

# 概要

## 名词解释

读写器:读卡设备

客户端:Windows 应用程序或者网页程序

## 基本需求

1. 读写器启用后,连上 TCP 服务器,主动将工作参数发送到服务器.
2. 服务器可以查询读写器工作状态,并可以在线设置读写器参数.
3. 读写器在一定时间内将读到的电子标签按照规定的数据格式通过 TCP 方式上传到服务器.
4. 读写断电数据保存数据不能丢失.
5. 如果上传后服务器失败,读写器需要保存电子标签的数据.
6. 服务器可以在线重启读写器.

## 配置参数

1. TCP 服务器地址.
2. 同一电子标签在多长时间上传.
3. 读写器天线功率.
4. 设备号,用于设备唯一.

## 端口定义

服务器在服务端开启三两个个端口:一个用于上传标签数据,一个用于发送命令.

配置读写器,只配置上传标签地址,发送命令地址端口是在上传端口加 1.

如上传标签的地址为:102.11.2.11:8088

则发送命令的地址为:102.11.2.11:8089

## 数据格式

# 服务器与读写器数据格式

---

数据以<字符为开始符,字符>结束符.消息实体为 JSON 对象.

例:

```
1. <{
2.     "msg_id" : "00001",
3.     "cmd" : "10001",
4.     "source" : "console",
5.     "data" : {
6.         "result" : "ok"
7.     }
8. }>
```

## 通用数据格式

参数名	类型	说明
msg_id	string	消息号
cmd	string	命令类型
source	string	数据源,读写器接收到此类型时需要回传
data	object	数据类型

## 服务器与读写器命令定义

参数值	解释
10001	查询读写器状态
10002	设置读写器参数
10003	重启读写器
10004	读写器参数下载
10005	上传标签数据
10006	开始读取标签
10007	停止读取标签
10008	获取系统时间

# 1.查询读写器状态

简要描述：

- 服务器主动查询读写器工作状态

命令类型

- 10001

命令方向

- 服务器->读写器

请求方式：

- TCP

请求参数：

无

返回参数：

参数名	必选	类型	说明
device_id	是	string	设备号
upload_time	是	int	当前标签上传间隔时间, 毫秒为单位
server_addr	是	string	服务器地址
auto_start_read	是	int	是否自动开始读取, 读取: <span>1</span> , 不读取: <span>0</span>
is_reading_tag	是	int	是否在读标签 读取: <span>1</span> , 未读取: <span>0</span>
enable_cache	否	int	是否暂存数据, 暂存: <span>1</span> , 不暂存: <span>0</span>
antenna	是	string	天线状态格式:[天线号, 是否启用, 功率], 每根天线用分号隔开. 启用: <span>1</span> , 未启用: <span>0</span> , 如: <span>1,1,2000;2,0,3000;3,1,2000;4,1,3000</span>
is_ok	是	int	读写器工作状态, 正常: <span>1</span> , 异常: <span>0</span>
error_code	否	int	读写器异常码, 具体由读写器定义

请求示例

```
1.  {
2.     "msg_id":"00001",
```

```
3.     "source": "console",
4.     "cmd": "10001"
5. }
```

## 返回示例

```
1.  {
2.     "msg_id": "00001",
3.     "cmd": "10001",
4.     "source": "console",
5.     "data": {
6.         "device_id": "KL010096",
7.         "upload_time": 60000,
8.         "is_reading_tag": 1,
9.         "auto_start_read": 1,
10.        "enable_cache": 1,
11.        "server_addr": "102.11.23.25:8088",
12.        "antenna": "1,1,2000;2,0,3000;3,1,2000;4,1,3000",
13.        "is_ok": 1
14.    }
15. }
```

## 2. 设置读写器参数

简要描述：

- 服务器将参数发送到读写器,读写器接收并设置.

命令类型

- 10002

命令方向

- 服务器->读写器

请求方式：

- TCP
- 请求参数：

参数名	必选	类型	默认值	说明
upload_time	否	int		当前标签上传间隔时间, 毫秒为单位
enable_cache	否	int	1	是否缓存数据, 缓存: 1, 不缓存: 0
auto_start_read	是	int	1	是否自动开始读取, 读取: 1, 不读取: 0
server_addr	否	string		服务器地址为空不设置
antenna	否	string		天线状态格式:[天线号, 是否启用, 功率], 每根天线用分号隔开. 启用: 1, 未启用: 0, 如: 1,1,2000;2,0,3000;3,1,2000;4,1,3000

- 返回参数：

参数名	必选	类型	说明
device_id	是	string	设备号
result	是	int	设置结果, 成功: 1, 天线不可用: 2, 失败: 0

请求示例

```
1.  {
2.     "msg_id": "00001",
3.     "cmd": "10002",
4.     "source": "console",
5.     "data": {
6.         "upload_time": 60000,
7.         "enable_cache": 1,
8.         "auto_start_read": 1,
9.         "server_addr": "102.11.23.25:8088",
10.        "antenna": "1,1,2000;2,0,3000;3,1,2000;4,1,3000",
11.    }
12. }
```

返回示例

```
1.  {
2.     "msg_id": "00001",
3.     "cmd": "10002",
4.     "source": "console",
5.     "data": {
6.         "device_id": "KL010096"
7.         "result": 1
8.     }
9. }
```



### 3.开始读取标签

简要描述：

- 读写器开始读取标签

命令类型

- 10006

命令方向

- 服务器->读写器

请求方式：

- TCP

请求参数：

空

返回参数：

参数名	必选	类型	说明
device_id	是	string	设备号

参数名	必选	类型	说明
result	是	int	开始读取结果, 成功: <input type="text" value="1"/> , 失败: <input type="text" value="0"/>

请求示例

```
1.  {
2.    "msg_id":"00001",
3.    "cmd":"10006",
4.    "source":"console"
5. }
```

返回示例

```
1.  {
2.    "msg_id":"00001",
3.    "cmd":"10006",
4.    "source":"console",
5.    "data":{
6.      "device_id":"KL010096",
7.      "result":1
8.    }
9. }
```

4.停止读取标签

简要描述：

- 读写器停止读取标签

## 命令类型

- 10007

## 命令方向

- 服务器->读写器

## 请求方式：

- TCP

## 请求参数：

空

## 返回参数：

参数名	必选	类型	说明
device_id	是	string	设备号
result	是	int	停止读取结果, 成功: 1, 失败: 0

## 请求示例

```
1. {
```

```
2.     "msg_id":"00001",
3.     "cmd":"10007",
4.     "source":"console"
5. }
```

返回示例

```
1.  {
2.     "msg_id":"00001",
3.     "cmd":"10007",
4.     "source":"console",
5.     "data":{
6.         "device_id":"KL010096"
7.         "result":1
8.     }
9. }
```

## 5.重启读写器

简要描述：

- 重启读写器

命令类型

- 10003

命令方向

- 服务器->读写器

请求方式：

- TCP

请求参数：

空

返回参数：

参数名	必选	类型	说明
device_id	是	string	设备号
result	是	int	重启结果, 成功: 1, 失败: 0

请求示例

```
1.  {
2.    "msg_id":"00001",
3.    "source":"console",
4.    "cmd":"10003"
5. }
```

返回示例

```
1.  {
2.    "msg_id":"00001",
3.    "cmd":"10003",
4.    "source":"console",
5.    "data":{
```

```
6.         "device_id": "KL010096"  
7.         "result": 1  
8.     }  
9. }
```

## 6.读写器参数下载

简要描述：

- 读写器从服务器下载参数,并设置,如果下载失败,则不用设置.
- 非缓存模式下,读写器不用进行数据处理,采集的数据立即上传.
- 缓存模式下,标签上传时间不能低于 1 秒,低于 1 秒,读写器自动关闭缓存模式.

命令类型

- 10004

请求方式：

- TCP

命令方向

- 读写器->服务器

请求参数：

参数名	必选	类型	说明
device_id	是	string	设备号

返回参数：

参数名	必选	类型	默认值	说明
upload_time	是	int		标签上传间隔时间, 毫秒为单位
enable_cache	否	int	1	是否暂存数据, 暂存: 1, 不暂存: 0
auto_start_read	是	int	1	是否自动开始读取, 读取: 1, 不读取: 0
server_addr	是	string		服务器地址
antenna	是	string		天线状态格式: [天线号, 是否启用, 功率], 每根天线用分号隔开. 启用: 1, 未启用: 0, 如: 1,1,2000;2,0,3000;3,1,2000;4,1,3000

## 请求示例

```
1.  {
2.    "msg_id":"00001",
3.    "cmd":"10004",
4.    "source":"console",
5.    "data":{
6.      "device_id":"KL010096"
7.    }
8. }
```

## 返回示例

```
1.  {
2.    "msg_id":"00001",
3.    "cmd":"10004",
4.    "source":"console",
5.    "data":{
6.      "upload_time":60000,
7.      "enable_cache":1,
8.      "auto_start_read":1,
9.      "server_addr":"102.11.23.25:8088",
10.     "antenna":"1,1,2000;2,0,3000;3,1,2000;4,1,3000"
11.   }
12. }
```

## 7.上传标签数据

简要描述：



- 上传标签数据

## 命令类型

- 10005

## 请求方式：

- TCP

## 命令方向

- 读写器->服务器

## 请求参数：

参数名	必选	类型	说明
device_id	是	string	设备号
count	是	int	标签数量
epcs	是	string	标签数据, 以分号分隔; 格式[EPC, 时间, 次数, 天线]

## 返回参数：

参数名	必选	类型	说明
result	是	int	1 成功, 0 失败

### 请求示例

```
1.  {
2.    "msg_id":"00001",
3.    "cmd":"10005",
4.    "source":"client",
5.    "data":{
6.      "device_id":"KL01106",
7.      "count":2,
8.      "epc":"20000021110001,2018-01-22
16:20:21,1,1;20000021110002,2018-01-22 16:20:22,2,1"
9.    }
10. }
```

### 返回示例

```
1.  {
2.    "msg_id":"00005",
3.    "cmd":"10005",
4.    "source":"client",
5.    "data":{
6.      "result":1
7.    }
8.  }
```

## 8.获取服务器时间

简要描述：

- 读写器从获取服务器时间.并上传读写器状态.

命令类型

- 10008

请求方式：

- TCP

命令方向

- 读写器->服务器

请求参数：

参数名	必选	类型	说明
device_id	是	string	设备号
upload_time	是	int	当前标签上传间隔时间, 毫秒为单位

参数名	必选	类型	说明
server_addr	是	string	服务器地址
auto_start_read	是	int	是否自动开始读取, 读取: 1, 不读取: 0
is_reading_tag	是	int	是否在读标签 读取: 1, 未读取: 0
enable_cache	否	int	是否暂存数据, 暂存: 1, 不暂存: 0
antenna	是	string	天线状态格式:[天线号, 是否启用, 功率], 每根天线用分号隔开. 启用: 1, 未启用: 0, 如: 1,1,2000;2,0,3000;3,1,2000;4,1,3000
is_ok	是	int	读写器工作状态, 正常: 1, 异常: 0
error_code	否	int	读写器异常码, 具体由读写器定义

返回参数:

参数名	必选	类型	说明
server_time	是	string	服务器时间

## 请求示例

```
1.  {
2.    "msg_id":"00001",
3.    "cmd":"10008",
4.    "source":"client",
5.    "data":{
6.      "device_id":"KL010096",
7.      "upload_time":60000,
8.      "is_reading_tag":1,
9.      "auto_start_read":1,
10.     "enable_cache":1,
11.     "server_addr":"102.11.23.25:8088",
12.     "antenna":"1,1,2000;2,0,3000;3,1,2000;4,1,3000",
13.     "is_ok":1
14.   }
15. }
```

## 返回示例

```
1.  {
2.    "msg_id":"00001",
3.    "cmd":"10008",
4.    "source":"client",
5.    "data":{
6.      "server_time":"2018-01-22 13:44:55"
7.    }
8.  }
```