2	2
数值				……				

D0：

上报方式，占用1个字节，不同的字节数值代表不同的功能，默认间隔上报，HEX码。

01H-定时上报（D26-D29 无效）；

02H-间隔上报（D1-D25 无效）。

说明：终端出厂默认间隔上报时间为 30 分钟，间隔采样时间为 30 分钟。

D1：

定时上报的日次数，取值范围1-12，默认为1，占用1字节，HEX码。

D2~D25：

定时上报时间,定时上报时间设定不分先后顺序,终端按照时间先后顺序进行定时上报。

默认为0。对应时间段数值为9999，则代表该时间段无设置。占用24字节，BCD码。

D2~D3：定时上报时间1，小时分钟；

D4~D5：定时上报时间2，小时分钟；

… …

D24~D25：定时上报时间 12，小时分钟。

D26~D27：

间隔上报的间隔时间，分钟，占用2个字节，低字节在前，高字节在后，HEX 码，默认为1440 分钟。

D28~D29：

间隔上报的首次上报时间，小时分钟，占用2个字节，BCD码。默认为0。

D30~D31：

采集的间隔时间，分钟，占用2个字节，低字节在前，高字节在后，HEX 码，默认为30 分钟。（定时上报或间隔上报，采集间隔时间均有效）

表8.2.5-3 设置采集上报周期应答帧格式

从发	终端>>>系统

序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	秘钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1
数值	67H				1-9H	49H	10H00H			EDH

指令说明：

远传终端应答“设置配置”命令。

D0：

应答标识，数值为00H：设置成功；01H：设置失败。HEX 码。

D1-D15:

固定值：00H00H…00H

8.2.6 读取采集上报周期（4AH）

表8.2.6-1 读取采集上报周期帧格式

主发	系统>>>终端									
序号	1	2	3	4-11	12	13	14-15	16-47	48-49	50
名称	包头	协议版本	设备类型	设备编号	秘钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0~D15	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1
数值	83H				0-9	4AH	10H00H			EDH

指令说明：

读取远传终端采集上报周期参数。

D0: 固定值01，HEX码

D1-D15: 00H00H…00H, HEX码

表8.2.6-2 读取采集上报周期应答帧格式

从发	终端>>>系统									
序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	秘钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0~D31	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	32	2	1
数值	67H				1-9H	4AH	20H00H			EDH

D0~D31：读取采集上报周期应答数据域详见下表8.2.6-3。

表8.2.6-3 读取采集上报周期应答数据域

数据域	D0~D31							
符号	D0	D1	D2~D25					D30~D31
			D2~D3	……	D24~D25	D26~D27	D28~D29	
名称	上报方式	定时上报日次数	定时上报时间			间隔上报的间隔时间	间隔上报首次上报时间	采集的间隔时间
			定时上报时间 1	……	定时上报时间 12			
类型	HEX	HEX	BCD	……	BCD	HEX	BCD	HEX
字节	1	1	2	……	2	2	2	2
数值				……				

D0：

上报方式，占用1个字节，不同的字节数值代表不同的功能，默认间隔上报，HEX码。

01H-定时上报（D26-D29 无效）；

02H-间隔上报（D1-D25 无效）。

说明：终端出厂默认间隔上报时间为 1440 分钟，间隔采样时间为 30 分钟。

D1：

定时上报的日次数，取值范围1-12，默认为1，占用1字节，HEX码。

D2~D25：

定时上报时间,定时上报时间设定不分先后顺序,终端按照时间先后顺序进行定时上报。

默认为0。对应时间段数值为9999，则代表该时间段无设置。占用24字节，BCD码。

D2~D3：定时上报时间1，小时分钟；

D4~D5：定时上报时间2，小时分钟；

……

D24~D25：定时上报时间 12，小时分钟。

D26~D27：

间隔上报的间隔时间，分钟，占用2个字节，低字节在前，高字节在后，HEX 码，默认为1440 分钟。

D28~D29：

间隔上报的首次上报时间，小时分钟，占用2个字节，BCD码。默认为0。

D30~D31：

采集的间隔时间，分钟，占用2个字节，低字节在前，高字节在后，HEX 码，默认为30 分钟。（定时上报或间隔上报，采集间隔时间均有效）。

8.2.7 校对时钟（05H）

表8.2.7-1　校对时钟帧格式

主发	系统>>>终端									
序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	密钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0~D15	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1
数值	83H				1-9H	05H	10H00H			EDH

指令说明：

系统给终端下发的系统当前时间（北京时间），用于给终端时钟校时。

系统每天接收第一次上报后，解析出数据域中的采集时间，若误差与当前系统时间误差超过 2 分钟，则系统会自动下发校时指令。

D0~D15：校对时钟命令数据域说明详见下表 8.2.7-2。

表8.2.7-2　校对时钟数据域

数据域	D0~D15	
符号	D0-D5	D6-D15
名称	系统当前时钟	补位
类型	BCD	HEX
字节	6	10
数值		00H00H…00H

D0~D5：

系统下发给终端的当前时钟。年月日时分秒，年在高位，秒在低位。6 字节，BCD 码。

例：20160704152250

D0：16H-代表 16 年；

D1：07H-代表 7 月；

D2：04H-代表 4 日；

D3：15H-代表 15 时；

D4：22H-代表 22 分；

D5：50H-代表 50 秒；

D6-D15:

固定值：00H00H…00H

表8.2.7-3　校对时钟应答帧格式

从发	终端>>>系统									
序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	密钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0-D15	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX		HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1
数值	67H				1-9H	05H	10H00H			EDH

指令说明：

远传终端应答“校对时钟”命令。

D0：

应答标识，数值为 00H：校时成功，01H：校时失败。HEX 码。

D1-D15:

固定值：00H00H…00H

8.2.8 设置报警浓度值（58H）

表8.2.8-1　设置报警浓度值帧格式

主发	系统>>>终端									
序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	密钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0~D15	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1
数值	83H				1-9H	58H	10H00H			EDH

指令说明：

设置远传终端的报警浓度值。

D0~D4：报警值设置命令数据域说明详见下表。

表 8.2.8-2　报警浓度值下发数据域

数据域	D0 ~ D15			
符号	D0	D1~D4	D5~D8	D9~D15
名称	数据类型	低限报警值	高限报警值	补位
类型	HEX	HEX	HEX	HEX
字节	1	4	4	7
数值				00H00H…00H

D0:

数据类型（与 22H 数据类型一致）不同的字节数值代表不同的数据种类，1 字节，HEX 码,每一路探测器数据类型与下发数据类型相同时，报警值统一设置为下发报警值。

01H：甲烷 CH4 传感器；

02H：一氧化碳 CO 传感器；

03H：丙烷 C3H8 传感器；

04H：氢气 H2 传感器；

05H：温度传感器；

06H：湿度传感器；

07H：PM2.5 传感器；

08H：预留；

D1~D4：

低限报警浓度数据 4 字节，数据不体现小数，但数值默认缩小 100 倍，即具有 2 位小数，甲烷 CH4 默认低限报警浓度为 25%LEL，国标允许设置范围 5%LEL~25%LEL；

D5~D8：

高限报警浓度数据 4 字节，数据不体现小数，但数值默认缩小 100 倍，即具有 2 位小数，甲烷 CH4 默认高限报警浓度为 50%LEL，国标不允许更改。

D9~D15：

固定值：00H00H…00H

表8.2.8-3　设置报警浓度应答帧格式

从发	终端>>>系统									
序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	密钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0-D15	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX		HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1
数值	67H				1-9H	58H	10H00H			EDH

指令说明：

远传终端应答“校对时钟”命令。

D0：

应答标识，数值为 00H：设置成功，01H：设置失败。HEX 码。

D1-D15:

固定值：00H00H…00H

8.2.9 读取报警浓度值（6AH）

表8.2.9-1　读取报警浓度值帧格式

主发	系统>>>终端

序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	密钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0~D15	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1
数值	83H				1-9H	6AH	10H			EDH

指令说明：

设置远传终端的报警浓度值。

D0~D4：报警值设置命令数据域说明详见下表。

表 8.2.9-2　读取报警浓度值上报数据域

数据域	D0 ~ D15	
符号	D0	D1~D15
名称	数据类型	补位
类型	HEX	HEX
字节	1	15
数值	01	00H00H…00H

D0:

数据类型（与 22H 数据类型一致）不同的字节数值代表不同的数据种类，1 字节，HEX 码,每一路探测器数据类型与下发数据类型相同时，报警值统一设置为下发报警值。

01H：甲烷 CH4 传感器；

02H：一氧化碳 CO 传感器；

03H：丙烷 C3H8 传感器；

04H：氢气 H2 传感器；

05H：温度传感器；

06H：湿度传感器；

07H：PM2.5 传感器；

08H：预留；

D1~D15：

固定值：00H00H…00H

表8.2.9-3　设置报警浓度应答帧格式

从发	终端>>>系统									
序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	密钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0-D15	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX		HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1
数值	67H				1-9H	6AH	10H00H			EDH

指令说明：

设置远传终端的报警浓度值。

D0~D4：报警值设置命令数据域说明详见下表。

表 8.2.8-2　读取报警浓度值数据域

数据域	D0 ~ D15			
符号	D0	D1~D4	D5~D8	D9~D15
名称	数据类型	低限报警值	高限报警值	补位
类型	HEX	HEX	HEX	HEX
字节	1	4	4	7
数值				00H00H…00H

D0:

数据类型（与 22H 数据类型一致）不同的字节数值代表不同的数据种类，1 字节，HEX 码,每一路探测器数据类型与下发数据类型相同时，报警值统一设置为下发报警值。

01H：甲烷 CH4 传感器；

02H：一氧化碳 CO 传感器；

03H：丙烷 C3H8 传感器；

04H：氢气 H2 传感器；

05H：温度传感器；

06H：湿度传感器；

07H：PM2.5 传感器；

08H：预留；

D1~D4：

低限报警浓度数据 4 字节，数据不体现小数，但数值默认缩小 100 倍，即具有 2 位小

数。

D5~D8：

高限报警浓度数据 2 字节，数据不体现小数，但数值默认缩小 100 倍，即具有 2 位小数。

D9~D15：

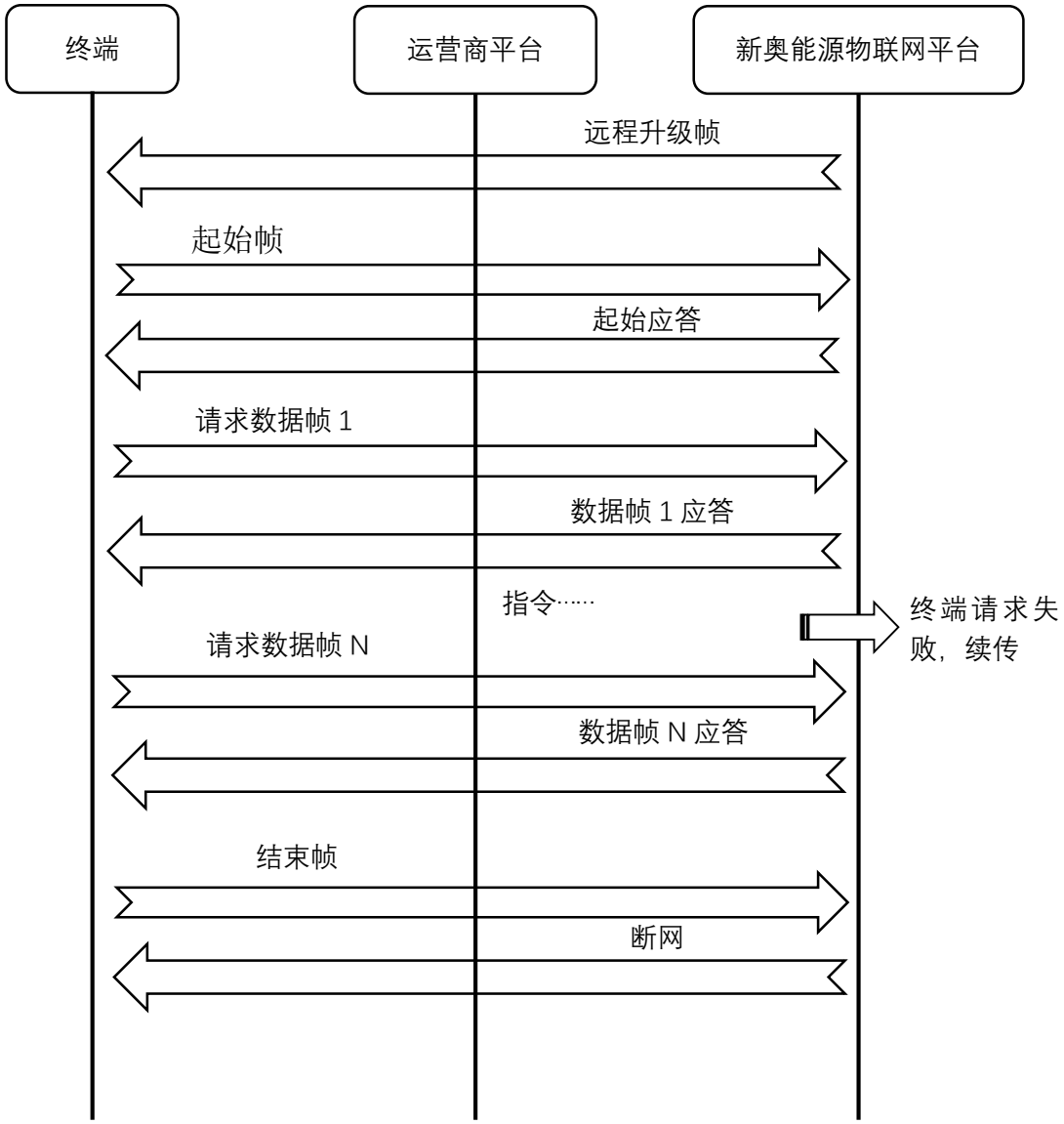
固定值：00H00H…00H

8.2.10 远程升级（5AH）

远程升级首先由系统下发远程升级帧，终端接收后转为以终端为主，系统为辅，一问一答式形式进行升级。远程升级分别为远程升级帧、起始帧，数据帧，结束帧。

升级流程:

终端收到系统远程升级帧后，开始自行下载固件(中途失败后终端自行续传)，固件下载完毕后终端验证代码校验，校验通过告知系统升级结束，系统删除升级任务(记录升级成功)，升级过程中设备状态发生变化应能正常上报。



1. 远程升级帧

系统下发远程升级帧。

表 8.2.10-1 远程升级帧格式

主发	**系统>>>终端**									
序号	1	2	3	4-11	12	13	14-15	16-47	48-49	50
名称	包头	协议版本	设备类型	设备编号	秘钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0-D15	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1

数值	83H				0-9H	5AH	10H 00H			EDH

D0：

数值为 00H，HEX 码。

D1-D15:

固定值：00H00H…00H

2. 起始帧

终端接收系统下发远程升级帧后，请求数据帧。

终端发送:

表 8.2.10-2 起始帧请求格式

从发	终端>>>系统									
序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	秘钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0~D15	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1
数值	67H				0-9H	5AH	10H 00H			EDH

D0~D15: 起始帧请求数据域详见下表 8.2.10-3。

表 8.2.10-3 起始帧请求数据域

数据域	D0-D15		
符号	D0	D1	D2-D15
名称	起始帧标识	最大字节数倍率	补位
类型	HEX	HEX	HEX
字节	1	1	15
数值	01H		00H00H…00H

D0:

起始帧标识，1 个字节，固定为 01H，HEX 码。

D1:

最大字节数倍率 1 个字节。

最大可接收数据长度为 128 字节的整数倍 N，终端请求系统在数据帧按 128*N 字节拆包。终端应根据自身传输能力尽可能请求系统组大包传输，系统根据此参数在数据帧按（1~N）*128 字节组包

D2-D15:

补位：固定值，00H00H…00H。

表 8.2.10-4 起始帧应答帧格式

主发	系统>>>终端									
序号	1	2	3	4-11	12	13	14-15	16-47	48-49	50
名称	包头	协议版本	设备类型	设备编号	秘钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0~D31	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	32	2	1
数值	83H				0-9H	5AH	20H 00H			EDH

D0~D31: 起始帧应答帧数据域详见下表 8.2.10-5。

表 8.2.10-5 起始帧应答数据域

数据域	D0-D15							
符号	D0	D1	D2	D3-D4	D5-D8	D9-D12	D13-D16	D17-D31
名称	起始帧标识	是否有升级任务	目标 CPU 板	总包数	任务 ID	固件字节数	固件校验和	补位
类型	HEX	HEX	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	2	4	4	4	15
数值	01H							00H00H…00H

D0:

起始帧标识，1 个字节，固定 01H。

D1:

是否有固件需要升级 1 个字节，00H 无固件，01H 有固件。

D2:

目标 CPU 板，对于多主板设备，用于区分此升级目标主板

D3-D4:

总包数，明确此固件拆包数量(按照接收到的倍率计算)，小端对齐。

D5-D8:

固件任务 id，小端对齐。

D9-D12:

固件总字节数，小端对齐

D13-16:

固件校验和，小端对齐。

D17-D31:

固定值：00H00H…00H。

3. 数据帧

终端遇到有固件需要升级时，发送此命令，如果请求失败后，终端应自行再次请求。

表 8.2.10-6 数据帧请求格式

从发	终端>>>系统									
序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	秘钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0~D15	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1
数值	67H				0-9H	5AH	10H 00H			EDH

D0~D15: 数据帧数据域详见下表 8.2.10-7。

表 8.2.10-7 数据帧数据域

数据域	D0-D15			
符号	D0	D1-D4	D5-D6	D7-D15
名称	数据帧标识	升级任务 ID	包号	补位
类型	HEX	HEX	HEX	HEX
字节	1	4	2	8
数值	02H			00H00H…00H

终端发送:

D0:

数据帧标识，1 个字节 固定为 02H

D1-D4：

固件升级任务 id 小端模式

D5,D6:

数据帧包号小端模式取值从 1 开始

D7-D15:

固定值：00H00H…00H。

系统应答:

表 8.2.10-8 数据帧应答帧格式

主发	系统>>>终端									
序号	1	2	3	4-11	12	13	14-15	16… 16*(N+1)-1	16*(N+1)-16*(N+1)+1	16*(N+1)+2

名称	包头	协议版本	设备类型	设备编号	秘钥号	命令码	长度	数据域	校验	包尾
符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	16*N	2	1
数值	83H				0-9H	5AH	16*N			EDH

D0~DL-1: 数据帧应答帧数据域详见下表 8.2.10-9。

表 8.2.10-9 数据帧应答帧数据域

数据域	D0-DL-1				
符号	D0	D1	D2-D3	D4-D15	D16-DL-1
名称	数据帧标识	状态码	包号	补位	升级文件内容
类型	HEX	HEX	HEX	HEX	HEX
字节	1	1	2	12	128*N
数值	02H			00H	

D0:

数据帧标识，1 个字节固定为 02H。

D1:

状态码：0xaa 正常，0x01 固件升级任务 id 错误，0x02 包号错误。

D2-D3:

数据帧包号，小端模式，取值从 1 开始

D4-D15:

固定值：00H00H…00H。

D16-DL-1:

每包数据 128*N 字节，最后一包不足 128 按实际值计算。

注: 终端接收到状态码为 0x01 时，代表服务器端的升级任务发生了改变，需重新发送起始帧

4. 结束帧

终端发送:

表 8.2.10-10 结束帧请求格式

从发	终端>>>系统									
序号	1	2	3	4-11	12	13	14-15	16-31	32-33	34
名称	包头	协议版本	设备类型	设备编号	秘钥号	命令码	长度	数据域	校验	包尾

符号	Head	Ver	Type	A0-A7	Key	CMD	L	D0~DL-1	CRC	End
类型	HEX	HEX	HEX	BCD	HEX	HEX	HEX	HEX	HEX	HEX
字节	1	1	1	8	1	1	2	16	2	1
数值	67H				0-9H	5AH	10H 00H			EDH

D0~D15: 结束帧数据域详见下表 8.2.10-11。

表 8.2.10-11 结束帧数据域格式

数据域	D0-D15		
符号	D0	D1	D2-D15
名称	结束帧标识	状态码	补位
类型	HEX	HEX	HEX
字节	1	1	15
数值	03H		00H00H…00H

D0:

结束帧标识 1 个字节 固定为 03H

D1:

状态码 0xaa 正常

0x03 固件大小错误

0x04 固件校验错误

0x05 固件错误(非本厂家固件)

D2~D15：

固定值：00H00H…00H。

系统接收到此命令后 下发断网命令

附录 A 加密算法

A.0.1 CRC16 算法函数（C 语言代码）

该算法只对本标准档描述协议有效。

函数说明：长度为 len 字节的 p_data 数据进行 CRC 计算，产生 16 位校验值

参数描述：p_data：待计算数据指针

len：待计算数据长度

p_crc：CRC校验值

返回值 0：成功 -1：失败

备注：最终计算结果为 16 位校验值。

```
uint16_t tp_sec_f2_crc16(const uint8_t * p_data, uint32_t len, uint16_t * p_crc)
{
    uint32_t i;
    uint16_t crc = 0xffff;
    if (NULL==p_data || NULL==p_crc || 0==len)
    {
        return -1;
    }
    for (i = 0; i < len; i++)
    {
        crc = (unsigned char)(crc >> 8) | (crc << 8);
        crc ^= p_data[i];
        crc ^= (unsigned char)(crc & 0xff) >> 4;
        crc ^= (crc << 8) << 4;
        crc ^= ((crc & 0xff) << 4) << 1;
    }
    *p_crc = crc;
    return 0;
}
```

A.0.2 AES 算法

1 嵌入式终端固件 C 语言算法

1 Aes_128.c:

```
/*
 *  Aes_128.c
 *
 *  Created on: Dec 30, 2016
 *          Author: Xiaoguang Yang
```

```
 *
 *    Description: Implementation of the AES-128 as defined by the FIPS PUB 197:
 *    the official AES standard
 */


#include <string.h>



// foreward sbox
const unsigned char sbox[256] =     {
//0        1       2        3       4       5       6       7        8       9       A        B       C
D       E       F
0x63, 0x7c, 0x77, 0x7b, 0xf2, 0x6b, 0x6f, 0xc5, 0x30, 0x01, 0x67, 0x2b, 0xfe, 0xd7, 0xab,
0x76, //0
0xca, 0x82, 0xc9, 0x7d, 0xfa, 0x59, 0x47, 0xf0, 0xad, 0xd4, 0xa2, 0xaf, 0x9c, 0xa4, 0x72,
0xc0, //1
0xb7, 0xfd, 0x93, 0x26, 0x36, 0x3f, 0xf7, 0xcc, 0x34, 0xa5, 0xe5, 0xf1, 0x71, 0xd8, 0x31,
0x15, //2
0x04, 0xc7, 0x23, 0xc3, 0x18, 0x96, 0x05, 0x9a, 0x07, 0x12, 0x80, 0xe2, 0xeb, 0x27, 0xb2,
0x75, //3
0x09, 0x83, 0x2c, 0x1a, 0x1b, 0x6e, 0x5a, 0xa0, 0x52, 0x3b, 0xd6, 0xb3, 0x29, 0xe3, 0x2f,
0x84, //4
0x53, 0xd1, 0x00, 0xed, 0x20, 0xfc, 0xb1, 0x5b, 0x6a, 0xcb, 0xbe, 0x39, 0x4a, 0x4c, 0x58, 0xcf,
//5
0xd0, 0xef, 0xaa, 0xfb, 0x43, 0x4d, 0x33, 0x85, 0x45, 0xf9, 0x02, 0x7f, 0x50, 0x3c, 0x9f,
0xa8, //6
0x51, 0xa3, 0x40, 0x8f, 0x92, 0x9d, 0x38, 0xf5, 0xbc, 0xb6, 0xda, 0x21, 0x10, 0xff, 0xf3,
0xd2, //7
0xcd, 0x0c, 0x13, 0xec, 0x5f, 0x97, 0x44, 0x17, 0xc4, 0xa7, 0x7e, 0x3d, 0x64, 0x5d, 0x19,
0x73, //8
```

0x60, 0x81, 0x4f, 0xdc, 0x22, 0x2a, 0x90, 0x88, 0x46, 0xee, 0xb8, 0x14, 0xde, 0x5e, 0x0b, 0xdb, //9

0xe0, 0x32, 0x3a, 0x0a, 0x49, 0x06, 0x24, 0x5c, 0xc2, 0xd3, 0xac, 0x62, 0x91, 0x95, 0xe4, 0x79, //A

0xe7, 0xc8, 0x37, 0x6d, 0x8d, 0xd5, 0x4e, 0xa9, 0x6c, 0x56, 0xf4, 0xea, 0x65, 0x7a, 0xae, 0x08, //B

0xba, 0x78, 0x25, 0x2e, 0x1c, 0xa6, 0xb4, 0xc6, 0xe8, 0xdd, 0x74, 0x1f, 0x4b, 0xbd, 0x8b, 0x8a, //C

0x70, 0x3e, 0xb5, 0x66, 0x48, 0x03, 0xf6, 0x0e, 0x61, 0x35, 0x57, 0xb9, 0x86, 0xc1, 0x1d, 0x9e, //D

0xe1, 0xf8, 0x98, 0x11, 0x69, 0xd9, 0x8e, 0x94, 0x9b, 0x1e, 0x87, 0xe9, 0xce, 0x55, 0x28, 0xdf, //E

0x8c, 0xa1, 0x89, 0x0d, 0xbf, 0xe6, 0x42, 0x68, 0x41, 0x99, 0x2d, 0x0f, 0xb0, 0x54, 0xbb, 0x16 }; //F

// inverse sbox

const unsigned char rsbox[256] =

{ 0x52, 0x09, 0x6a, 0xd5, 0x30, 0x36, 0xa5, 0x38, 0xbf, 0x40, 0xa3, 0x9e, 0x81, 0xf3, 0xd7, 0xfb

, 0x7c, 0xe3, 0x39, 0x82, 0x9b, 0x2f, 0xff, 0x87, 0x34, 0x8e, 0x43, 0x44, 0xc4, 0xde, 0xe9, 0xcb

, 0x54, 0x7b, 0x94, 0x32, 0xa6, 0xc2, 0x23, 0x3d, 0xee, 0x4c, 0x95, 0x0b, 0x42, 0xfa, 0xc3, 0x4e

, 0x08, 0x2e, 0xa1, 0x66, 0x28, 0xd9, 0x24, 0xb2, 0x76, 0x5b, 0xa2, 0x49, 0x6d, 0x8b, 0xd1, 0x25

, 0x72, 0xf8, 0xf6, 0x64, 0x86, 0x68, 0x98, 0x16, 0xd4, 0xa4, 0x5c, 0xcc, 0x5d, 0x65, 0xb6, 0x92

, 0x6c, 0x70, 0x48, 0x50, 0xfd, 0xed, 0xb9, 0xda, 0x5e, 0x15, 0x46, 0x57, 0xa7, 0x8d, 0x9d, 0x84

, 0x90, 0xd8, 0xab, 0x00, 0x8c, 0xbc, 0xd3, 0x0a, 0xf7, 0xe4, 0x58, 0x05, 0xb8, 0xb3, 0x45, 0x06

, 0xd0, 0x2c, 0x1e, 0x8f, 0xca, 0x3f, 0x0f, 0x02, 0xc1, 0xaf, 0xbd, 0x03, 0x01, 0x13, 0x8a,

```
0x6b
    , 0x3a, 0x91, 0x11, 0x41, 0x4f, 0x67, 0xdc, 0xea, 0x97, 0xf2, 0xcf, 0xce, 0xf0, 0xb4, 0xe6,
0x73
    , 0x96, 0xac, 0x74, 0x22, 0xe7, 0xad, 0x35, 0x85, 0xe2, 0xf9, 0x37, 0xe8, 0x1c, 0x75, 0xdf,
0x6e
    , 0x47, 0xf1, 0x1a, 0x71, 0x1d, 0x29, 0xc5, 0x89, 0x6f, 0xb7, 0x62, 0x0e, 0xaa, 0x18, 0xbe,
0x1b
    , 0xfc, 0x56, 0x3e, 0x4b, 0xc6, 0xd2, 0x79, 0x20, 0x9a, 0xdb, 0xc0, 0xfe, 0x78, 0xcd, 0x5a,
0xf4
    , 0x1f, 0xdd, 0xa8, 0x33, 0x88, 0x07, 0xc7, 0x31, 0xb1, 0x12, 0x10, 0x59, 0x27, 0x80, 0xec,
0x5f
    , 0x60, 0x51, 0x7f, 0xa9, 0x19, 0xb5, 0x4a, 0x0d, 0x2d, 0xe5, 0x7a, 0x9f, 0x93, 0xc9, 0x9c,
0xef
    , 0xa0, 0xe0, 0x3b, 0x4d, 0xae, 0x2a, 0xf5, 0xb0, 0xc8, 0xeb, 0xbb, 0x3c, 0x83, 0x53, 0x99,
0x61
    , 0x17, 0x2b, 0x04, 0x7e, 0xba, 0x77, 0xd6, 0x26, 0xe1, 0x69, 0x14, 0x63, 0x55, 0x21, 0x0c,
0x7d };


    // round constant
    const unsigned char Rcon[10] = {
        0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80, 0x1b, 0x36};
    // multiply by 2 in the galois field
    unsigned char galois_mul2(unsigned char value)
    {
       signed char temp;
       // cast to signed value
       temp = (signed char) value;
       // if MSB is 1, then this will signed extend and fill the temp variable with 1's
       temp = temp >> 7;
```

```
    // AND with the reduction variable
    temp = temp & 0x1b;
    // finally shift and reduce the value
    return ((value << 1)^temp);
}

// AES encryption and decryption function
// The code was optimized for memory (flash and ram)
// Combining both encryption and decryption resulted in a slower implementation
// but much smaller than the 2 functions separated
// This function only implements AES-128 encryption and decryption (AES-192 and
// AES-256 are not supported by this code)
/*
      in_data:原始数据地址
        state:加解密操作后数据地址
        aakey:使用的秘钥
        dir :加密/解密
        注意:每次只能加密/解密 16 个字节
*/
void aes_enc_dec(unsigned char *in_data, unsigned char *state, unsigned char *aakey,
    unsigned char dir)
{
  unsigned char buf1, buf2, buf3, buf4, round, i;
  unsigned char key_buffer[16];
  memcpy(state,in_data,16);
  memcpy(key_buffer,aakey,16);
  // In case of decryption
  if (dir) {
    // compute the last key of encryption before starting the decryption
    for (round = 0 ; round < 10; round++) {
```

```
        //key schedule
        key_buffer[0] = sbox[key_buffer[13]]^key_buffer[0]^Rcon[round];
        key_buffer[1] = sbox[key_buffer[14]]^key_buffer[1];
        key_buffer[2] = sbox[key_buffer[15]]^key_buffer[2];
        key_buffer[3] = sbox[key_buffer[12]]^key_buffer[3];
        for (i=4; i<16; i++) {
          key_buffer[i] = key_buffer[i] ^ key_buffer[i-4];
        }
      }

      //first Addroundkey
      for (i = 0; i <16; i++){
        state[i]=state[i] ^ key_buffer[i];
      }
    }

    // main loop
    for (round = 0; round < 10; round++){
      if (dir){
        //Inverse key schedule
for (i=15; i>3; --i) {
        key_buffer[i] = key_buffer[i] ^ key_buffer[i-4];
        }
        key_buffer[0] = sbox[key_buffer[13]]^key_buffer[0]^Rcon[9-round];
        key_buffer[1] = sbox[key_buffer[14]]^key_buffer[1];
        key_buffer[2] = sbox[key_buffer[15]]^key_buffer[2];
        key_buffer[3] = sbox[key_buffer[12]]^key_buffer[3];
      } else {
        for (i = 0; i <16; i++){
```

```
      // with shiftrow i+5 mod 16
 state[i]=sbox[state[i] ^ key_buffer[i]];
  }
  //shift rows
  buf1 = state[1];
  state[1] = state[5];
  state[5] = state[9];
  state[9] = state[13];
  state[13] = buf1;

  buf1 = state[2];
  buf2 = state[6];
  state[2] = state[10];
  state[6] = state[14];
  state[10] = buf1;
  state[14] = buf2;

  buf1 = state[15];
  state[15] = state[11];
  state[11] = state[7];
  state[7] = state[3];
    state[3] = buf1;
}
//mixcol - inv mix
if ((round > 0 && dir) || (round < 9 && !dir)) {
  for (i=0; i <4; i++){
    buf4 = (i << 2);
    if (dir){
      // precompute for decryption
```

```
                buf1 = galois_mul2(galois_mul2(state[buf4]^state[buf4+2]));
                buf2 = galois_mul2(galois_mul2(state[buf4+1]^state[buf4+3]));
                state[buf4] ^= buf1; state[buf4+1] ^= buf2; state[buf4+2] ^= buf1; state[buf4+3] ^=
buf2;
            }
            // in all cases
            buf1 = state[buf4] ^ state[buf4+1] ^ state[buf4+2] ^ state[buf4+3];
            buf2 = state[buf4];
            buf3 = state[buf4]^state[buf4+1]; buf3=galois_mul2(buf3); state[buf4] = state[buf4] ^
buf3 ^ buf1;
            buf3 = state[buf4+1]^state[buf4+2]; buf3=galois_mul2(buf3); state[buf4+1] =
state[buf4+1] ^ buf3 ^ buf1;
            buf3 = state[buf4+2]^state[buf4+3]; buf3=galois_mul2(buf3); state[buf4+2] =
state[buf4+2] ^ buf3 ^ buf1;
            buf3 = state[buf4+3]^buf2; buf3=galois_mul2(buf3); state[buf4+3] =
state[buf4+3] ^ buf3 ^ buf1;
          }
        }

        if (dir) {
          //Inv shift rows
          // Row 1
    buf1 = state[13];
          state[13] = state[9];
          state[9] = state[5];
          state[5] = state[1];
          state[1] = buf1;
          //Row 2
          buf1 = state[10];
```

```
      buf2 = state[14];
      state[10] = state[2];
      state[14] = state[6];
      state[2] = buf1;
      state[6] = buf2;
      //Row 3
      buf1 = state[3];
      state[3] = state[7];
      state[7] = state[11];
      state[11] = state[15];
      state[15] = buf1;

      for (i = 0; i <16; i++){
        // with shiftrow i+5 mod 16
        state[i]=rsbox[state[i]] ^ key_buffer[i];
      }
    } else {
      //key schedule
      key_buffer[0] = sbox[key_buffer[13]]^key_buffer[0]^Rcon[round];
      key_buffer[1] = sbox[key_buffer[14]]^key_buffer[1];
      key_buffer[2] = sbox[key_buffer[15]]^key_buffer[2];
      key_buffer[3] = sbox[key_buffer[12]]^key_buffer[3];
      for (i=4; i<16; i++) {
        key_buffer[i] = key_buffer[i] ^ key_buffer[i-4];
      }
    }
  }
  if (!dir) {
  //last Addroundkey
```

```
      for (i = 0; i <16; i++){
        // with shiftrow i+5 mod 16
        state[i]=state[i] ^ key_buffer[i];
      } // enf for
    } // end if (!dir)
} // end function
```

2 Aes_128.h:

```
/*
  AES 加解密程序头文件
  */
#ifndef TI_OPT_AES_H_
#define TI_OPT_AES_H_
#define ENCRYPT      0     //加密
#define DECRYPT      1     //解密
#define   DEBUG_PRINTF_EN   0 //
void aes_enc_dec(unsigned char *in_data, unsigned char *state, unsigned char *aakey, unsigned char dir);
/*
单次调用只能加密/解密 128bit(16byte)数据.
in_data ：原始数据
state    ：加解密操作后数据
aakey    ：使用的秘钥
dir      ：0---加密
1---解密
*/
#endif /* TI_OPT_AES_H_ */
```

3）AES 软件加密算法使用说明：

该程序支持 128BIT 秘钥以及 128BIT 数据的加密和解密。

void aes_enc_dec(unsigned char *in_data, unsigned char *state, unsigned char *aakey, unsigned char dir)　只能对 16 字节数据块进行加密和解密操作；

如小于 16 字节则将原始数据补足为 16 字节；

如长度大于 16 字节，需要将原始数据补足为 16 字节的整数倍，每 16 个字节调用一次 aes_enc_dec 函数。

例：

原数据 test3 长度为 32 字节，则调用方式为:

aes_enc_dec(test3,test4,key1,DECRYPT);

aes_enc_dec(test3+16,test4+16,key1,DECRYPT);

2　系统端 JAVA 语言算法

AESUtils.java:

```
package cn.enn.smartcloud.core.utils;
import org.apache.commons.lang3.StringUtils;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
import java.util.UUID;
/**
 * Created by Administrator on 2015/4/9.
 */
public class AESUtils {

    private static final String CipherMode = "AES/ECB/NoPadding";
    private static final String KEY_PAIR_GENERATOR = "AES";
    private static final int MOD = 16;
    private static Cipher cipher = null;

    static {
        try {
```

```
            cipher = Cipher.getInstance(CipherMode);
        } catch (Exception ee) {
            ee.printStackTrace();
        }
    }

    public static String encrypt(String data, String key) {
        String ret = null;
        try {
            SecretKeySpec keySpec = createKey(key);
            cipher.init(Cipher.ENCRYPT_MODE, keySpec);
            byte[] result = cipher.doFinal(dataPadding(data));
            ret = EncryptUtils.hex2str(result);
        } catch (Exception ee) {
            ee.printStackTrace();
            ret = null;
        }
        return ret != null ? ret.toUpperCase() : null;
    }

    public static String decrypt(String data, String key) {
        String ret = null;
        try {
            SecretKeySpec keySpec = createKey(key);
            cipher.init(Cipher.DECRYPT_MODE, keySpec);
            byte[] result = cipher.doFinal(EncryptUtils.str2hex(data));
            ret = EncryptUtils.hex2str(result);
        } catch (Exception ee) {
            ee.printStackTrace();
```

```
            ret = null;
        }
        return ret != null ? ret.toUpperCase() : null;
    }

    private static SecretKeySpec createKey(String password) {
        byte[] data = keyPadding(password);
        return new SecretKeySpec(data, KEY_PAIR_GENERATOR);
    }

    private static byte[] dataPadding(String data) throws Exception{
        byte[] ret = null;

        if (StringUtils.isNotBlank(data)) {
            byte[] tmp = EncryptUtils.str2hex(data);
            if (tmp != null) {
                int tmpLength = tmp.length;
                int padLength = (tmpLength / MOD + ((tmpLength % MOD) > 0 ?
1 : 0)) * MOD;
                if (tmpLength % MOD > 0) {
                    ret = new byte[padLength];
                    for (int i = 0; i < tmpLength; i++) {
                        ret[i] = tmp[i];
                    }
                    for (int i = tmpLength; i < padLength; i++) {
                        ret[i] = (byte)(padLength - tmpLength);
                    }
        } else {
                    ret = tmp;
```

```
                }
            }
        }
        return ret;
    }
    private static byte[] keyPadding(String key) {
        byte[] ret = null;
        if (StringUtils.isNotBlank(key)) {
            byte[] tmp = EncryptUtils.str2hex(key);
            if (tmp != null) {
                int tmpLen = tmp.length;
                if (tmpLen < MOD) {
                    ret = new byte[MOD];
                    for (int i = tmpLen; i < MOD; i++) {
                        ret[i] = (byte)(MOD - tmpLen);
                    }
                } else {
                    ret = tmp;
                }
            }
        }
        return ret;
    }
    public static String getKey() {
        return UUID.randomUUID().toString().replace("-", "");
    }
    public static String xor(String op1, String op2) {
        byte[] b1 = EncryptUtils.str2hex(op1);
        byte[] b2 = EncryptUtils.str2hex(op2);
```

```
            for (int i = 0; i < b1.length; i++) {
                b1[i] = (byte)(b1[i] ^ b2[i]);
            }
            return EncryptUtils.hex2str(b1);
        }
    }
```

附录 B　各设备类型采集点说明

智能压力采集仪

固有属性	数值类型	单位
工作电压	双精度型	V
静电电流	双精度型	uA
工作电流	双精度型	mA
重量	整型	g
规格型号	字符型	
数据传输方式	整型	
上报模式	整型	
量程	整型	
电源	整型	
防水等级	字符型	
参数设置	字符型	
压力精度等级	双精度型	
工作温度	整型	℃
压力传感器补偿温度	整型	℃
防爆等级	字符型	
工作湿度	字符型	%
显示方式	整型	
外形尺寸	整型	mm
设备编号	字符型	

量测属性名称	必选/可选	数据类型	单位
管道压力	必选	瞬时值	kPa
环境温度	可选	瞬时值	℃
大气压力	可选	瞬时值	kPa
供电方式+电池电量	必选	瞬时值	mV
信号强度	必选	瞬时值	dBm
采集时间	必选	瞬时值	年月日时分秒
报警状态	必选	状态值	boolean
采集间隔	可选	瞬时值	min

云台式激光可燃气体探测器

固有属性	数值类型	单位
测量气体	字符型	
检测原理	字符型	
报警阀值	整型	ppm*m
指示激光安全等级	字符型	
检测激光安全等级	字符型	
测量量程	整型	ppm*m

静态检测限	整型	ppm*m
基本误差	双精度型	%F.S
响应时间	双精度型	s
遥测距离	整型	m
激光器自动标定功能	字符型	
工作电压	整型	V
典型功耗	整型	W
防爆方式	字符型	
防爆标志	字符型	
工作场所	字符型	
壳体材料	字符型	
固定方式	字符型	
电气接口	字符型	
防护等级	字符型	
工作温度	整型	℃
相对湿度	整型	%RH
操作方式	字符型	
管理终端	字符型	
云台水平方向运行范围	整型	°
云台垂直方向运行范围	整型	°
云台传动方式	字符型	
云台角度准确性	双精度型	°
云台预置点最大数量	整型	个
防电涌保护器标称放电电流（8/20μs）	整型	kA
防电涌保护器最大放电电流（8/20μs）	整型	kA
防电涌保护器电压保护水平（线-线/线-地）	整型	V
摄像机图像传感器	字符型	
摄像机最大光圈	字符型	
摄像机最低照度	字符型	
摄像机光学变焦倍数	整型	倍
摄像机焦距	字符型	
摄像机分辨率	字符型	
摄像机帧率	整型	fps
摄像机信噪比	整型	dB
摄像机红外补光（日/夜转换）	字符型	
整机网络接口	字符型	
浓度数据接口协议	字符型	
视频数据对外通信协议	字符型	
视频数据编码	字符型	

量测属性名称	必选/可选	数据类型	单位
浓度值	必选	瞬时值	ppm*m
泄露报警	必选	状态值	Boolean
故障状态	必选	状态值	Boolean
云台运行角速度	可选	瞬时值	°/s
云台运行位置	可选	状态值	
设备运行状态	可选	状态值	
报警日期	可选	瞬时值	YYYY-MM-DD
报警时间	可选	瞬时值	HH:MM:SS

阀门井

固有属性	数值类型	单位
供电方式	字符型	V
电池寿命	整型	
工作方式	字符型	
采样方式	字符型	
传感器类型	字符型	
检测参数	整型	温度：℃，甲烷：%LEL
检测精度	字符型	
显示方式	字符型	
工作温度	整型	℃
工作湿度	字符型	
环境压力	整型	
检测对象	字符型	
防护等级	字符型	
防爆等级	字符型	
产品尺寸/外形尺寸	整型	mm
数据传输	字符型	
参考标准	字符型	
重量	整型	kg

量测属性名称	必选/可选	数据类型	单位
气体浓度	必选	瞬时值	LEL
环境温度	必选	瞬时值	℃
可燃气体浓度报警	必选	状态值	
电池电量	必选	瞬时值	V
信号强度	必选	瞬时值	dBm
采集时间	必选	瞬时值	年月日时分秒

智能阴极保护终端

固有属性	数值类型	单位
报警信号	整型	
电池电量	整型	v

防护等级	整型	
远程通断	整型	
设备工作状态	整型	
位置信息	字符型	
设备编号	字符型	

量测属性名称	必选/可选	数据类型	单位
通电电位	必选	瞬时值	V
断电电位	必选	瞬时值	V
自然电位	必选	瞬时值	V
交流干扰电压	必选	瞬时值	V
直流电流	必选	瞬时值	A
交流电流	必选	瞬时值	A
电池电量	必选	瞬时值	mV
信号强度	必选	瞬时值	dBm
采集时间	必选	瞬时值	年月日时分秒
交流电流密度	可选	瞬时值	I/m2

家用气体报警器

固有属性	数值类型	单位
工作电压	双精度型	V
静电电流	双精度型	uA
工作电流	双精度型	mA
重量	整型	g
数据传输方式	整型	
气体检测量程	双精度型	%LEL，默认 20.0%LEL
电源	整型	
防水等级	字符型	
传感器类型	整型	半导体，催化等
浓度报警值	双精度型	%LEL,默认 8.0%LEL
浓度报警误差范围	双精度型	%LEL,±3.0%LEL
气体检测类型	整型	默认甲烷，液化气为丙烷
气体浓度响应时间（T90）	整型	≤10s（t_{90}）
传感器使用寿命	整型	默认 5 年
工作温度	整型	℃
防爆等级	字符型	
工作湿度	字符型	%
显示方式	整型	
外形尺寸	整型	mm
设备编号	字符型	

量测属性名称	必选/可选	数据类型	单位

气体浓度	必选	瞬时值	%LEL
环境温度	可选	瞬时值	℃
供电方式+电池电量	必选	瞬时值	mV
信号强度	必选	瞬时值	dBm
传感器故障状态	必选	状态值	boolean
可燃气体报警状态	必选	状态值	boolean