上位机与终端通信JSON\_V1.4

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 日期 | 修订记录 | 作者 |
| V1.0 | 2023-03-15 | 初始版本 | 程茂林 |
| V1.1 | 2023-03-17 | 1.优化字段，并添加字段定义；  2.增加告警信息；  3.除报文时标外，增加各参数时标；  4.合并值和单位字段； | 程茂林、吴江 |
| V1.2 | 2023-03-22 | 1.细化告警字段定义。 | 程茂林、吴江 |
| V1.3 | 2023-03-23 | 1.参数增加kks字段，由用户提供每个参数测点的kks码。  2.params中存在同一参数不同时间的值。  3.ref定义为别名字段，kks码为唯一标识。  4.各系统会存在时间差，上位机需记录接收时间。 | 程茂林、黄天雄、郭祖仪 |
| V1.4 | 2023-04-06 | 1.增加通信协议定义。  2.修改时间字段为可读格式。  3.修改包时间字段”pkgt”->”t”。 | 程茂林、吕畅 |
| V1.5 | 2023-04-11 | 1.标准化json格式 | 岳豪 |
| V1.6 | 2023-04-14 | 1.新增vtype字段，标识数据类型  2.新增websocket通信相关信息 | 岳豪、吕畅 |

# 一、概述

安全网关与终端的ssl客户端建立安全通道，为终端、标段二、上位机之间的无线通信提供加密传输。

上位机为终端和标段二的数据提供界面展示，需接收终端和标段二的数据上报。上位机为WebSocket服务端，终端及标段二为WebSocket客户端。

通信内容为json文本，json详细内容参考字段定义和示例，其中所有汉字字符均采用UTF8编码。

# 二、字段定义

## “name”字段：

"name":"string",<--终端提供命令行设置，标段二上时报由其自主控制-->

## “id”字段：

"id":"string",<--出厂初始化设置，过程中不允许修改，标段二上时报由其自主控制-->

## “terminal”字段：

"terminal":"string",<--终端提供命令行设置，更换IO板卡时由命令行更改，string=COL、ENV、VOL、CUR，ASE、PSE,为六种采集终端collect、environment、voltage、current、Activesensor、Passivesensor的缩写，标段二上报时固定为COL-->

## “mtype”字段：

"mtype":"string",<--消息类型string=UNKN、MSGE、EVENT-->

## “t”字段：

"t":" yyyy-mm-dd hh:mm:ss.SSS ",<--报文时标,例如2023-04-05 09:28:47.872 每段报文、每个参数、每个告警事件应带时标字段-->

## “params”组：

除临时使用类型终端，其区域、设备、参数等名称均在实施时配置完成，标段二的ref及单位由其自行定义；对临时使用的终端，提供命令行修改功能，对临时测试的参数进行台账命名。

### “kks”字段：

"kks":"string",<--根据用户配置的kks码组织报文-->

### “ref”字段：

"ref":"区域/设备.参数",<--别名字段；ENV、VOL、CUR，ASE、PSE类型的终端区域=$name，”，标段二的区域、设备、参数由其自主控制-->

### “vtype”字段：

"vtype":"区分参数值的类型，例如温度、湿度、水压、气压、电流、电压、PM25等"

### “val”字段：

"val":"参数值-单位",

### “t”字段：

"t":"年-月-日 时:分:秒:毫秒",<--参数时标，与报文时标定义相同-->

## “events”组：

对终端设备的系统事件、参数越限等信息，通过events上报，告知上位机。参数范围上下限，在实施时由终端侧配置定义。标段二的事件内容由其自行定义，并根据events格式组织上报。

### “type”字段：

"type":"strint",<--string=SYS-系统、COM-通信、VAL-参数等事件主体，标段二根据事件情况选择合适的事件主体-->

### “level”字段：

"level":"strint",<--string=UNKN、INFO、WARN、ERRO等，标段二根据事件情况选择合适的事件等级-->

### “kks”字段：

"kks":"string",<--根据用户配置的kks码组织报文，不是参数事件时设置为空-->

### “msg”字段：

"msg":"string",<--string=对当前事件的字符串描述，如：“系统重启”、“cpu占用过高”、“发变811层/机组1.水压1值0.78超告警上线0.75”等，标段二根据事件情况组织合适的告警信息-->

### “t”字段：

"t":"年-月-日 时:分:秒:毫秒",<--事件发生时间，与参数时标定义相同-->

# 三、参数上报示例

## 标段二参数

{

    "name": "终端1",

    "id": "202303110001xl",

    "terminal": "COL",

    "mtype": "MSGE",

    "t": "2023-04-05 09:28:47.475",

    "params": [

        {

            "kks": "kks编码",

            "ref": "发变811层/机组1.水压1",

            "vtype": "水压",

            "val": "0.56-Mpa",

            "t": "2023-04-05 09:28:47.872"

        },

        {

            "kks": "kks编码",

            "ref": "发变811层/机组1.水压2",

            "vtype": "水压",

            "val": "0.55-Mpa",

            "t": "2023-04-05 09:28:47.879"

        },

        {

            "kks": "kks编码",

            "ref": "发变811层/机组1.水压20",

            "vtype": "水压",

            "val": "0.57-Mpa",

            "t": "2023-04-05 09:28:47.879"

        },

        {

            "kks": "kks编码",

            "ref": "发变816层母线洞/GEBS.SF6气压",

            "vtype": "气压",

            "val": "0.62-Mpa",

            "t": "2023-04-05 09:28:47.879"

        },

        {

            "kks": "kks编码",

            "ref": "主变828层/GCB.SF6气压",

            "vtype": "气压",

            "val": "0.62-Mpa",

            "t": "2023-04-05 09:28:47.879"

        },

        {

            "kks": "kks编码",

            "ref": "GIS837层/GIS气室.SF6气压1",

            "vtype": "气压",

            "val": "0.63-Mpa",

            "t": "2023-04-05 09:28:47.879"

        },

        {

            "kks": "kks编码",

            "ref": "GIL竖井/GIL气室.SF6气压1837\_1",

            "vtype": "气压",

            "val": "480-Kpa",

            "t": "2023-04-05 09:28:47.879"

        },

        {

            "kks": "kks编码",

            "ref": "GIL竖井/GIL气室.SF6气压8111\_6",

            "vtype": "气压",

            "val": "480-Kpa",

            "t": "2023-04-05 09:28:47.879"

        },

        {

            "kks": "kks编码",

            "ref": "出线场/GIL气室.SF6气压1145\_15",

            "vtype": "气压",

            "val": "480-Kpa",

            "t": "2023-04-05 09:28:47.879"

        }

    ]

}

## 环境参数

{

    "name": "水车室",

    "id": "202303110002xl",

    "terminal": "ENV",

    "mtype": "MSGE",

    "t": "2023-04-05 09:28:47.894",

    "params": [

        {

            "kks": "kks编码",

            "ref": "水车室/ENV.温度",

            "vtype": "温度",

            "val": "21.4-℃",

            "t": "2023-04-05 09:28:47.894"

        },

        {

            "kks": "kks编码",

            "ref": "水车室/ENV.湿度",

            "vtype": "湿度",

            "val": "89-%",

            "t": "2023-04-05 09:28:47.894"

        },

        {

            "kks": "kks编码",

            "ref": "水车室/ENV.PM25",

            "vtype": "PM25",

            "val": "102-ug/m³",

            "t": "2023-04-05 09:28:47.894"

        }

    ]

}

## 电压参数

{

    "name": "发变811层",

    "id": "202303110003xl",

    "terminal": "VOL",

    "mtype": "MSGE",

    "t": "2023-04-05 09:28:47.475",

    "params": [

        {

            "kks": "kks编码",

            "ref": "发变811层/VOL.V1",

            "vtype": "电压",

            "val": "48.2-V",

            "t": "2023-04-05 09:28:47.475"

        }

    ]

}

## 电流参数

{

    "name": "发变811层",

    "id": "202303110004xl",

    "terminal": "CUR",

    "mtype": "MSGE",

    "t": "2023-04-05 09:28:47.475",

    "params": [

        {

            "kks": "kks编码",

            "ref": "发变811层/CUR.AC",

            "vtype": "电流",

            "val": "4.8-A",

            "t": "2023-04-05 09:28:47.475"

        }

    ]

}

## 有源传感参数

{

    "name": "发变811层",

    "id": "202303110005xl",

    "terminal": "ASE",

    "mtype": "MSGE",

    "t": "2023-04-05 09:28:47.475",

    "params": [

        {

            "kks": "kks编码",

            "ref": "发变811层/ASE.ASE1",

            "vtype": "",

            "val": "23-Hz",

            "t": "2023-04-05 09:28:47.475"

        },

        {

            "kks": "kks编码",

            "ref": "发变811层/ASE.ASEN",

            "vtype": "",

            "val": "5000-m³/min",

            "t": "2023-04-05 09:28:47.475"

        }

    ]

}

## 无源传感参数

{

    "name": "终端1",

    "id": "202303110006xl",

    "terminal": "PSE",

    "mtype": "MSGE",

    "t": "2023-04-05 09:28:47.475",

    "params": [

        {

            "kks": "kks编码",

            "ref": "发变811层/PSE.PSE1",

            "vtype": "",

            "val": "23-Hz",

            "t": "2023-04-05 09:28:47.475"

        },

        {

            "kks": "kks编码",

            "ref": "发变811层/PSE.PSEN",

            "vtype": "",

            "val": "2000-转/min",

            "t": "2023-04-05 09:28:47.475"

        }

    ]

}

# 四、事件上报示例

{

    "name": "发变811层",

    "id": "202303110003xl",

    "terminal": "VOL",

    "mtype": "EVENT",

    "t": "2023-04-05 09:28:47.475",

    "events": [

        {

            "type": "SYS",

            "level": "INFO",

            "kks": "",

            "t": "2023-04-05 09:28:47.127",

            "msg": "系统重启"

        },

        {

            "type": "SYS",

            "level": "INFO",

            "kks": "",

            "t": "2023-04-05 09:28:47.127",

            "msg": "进程重启"

        },

        {

            "type": "COM",

            "level": "INFO",

            "kks": "",

            "t": "2023-04-05 09:28:47.127",

            "msg": "ConnectSrv192.168.100.24:9000Success"

        },

        {

            "type": "VAL",

            "level": "WARN",

            "kks": "kks编码",

            "t": "2023-04-05 09:28:47.127",

            "msg": "发变811层/VOL.V1值248超越上限值240"

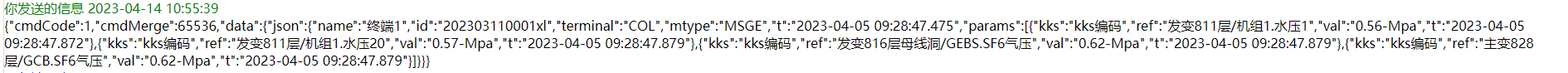
        }

    ]

}

# websocket通信相关

## 5.1 参数示例说明



假设你要发送的参数value如下所示：

{  
    "name":"终端1",  
    "id":"202303110001xl",  
    "terminal":"COL",  
    "mtype":"MSGE",  
    "t":"2023-04-05 09:28:47.475",  
    "params":[  
        {  
            "kks":"kks编码",  
            "ref":"发变811层/机组1.水压1",  
            "val":"0.56-Mpa",  
            "t":"2023-04-05 09:28:47.872"  
        },  
        {  
            "kks":"kks编码",  
            "ref":"发变811层/机组1.水压20",  
            "val":"0.57-Mpa",  
            "t":"2023-04-05 09:28:47.879"  
        },  
        {  
            "kks":"kks编码",  
            "ref":"发变816层母线洞/GEBS.SF6气压",  
            "val":"0.62-Mpa",  
            "t":"2023-04-05 09:28:47.879"  
        },  
        {  
            "kks":"kks编码",  
            "ref":"主变828层/GCB.SF6气压",  
            "val":"0.62-Mpa",  
            "t":"2023-04-05 09:28:47.879"  
        }  
    ]  
}

之后把数据组装成如下格式，value是你想要传输的数据：

{  
 cmdCode: 1,  
 cmdMerge: 65537,  
 data: {json:value}  
}

cmdCcode默认是1

cmdMerge这里约定：当使用参数上报时是：65537，当使用事件上报时是：65538

然后开始编码、传输即可

## 5.2 编码步骤以及相应的伪代码说明

**步骤：**

1. 声明一个对象object，保存你要传输的数据
2. 再声明一个对象data，然后将对象object赋给data的json属性，即data.json=object
3. 将data转换为json字符串格式，用字符串变量str保存
4. 将字符串变量str转码为字节数组byteArr(使用UTF-8编码)
5. 创建一个对象message，将字节数组byteArr付给message的data属性，即message.data=byteArr，再加上路由参数cmdCode和cmdMerge
6. 重复步骤3、4，生成字节数组messageArr
7. 将messageArr传输给上位机，使用websocket传输二进制数据

**相应伪代码：**

1. Object object = “你要传输的数据”
2. Object data = new Object(); data.json = object;
3. String str = toJsonString(data);
4. ByteArray byteArr = stringToByteArray(str);
5. Object message = new Object(cmdCode, cmdMerge); message.data = byteArr;
6. String str2 = toJsonString(message);
7. ByteArray messageArr = stringToByteArray(str2);

**字符串与字节数组相互转换的算法：**

仅供参考，可以自己实现

*//将字符串转为 Array byte数组***function** *stringToByte*(str) {  
 *//创建一个空字节数组bytes* **let** bytes = **new *Array***();  
 *//创建一个变量len保存字符串的长度* **let** len = str.**length**;  
 *//创建一个变量c，用于保存字符* **let** c;  
 **for** (**let** i = 0; i < len; i++) {  
 c = str.charCodeAt(i);  
 **if** (c >= 0x010000 && c <= 0x10FFFF) {  
 bytes.push(((c >> 18) & 0x07) | 0xF0);  
 bytes.push(((c >> 12) & 0x3F) | 0x80);  
 bytes.push(((c >> 6) & 0x3F) | 0x80);  
 bytes.push((c & 0x3F) | 0x80);  
 } **else if** (c >= 0x000800 && c <= 0x00FFFF) {  
 bytes.push(((c >> 12) & 0x0F) | 0xE0);  
 bytes.push(((c >> 6) & 0x3F) | 0x80);  
 bytes.push((c & 0x3F) | 0x80);  
 } **else if** (c >= 0x000080 && c <= 0x0007FF) {  
 bytes.push(((c >> 6) & 0x1F) | 0xC0);  
 bytes.push((c & 0x3F) | 0x80);  
 } **else** {  
 bytes.push(c & 0xFF);  
 }  
 }  
 **return** bytes;  
}

*//byte数组转字符串***function** *byteToString*(arr) {  
 *//创建一个变量str用于保存生成的字符串* **let** str = **''**;  
 **let** \_arr = arr;  
 **for** (**let** i = 0; i < \_arr.**length**; i++) {  
 **let** one = \_arr[i].toString(2),  
 v = one.match(/^1+?(?=0)/);  
 **if** (v && one.**length** === 8) {  
 **let** bytesLength = v[0].**length**;  
 **let** store = \_arr[i].toString(2).slice(7 - bytesLength);  
 **for** (**let** st = 1; st < bytesLength; st++) {  
 store += \_arr[st + i].toString(2).slice(2);  
 }  
 *//fromCharCode函数用于从一些Unicode字符值中返回一个字符串。  
 //例如：String.fromCharCode(65) =='A'  
 //例如：String.fromCharCode(66) =='B'  
 //例如：String.fromCharCode(67) =='C'* str += ***String***.fromCharCode(*parseInt*(store, 2));  
 i += bytesLength - 1;  
 } **else** {  
 str += ***String***.fromCharCode(\_arr[i]);  
 }  
 }  
 **return** str;  
}