# 基于HTTP协议的数据交换格式

王德麾

## 一、数据上传帧

V = 版本号 （默认值 0.2.0）

E = 加密算法名称 （默认值为 NULL）

C = 压缩算法名称 （默认值为 NULL）

T = 数据类型 （默认值为 data）

CN = 编码类型 （文件可显字符上传时）

NOTE = 备注说明 （在某些特殊场合使用）

F = 文件名 （T取某些值时采用）

I1 = guid1 （设备guid1）

I2 = guid2+timestamp(加密) （身份验证）

TIME = 时间戳 ，对应每个设备数据，逗号间隔

ID = 上传二进制文件时，表示此参数id

L = W中数据长度，单位：字节 （T=data时，可省略）

S = 数据存储方式 （默认为，更新存储）

P = 文件中指定位置，文件头为0，单位：字节 （S取某些值时设置）

W = 数据

T = data、alarm时格式为：

**[设备1数据块][设备2数据块]…..**

设备数据块格式：

**dev\_id ;(数据1块)（数据2块）**

数据n块格式：

**数据标识id,value,可选项**

其中，可选项为取值为：

1. 单位：u单位符号；

2. 时间偏移：t数值，默认为0；（例：t-1 偏移值为-1； t2 偏移值为+2）

3. 数据类型：d类型名。

4. 数据组号：g数值，如：g1

**注：**存在默认值的key，可以省略不传。

TIME参数中时间戳可以为：

时间戳绝对值；

相对值+/-偏移量，其基值是之前设备块中，最近一个采用绝对值形式的时间戳。

E、C、T、S、d具体取值，见附件1。

## 二、设备描述信息帧

## 三、控制指令交换帧

## 四、设备端监听端口的注册（接收控制指令）

访问： 0x85.php

必须key： I1、 I2（后续加入，C和E变为必须）

可选key： C、 E

注册流程为：

1. 设备发送 I1、I2至服务器端；

2. 等待接收服务器端返回数据；

3. 如请求被接收，则返回新的IP、端口号；

设备

服务器端返回数据格式：

1. 非法的注册请求，不返回数据；

2. 合法的注册请求，返回：

E=××&W=IP;端口号

其中IP为： IPV4 - ××.××.××.××

IPV6 同IPV4，根据符号”.”数量决定；

返回的IP如与设备连接的服务器IP一致，则IP值为：SAME

E可省略，省略时取默认值，即无加密；

返回数据以“\r\n”结尾.

设备重连此ip、端口，发送：

I1=guid1， 以“\r\n”结尾。

# 附件1

|  |  |
| --- | --- |
| **T参数取值** | **d参数取值** |
| data（默认）  file  image  video  audio  alarm  order  dev/info | value（默认）  bin（16进制字符表示）  utf8(多数据，空格间隔)  ascii(16进制字符表示) |

|  |  |
| --- | --- |
| **S参数取值** | **CN** |
| 0更新保存（默认）  1累积保存  2追加  3从指定位置更新 | base64  plain |

注：S取1时，是否累积保存，还需根据用户权限确定；

file 指一般性的文件，即不是T中指出的任意一种文件类型

# 附件2

二进制文件上传格式

1. **涉及的key**

可选key： F、 TIME、 L、 S、 V、 A、 E、 C、I2（I2以后须设置为必须key）

必须key： T、I1

**注：各key出现顺序不限。**

1. **数据组织格式**

T=xxx 0X00 TIME=xxx 0x00 … 13 文件数据

即：key-value用ascii字符串形式编码，字符串末尾必须为0。Key-value段后用13与文件正文间隔。

# 附件3

各类型数据传输规范

**data数据上传**

访问： 0x82.php

必须key： I1、 TIME、 W、 I2（后续加入）

可选key： C、 E、 T

注： T=‘order’，W格式同data时的格式，携带的内容为设备想让服务器运行的指令；

**数据上传帧格式：**

POST /0x82.php HTTP/1.1\r\n

Host:host\_addr\r\n

Content-Type:application/x-www-form-urlencoded\r\n

Content-Length: Length \r\n\r\n

I1=\*\*&TIME=\*\*&W=\*\*….\r\n

其中: \r\n 对应的十进制的ASCII码分别为13，10，即换行、回车；

host\_addr 为主机的 DNS 或 IP；

Length 为 “I1=\*\*&TIME=\*\*&W=\*\*….\r\n”内容的长度，含\r\n；

当 TIME=F××时，表示以××赫兹频率采集的数据。此时W中传递的参数id须相同。第一个参数采集时间基准为0，以后依次增加1/××秒。（待实现）

**alarm数据上传**

访问： 0x82.php

必须key： T=’alarm’、 I1、 TIME、 W、 I2（后续加入，C和E变为必须）

可选key： C、 E

**设备信息上传**

访问： 0x83.php

必须key： T=’dev/info’、 I1、 TIME、 W、 I2（后续加入，C和E变为必须）

可选key： C、 E

W 中设备信息以 xml 文件格式描述，具体为：

<dev>

<n>dev\_name</n>

<m>model</m>

<c>company</c>

<k>key(可选)</k>

<tz>时区</tz>

<logo>设备logo图标名称</logo>

<lo>经度</lo>

<la>维度</la>

<h>高度</h>

<d>

<id></id> //数据id号，唯一，必需

<n></n> // 参数名称

<rem></rem> // 参数的remark

<ty></ty> //参数的数据类型

<unit></unit>

<sd></sd> // storage duration，保存周期，单位：小时，可选

<st></st> // 保存方式，是否累积

</d>

<d>

…

</d>

</dev>

**以可显示字符上传文件**

访问： 0x82.php

必须key： T、 I1、 W、I2（后续加入，C和E变为必须）

可选key： F、 C、 E、S、 TIME、 CN 、 L

注： 此时W中仅能有1个设备的1个参数值；

参数值直接为按照CN指定的编码方式编码的文件数据（CN默认为plain）

**二进制文件上传**

访问： 0x81.php

必须key： L、 T、 I1、 TIME、I2、ID（后续加入，C和E变为必须）

可选key： F、 C、 E、 S

**如没有F，则默认为jpg格式。**

**如没有定义S，则默认为0，即更新保存。**

**设备控制端口注册（待实现）**

访问： 0x82.php

必须key： T=’dev/funs’、 I1、 TIME、 W、 I2（后续加入，C和E变为必须）

可选key： C、 E

**控制指令交换（用户控制设备）（待实现）**

访问： 0x84.php

必须key： I1、 TIME、 W、 I2（后续加入，C和E变为必须）

可选key： C、 E

**设备操作信息上传**

访问： 0x83.php

必须key： T=’dev/funs’、 I1、 TIME、 W、 I2（后续加入，C和E变为必须）

可选key： C、 E

W 中设备控制信息以 xml 文件格式描述，具体为：

<funs>

<f>

<id>fun\_id</id>

<n>fun name</n>

<r>fun\_remark</r>

<p>

<n>p\_name</n>

<r>p\_remark</r>

<u>unit</u>

</p>

<p>

<n>p\_name</n>

….

</p>

</f>

<f>

……….

</f>

</funs>

设备采集数据信息上传xml文件格式（设备无需优先发送此信息，数据上传时系统会自动将相关信息添加至数据库）

<d> // 代表设备

<p>

<id>××</id>

<n>参数名称</n>

<u>单位</u>

<a>别名</a>

<r>备注</r>

</p>

</d>