设备类型定义

1、设备地址：

每个遵循吉风通信协议的设备用全球唯一地址做为设备的地址,地址由 8 个字节构成，设备的所有通信都用唯一地址标识：

1.2，设备地址共 8 字节，第一字节表示地址扮演的角色，第二字节表示国家区域，后 6 个字节表示设备地址，有效范围是 00：00：00：00：00：00 - FF：FF：FF：FF：FF：FF。

2、设备权限

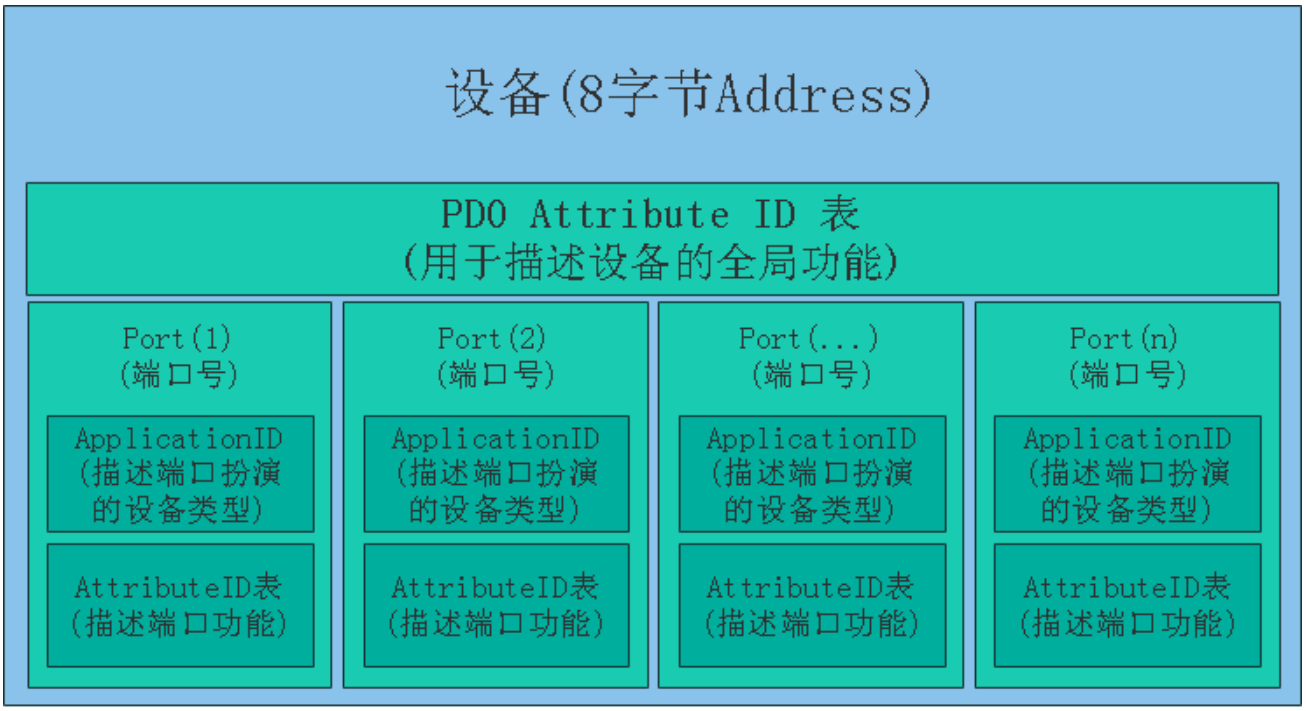
吉风通信协议对设备的每个具体操作可能会要求需要不同的操作权限。不同的操作权限将使用不同的通信密钥进行加密，下表列出所有的操作权限并对其进行说明：

权限值	密钥	描述
0x00	7777772e62696e676f696f742e636f6d	公共密钥，用于控制设备，所有设备一致，安全性差
0x01	服务器分配，与设备地址一一对应	仅用于服务器认证，不能控制与管理设备
0x02	设备每次入网时随机生成	用于管理设备与控制设备，具有最高权限
0x03	设备每次入网时随机生成	仅用于控制设备
0x04	设备每次重启或每隔 12 小时随机生成	仅用于控制设备，每 12 小时更新一次

对于临时成员密钥与默认密钥设备都有单独的开关控制是否开启这两个权限的设备控制功能。其中，公共密钥的设备权限默认关闭，默认情况下，设备并不能通过公共密钥操作设备，除非 sharelock 标志置 0。临时成员操作权限默认开启。

3、设备的功能描述

一个完整的设备应该包含以下信息：



a) 设备地址：用于描述通信设备唯一编号，用于设备位置的标识，每个设备的地址都是全球唯一的。
b) PDO AttributeID:用于描述设备的全局功能，比如支持文件系统，远程升级，信号强度，场景等功能。
c) Port:子设备编号(端口号)。
d) ApplicationID:子设备类型，比如开关，插座，传感器等，每个子设备只能有一个 ApplicationID。
e) AttributeID:子设备功能描述，一个子设备可能同时存在多个不同功能 AttributeID。

设备的全局功能由 PDO AttributeID 描述，仅需要向系统中注册相应的 PDO AttributeID 就可以实现对设备基本功能描述。设备基本功能描述不能描述子设备数量，子设备信息，每个 PDO AttributeID 的具体操作与通信数据格式由 option 与 cmd 决定，不同的 PDO AtrributeID 对 option 与 cmd 的定义可能 会不一样。以下将对其进行详细描述：

pluto 设备基本功能(PDO Attribute)		
aID		备注
0x00000000	设备 Server 信息服务，提供读写服务器信息功能。	每个设备必须包含的功能
0x00000001	设备基本描述信息，提供读写设备的描述信息功能。	
0x00000002	设备有效端口列表。提供读取设备有效端口的功能	
0x00000003	设备端口描述功能。提供读写设备端口信息的功能。	
0x00000004	设备智能情景。提供读写执行设备情景的功能。	
0x00000005	设备文件系统。提供读写设备内部所有文件的功能。	
0x00000006	设备黑白名单管理。提供读写用户操作权限的控制功能。	
0x00000007	设备报警记录。提供读写设备报警记录功能。	
0x00000008	设备操作记录。提供读写设备操作记录功能。	
0x0000000F	设备信标功能，提供标识与响应局域网内设备是否存在的功能。	
0x00000010	设备入网通知功能，设备入网后，主动发送通知信息给用户端。	

0x00000012	设备升级功能，提供设备的升级与维护	
0x00000100	增删减子设备功能	设备可选功能
0x00000101	保留	
0x00000102	设备通信信号质量	

3.1，AttributeID 值为 0x00000000。读写服务器信息，

用户可以自由更改服务器地址与信息。

支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x01	0x00	data={无} 读服务器信息	02
返回	0x81	0x00	data={ status[1]+json 字符串} 读服务器信息。 status={ 00：成功，01：失败}	-
发送	0x02	0x00	data={json 字符串} 写服务器信息，要求新的 json 字符串原来旧的 json 字符串中的 password 一致，否则更新失败。如果原设备没有服务器信息，不需要匹配 password 可直接写入	02
返回	0x82	0x00	data=status[1] status={ 00：成功，01：失败，04：无权限}	-

其 json 格式如下表所示：

```
{"PID": 1,
  "VID": 1,
  "product_date": "2018-07-06",
  "password": "123456",
  "sn": "2C6FED64480EF212AF933E5A46D1DEB0",
  "addr": "0100010100000013",
  "server_url": "www.glalaxy.com",
  "server_ipv4": "119.23.8.181",
  "server_udp_port": 16729,
  "server_tcp_port": 16728,
  "local_udp_port": 17729}
```

3.2，AttributeID 值为 0x00000001。读写设备的基本信息。

支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x01	0x00	data={无} 读服务器信息	02|03|04
返回	0x81	0x00	data={ status[1]+json 字符串} 读设备信息 status={ 00：成功，01：失败}	-

发送	0x02	0x00	data={json 字符串} 写入信息	02
返回			data=status[1] status={ 00：成功，01：失败，03：无效操作，04：无权限}	-

json 格式如下表所示：

{"version": 1,
"MFG": "2017-11-8 15:00:00",
"name": "Switch",
"dev_type": 3,
"encrypt": 0,
"remote_lock": 0,
"local_lock": 0,
"guest_lock": 0,
"share_lock": 1,
"history_record": 0,
"zone": 8,
"admin_key": "4C637853F803BFEA9B2418E91525EAA4",
"com_key": "2AF204783E9565E4A0359DACADE8484D",
"guest_key": "E821E52C6B77D0B0A51E46C31E62E487",
"attr_id": [1, 2, 3, 4, 5, 6, 7, 8, 15, 16, 18, 258]}

3.3，AttributeID 值为 0x00000002。读设备的所有有效端口。

支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x01	0x00	data={无} 读端口列表	02|03|04
返回	0x81	0x00	data={status[1]+端口列表，1 个字节表示 1 个端口} status={ 00：成功，01：失败，06 无端口}	-

3.4，AttributeID 值为 0x00000003。读设备的端口描述操作

支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x01	0x00	data={端口号} 读端口描述	02|03|04
返回	0x81	0x00	data={status[1]+json 字符串} status={ 00 成功，06 无端口}	-
发送	0x02	0x00	data={json 字符串} 写入端口信息	02
返回	0x82	0x00	data={status[1]} status={ 00 成功，01 失败，04 无权限}	-
发送	0x03	0x00	data={端口号} 删除端口	02

返回	0x83	0x00	status={ 00 成功，04 无权限，06 无端口}	-

json 格式如下表所示：

{“port”：01,

“name”：”switch”,

“name”：”switch”,

“app_id”:”123”,

"attr_id"： [256, 257, 18, 258] }

3.5，AttributeID 值为 0x00000004。情景操作

支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x01	0x00	读情景 data={crc[2]+name_length[2]+data_length[4]+data[n]} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={情景名长度，情景名从 data[0]开始} data_lenght={ 0} data={情景名}	02|03|04
返回	0x81	0x00	data=status[1]+crc[2]+name_length[2]+data_length[4]+data[n] status={00 成功，07 无此文件} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={情景名长度，情景名从 data[0]开始} data_lenght={ 情景内容长度} data={情景名+情景内容}	-
发送	0x02	0x00	写入情景 data=status[1]+crc[2]+name_length[2]+data_length[4]+data[n] status={00 成功，07 无此文件} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={情景名长度，情景名从 data[0]开始} data_lenght={ 情景内容长度} data={情景名+情景内容}	02
返回	0x82	0x00	data=status[1]+crc[2]+name_length[2]+data_length[4]+data[n] status={00 成功，01 失败，04 无权限} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={情景名长度，情景名从 data[0]开始} data_lenght={ 0} data={情景名}	-
发送	0x07	0x00	重命名 data=status[1]+crc[2]+name_length[2]+data_length[4]+data[n] status={00 成功，07 无此文件} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法}	02

			name_lenght={情景名长度，情景名从 data[0]开始} data_lenght={ 情景名长度} data={情景名+新情景名}	
返回	0x87	0x00	data=status[1]+crc[2]+name_length[2]+data_length[4]+data[n] status={00 成功，01 失败，04 无权限} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={情景名长度，情景名从 data[0]开始} data_lenght={ 0} data={情景名}	-
发送	0x03	0x00	删除情景 data={crc[2]+name_length[2]+data_length[4]+data[n]} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={情景名长度，情景名从 data[0]开始} data_lenght={ 0} data={情景名}	02
返回	0x83	0x00	data=status[1]+crc[2]+name_length[2]+data_length[4]+data[n] status={00 成功，04 无权限} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={情景名长度，情景名从 data[0]开始} data_lenght={ 0} data={情景名}	
发送	0x09	0x00	写入新状态 data=status[1]+crc[2]+name_length[2]+data_length[4]+data[n] status={0x12 某情景在情景执行完之后执行。data={ 被加入的情景名}，0x11 测试情景，0x0B 重新执行情景或从暂停状态中恢复运行。0x0C 暂停情景，0x0D 停止情景} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={情景名长度，情景名从 data[0]开始} data_lenght={ 0} data={情景名}	02|03
返回	0x89	0x00	ata=status[1]+crc[2]+name_length[2]+data_length[4]+data[n] status={00 成功，01 操作失败，03 无效操作，0F 无此情景} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={情景名长度，情景名从 data[0]开始} data_lenght={ 0} data={情景名 or 情景名+情景名}	
发送	0x06	0x00	读情景名 data={crc[2]+name_length[2]+data_length[4]+data[n]} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={情景名长度，情景名从 data[0]开始} data_lenght={ 0} data={情景名}	02|03|04
返回	0x86	0x00	data=status[1]+crc[2]+name_length[2]+data_length[4]+data[n]	

			status={0x0A 无效，0x0B 重新执行情景或从暂停状态中恢复运行。0x0C 暂停情景，0x0D 停止情景,0x0E 执行完成，0x0F 无此情景，0x11 测试，0x02 加入} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={情景名长度，情景名从 data[0]开始} data_lenght={ 0} data={情景名}	
发送	0x06	0x00	读情状态 data={crc[2]+name_length[2]+data_length[4]+data[n]} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={情景名长度，情景名从 data[0]开始} data_lenght={ 0} data={情景名}	02|03|04
返回	0x86	0x00	data=status[1]+crc[2]+name_length[2]+data_length[4]+data[n] status={0x0A 无效，0x0B 重新执行情景或从暂停状态中恢复运行。0x0C 暂停情景，0x0D 停止情景,0x0E 执行完成，0x0F 无此情景，0x11 测试，0x02 加入} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={情景名长度，情景名从 data[0]开始} data_lenght={ 0} data={情景名}	

3.6，AttributeID 值为 0x00000005。文件操作

支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x01	0x00	读文件 data={crc[2]+name_length[2]+data_length[4]+data[n]} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={文件名长度，文件名从 data[0]开始} data_lenght={ 0} data={文件名}	02|03|04
返回	0x81	0x00	data=status[1]+crc[2]+name_length[2]+data_length[4]+data[n] status={00 成功，07 无此文件} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={文件名长度，文件名从 data[0]开始} data_lenght={ 文件内容长度} data={文件名+文件内容}	-
发送	0x02	0x00	写入文件 data=status[1]+crc[2]+name_length[2]+data_length[4]+data[n] status={00 成功，07 无此文件} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={文件名长度，文件名从 data[0]开始} data_lenght={ 文件内容长度}	02

			data={文件名+文件内容}	
返回	0x82	0x00	data=status[1]+crc[2]+name_length[2]+data_length[4]+data[n] status={00 成功，01 失败，04 无权限} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={文件名长度，文件名从 data[0]开始} data_lenght={ 0} data={文件名}	-
发送	0x07	0x00	重命名 data=status[1]+crc[2]+name_length[2]+data_length[4]+data[n] status={00 成功，07 无此文件} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={文件名长度，文件名从 data[0]开始} data_lenght={ 文件名长度} data={文件名+新文件名}	02
返回	0x87	0x00	data=status[1]+crc[2]+name_length[2]+data_length[4]+data[n] status={00 成功，01 失败，04 无权限} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={文件名长度，文件名从 data[0]开始} data_lenght={ 0} data={新文件名}	-
发送	0x03	0x00	删除文件 data={crc[2]+name_length[2]+data_length[4]+data[n]} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={文件名长度，文件名从 data[0]开始} data_lenght={ 0} data={文件名}	02
返回	0x83	0x00	data=status[1]+crc[2]+name_length[2]+data_length[4]+data[n] status={00 成功，04 无权限} crc={data 字段的 CRC16 效验，使用 modbus CRC16 算法} name_lenght={文件名长度，文件名从 data[0]开始} data_lenght={ 0} data={文件名}	

3.7，AttributeID 值为 0x00000006。黑白名单管理

支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x01	0x00	读黑白名单表	02|03|04

			data={无}	
返回	0x81	0x00	data=status[1]+json 格式名单表 status={00 成功，03 黑白名单不启用}	-
发送	0x02	0x00	写入黑白名单表 data={json 格式的名单列表}	02
返回	0x82	0x00	data=status[1] status={00 成功，01 失败，04 无权限}	-
发送	0x03	0x00	删除黑白名单表 data={无}	02
返回	0x83	0x00	data=status[1] status={00 成功，01 失败，04 无权限}	
发送	0x08	0x00	读黑白名单使能状态 data={无}	02|03|04
返回	0x88	0x00	data=status[1]+string status={00 成功，01 失败，04 无权限} string={“invalid”禁止使用黑白名单，“enable”做为白名单，“disable”做为黑名单}	
发送	0x09	0x00	更改黑白名单状态 data={“invalid”禁止使用黑白名单，“enable”做为白名单，“disable”做为黑名单}	02
返回	0x89	0x00	data=status[1] status={00 成功，01 失败，03 黑白名单不启用,04 无权限}	-

名单表 json 格式如下所示：

{

“white_list”：1,

“user_list”:[{“user”：0000000000000002,

“port”：”0000000000000000 0000000000000000 0000000000000000 0000000000000002” },

{“user”：0000000000000001,

“port”：”1000000000000000 0000000000000000 0000000000000000 0000000000000002” },]

}

port 的每个位表示该是否允许此用户使用此端口，从右边开始第一个位表示 0 端口，依此类推。当 state 为 enable 时，对应端口位为 1 表示该用户可以控制此端口。state 为 disable 时对应端口位为 1 表示该用户禁止使用此端口，为 0 表示可以控制此端口。state 为 invalid 时黑白名单无效，所有用户都可以控制任意端口。

3.8，AttributeID 值为 0x00000007。设备报警记录

支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x01	0x00	读报警记录	02|03|04

			data={无}	
返回	0x81	0x00	data=status[1]+json 格式记录表 status={00 成功，01，失败，04 无权限}	-
发送	0x03	0x00	删除记录 data={无}	02
返回	0x83	0x00	data=status[1] status={00 成功，01，失败，04 无权限}	-

[{“tag”：”send”,
“date”：45616789,
“addr”：”0000000000000001”,
“cmd”：”010000000102”,
“data”：”01”},
{“tag”：”receive”,
“date”：45616789,
“addr”：”0000000000000002”,
“cmd”：”010000000102”,
“data”：”01”}]

3.9，AttributeID 值为 0x00000008。设备操作记录

支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x01	0x00	读记录 data={无}	02|03|04
返回	0x81	0x00	data=status[1]+json 格式记录表 status={00 成功，01，失败，04 无权限}	-
发送	0x03	0x00	删除记录 data={无}	02
返回	0x83	0x00	data=status[1] status={00 成功，01，失败，04 无权限}	-
发送	0x08	0x00	data={无}	02|03|04
返回	0x98	0x00	data=status[1]+command[1] status={00 成功，04 无权限} command ={00 禁止记录，01，使能记录 }	-
发送	0x09	0x00	使能设备操作记录 data={00 禁止记录，01 使能设备操作记录 }	02
返回	0x89	0x00	data=status[1] status={00 成功，01 失败，04 无权限}	-

[{“tag”：”send”,
“date”：45616789,
“addr”：”0000000000000001”,
“cmd”：”010000000102”,
“data”：”01”},
{“tag”：”receive”,

3.10，AttributeID 值为 0x0000000F。设备信标

支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x0C	0x00	读记录 data={任意数据}	02|03|04|00
返回	0x8C	0x00	data={任意数据}	-

3.11，AttributeID 值为 0x00000010。新设备加入通知

支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x0D	0x00	发送新设备加入通知 data={random_number[1]}	02|03|04|00
返回	0x8D	0x00	data={ random_number[1]}	-

3.12，AttributeID 值为 0x00000012。设备升级

支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x01	0x00	读软件版本信息 data={无}	02
返回	0x81	0x00	data={ status[1]+json 字符串} status={00 成功，01 失败，04 无权限} json:	-

			{
			“version”：1,
			“bin_name”：”ethernet_demo.bin”,
			“chip_name”：”ESP32”,
			“project_code”：”SW03-1807”,
			“VID”：1,
			“PID”：2
			}	

发送	0x02	0x00	发送升级命令
			data={json 字符串}
			json：
			{
			“state”:33,
			“try”:3,
			“server_ipv4”：”192.168.1.1”,
			“server_port”：”10080”,
			“server_url”：”/switch03/ethernet_demo.bin”
			}
				02

返回	0x82	0x00	data={status[1] status={00 成功，01 失败，04 无权限}	-
返回	0x0D	0x00	升级状态通知 data={ status[1]+percent } status={00 成功，32 停止升级，33 开始升级，34 升级完成，其它失败} percent={进度百分比}	-

3.13，AttributeID 值为 0x00000100.子设备操作

支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x02	0x00	data={enable[1]+duration[2]} enable：00 禁止添加，01 允许添加 duration：持续时间单位秒。	02
返回	0x82	0x00	data={ status[1]} status={00 成功，01 失败，04 无权限}	
新设备加入	0x0D	0x00	data={json 字符串} json： {“port”：01, “name”：”switch” ″aID″：　[256, 257, 18, 258] }	02

3.15，AttributeID 值为 0x00000102。获取设备通信信号质量

支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x01	0x00	data={空}	02|03|04|00
返回	0x81	0x00	data={ lqi[1]} lqi=0~255 信号质量	

3.16，AttributeID 值为 0x00000103。自定义数据

支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
接收	0x00	0x00	读生产商信息	必须定义
返回	0x80	0x00	vendorID[4]+version[2]（如果没有 vendorID 请使用 0xFFFFFFFF 填充）	
发送	0x00	0x00	data={空}（注:port=0x00，此命令为自定义命令，由开发者定义 cmd,opion，data 的值，通信返回方式也需自定义）	02|03|04|00

4、设备定义：

每一个芯片或处理器最大可以定义 253 个端口，每个端口只能表示为一个设备，而设备由 ApplicationID 来定义，每一个 ApplicationID 表示一种具体的设备。每个设备的通信格式与属性由 AttributeID 定义。每个 AttributeID 的具体操作与通信数据格式由 option 与 cmd 决定，不同的 AtrributeID 对 option 与 cmd 的定义可能会不一样。以下将对其进行详细描述：

4.1，ApplicationID：0x001010 开关

开关		
AttributeID		
0x009020	开关	必要属性
0x00D170	电能测量	可选属性
0x00D174	功率测量	
0x00D178	电压测量	
0x00D17C	电流测量	
0x00C130	设备低电压报警	
0x000102	通信信号质量	

4.2，ApplicationID：0x001030 插座

插座		
AttributeID		
0x009020	开关	必要属性
0x00D170	电能测量	可选属性
0x00D174	功率测量	
0x00D178	电压测量	
0x00D17C	电流测量	
0x00C130	设备低电压报警	
0x000102	通信信号质量	

4.3，ApplicationID：0x001031 排插

排插		
AttributeID		
0x009020	开关	必要属性
0x00D170	电能测量	可选属性
0x00D174	功率测量	
0x00D178	电压测量	
0x00D17C	电流测量	
0x00C130	设备低电压报警	
0x000102	通信信号质量	

4.4，ApplicationID：0x001041 窗帘

窗帘		
AttributeID		
0x009050	窗帘开关	必要属性
0x00D003	照度测量	可选属性
0x00C130	设备低电压报警	
0x000102	通信信号质量	

4.5，ApplicationID：0x001042 百叶窗帘

百叶窗帘		
AttributeID		
0x009052	百叶窗帘	必要属性

0x00D003	照度测量	可选属性
0x00C130	设备低电压报警	
0x000102	通信信号质量	

4.6，ApplicationID：0x001043 百分比窗帘

百分比窗帘		
AttributeID		
0x009051	百分比窗帘	必要属性
0x00D003	照度测量	可选属性
0x00C130	设备低电压报警	
0x000102	通信信号质量	

4.7，ApplicationID：0x001021 调光灯

调光灯		
AttributeID		
0x009030	电平等级	必要属性
0x00D170	电能测量	可选属性
0x00D174	功率测量	
0x00D178	电压测量	
0x00D17C	电流测量	
0x00C130	设备低电压报警	
0x00D003	照度测量	
0x000102	通信信号质量	

4.8，ApplicationID：0x001023 彩灯

彩灯		
AttributeID		
0x009040	RGBW 四基色	必要属性
0x009031	红色	可选属性
0x009032	绿色	
0x009033	蓝色	
0x009034	黄色	
0x00903F	白色	
0x00D170	电能测量	
0x00D174	功率测量	
0x00D178	电压测量	

0x00D17C	电流测量	
0x00C130	设备低电压报警	
0x00D003	照度测量	
0x000102	通信信号质量	

4.10，ApplicationID：0x001023 冷暖二色灯

冷暖二色灯		
AttributeID		
0x009030	电平等级	必要属性
0x009034	黄色	
0x00903F	白色	
0x00D170	电能测量	可选属性
0x00D174	功率测量	
0x00D178	电压测量	
0x00D17C	电流测量	
0x00C130	设备低电压报警	
0x00D003	照度测量	
0x000102	通信信号质量	

4.11，ApplicationID：0x001080 门锁

门锁		
AttributeID		
0x009080	门锁	必要属性
0x009081	RFID 卡	可选属性
0x009082	密码	
0x009083	指纹	
0x00C130	设备低电压报警	
0x00C100	门磁报警	
0x00C400	拆卸报警	
0x000102	通信信号质量	

4.12，ApplicationID：0x001100 红外宝

红外宝		
AttributeID		
0x009055	自有编码	必要属性

0x009056	第三方编码	可选属性
0x00C130	设备低电压报警	
0x000102	通信信号质量	

4.14，ApplicationID：0x001100 风扇

风扇		
AttributeID		
0x009060	档位 0~6 档	必要属性
0x009061	风扇转速百分比	可选属性
0x009062/0x009063	水平风向（摇头）	
0x009064/0x009065	垂直风向	
0x000102	通信信号质量	

4.15，ApplicationID：0x003001 漏电开关

漏电开关		
AttributeID		
0x009020	开关	必要属性
0x00D170	电能测量	可选属性
0x00D174	功率测量	
0x00D178	电压测量	
0x00D17C	电流测量	
0x00C130	设备低电压报警	
0x000102	通信信号质量	

4.16，ApplicationID：0x003002 空气开关

空气开关		
AttributeID		
0x009020	开关	必要属性
0x00D170	电能测量	可选属性
0x00D174	功率测量	
0x00D178	电压测量	
0x00D17C	电流测量	
0x00C130	设备低电压报警	
0x000102	通信信号质量	

4.17，ApplicationID：0x004000 火情报警器

火警报警器		
AttributeID		
0x00CFFF	安防状态通知	必要属性
0x00C000	火情报警	
0x00C400	拆卸报警	
0x00C130	低电压报警	可选属性
0x000102	通信信号质量	

4.18，ApplicationID：0x004001 烟雾报警器

烟雾报警器		
AttributeID		
0x00CFFF	安防状态通知	必要属性
0x00C001	烟雾报警	
0x00C400	拆卸报警	
0x00C130	低电压报警	可选属性
0x000102	通信信号质量	

4.19，ApplicationID：0x004002 PM2.5 报警器

烟 PM2.5 报警器		
AttributeID		
0x00CFFF	安防状态通知	必要属性
0x00C002	PM2.5 报警	
0x00C400	拆卸报警	
0x00C130	低电压报警	可选属性
0x000102	通信信号质量	

4.20，ApplicationID：0x004010 水灾报警器

水灾报警器		
AttributeID		
0x00CFFF	安防状态通知	必要属性
0x00C010	水灾报警	

0x00C400	拆卸报警	
0x00C130	低电压报警	可选属性
0x000102	通信信号质量	

4.21，ApplicationID：0x004011 水浸报警器

水浸报警器		
AttributeID		
0x00CFFF	安防状态通知	必要属性
0x00C011	水浸报警	
0x00C400	拆卸报警	
0x00C130	低电压报警	可选属性
0x000102	通信信号质量	

4.22，ApplicationID：0x004020 毒气报警器

毒气报警器		
AttributeID		
0x00CFFF	安防状态通知	必要属性
0x00C020	毒气报警	
0x00C400	拆卸报警	
0x00C130	低电压报警	可选属性
0x000102	通信信号质量	

4.23，ApplicationID：0x004021 一氧化碳报警器

一氧化碳报警器		
AttributeID		
0x00CFFF	安防状态通知	必要属性
0x00C021	一氧化碳报警	
0x00C400	拆卸报警	
0x00C130	低电压报警	可选属性
0x000102	通信信号质量	

4.24，ApplicationID：0x004030 天燃气体报警器

天燃气报警器		
AttributeID		
0x00CFFF	安防状态通知	必要属性
0x00C030	天燃气报警	
0x00C400	拆卸报警	
0x00C130	低电压报警	可选属性
0x000102	通信信号质量	

4.25，ApplicationID：0x004100 门磁报警器

门磁报警器		
AttributeID		
0x00CFFF	安防状态通知	必要属性
0x00C100	门磁报警	
0x00C400	拆卸报警	
0x00C130	低电压报警	可选属性
0x000102	通信信号质量	

4.26，ApplicationID：0x004110 人体感应报警器

人体感应报警器		
AttributeID		
0x00CFFF	安防状态通知	必要属性
0x00C110	人体感应报警	
0x00C400	拆卸报警	
0x00C130	低电压报警	可选属性
0x000102	通信信号质量	

4.27，ApplicationID：0x004120 紧急按钮

紧急按钮		
AttributeID		
0x00CFFF	安防状态通知	必要属性
0x00C120	紧急按钮报警	
0x00C400	拆卸报警	
0x00C130	低电压报警	可选属性

0x000102	通信信号质量	

4.28，ApplicationID：0x005000 温湿度传感

温度传感		
AttributeID		
0x00D000	温度测量	必要属性
0x00D001	相对湿度测量	
0x00C130	低电压报警	可选属性
0x000102	通信信号质量	

4.29，ApplicationID：0x005001 温度传感

温度传感		
AttributeID		
0x00D000	温度测量	必要属性
0x00D001	相对湿度测量	可选属性
0x00C130	低电压报警	
0x000102	通信信号质量	

4.30，ApplicationID：0x005002 相对湿度传感

相对湿度传感		
AttributeID		
0x00D001	相对湿度测量	必要属性
0x00D000	温度测量	可选属性
0x00C130	低电压报警	
0x000102	通信信号质量	

4.31，ApplicationID：0x005130 亮度传感

亮度传感		
AttributeID		
0x00D003	亮度测量	必要属性
0x00C130	低电压报警	可选属性
0x000102	通信信号质量	

4.32，ApplicationID：0x005170 电能表

电能表		
AttributeID		
0x00D170	电能测量	必要属性
0x00D174	功率测量	可选属性
0x00D178	电压测量	
0x00D17C	电流测量	
0x00D180	频率测量	
0x00C130	低电压报警	
0x000102	通信信号质量	

4.33，ApplicationID：0x004FFE 报警器(警笛)

亮度传感		
AttributeID		必要属性
0x00CFFD	警灯	
0x00CFFE	鸣警	
0x00CFFF	安防状态通知	可选属性
0x00C400	拆卸报警	
0x00C130	低电压报警	
0x000102	通信信号质量	

4.34，ApplicationID：0x004FFF 安防设备

报警器		
AttributeID		
0x00CFFF	安防状态通知	必要属性
0x00C400	拆卸报警	
0x00C130	低电压报警	可选属性
0x000102	通信信号质量	

5、AttributeID 定义：

5.1，AttributeID 值为 0x009000 按键

0x009000 支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x80	0x00	上报按键状态 data={key[1]+shift[1]+duration[2]} key：按键值，取值 1~255。 shift：按下状态,00 无效，01 按下，02 保持，03 弹起。 duration：按下持续时间单位毫秒。	02|03|04|00
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.2，AttributeID 值为 0x009020 开关

0x009020 支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x80	0x00	上报开关状态 data={ status[1]} status={00 关，01 开}	
接收	0x01	0x00	读开关状态 data={空}	02|03|04|00
返回	0x81	0x00	data={ status[1]} status={00 关，01 开}	-
接收	0x02	0x00	写开关状态	02|03|04|00
返回	0x82	0x00	data={ status[1]} status={00 关，01 开}	-
接收	0x04	0x00	翻转开关 data={空}	02|03|04|00
返回	0x84	0x00	data={ status[1]} status={00 关，01 开}	-
接收	0x40	0x00	读生产商信息	支持个性化
返回	0xC0	0x00	vendorID[4]+version[2]	的功能的设备这

接收	0x41	0x00	读自定义数据	几条命令必须同时实现
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.3，AttributeID 值为 0x009030 电平等级

0x009030 支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x80	0x00	上报电平等级 data={ status[1]} status={00 关，取值 01~255}	
接收	0x80	0x00	上报开关状态 data={ status[1]} status={00 关，01 开}	02|03|04|00
返回	0x81	0x00	data={ level[1]} level ={00 关，取值 01~255 }	-
接收	0x02	0x00	写入电平等级 data={ level[1]} level ={00 关，取值 01~255 }	02|03|04|00
返回	0x82	0x00	data={ level[1]} level ={00 关，取值 01~255}	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.4，AttributeID 值为 0x009031 红色

0x009031 支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x80	0x00	上报颜色等级 data={ red_color [1]} red_color ={00 关，取值 01~255}	

接收	0x01	0x00	读颜色等级 data={空}	02|03|04|00
返回	0x81	0x00	data={ red_color[1]} red_color ={00 关，取值 01~255 }	-
接收	0x02	0x00	写入颜色等级 data={ red_color [1]} red_color ={00 关，取值 01~255 }	02|03|04|00
返回	0x82	0x00	data={ red_color [1]} red_color ={00 关，取值 01~255}	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.5，AttributeID 值为 0x009032 绿色

0x009032 支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x80	0x00	上报颜色等级 data={ color [1]} color ={00 关，取值 01~255}	
接收	0x01	0x00	读颜色等级 data={空}	02|03|04|00
返回	0x81	0x00	data={ green_color[1]} green_color ={00 关，取值 01~255 }	-
接收	0x02	0x00	写入颜色等级 data={ green_color [1]} green_color ={00 关，取值 01~255 }	02|03|04|00
返回	0x82	0x00	data={ green_color [1]} green_color ={00 关，取值 01~255}	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据	

			data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.6，AttributeID 值为 0x009033 蓝色

0x009033 支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x80	0x00	上报颜色等级 data={ color [1]} color ={00 关，取值 01~255}	
接收	0x01	0x00	读颜色等级 data={空}	02|03|04|00
返回	0x81	0x00	data={ blue_color[1]} blue_color ={00 关，取值 01~255 }	-
接收	0x02	0x00	写入颜色等级 data={ blue_color [1]} blue_color ={00 关，取值 01~255 }	02|03|04|00
返回	0x82	0x00	data={ blue_color [1]} blue_color ={00 关，取值 01~255}	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.7，AttributeID 值为 0x009034 黄色

0x009034 支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x80	0x00	上报颜色等级 data={ color [1]} color ={00 关，取值 01~255}	
接收	0x01	0x00	读颜色等级 data={空}	02|03|04|00
返回	0x81	0x00	data={ yellow_color[1]} yellow _color ={00 关，取值 01~255 }	-

接收	0x02	0x00	写入颜色等级 data={ yellow _color [1]} yellow _color ={00 关，取值 01~255 }	02|03|04|00
返回	0x82	0x00	data={ yellow_color [1]} yellow_color ={00 关，取值 01~255}	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.8，AttributeID 值为 0x00903F 白色

0x00903F 支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x80	0x00	上报颜色等级 data={ color [1]} color ={00 关，取值 01~255}	
接收	0x01	0x00	读颜色等级 data={空}	02|03|04|00
返回	0x81	0x00	data={ white[1]} white ={00 关，取值 01~255 }	-
接收	0x02	0x00	写入颜色等级 data={ white[1]} white ={00 关，取值 01~255 }	02|03|04|00
返回	0x82	0x00	data={white [1]} white ={00 关，取值 01~255}	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.9，AttributeID 值为 0x009040 彩色

0x009040 支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x80	0x00	上报颜色等级 data={ Red[1]+Green[1]+Blue[1]+White[1]} Red ={00 关，取值 01~255 } Green ={00 关，取值 01~255 } Blue ={00 关，取值 01~255 } White ={00 关，取值 01~255 }	
接收	0x01	0x00	读颜色 data={空}	02|03|04|00
返回	0x81	0x00	data={ Red[1]+Green[1]+Blue[1]+White[1]} Red ={00 关，取值 01~255 } Green ={00 关，取值 01~255 } Blue ={00 关，取值 01~255 } White ={00 关，取值 01~255 }	-
接收	0x02	0x00	写入颜色等级 data={ Red[1]+Green[1]+Blue[1]+White[1]} Red ={00 关，取值 01~255 } Green ={00 关，取值 01~255 } Blue ={00 关，取值 01~255 } White ={00 关，取值 01~255 }	02|03|04|00
返回	0x82	0x00	data={ Red[1]+Green[1]+Blue[1]+White[1]} Red ={00 关，取值 01~255 } Green ={00 关，取值 01~255 } Blue ={00 关，取值 01~255 } White ={00 关，取值 01~255 }	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.10，AttributeID 值为 0x009050 窗帘开关状态

0x009050 支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	

发送	0x80	0x00	上报开关状态 data={ status[1]} status：01 开，02 关，00 停止	
发送	0x80	0x01	上报百分比 data={ status[1]} status：01 开，02 关，00 停止	
接收	0x01	0x00	读状态 data={空}	02|03|04|00
返回	0x81	0x00	data={ status[1]} status：01 开，02 关，00 停止	-
接收	0x02	0x00	写入状态 data={ status[1]} status：01 开，02 关，00 停止	02|03|04|00
返回	0x82	0x00	data={ status[1]} status：01 开，02 关，00 停止	-
接收	0x03	0x00	data=[status+duaration[4]] status：01 开，02 关，00 停止 duaration：持续时间，单位毫秒	02|03|04|00
返回	0x83	0x00	data={ status[1]} status：01 失败，00 成功	返回
接收	0x01	0x01	读百分比 data={空}	
返回	0x81	0x01	data={ percent[1]} percent：0~100 百分比	
接收	0x02	0x01	写入百分比 data={ percent[1]} percent：0~100 百	02|03|04|00
返回	0x82	0x01	data={ percent[1]} percent：0~100 百	-
接收	0x0A	0x00	读行进方向 data={空}	02|03|04|00
返回	0x8A	0x00	data={status[1]} status：00 正方向行进，01 反方向行进	-
接收	0x0B	0x00	设置行进方向 data={status[1]} status：00 正方向行进，01 反方向行进	02
返回	0x8B	0x00	data={status[1]} status：00 正方向行进，01 反方向行进	-
接收	0x0C	0x00	读预设总行程时间 data={空}	02|03|04|00
返回	0x8C	0x00	data={time[4]}	-

			time：总行程时长，单位：毫秒	
接收	0x0D	0x00	设置预设总行程时间 data={time[4]} time：总行程时长，单位：毫秒	02
返回	0x8D	0x00	data={status[1]} status：00 正方向行进，01 反方向行进	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.11，AttributeID 值为 0x009051 窗帘百分比

0x009051 支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x80	0x00	上报百分比 data={ status[1]} status：01 开，02 关，00 停止	
接收	0x01	0x00	读百分比 data={空}	02|03|04|00
返回	0x81	0x00	data={ percent[1]} percent：00 关，1~255 开百分比	-
接收	0x02	0x00	写入百分比 data={ percent[1]} percent：00 关，1~255 开百分比	02|03|04|00
返回	0x82	0x00	data={ percent[1]} percent：00 关，1~255 开百分比	-
接收	0x0A	0x00	读行进方向 data={空}	02|03|04|00
返回	0x8A	0x00	data={status[1]} status：00 正方向行进，01 反方向行进	-
接收	0x0B	0x00	data={status[1]} status：00 正方向行进，01 反方向行进	02
返回	0x8B	0x00	data={status[1]} status：00 正方向行进，01 反方向行进	-
接收	0x40	0x00	读生产商信息	支持个性化

返回	0xC0	0x00	vendorID[4]+version[2]	的功能的设备这几条命令必须同时实现
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.12，AttributeID 值为 0x009052 百叶窗帘

0x009052 支持的命令与选项				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x80	0x00	上报百分比 data={ percent[1]+angle[1]} percent：00 关，1~255 开百分比 angle[1]：百叶角度，取值 0~180	
接收	0x01	0x00	读百分比 data={空}	02|03|04|00
返回	0x81	0x00	data={ percent[1]+angle[1]} percent：00 关，1~255 开百分比 angle[1]：百叶角度，取值 0~180	-
接收	0x02	0x00	写入百分比 data={ percent[1]+angle[1]} percent：00 关，1~255 开百分比 angle[1]：百叶角度，取值 0~180	02|03|04|00
返回	0x82	0x00	data={ percent[1]+angle[1]} percent：00 关，1~255 开百分比 angle[1]：百叶角度，取值 0~180	-
接收	0x0A	0x00	读行进方向 data={空}	02|03|04|00
返回	0x8A	0x00	data={status[1]} status：00 正方向行进，01 反方向行进	-
接收	0x0B	0x00	data={status[1]} status：00 正方向行进，01 反方向行进	02
返回	0x8B	0x00	data={status[1]} status：00 正方向行进，01 反方向行进	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	

接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.13，AttributeID 值为 0x009055 吉风红外编码

0x009055 吉风红外编码				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x80	0x00	上报编码 data={ code[n]} code：红外编码。	
接收	0x01	0x00	学习红外编码 data={空}	02|03|04|00
返回	0x81	0x00	data={status[1]+code[n]} status：00 成功，1 失败 code：红外编码。	-
接收	0x02	0x00	发送编码 data={ code[n]} code：红外编码。	02|03|04|00
返回	0x82	0x00	data={ status[1]} status：状态 0 成功，1 失败	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.14，AttributeID 值为 0x009056 第三方红外编码

0x009056 第三方红外编码				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x80	0x00	上报编码 data={ code[n]} code：红外编码。	
接收	0x01	0x00	学习红外编码 data={空}	02|03|04|00
返回	0x81	0x00	data={status[1]+code[n]}	-

			status：00 成功，1 失败 code：红外编码。	
接收	0x02	0x00	发送编码 data={ code[n]} code：红外编码。	02|03|04|00
返回	0x82	0x00	data={ status[1]} status：状态 0 成功，1 失败	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.15，AttributeID 值为 0x009060 风扇

0x009060 风扇				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x80	0x00	上报状态 data={level[1]+mode[1]} level：00-关，档位，1~6 档。 mode：自然风，00 关，1~6 级。	
接收	0x01	0x00	读风扇状态 data={空}	02|03|04|00
返回	0x81	0x00	data={level[1]+mode[1]} level：00-关，档位，1~6 档。 mode：自然风，00 关，1~6 级。	-
接收	0x02	0x00	控制 data={level[1]+mode[1]} level：00-关，档位，1~6 档。 mode：自然风，00 关，1~6 级。	02|03|04|00
返回	0x82	0x00	data={level[1]+mode[1]} level：00-关，档位，1~6 档。 mode：自然风，00 关，1~6 级。	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	

接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.16，AttributeID 值为 0x009061 无级调速风扇

0x009061 无级调速风扇				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x80	0x00	上报状态 data={level[1]+mode[1]} level：00-关，风速 1~255。 mode：自然风，00 关，1~6 级。	
接收	0x01	0x00	读风扇状态 data={空}	02|03|04|00
返回	0x81	0x00	data={level[1]+mode[1]} level：00-关，风速 1~255。 mode：自然风，00 关，1~6 级。	-
接收	0x02	0x00	控制 data={level[1]+mode[1]} level：00-关，风速 1~255。 mode：自然风，00 关，1~6 级。	02|03|04|00
返回	0x82	0x00	data={level[1]+mode[1]} level：00-关，风速 1~255。 mode：自然风，00 关，1~6 级。	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.17，AttributeID 值为 0x009062 水平风向(风扇、空调)

0x009062 水平风向				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x80	0x00	上报状态 data={level[1]} level:0 关，1:LL，2:L,3:M,4:H,5:HH	

接收	0x01	0x00	读风向 data={空}	02|03|04|00
返回	0x81	0x00	data={level[1]} 1、 level:0 关，1:LL，2:L,3:M,4:H,5:HH	-
接收	0x02	0x00	data={level[1]} 1、 level:0 关，1:LL，2:L,3:M,4:H,5:HH	02|03|04|00
返回	0x82	0x00	data={level[1]} 1、 level:0 关，1:LL，2:L,3:M,4:H,5:HH	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定义数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.18，AttributeID 值为 0x009064 垂直风向(风扇、空调)

0x009064 垂直风向				操作权限
	cmd	option	data[x]的数据格式与描述	
	cmd	option	data[x]的数据格式与描述	
发送	0x80	0x00	上报状态 data={level[1]} level:0 关，1:LL，2:L,3:M,4:H,5:HH	
接收	0x01	0x00	读风向 data={空}	02|03|04|00
返回	0x81	0x00	data={level[1]} 2、 level:0 关，1:LL，2:L,3:M,4:H,5:HH	-
接收	0x02	0x00	data={level[1]} 2、 level:0 关，1:LL，2:L,3:M,4:H,5:HH	02|03|04|00
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定义数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.19，AttributeID 值为 0x009080 门锁

0x009080 门锁				操作权限
	cmd	option	data[x]的数据格式与描述	
发送	0x80	0x00	上报状态 data={type[1]+lock_mask[1]+menci_flag[1]} type:开锁方式：0 无效，1 密码，2 指纹，3 RFIDCard,4 远程。 lock_mask：锁标志 00 开，bit1 上锁，bit2 反锁,bit3 安全锁。 menci_flag：门磁 01 门闭合，00 门打开。	
接收	0x01	0x00	读锁状态 data={空}	02|03|04|00
返回	0x81	0x00	data={type[1]+lock_mask[1]+menci_flag[1]} type:开锁方式：0 无效，1 密码，2 指纹，3 RFIDCard,4 远程。 lock_mask：锁标志 00 开，bit1 上锁，bit2 反锁,bit3 安全锁。 menci_flag：门磁 01 门闭合，00 门打开。	-
接收	0x02	0x00	控制锁 data={ lock_mask[1]} lock_mask：锁标志 00 开，bit1 上锁，bit2 反锁,bit3 安全锁。	02|03|04|00
返回	0x82	0x00	data={type[1]+lock_mask[1]+menci_flag[1]} type:开锁方式：0 无效，1 密码，2 指纹，3 RFIDCard,4 远程。 lock_mask：锁标志 00 开，bit1 上锁，bit2 反锁,bit3 安全锁。 menci_flag：门磁 01 门闭合，00 门打开。	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.20，AttributeID 值为 0x009081 门锁-RFID 卡操作

0x009081 门锁-RFID				操作权限
	cmd	option	data[x]的数据格式与描述	

发送	0x80	0x00	上报刷卡信息
			data={JSON 字符串 }
			JSON:
			{
			“user”:”xxx”,
			}	

接收	0x01	0x00	读 RFID 卡数量 data={空}	02
返回	0x81	0x00	data=num[2] num：RFID 卡数量	-
接收	0x02	0x00	读 RFID 卡信息 data=startID[2]+endID[2]。	02

返回	0x82	0x00	data={status[1]+ID[2]+JSON 字符串 }
			status：00 成功，01 失败。
			ID：返回第几个 RFID 卡。
			JSON：
			{
			“UID”:”xxxx”,
			“data”:”xxx”,
			“user”:”xxx”,
			“date”：“yy-mm-dd,hh:mm:ss”
			“expire”:xxx,
			“permit”:xx
			}
				-

接收	0x03	0x00	以 ID 寻址方式删除 RFID 卡 data={startID[2]+endID[2]}	02
返回	0x83	0x00	data=status[1] status：00 成功，01 失败。	-
接收	0x04	0x00	以 user 方式删除 RFID 卡 data={user[n]} user ：用户名字符串	02
返回	0x84	0x00	data=status[1] status：00 成功，01 失败。	-

接收	0x05	0x00	添加 RFID 卡
			data={JSON 字符串 }
			JSON：
			{
			“UID”:”xxxx”,
			“data”:”xxx”,
			“user”:”xxx”,
			“date”：“yy-mm-dd,hh:mm:ss”
			“expire”:xxx,
			“permit”:xx
			}
				02

返回	0x85	0x00	data=status[1] status：00 成功，01 失败。	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	

返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	时实现
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.21，AttributeID 值为 0x009082 门锁-密码操作

0x009082 门锁密码操作				操作权限
	cmd	option	data[x]的数据格式与描述	

发送	0x80	0x00	上报输入密码
			data={JSON 字符串 }
			JSON:
			{
			“user”:”xxx”,
			}	

接收	0x01	0x00	读密码数量 data={空}	02
返回	0x81	0x00	data=num[2] num：密码数量	-
接收	0x02	0x00	读密码信息 data=startID[2]+endID[2]。	02

返回	0x82	0x00	data={status[1]+ID[2]+JSON 字符串 }
			status：00 成功，01 失败。
			ID：返回第几个用户密码。
			JSON:
			{
			“psw”:”xxxx”,
			“user”:”xxx”,
			“date”：“yy-mm-dd,hh:mm:ss”
			“expire”:xxx,
			“permit”:xx
			}
				-

接收	0x03	0x00	以 ID 寻址方式删除密码 data={startID[2]+endID[2]}	02
返回	0x83	0x00	data=status[1] status：00 成功，01 失败。	-
接收	0x04	0x00	以 user 方式删除密码 data={user[n]} user ：用户名字符串	02
返回	0x84	0x00	data=status[1]	-

			status：00 成功，01 失败。	

接收	0x05	0x00	添加密码
			data={JSON 字符串 }
			JSON：
			{
			“psw”:”xxxx”,
			“user”:”xxx”,
			“date”：“yy-mm-dd,hh:mm:ss”
			“expire”:xxx,
			“permit”:xx
			}
				02

返回	0x85	0x00	data=status[1] status：00 成功，01 失败。	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.22，AttributeID 值为 0x009083 门锁-指纹操作

0x009083 门锁指纹操作				操作权限
	cmd	option	data[x]的数据格式与描述	

发送	0x80	0x00	上报输入指纹
			data={JSON 字符串 }
			JSON：
			{
			“user”:”xxx”,
			}	

接收	0x01	0x00	读指纹数量 data={空}	02
返回	0x81	0x00	data=num[2] num：指纹数量	-
接收	0x02	0x00	读指纹信息 data=startID[2]+endID[2]。	02
返回	0x82	0x00	data={status[1]+ID[2]+JSON 字符串 } status：00 成功，01 失败。 ID：返回第几个用户指纹。	-

			JSON:
			{
			“fingerprint”:”xxxx”,
			“user”:”xxx”,
			“date”:“yy-mm-dd,hh:mm:ss”
			“expire”:xxx,
			“permit”:xx
			}	

接收	0x03	0x00	以 ID 寻址方式删除指纹 data={startID[2]+endID[2]}	02
返回	0x83	0x00	data=status[1] status：00 成功，01 失败。	-
接收	0x04	0x00	以 user 方式删除指纹 data={user[n]} user ：用户名字符串	02
返回	0x84	0x00	data=status[1] status：00 成功，01 失败。	-

接收	0x05	0x00	添加指纹
			data={JSON 字符串 }
			JSON:
			{
			“fingerprint”:”xxxx”,
			“user”:”xxx”,
			“date”:“yy-mm-dd,hh:mm:ss”
			“expire”:xxx,
			“permit”:xx
			}
				02

返回	0x85	0x00	data=status[1] status：00 成功，01 失败。	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.23，AttributeID 值为 0x00C000 报警-火警

0x00C000 报警-火警	操作权限

	cmd	option	data[x]的数据格式与描述	
返回	0x85	0x00	data=status status:00 取消报警 status:01 报警	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.24，AttributeID 值为 0x00C001 报警-烟雾

0x00C001 报警-烟雾				操作权限
	cmd	option	data[x]的数据格式与描述	
返回	0x85	0x00	data=status status:00 取消报警 status:01 报警	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.25，AttributeID 值为 0x00C002 报警-PM2.5

0x00C002 报警-PM2.5				操作权限
	cmd	option	data[x]的数据格式与描述	
返回	0x85	0x00	data=status status:00 取消报警 status:01 报警	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据	

			data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.26，AttributeID 值为 0x00C010 报警-水灾

0x00C010 报警-水灾				操作权限
	cmd	option	data[x]的数据格式与描述	
返回	0x85	0x00	data=status status:00 取消报警 status:01 报警	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.27，AttributeID 值为 0x00C011 报警-水浸

0x00C011 报警-水灾				操作权限
	cmd	option	data[x]的数据格式与描述	
返回	0x85	0x00	data=status status:00 取消报警 status:01 报警	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.28，AttributeID 值为 0x00C020 报警-毒气

0x00C020 报警-水灾				操作权限
	cmd	option	data[x]的数据格式与描述	
返回	0x85	0x00	data=status status:00 取消报警 status:01 报警	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.29，AttributeID 值为 0x00C021 报警-一氧化碳

0x00C021 报警-一氧化碳				操作权限
	cmd	option	data[x]的数据格式与描述	
返回	0x85	0x00	data=status status:00 取消报警 status:01 报警	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.30，AttributeID 值为 0x00C030 报警-可燃气体

0x00C030 报警-可燃气体			
	cmd	option	data[x]的数据格式与描述
返回	0x85	0x00	data=status status:00 取消报警 status:01 报警

接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.31，AttributeID 值为 0x00C100 报警-门磁

0x00C100 报警-门磁				
	cmd	option	data[x]的数据格式与描述	
返回	0x85	0x00	data=status status:00 取消报警 status:01 报警	
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.32，AttributeID 值为 0x00C110 报警-人体红外感应

0x00C110 报警-人体红外感应				
	cmd	option	data[x]的数据格式与描述	
返回	0x85	0x00	data=status status:00 取消报警 status:01 报警	
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.33，AttributeID 值为 0x00C120 报警-紧急按钮

0x00C120 报警-紧急按钮				
	cmd	option	data[x]的数据格式与描述	
返回	0x85	0x00	data=status status:00 取消通知 status:01 通知	
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.34，AttributeID 值为 0x00C130 报警-低电压

0x00C120 报警-低电压				
	cmd	option	data[x]的数据格式与描述	
返回	0x85	0x00	status:00 取消通知 status:01 通知	
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.35，AttributeID 值为 0x00C400 报警-拆卸

0x00C120 报警-拆卸			
	cmd	option	data[x]的数据格式与描述
返回	0x85	0x00	data=status status:00 取消通知

			status:01 通知	
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.36，AttributeID 值为 0x00CFFD 报警-警灯

0x00CFFD 报警-警灯				操作权限
	cmd	option	data[x]的数据格式与描述	
接收	0x01	0x00	读状态 data=空	02|03|04|00
返回	0x81	0x00	data=mode[1] mode：模式，0 关，1 出警，2，急救，3 防空	-
接收	0x02	0x00	data=mode[1] mode：模式，0 关，1 出警，2，急救，3 防空	02|03|04|00
返回	0x82	0x00	data=mode[1] mode：模式，0 关，1 出警，2，急救，3 防空	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.37，AttributeID 值为 0x00CFFE 报警-警笛

0x00CFFE 报警-警笛				操作权限
	cmd	option	data[x]的数据格式与描述	
接收	0x01	0x00	读状态 data=空	02|03|04|00
返回	0x81	0x00	data=mode[1]+level[1]	-

			mode：模式，0 默认，1 出警，2，急救，3 防空 level：音量，0 关，1 小，2，中，3 大	
接收	0x02	0x00	data=mode[1]+level[1] mode：模式，0 默认，1 出警，2，急救，3 防空 level：音量，0 关，1 小，2，中，3 大	02|03|04|00
返回	0x82	0x00	data=mode[1]+level[1] mode：模式，0 默认，1 出警，2，急救，3 防空 level：音量，0 关，1 小，2，中，3 大	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.38，AttributeID 值为 0x00CFFF 安防报警

0x00CFFF 安防报警				操作权限
	cmd	option	data[x]的数据格式与描述	
返回	0x85	0x00	data=status status:00 取消报警 status:01 报警	-
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.39，AttributeID 值为 0x00D000 温度

0x00D000 温度				操作权限
	cmd	option	data[x]的数据格式与描述	
接收	0x01	0x00	读温度 data=空	02|03|04|00
返回	0x81	0x00	data=temp	-

			temp:温度数据，浮点型字符串，单位：摄氏度。如：12.36	
发送	0x80	0x00	上报温度变更 data=temp temp:温度数据，浮点型字符串，单位：摄氏度。如：12.36	-
接收	0x0A	0x00	读采集速率 -	
返回	08A	0x00	读采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x0B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
返回	08B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.40，AttributeID 值为 0x00D001 相对湿度

0x00D001 湿度				操作权限
	cmd	option	data[x]的数据格式与描述	
接收	0x01	0x00	读湿度 data=空	02|03|04|00
返回	0x81	0x00	data=value value:湿度数据，浮点型字符串，单位：RH%。如：12.36	-
发送	0x80	0x00	上报湿度变更 data= value value:湿度数据，浮点型字符串，单位：RH%。如：12.36	-
接收	0x0A	0x00	读采集速率 -	
返回	08A	0x00	读采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x0B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
返回	08B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x40	0x00	读生产商信息	支持个性化

返回	0xC0	0x00	vendorID[4]+version[2]	的功能的设备这几条命令必须同时实现
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.41，AttributeID 值为 0x00D003 光照度

0x00D003 光照度				操作权限
	cmd	option	data[x]的数据格式与描述	
接收	0x01	xx	读照度 data=空 opiont={00:单位为 lex,01:单位为 lm}	02|03|04|00
返回	0x81	xx	data= value(ascii code) opiont={00:单位为 lex,01:单位为 lm} value:照度数据，浮点型字符串，单位：lex。如：12.36	-
发送	0x80	xx	上报照度变更 data= value(ascii code) opiont={00:单位为 lex,01:单位为 lm} value:照度数据，浮点型字符串，如：12.36	-
接收	0x0A	0x00	读采集速率 -	
返回	08A	0x00	读采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x0B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
返回	08B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.42，AttributeID 值为 0x00D105 压力

0x00D105 压力				操作权限
	cmd	option	data[x]的数据格式与描述	
接收	0x01	xx	读压力 data=空	02|03|04|00
返回	0x81	xx	data= value(ascii code) option ={0:Pa,1:KPa,2:MPa,3:GPa} value:压力数据，浮点型字符串，如：12.36	-
发送	0x80	xx	上报压力变更 data= unit[1]+value(ascii code) option ={0:Pa,1:KPa,2:MPa,3:GPa} value:压力数据，浮点型字符串，如：12.36	-
接收	0x0A	0x00	读采集速率 -	
返回	08A	0x00	读采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x0B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
返回	08B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.43，AttributeID 值为 0x00D160 流体计量(m^3)

0x00D160 流体计量				操作权限
	cmd	option	data[x]的数据格式与描述	
接收	0x01	xx	读流量 data=空	02|03|04|00
返回	0x81	xx	data= unit[1]+value(ascii code) option={0:mL,1:L,2:m^3,3:Km^3} value:流量数据，浮点型字符串，如：12.36	-
发送	0x80	xx	上报流量变更 option={0:mL,1:L,2:m^3,3:Km^3}	-

			value:流量数据，浮点型字符串,如：12.36	
接收	0x0A	0x00	读采集速率 -	
返回	08A	0x00	读采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x0B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
返回	08B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.44，AttributeID 值为 0x00D170 电能计量(kw*h)

0x00D170 电能计量				操作权限
	cmd	option	data[x]的数据格式与描述	
接收	0x01	0x00	读 data=空	02|03|04|00
返回	0x81	0x00	data= value value:能耗数据，浮点型字符串，单位：kw*h。如：12.36	-
发送	0x80	0x00	上报能耗变更 data= value value: 能耗数据，浮点型字符串，单位：kw*h。如：12.36	-
接收	0x0A	0x00	读采集速率 -	
返回	08A	0x00	读采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x0B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
返回	08B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据	

			data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.45，AttributeID 值为 0x00D174 功率计量(W)

0x00D174 功率计量				操作权限
	cmd	option	data[x]的数据格式与描述	
接收	0x01	xx	读 data=空	02|03|04|00
返回	0x81	xx	data= unit[1]+value(ascii code) option={0:W,1:mW,2:uW,3:nW,4:pW} value:功率数据，浮点型字符串，如：12.36	-
发送	0x80	xx	上报功率变更 data= unit[1]+value(ascii code) option={0:W,1:mW,2:uW,3:nW,4:pW} value: 功率数据，浮点型字符串，如：12.36	-
接收	0x0A	0x00	读采集速率 -	
返回	08A	0x00	读采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x0B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
返回	08B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.46，AttributeID 值为 0x00D178 电压计量(V)

0x00D178 电压计量				操作权限
	cmd	option	data[x]的数据格式与描述	
接收	0x01	xx	读	02|03|04|00

			data=空	
返回	0x81	xx	data= unit[1]+value(ascii code) option={0:V,1:mV,2:uV,3:nV,4:pV} value:电压数据，浮点型字符串，如：12.36	-
发送	0x80	xx	上报电压 data= unit[1]+value(ascii code) option={0:V,1:mV,2:uV,3:nV,4:pV} value: 电压数据，浮点型字符串，如：12.36	-
接收	0x0A	0x00	读采集速率 -	
返回	08A	0x00	读采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x0B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
返回	08B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.47，AttributeID 值为 0x00D17C 电流计量(A)

0x00D17C 电流计量				操作权限
	cmd	option	data[x]的数据格式与描述	
接收	0x01	xx	读 data=空	02|03|04|00
返回	0x81	xx	data= unit[1]+value(ascii code) option={0:A,1:mA,2:uA,3:nA,4:pA} value:电流数据，浮点型字符串，如：12.36	-
发送	0x80	xx	上报电流 data= unit[1]+value(ascii code) option={0:A,1:mA,2:uA,3:nA,4:pA} value: 电流数据，浮点型字符串，如：12.36	-
接收	0x0A	0x00	读采集速率 -	
返回	08A	0x00	读采集速率	

			data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x0B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
返回	08B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据 data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	

5.48，AttributeID 值为 0x00D180 频率计量(A)

0x00D180 频率计量				操作权限
	cmd	option	data[x]的数据格式与描述	
接收	0x01	xx	读 data=空	02|03|04|00
返回	0x81	xx	data= unit[1]+value(ascii code) option={0:Hz,1:KHz,2:MHz,3,GHz} value:频率数据，浮点型字符串。如：12.36	-
发送	0x80	xx	上报频率 data= unit[1]+value(ascii code) option={0:Hz,1:KHz,2:MHz,3,GHz} value: 频率数据，浮点型字符串，如：12.36	-
接收	0x0A	0x00	读采集速率 -	
返回	08A	0x00	读采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x0B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
返回	08B	0x00	调整采集速率 data[1]={0:默认速率，1~9 级，10 自动}	
接收	0x40	0x00	读生产商信息	支持个性化的功能的设备这几条命令必须同时实现
返回	0xC0	0x00	vendorID[4]+version[2]	
接收	0x41	0x00	读自定义数据	
返回	0xC1	0x00	自定义数据 data[x]={生产商自定义个性化数据}	
接收	0x42	0x00	写入自定数据	

			data[x]={生产商自定义个性化数据}	
返回	0xC2	0x00	state[1]={0:成功，1:失败}	