|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 【密级程度】 公开 √ 机密 绝密 \_\_   |  | | --- | | **网络管理** |   文件编号：  版 本 号： V1.0  拟制日期： 2017-09-19  发布日期：   |  |  |  | | --- | --- | --- | | 批准 | 审核 | 编制 | |  |  | 罗清文 | | 日期： | 日期： | 日期：2017-09-18 |  |  |  | | --- | --- | | 发文部门 | 研发部 | | 收文部门 | 研发文控 √ | | 市场部 | | 采购部 | | 品质部 | |

# 目录

[目录 2](#_Toc493679156)

[1 概述 3](#_Toc493679157)

[1.1. 主要功能 3](#_Toc493679158)

[1.2. 功能描述 3](#_Toc493679159)

[2 数据包定义 3](#_Toc493679160)

[2.1 找卡 3](#_Toc493679161)

[2.2 有线网络信息 6](#_Toc493679162)

[2.3 3/4G信息 8](#_Toc493679163)

[2.4 WIFI 11](#_Toc493679164)

**网络管理**

1. 概述

## 主要功能

本文主要叙述与下位机网络相关的内容，主要由：找卡、有线网络信息、3/4G信息、WIFI等部分组成。

## 功能描述

1. 找卡：上位机可以通过该功能发现局域网中的控制卡。
2. 有线网络信息：上位机可以通过该功能获取下位机有线网络地址信息，也可以修改下位机的有线网络地址信息。
3. 3/4G信息：上位机可以通过该功能获取下位机的3/4G信息，也可以修改3/4G相关的APN参数信息。
4. WIFI：上位机可以通过该功能获取下位机的WIFI信息，也可以修改WIFI的参数信息。
5. 数据包定义

## 找卡

1. 找卡请求

#pragma pack(1)

//结构定义

typedef struct HUdpSearchAsk

{

huint32 version; ///< 版本号

huint16 cmd; ///< 命令值

} HUdpSearchAsk;

#pragma pack()

|  |  |  |
| --- | --- | --- |
| 字段 | 版本号(Version) | 命令(Cmd) |
| 字节数 | 4字节 | 2字节 |
| 取值 | LOCAL\_UDP\_VERSION | kSearchDeviceAsk |

1. 找卡反馈

#pragma pack(1)

//结构定义

typedef struct HUdpSearchAnswer

{

huint32 version; ///< 版本号

huint16 cmd; ///< 命令值

hint8 id[MAX\_DEVICE\_ID\_LENGHT]; ///< 设备ID

huint32 change; ///< 设备信息改变标志

} HUdpSearchAnswer;

#pragma pack()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 版本号(Version) | 命令(Cmd) | 设备ID(ID) | 标志(Change) |
| 字节数 | 4字节 | 2字节 | 15字节 | 4字节 |
| 取值 | LOCAL\_UDP\_VERSION | kSearchDeviceAnswer | ID | change |

说明：

1. ID表示设备ID；设备ID字符串不足15字节时，使用’\0’进行补充。
2. change表示设备信息改变标志。当控制卡中的设备信息发生改变时，该值会自动加一，上位机发现下位机的change字段发生改变后，应该重新获取设备基本信息。change的初始值为0。
3. 获取设备信息请求

#pragma pack(1)

//结构定义

typedef struct HUdpSDKCmd

{

huint32 version; ///< 版本号

huint16 cmd; ///< 命令值

hint8 id[MAX\_DEVICE\_ID\_LENGHT]; ///< 设备ID

hint8 xml[1] ///< sdk接口数据

} HUdpSDKCmdAsk, HUdpSDKCmdAnswer;

#pragma pack()

<!-- data的取值为 -->

<?xml version=**"1.0"** encoding=**"utf-8"**?>

<sdk guid=**"##GUID"**>

<in method=**"GetDeviceInfo"**/>

</sdk>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 版本号(Version) | 命令(Cmd) | 设备ID(ID) | SDK接口数据 |
| 字节数 | 4字节 | 2字节 | 15字节 | N字节 |
| 取值 | LOCAL\_UDP\_VERSION | kSDKCmdAsk | ID | data |

1. 获取设备信息反馈

<!-- data的取值为 -->

<?xml version=**"1.0"** encoding=**"utf-8"**?>

<sdk guid=**"##GUID"**>

<out method=**"GetDeviceInfo"** result=**"##result"**>

<device cpu=**"##cpu"** model=**"##model"** id=**"##id"** name=**"##name"**/>

<version fpga=**"##fpga"** app=**"##app"** kernel=**"##kernel"**/>

<screen width=**"##width"** rotation=**"##rotation"** height=**"##height"**/>

</out>

</sdk>

<!--

详情:

1. sdk.guid固定值为"##GUID"

2. out.method固定值为"GetDeviceInfo"

3. out.result取值参考<<SDK协议流程.docx>>文档中的2.3小结(错误码定义)

4. out.result取值不为"kSuccess"时, out无子节点

5. device.cpu: 设备cpu类型, 取值范围

{"Freescale.iMax6", "ZTE.902c", "TI.335x"}

6. device.model: 设备型号, 取值范围

{"A601", "A602", "A603", "D1", "D3", "C1i", "C3i", "C10",

"C30", "D10", "D20", "D30", "A30", "A30+", "C10+", "C30+"}

7. device.id: 设备ID

8. device.name: 设备名

9. version.fpga: fpga固件的版本号

10.version.app: 应用版本号

11.version.kernel: 系统版本号

12.screen.width: 显示屏宽度 (单位像素)

13.screen.height: 显示屏高度 (单位像素)

14.screen.rotation: 显示屏旋转标志, 取值范围{"0", "90", "180", "270"}

-->

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 版本号(Version) | 命令(Cmd) | 设备ID(ID) | SDK接口数据 |
| 字节数 | 4字节 | 2字节 | 15字节 | N字节 |
| 取值 | LOCAL\_UDP\_VERSION | kSDKCmdAnswer | ID | data |

## 有线网络信息

1. 获取有线网络地址信息请求(参考2.1找卡中“获取设备信息请求”小结)

<!-- data的取值为 -->

<?xml version=**"1.0"** encoding=**"utf-8"**?>

<sdk guid=**"##GUID"**>

<in method=**"GetEth0Info"**/>

</sdk>

1. 获取有线网络地址信息应答(参考2.1找卡中“获取设备信息反馈”小结)

<!-- data的取值为 -->

<?xml version=**"1.0"** encoding=**"utf-8"**?>

<sdk guid=**"##GUID"**>

<out method=**"GetEth0Info"** result=**"##result"**>

<eth valid=**"##valid"**>

<enable value=**"##enable"**/>

<dhcp auto=**"##dhcp"**/>

<address ip=**"##ip"** netmask=**"##netmask"** gateway=**"##gateway"** dns=**"##dns"**/>

</eth>

</out>

</sdk>

<!--

详情:

1. sdk.guid固定值为"##GUID"

2. out.method固定值为"GetEth0Info"

3. out.result取值参考<<SDK协议流程.docx>>文档中的2.3小结(错误码定义)

4. out.result取值不为"kSuccess"时, out无子节点

5. eth.valid固定值为"true"

6. enable.value取值范围{"true"(有线网络接入), "false"(有线网络未接入)}

7. dhcp.auto取值范围{"true"(dhcp获取ip地址), "false"(静态ip地址)}

8. address.ip: 有线网络ip地址

9. address.netmask: 有线网络子网掩码

10.address.gateway: 默认路由

11.address.dns: dns服务器地址

-->

1. 设置有线网络地址请求(参考2.1找卡中“获取设备信息请求”小结)

<!-- data的取值为 -->

<?xml version=**"1.0"** encoding=**"utf-8"**?>

<sdk>

<in method=**"SetEth0Info"**>

<eth valid=**"##valid"**>

<enable value=**"##enable"**/>

<dhcp auto=**"##dhcp"**/>

<address ip=**"##ip"** netmask=**"##netmask"** gateway=**"##gateway"** dns=**"##dns"**/>

</eth>

</in>

</sdk>

<!--

详情:

1. sdk.guid固定值为"##GUID"

2. in.method固定值为"SetEth0Info"

3. eth.valid固定值为"true"

4. enable.value固定值为"true"

5. dhcp.auto取值范围{"true"(dhcp获取ip地址), "false"(静态ip地址)}

6. dhcp.auto取值为"true"时, address中的属性值{ip, netmask, gateway, dns}均可以设置为""

7. dhcp.auto取值为"false"时, address中的属性值{ip, netmask, gateway, dns}必须设置

8. address.ip: 有线网络ip地址

9. address.netmask: 有线网络子网掩码

10.address.gateway: 默认路由

11.address.dns: dns服务器地址

-->

1. 设置有线网络地址反馈(参考2.1找卡中“获取设备信息反馈”小结)

<!-- data的取值为 -->

<?xml version=**"1.0"** encoding=**"utf-8"**?>

<sdk guid=**"##GUID"**>

<out method=**"SetEth0Info"** result=**"##result"**/>

</sdk>

<!--

详情:

1. sdk.guid固定值为"##GUID"

2. out.method固定值为"GetEth0Info"

3. out.result取值参考<<SDK协议流程.docx>>文档中的2.3小结(错误码定义)

-->

## 3/4G信息

1. 获取3/4G信息请求(参考<<SDK协议流程.docx>>中4.2SDK协议交互小结)

<!-- data的取值为 -->

<?xml version=**"1.0"** encoding=**"utf-8"**?>

<sdk guid=**"##GUID"**>

<in method=**"GetPppoeInfo"**/>

</sdk>

<!--

详解:

1. sdk.guid为通过SDK协议版本协商交互获取到的guid值

-->

1. 获取3/4G信息应答(参考<<SDK协议流程.docx>>中4.2SDK协议交互小结)

<!-- data的取值为 -->

<?xml version=**"1.0"** encoding=**"utf-8"**?>

<sdk guid=**"##GUID"**>

<out method=**"GetPppoeInfo"** result=**"##value"**>

<pppoe valid=**"##value"**>

<enable value=**"##value"**/>

<apn value=**"##value"**/>

<manufacturer value=**"##value"**/>

<version value=**"##value"**/>

<model value=**"##value"**/>

<imei value=**"##value"**/>

<number value=**"##value"**/>

<operators value=**"##value"**/>

<signal value=**"##value"**/>

<dbm value=**"##value"**/>

<insert value=**"##value"**/>

<status value=**"##value"**/>

<network value=**"##value"**/>

<code value=**"##value"**/>

</pppoe>

</out>

</sdk>

<!--

详解:

1. sdk.guid为通过SDK协议版本协商交互获取到的guid值

2. out.method固定值为GetPppoeInfo

3. out.result取值参考<<SDK协议流程.docx>>文档中的2.3小结(错误码定义)

4. out.result取值不为"kSuccess"时, out无子节点

5. pppoe.valid取值范围{"true"(有3/4G模块接入), "false"(无3/4G模块接入)}; pppoe.valid为"false"时无子节点

6. enable.value取值范围{"true"(有3/4G网络接入), "false"(无3/4G网络接入)}

7. apn.value: apn

8. manufacturer.value: 模块生产商

9. version.value: 模块版本

10.model.value: 模块型号

11.imei.value: 模块IMEI

12.number.value: SIM卡电话号码

13.operators.value: 运营商

14.signal.value: 信号强度, 取值范围[1, 5]; 1表示信号强度最差; 5表示信号强度最好

15.dbm.value: 信号强度 (单位dbm)

16.insert.value: SIM卡是否插入, 取值范围{"true"(有SIM卡插入), "false"(无SIM卡插入)}

17.status.value: 网络注册状态, 取值范围

{"unregister"(未注册), "register local"(已注册, 本地网络), "searching"(搜索中), "reject"(拒绝注册), "unknow"(未知错误),

"register roaming"(已注册, 漫游网络), "init"(初始化状态)}

18.network.value: 网络制式, 取值范围

{"init"(初始化状态), "unknow"(未知网络), "2G"(2G), "2.5G"(2.5G), "3GPP"(3GPP家族), "3G TD"(移动3G), "3.5G HSDPA",

"3.5G HSUPA", "3.5G HSPAPlus", "4G LTE", "4G TDD", "4G FDD"()}

19.code.value: 错误码(保留)

-->

1. 设置APN请求(参考<<SDK协议流程.docx>>中4.2SDK协议交互小结)

<!-- data的取值为 -->

<?xml version=**"1.0"** encoding=**"utf-8"**?>

<sdk guid=**"##GUID"**>

<in method=**"SetApn"**>

<apn value=**"##value"**/>

</in>

</sdk>

<!--

详解:

1. sdk.guid为通过SDK协议版本协商交互获取到的guid值

2. apn.value: apn

-->

1. 设置APN应答(参考<<SDK协议流程.docx>>中4.2SDK协议交互小结)

<!-- data的取值为 -->

<?xml version=**"1.0"** encoding=**"utf-8"**?>

<sdk guid=**"##GUID"**>

<out method=**"SetApn"** result=**"##result"**/>

</sdk>

<!--

详情:

1. sdk.guid为通过SDK协议版本协商交互获取到的guid值

2. out.method固定值为"SetApn"

3. out.result取值参考<<SDK协议流程.docx>>文档中的2.3小结(错误码定义)

-->

## WIFI

1. 获取WIFI信息请求(参考<<SDK协议流程.docx>>中4.2SDK协议交互小结)

<!-- data的取值为 -->

<?xml version=**"1.0"** encoding=**"utf-8"**?>

<sdk guid=**"##GUID"**>

<in method=**"GetWifiInfo"**/>

</sdk>

<!--

详解:

1. sdk.guid为通过SDK协议版本协商交互获取到的guid值

-->

1. 获取WIFI信息应答(参考<<SDK协议流程.docx>>中4.2SDK协议交互小结)

<!-- data的取值为 -->

<?xml version=**"1.0"** encoding=**"utf-8"**?>

<sdk guid=**"##GUID"**>

<out method=**"GetWifiInfo"** result=**"##value"**>

<wifi valid=**"##value"**>

<enable value=**"##value"**/>

<mode value=**"##value"**/>

<ap>

<ssid value=**"##value"**/>

<passwd value=**"##value"**/>

<channel value=**"##value"**/>

<encryption value=**"##value"**/>

<dhcp auto=**"##value"**/>

<address ip=**"##value"** netmask=**"##value"** gateway=**"##value"** dns=**"##value"**/>

</ap>

<station>

<current index=**"##value"**/>

<list>

<item>

<ssid value=**"##value"**/>

<passwd value=**"##value"**/>

<signal value=**"##value"**/>

<apmac value=**"##value"**/>

<dhcp auto=**"##value"**/>

<address ip=**"##value"** netmask=**"##value"** gateway=**"##value"** dns=**"##value"**/>

</item>

<list>

</station>

</wifi>

</out>

</sdk>

<!--

详解:

1. sdk.guid为通过SDK协议版本协商交互获取到的guid值

2. out.method固定值为GetWifiInfo

3. out.result取值参考<<SDK协议流程.docx>>文档中的2.3小结(错误码定义)

4. out.result取值不为"kSuccess"时, out无子节点

5. wifi.valid取值范围{"true"(有WIFI模块接入), "false"(无WIFI模块接入)}; wifi.valid为"false"时无子节点

6. enable.value取值范围{"true"(有WIFI网络接入), "false"(无WIFI网络接入)}

7. mode.value: wifi工作模式, 取值范围{"ap"(ap模式), "station"(客户端模式)}

8. ap.ssid.value: ssid, 固定值(设备ID)

9. ap.passwd.value: 密码

10.ap.channel.value: 信道

11.ap.encryption.value: 加密方式, 固定值"WPA-PSK"

12.ap.dhcp: 忽略

13.ap.address: 忽略

14.station.current.index: 表示当前使用的ap节点在list中的索引值, -1时表示未选中ap节点

15.station.list包含多个item, item表示wifi发现和已经保存的ap节点信息

16.station.list.item.ssid.value: ap的ssid

17.station.list.item.passwd.value: 密码

18.station.list.item.signal.value: 信号强度

19.station.list.item.apmac.value: ap设备的mac地址  
20.station.list.item.dhcp.auto: 取值范围{"true"(dhcp获取ip地址), "false"(静态ip地址)}

21.station.list.item.address.ip: WIFI网络ip地址

22.station.list.item.address.netmask: WIFI网络子网掩码

23.station.list.item.address.gateway: 默认路由

24.station.list.item.address.dns: dns服务器地址

-->

1. 设置WIFI模式请求(参考<<SDK协议流程.docx>>中4.2SDK协议交互小结)

<!-- data的取值为 -->

<?xml version=**"1.0"** encoding=**"utf-8"**?>

<sdk guid=**"##GUID"**>

<in method=**"SetWifiInfo"**>

<mode value=**"##value"**/>

<ap>

<ssid value=**"##value"**/>

<passwd value=**"##value"**/>

<channel value=**"##value"**/>

<encryption value=**"##value"**/>

<dhcp auto=**"##value"**/>

<address ip=**"##value"** netmask=**"##value"** gateway=**"##value"** dns=**"##value"**/>

</ap>

<station>

<ssid value=**"##value"**/>

<passwd value=**"##value"**/>

<mac value=**"##value"**/>

</station>

</in>

</sdk>

<!--

详解:

1. sdk.guid为通过SDK协议版本协商交互获取到的guid值

2. mode.value: wifi工作模式, 取值范围{"ap"(ap模式), "station"(客户端模式)}

3. ap.ssid.value: ssid, 固定值(设备ID)

4. ap.passwd.value: 密码

5. ap.channel.value: 信道

6. ap.encryption.value: 加密方式, 固定值"WPA-PSK"

7. ap.dhcp.auto: 固定值"true"

8. ap.addreess: 固定值<address ip="0.0.0.0" netmask="0.0.0.0" gateway="0.0.0.0" dns="0.0.0.0"/>

9. station.ssid.valud: ssid

10.station.passwd.valud: 密码

11.station.mac: ap设备的mac地址

-->

1. 设置WIFI模式应答(参考<<SDK协议流程.docx>>中4.2SDK协议交互小结)

<!-- data的取值为 -->

<?xml version=**"1.0"** encoding=**"utf-8"**?>

<sdk guid=**"##GUID"**>

<out method=**"SetWifiInfo"** result=**"##result"**/>

</sdk>

<!--

详情:

1. sdk.guid为通过SDK协议版本协商交互获取到的guid值

2. out.method固定值为"SetWifiInfo"

3. out.result取值参考<<SDK协议流程.docx>>文档中的2.3小结(错误码定义)

-->