★ 内部资料，敬请保密 ★

康泰物联网项目
内层控制协议

版本 1.1Beta

浙江欧佰信息技术有限公司　技术部

版本变更说明

版本号	发布时间	变更说明
1.0Beta	2014-10-31	1.0Beta 版
1.1Beta	2015-06-11	1、修改 wifi 定时，分离出新的 wifi 倒计时功能 2、新增 RF 定时的功能

1、协议概述

（1）该文档描述了康泰物联网项目的内层控制协议。项目开发的时候，请结合《康泰物联网系统外层协议说明》进行开发。

（2）该文档描述的控制协议，均与特定的 WiFi 设备相关。因此，发送该命令时，帧头中的 Mac 地址，应该使用相对应 WiFi 设备的 Mac 地址。帧头中的厂家代码、设备类型、授权码，应该使用从设备获取到的信息。

2、对若干数据的定义

若干数据定义如下：

（1）设备类型

对于设备——保密数据，另行告知。

对于 APP——保密数据，另行告知。

（2）厂家代码和授权码

保密数据，另行告知。

（3）L 口 UDP 端口

28530

（4）T 口负载均衡服务器

cloud.kangtai.com.cn:27531

（5）U 口负载均衡服务器

cloud.kangtai.com.cn:29531

3、插座控制协议说明

3.1 0x01 设置 GPIO 状态(T: TCP | L: UDP | U: TCP)

Request: | 01 | Pin |

Response: | 01 | Pin|

参数说明：

Pin：4 - Byte，指定引脚的功能&状态

命令说明：

一个 WiFi 模块可以有 N 个 GPIO 口。如果只接 1 个引脚，则为单孔插座；如果接 2 个引脚，则为 2 孔的插线板；如果接 4 个引脚，则为 4 孔的插线板，以此类推。该命令由服务器或手机 APP 主动发起，可同时设置多个 GPIO 口，每个 GPIO 设置

占用 4 - Byte。
Device 或 Server 收到请求后，回复设置后的 GPIO 状态，并且只发送请求中设置的 GPIO，其余 GPIO 不返回。
4 字节 Pin 的结构体如下：

```
typedef struct pin_struct{
 unsigned char flag;
 unsigned char fre;   //固定为 0x00
 unsigned char duty;   //输出高电平为 0XFF，低电平为 0X00
 unsigned char res;  //保留字节，固定为 FF
}pin_struct;
```

其中 pin_struct.flag 的含义如下：
Bit7~6 预留。
Bit5 目前请置 0。
低五位（Bit4~Bit0）表示 GPIO 序号。
例如：
flag=00 0 00000 表示：第 0 个插孔；
flag=00 0 00001 表示：第 1 个插孔；
flag=00 0 00011 表示：第 3 个插孔；

3.2 0x02 查询 GPIO 状态(T: TCP | L: UDP | U: TCP)

Request: | 02 | Pin |
Response: | 02 | Pin |
参数说明：
参见 3.1 节 0x01 设置 GPIO 状态
命令说明：
请求包仅需设置 pin_struct.flag 的 Bit4~0，即仅设置 IO 号，其余内容清零。
响应包仅需回复请求包中指定的 Pin，若请求包未指定 Pin，则全部返回。

3.3 0x03 设置定时操作(T: TCP | L: UDP | U: TCP)

Request: | 03 |Pin_num| Num | Flag | Hour | Min |pin |
Response: | 03 | Pin_num|Num |
参数说明：

Pin_num：1 - Byte 引脚编号，固定为 0x00
Num：1 - Byte，定时任务序号，取值范围为 1~10。
Flag：1 - Byte，定时任务标志。

Bit7 为定时任务状态（1~开启/0~关闭），若单次定时事件触发，则将对应定时任务的 Bit7 清零

Bit6~0 分别对应周日到周一（Bit6 对应星期天，Bit5 对应星期六，以此类推，Bit0 对应星期一），Bit6~0 的相应位被置位，则表示该定时为重复定时，定时事件触发后 Bit7 不清零，直到用户手动清零 Bit7，否则一直重复。
Hour：1 - Byte，小时，取值范围 0 ~ 23。
Min：1 - Byte，分钟，取值范围 0 ~ 59。
Pin：4 - Byte，参见“0x01 设置 GPIO 状态”

3.4 0x04 查询定时操作(T: TCP | L: UDP | U: TCP)

Request: | 04 | Pin_num|Num| ... |
Response: | 04 | Pin_num|Num | Flag | Hour | Min | Pin | ... |

参数说明：

参见 3.3 节 0x03 设置定时操作
…：重复阴影部分的格式。一次可以查询一个 GPIO 口的多组定时数据。

命令说明：

可同时查询多个定时任务，若请求包中只含一个 Num 且值为 0x00，则为查询全部定时，设备会将全部定时任务返回（包括已关闭的定时任务）。
若返回的 Hour 为 FF，则表示该 Num 为无效，其他字段均忽略，可能是第三方通过软件已经把定时删除，APP 端则将该 Num 值的定时数据删除

3.5 0x05 删除定时操作(T: TCP | L: UDP | U: TCP)

Request: | 05 | Pin_num|Num |
Response: | 05 | Pin_num| Num |

参数说明：

参见 3.3 节 0x03 设置定时操作

命令说明：

删除定时操作才会将此定时任务彻底删除，释放对应定时器 Num 资源。

3.6 0x06 设备 GPIO 事件(T: TCP | L: UDP Broadcast | U: TCP)

Request: | 06 | Pin |…|

Response: 无

参数说明：

参见 3.1 节 0x01 设置 GPIO 状态

… ：重复 Pin 的格式

命令说明：

当设备 GPIO 的状态发生改变时,Device 会主动上报此事件给用户(Server/User)，Server 会把数据推送给订阅了该事件的 APP。

请记得在 APP 中订阅、取消订阅该事件。参见《康泰物联网系统外层协议说明》中的 6.3 节 0x83 订阅/取消订阅设备事件。其中，Cmd=0x06，Param=0x00，表示订阅从第 0 个起的全部 GPIO 口事件。

3.7 0x07 设备定时事件(T: TCP | L: UDP Broadcast | U: TCP)

Request: | 07 | Pin_num|Num | Flag | Hour | Min | Pin |

Response: 无

参数说明：

参见 3.3 节 0x03 设置定时操作

命令说明：

当触发定时事件后（即定时任务到时）或者定时信息改变的时候，Device 会主动上报此事件给用户(Server/User)，Server 会把数据推送给订阅了该事件的 APP。

请记得在 APP 中订阅、取消订阅该事件。参见《康泰物联网系统外层协议说明》中的 6.3 节 0x83 订阅/取消订阅设备事件。其中，Cmd=0x07，Param 为 1 字节，代表引脚编号，即这里的 Pin_num。

注意：单次定时到点，会置定时任务为关闭状态，重复定时则不会。

触发定时事件后是否触发 GPIO 事件取决于 GPIO 状态是否改变,即若定时任务没有使 GPIO 状态改变则只上报定时事件，不上报 GPIO 事件。

若返回的 Hour 为 FF，则表示该 Num 为无效，其他字段均忽略，可能是第三方通过软件已经把定时删除，APP 端则将该 Num 值的定时数据删除。

3.9 0x09 设置防盗模式(T:TCP | L: UDP | U: TCP)

Request: | 09 | Flag | Start_time | Stop_time | Min_interval |

Response: | 09 | Result |

参数说明：

Flag:1 字节,Bit7 表示是否启用该防盗模式,1 为启用,0 为停用。Bit6~Bit0 预留，请置 0。

Start_time：防盗开始时间，4 字节无符号整数，其值为从 1970 年到设定时间所经过的秒数。

Stop_time：防盗结束时间，4 字节无符号整数，其值为从 1970 年到设定时间所经过的秒数。

Result：1 字节，0x00 表示成功，非 0x00 表示失败。

3.10 0x0A 查询防盗模式(T:TCP | L: UDP | U: TCP)

Request: | 0A |

Response: | 0A | Flag | Start_time | Stop_time |

参数说明：

参见 3.9 节 0x09 设置防盗模式

3.11 0x0B 查询电量参数(T:TCP | L: UDP | U: TCP)

Request: | 0B |

Response: | 0B | V | I | P |U|

参数说明：

V：当前电压。精度 1V， 2 个字节，BCD 码格式，例如：02 20 表示 220V。

I：当前电流。精度 0.01A ，2 个字节，BCD 码格式，例如 01 20 表示 1.2A。

P：当前功率。精度 0.0001KW，4 个字节，BCD 码格式，例如 00 10 10 10 表示 10.1010KW。

U：累计电能。精度：0.01KWH，4 个字节，BCD 码格式，例如 00 00 01 20 表示 1.20KWH。该值代表设备从出厂到目前为止累计消耗的电能。

3.12 0x0C 上报电能数据(T:TCP | L: UDP | U: TCP)

Request: | 0C | U |

Response: | 0C | Result |

参数说明：

U 表示累计电能，4 个字节，BCD 码格式。例如 U: 08 89 00 10，表示到目前为止累计耗电 88900.10KWH。

设备默认每隔 30 分钟将此值上传给服务器。在设备上，此值只做累加，即当前

的 U 值减去上一次的值即为这段时间内的电能，
如果服务器正确接收到了该值，则返回 Result=0x00。如果设备在 10 秒内没有收到服务器的正确返回，则重新向服务器上报最新的电能数据，直到上报成功。

3.13 0x0D RF 控制功能(T:TCP | L: UDP | U: TCP)

Request: | 0D | Data |
Response: | 0D | Result |
参数说明：
Data 为串口数据，具体请查看康泰提供的《UART 协议文档》。WiFi 模块接收到该命令后，把串口数据转发给 RF 发射模块。

3.14 0x0E 设置 RF 定时功能(T:TCP | L: UDP | U: TCP)

Request: | 0E | Addr | Num | Flag | Hour | Min | Data |
Response: | 0E | Result |
参数说明：
Addr：2 - Byte，RF 设备的地址码。具体请查看康泰提供的《UART 协议文档》。
Num：1 - Byte，定时任务序号，取值范围为 1~10。即：每个 RF 设备（对应一个地址码）最多可设置 10 组定时。
Flag：1 - Byte，定时任务标志。
Bit7 为定时任务状态（1~开启/0~关闭），若单次定时事件触发，则将对应定时任务的 Bit7 清零
Bit6~0 分别对应周日到周一（Bit6 对应星期天，Bit5 对应星期六，以此类推，Bit0 对应星期一），Bit6~0 的相应位被置位，则表示该定时为重复定时，定时事件触发后 Bit7 不清零，直到用户手动清零 Bit7，否则一直重复。
Hour：1 - Byte，小时，取值范围 0 ~ 23。
Min：1 - Byte，分钟，取值范围 0 ~ 59。
Data：为串口数据，具体请查看康泰提供的《UART 协议文档》。WiFi 模块接收到该命令后，把串口数据转发给 RF 发射模块。

3.15 0x0F 查询 RF 定时功能(T:TCP | L: UDP | U: TCP)

Request: | 0F | Addr | Num | … |
Response: | 0F | Addr | Num | Flag | Hour | Minute | Data | … |
参数说明：
参见 3.14 节 0E 设置定时操作
…：重复阴影部分的格式。阴影部分的出现次数为 0~10。

命令说明：

可同时查询多个定时任务，若请求包中只含一个 Num 且值为 0x00，则为查询该地址码下的全部定时，设备会将该地址码下的全部定时任务返回（包括已关闭的定时任务）。

若返回的 Hour 为 FF，则表示该 Num 为无效，其他字段均忽略，可能是第三方通过软件已经把定时删除，APP 端则将该 Num 值的定时数据删除。

3.16 0x10 删除 RF 定时功能(T:TCP | L: UDP | U: TCP)

Request: | 10 | Addr | Num |

Response: | 10 | Result |

参数说明：

参见 3.14 节 0E 设置 RF 定时操作

命令说明：

删除定时操作会将此地址码下的定时任务彻底删除,释放对应定时器 Num 资源。

如果 Num 值为 0x00，则表示删除该地址码下全部的定时信息。

3.17 0x11 设置倒计时(T: TCP | L: UDP | U: TCP)

Request: | 11 | Pin_Num | Num | Flag | Remain_time | Pin |

Response: | 11 | Result |

参数说明：

Pin_num：1 - Byte 引脚编号，固定为 0x00

Num：1- Byte，第几组，组数暂定为一组，所以 Num 固定为 1

Flag：1 - Byte，最高位为是否有效，1 为有效，0 为无效，其余 bit 保留，固定为 0。

Remain_time：4 - Byte，为倒计时设置的秒数（如：设置 3 分钟的倒计时，Remain_time 的值为 3*60=180）。

Pin：4 - Byte，参见“0x01 设置 GPIO 状态”。

3.18 0x12 查询倒计时(T: TCP | L: UDP | U: TCP)

Request: | 12 | Pin_Num | Num |

Response: | 12 | Pin_Num | Num | Flag | Remain_time | Data | ... |

参数说明：

（参见 3.17 节 11 设置倒计时操作）

...：表示重复阴影部分的格式。

Remain_time : 剩余的倒计时秒数。

3.19 0x13 删除倒计时(T: TCP | L: UDP | U: TCP)

Request: | 13 | Pin_Num | Num |

Response: | 13 | Result |

参数说明：

（参见 3.17 节 11 设置倒计时操作）

命令说明：

删除倒计时操作会将此倒计时任务彻底删除，释放对应倒计时 Num 资源。

4、用户账号相关协议

该项目要求用户需要注册、登录后方可使用。登录的时候,会把相关的数据从服务器同步到该账号下。用户使用同一账号在另一个手机上登录,则显示该账号的相关数据。

（1）该部分内容使用 HTTP 协议实现。

（2）HTTP 返回的数据，均是 JSON 格式。

（3）HTTP 提交和返回的数据，均采用 UTF-8 编码。

（4）HTTP 提交的请求，参数中需要有一个 accessKey，表示访问服务器的授权码，该授权码由欧佰提供。

（5）HTTP 返回的数据，由”success”来表示其操作是否成功，如果不成功，则还有一个”msg”属性，用来表示出错的原因。

例如，操作成功返回：

```
{
    “success” : true
}
```

操作失败返回：

```
{
    “success” : false,
    “msg” : “用户名已经存在”
}
```

（6）HTTP 正确返回的状态码使用 200。

（7）传输的用户密码，均是 MD5 加密后的字符串。

4.1 用户注册

URL：

http://xxx.com/api/account/signup

Method：

POST

参数：

“accessKey” : “abcdefg0123456789”,

“username” : “xbwen@hotmail.com” ,

“password” : “MD5_of_123456”

返回：

```
{
    “success” : true
}
```

4.2 用户登录

URL：

http://xxx.com/api/account/login

Method：

POST

参数：

“accessKey” : “abcdefg0123456789”,

“username” : “xbwen@hotmail.com” ,

“password” : “MD5_of_123456”

返回：

```
{
    “success” : true
}
```

4.3 修改密码

URL：

http://xxx.com/api/account/password/change

Method：

POST

参数：

```
    “accessKey” : “abcdefg0123456789”,
    “username” : “xbwen@hotmail.com” ,
    “old_password” : “MD5_of_123456”,
    “new_password” : “MD5_of_123456789”
```

返回：

```
{
    “success” : true
}
```

4.4 忘记密码

URL：

http://xxx.com/api/account/password/forget

Method：

POST

参数：

```
    “accessKey” : “abcdefg0123456789”,
    “username” : “xbwen@hotmail.com”
```

返回：

```
{
    “success” : true
}
```

说明：

服务器接收到该命令后，会向用户的邮箱发送一封 Email，里面包含修改密码的超链接。

4.5 获取 WiFi 设备列表

URL：

http://xxx.com/api/device/wifi/list

Method：

GET

参数：

```
    “accessKey” : “abcdefg0123456789”,
```

```
    “username” : “xbwen@hotmail.com” ,
    “password” : “MD5_of_123456”
```

返回：

```
{
    “success” : true,
    “list” : [
    {
        “macAddress” : “AABBCCDDEEFF” ,  //MAC 地址
        “companyCode” : “F1”,  //厂家代码
        “deviceType” : “D1”,   //设备类型
        “authCode” : “AABB”,  //授权码
        “deviceName” : “电视机”,  //设备名称
        “imageName” : “AAAA-BBBB-CCCC-DDDD.png” ,  //图片文件名
        “orderNumber” : 1,  //排序号
        “lastOperation” : 129887654309871  //最后更新时间
    },
    …
    ]
}
```

说明：

（1）返回的 list 是一个 JSON 数组。

（2）最后更新时间（lastOperation）是 long 型的数值，使用 UTC 时间计算，是服务器上新增或最后一次修改该条记录的时间的毫秒数。

4.6 编辑 WiFi 设备

URL：

```
    http://xxx.com/api/device/wifi/edit
```

Method：

```
    POST
```

参数：

```
    “accessKey” : “abcdefg0123456789”,
    “username” : “xbwen@hotmail.com” ,
    “password” : “MD5_of_123456”,
    “macAddress” : “AABBCCDDEEFF” ,  //MAC 地址
```

“companyCode” : “F1”, //厂家代码
“deviceType” : “D1”, //设备类型
“authCode” : “AABB”, //授权码
“deviceName” : “电视机”, //设备名称
“imageName” : “AAAA-BBBB-CCCC-DDDD.png” , //图片文件名
“orderNumber” : 2, //排序号
“lastOperation” : 129887654309871 //最后更新时间

返回：

```
{
    “success” : true
}
```

说明：

（1）服务器需要根据提交上来的用户名、MAC 地址来判断该条记录是否已经存在，如果不存在，则是添加操作；如果存在，则是修改操作。

（2）最后更新时间（lastOperation）是 long 型的数值，使用 UTC 时间计算，是 APP 上新增或最后一次修改该条记录的时间的毫秒数。如果服务器发现是修改操作，需要根据提交上来的 lastOperation 进行判断，是否要覆盖服务器要原有的数据。

4.7 删除 WiFi 设备

URL：

http://xxx.com/api/device/wifi/delete

Method：

POST

参数：

“accessKey” : “abcdefg0123456789”,
“username” : “xbwen@hotmail.com” ,
“password” : “MD5_of_123456”,
“macAddress” : “AABBCCDDEEFF” , //MAC 地址
“lastOperation” : 129887654309871 //最后更新时间

返回：

```
{
    “success” : true
```

}

说明：

（1）服务器需要根据提交上来的用户名、MAC 地址来判断删除哪一条记录。

（2）最后更新时间（lastOperation）是 long 型的数值，使用 UTC 时间计算，是 APP 上用户删除该条记录的时间的毫秒数。服务器需要根据提交上来的 lastOperation 进行判断，是否要删除服务器上的对应数据。

4.8 获取 RF 设备列表

URL：

http://xxx.com/api/device/rf/list

Method：

GET

参数：

```
“accessKey” : “abcdefg0123456789”,
“username” : “xbwen@hotmail.com” ,
“password” : “MD5_of_123456”
```

返回：

```
{
    “success” : true,
    “list” : [
    {
        “macAddress” : “AABBCCDDEEFF” , //MAC 地址
        “addressCode” : “AABBCC”, //433 设备地址码
        “type” : 1,   //433 设备类型，1-开关，2-调光器，3-窗帘机，4-恒温器
        “deviceName” : “电视机”, //设备名称
        “imageName” : “AAAA-BBBB-CCCC-DDDD.png” , //图片文件名
              “orderNumber” : 2, //排序号
        “lastOperation” : 129887654309871  //最后更新时间
    },
    …
    ]
}
```

说明：

（1）返回的 list 是一个 JSON 数组。

（2）最后更新时间（lastOperation）是 long 型的数值，使用 UTC 时间计算，是服务器上新增或最后一次修改该条记录的时间的毫秒数。

4.9 编辑 RF 设备

URL：

http://xxx.com/api/device/rf/edit

Method：

POST

参数：

“accessKey” : “abcdefg0123456789”,

“username” : “xbwen@hotmail.com” ,

“password” : “MD5_of_123456”,

“macAddress” : “AABBCCDDEEFF” , //MAC 地址

“addressCode” : “AABBCC”, //433 设备地址码

“type” : 1, //433 设备类型，1-开关，2-调光器，3-窗帘机，4-恒温器

“deviceName” : “电视机”, //设备名称

“imageName” : “AAAA-BBBB-CCCC-DDDD.png” , //图片文件名

“orderNumber” : 2, //排序号

“lastOperation” : 129887654309871 //最后更新时间

返回：

```
{
    “success” : true
}
```

说明：

（1）服务器需要根据提交上来的用户名、MAC 地址、RF 地址码来判断该条记录是否已经存在，如果不存在，则是添加操作；如果存在，则是修改操作。

（2）最后更新时间（lastOperation）是 long 型的数值，使用 UTC 时间计算，是 APP 上新增或最后一次修改该条记录的时间的毫秒数。如果服务器发现是修改操作，需要根据提交上来的 lastOperation 进行判断，是否要覆盖服务器要原有的数据。

4.10 删除 RF 设备

URL：

http://xxx.com/api/device/rf/delete

Method：

POST

参数：

“accessKey” : “abcdefg0123456789”,

“username” : “xbwen@hotmail.com” ,

“password” : “MD5_of_123456”,

“macAddress” : “AABBCCDDEEFF” , //MAC 地址

“addressCode” : “AABBCC”, //433 设备地址码

“lastOperation” : 129887654309871 //最后更新时间

返回：

```
{
    “success” : true
}
```

说明：

（1）服务器需要根据提交上来的用户名、MAC 地址、RF 地址码来判断删除哪一条记录。

（2）最后更新时间（lastOperation）是 long 型的数值，使用 UTC 时间计算，是 APP 上用户删除该条记录的时间的毫秒数。服务器需要根据提交上来的 lastOperation 进行判断，是否要删除服务器上的对应数据。

4.11 上传图片

URL：

http://xxx.com/device/image/upload

Method：

POST

参数：

“accessKey” : “abcdefg0123456789”,

“username” : “xbwen@hotmail.com” ,

“password” : “MD5_of_123456”,

“imageName” : “AAAA-BBBB-CCCC-DDDD.png” , //图片文件名

“file” : File //图片文件

返回：

{

“success” : true

}

说明：

（1）该接口在上传文件的同时，还要求提交必须的参数。

（2）图片文件名，是 APP 上自动生成的，需要确保唯一，建议使用 UUID。

（3）注意：由于服务器端实现的需要，这里的 URL 路径中并没有/api。

4.12 下载图片

URL：

http://xxx.com/UploadedFile/{imageName}

Method：

GET

参数：

imageName：图片文件名称，放置在 URL 中。

返回：

返回的不是 JSON 数据，而是二进制的数据流。

说明：

对于服务器端，图片是存在 mongoDB 中的，/UploadedFile/是在 web.xml 中配置的一个 Servlet。

4.13 查询最近 24 小时的功率

URL：

http://xxx.com/api/device/watt

Method：

GET

参数：

“accessKey” : “abcdefg0123456789”,

“username” : “xbwen@hotmail.com” ,

“password” : “MD5_of_123456”,

“macAddress” : “AABBCCDDEEFF” , //MAC 地址

“timeZone” : “GMT+8” //手机所在时区

返回：

```
{
    “success” : true,
    “data” : [
       {22 ： 10.0}, //小时：瓦数
       {23 : 2.0},
       {0 : 0.5},
       {1 : 0.5},
       {3 : 0.5}
       …
   ]
}
```

说明：

（1）返回的 data 是一个 JSON 数组，代表最近 24 小时每小时的平均功率。

（2）data 不一定包含 24 个元素。对于没有数据的时间点，服务器不会返回，APP 在显示的时候自动置 0。

（3）返回的小时，是手机 APP 所在时区的小时。

4.14 按天查询某个时间段内的电能

URL：

```
   http://xxx.com/api/device/energy/day
```

Method：

```
   GET
```

参数：

```
    “accessKey” : “abcdefg0123456789”,
    “username” : “xbwen@hotmail.com” ,
    “password” : “MD5_of_123456”,
    “macAddress” : “AABBCCDDEEFF” , //MAC 地址
    “timeZone” : “GMT+8”,
    “days” : 30 //30 代表最近 30 天
```

返回：

```
{
    “success” : true,
    “data” : [
```

{28 ： 10.0}， //日期：电量
{29 ： 20.5}，
{30 ： 9.0}，
{1 ： 5}，
{3 ： 6}
…
]
}
说明：
说明：
（1）返回的 data 是一个 JSON 数组，代表最近 30 天每天的总电量。
（2）data 不一定包含 30 个元素。对于没有数据的时间点，服务器不会返回，APP 在显示的时候自动置 0。
（3）返回的日期，是手机 APP 所在时区的日期。

4.15 按月查询最近一年的电能

URL：
http://xxx.com/api/device/energy/month
Method：
GET
参数：
“accessKey” ： “abcdefg0123456789”，
“username” ： “xbwen@hotmail.com” ，
“password” ： “MD5_of_123456”，
“macAddress” ： “AABBCCDDEEFF” ， //MAC 地址
返回：
{
“success” ： true,
“data” ： [
{11 ： 10.0}， //月份：电量
{12 ： 20.5}，
{1 ： 9.0}，
{3：5}，
…

```
    ]
}
```

说明：

（1）返回的 data 是一个 JSON 数组，代表最近一年每个月的总电量。

（2）data 不一定包含 12 个元素。对于没有数据的时间点，服务器不会返回，APP 在显示的时候自动置 0。