|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 【密级程度】 公开 √ 机密 绝密 \_\_   |  | | --- | | **文件管理** |   文件编号：  版 本 号： V1.0  拟制日期： 2017-09-18  发布日期：   |  |  |  | | --- | --- | --- | | 批准 | 审核 | 编制 | |  |  | 罗清文 | | 日期： | 日期： | 日期：2017-09-17 |  |  |  | | --- | --- | | 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](#_Toc493671427)

[1. 概述 3](#_Toc493671428)

[1.1 主要功能 3](#_Toc493671429)

[1.2 下发一个文件的流程 3](#_Toc493671430)

[1.3 回读一个文件的流程 4](#_Toc493671431)

[2. 数据包定义 5](#_Toc493671432)

[2.1 文件下发 5](#_Toc493671433)

[2.2 文件回读 8](#_Toc493671434)

[2.3 回读文件列表 9](#_Toc493671435)

[2.4 删除文件 10](#_Toc493671436)

**文件管理**

1. 概述

## 主要功能

本文主要叙述与下位机文件管理相关的内容，支持图片、视频、配置文件、字体等文件的下发、回读以及删除；支持图片、视频、配置文件、字体等文件列表的回读。

## 下发一个文件的流程



说明：

1. 上位机和下位机建立好连接以及协商好传输协议版本后就可以进行文件传输了。
2. 首先上位机发送文件开始传输请求给下位机，通知下位机将要传输一个文件了。
3. 下位机将反馈信息发送给上位机，如果出错，上位机应该检查错误码并中断传输；否者，上位机可以发送文件内容给下位机了；
4. 如果下位机在接收上位机发送的文件内容时出错，那么下位机会发送错误码给上位机，上位机在此过程中接收到错误码时应检查错误码并中断传输；
5. 如果上位机在发送完文件内容后，未接收到下位机的错误码反馈，那么上位机应发送文件结束传输请求给下位机，下位机检查文件传输结果后，将检查的信息反馈给上位机；
6. 至此完成一个文件的一次下发操作。

## 回读一个文件的流程



说明：

1. 上位机和下位机建立好连接以及协商好传输协议版本后就可以进行文件回读了。
2. 首先上位机发送回读文件开始传输请求给下位机，通知下位机将要回读一个文件了。
3. 如果下位机找到了对应的文件，那么下位机会将对应文件的基本信息反馈给上位机。
4. 如果下位机未找到对应的文件，那么下位机将反馈错误码给上位机。
5. 上位机接收到错误码的反馈信息后应该检查错误码并中断数据传输。
6. 上位机接收到下位机反馈的回读文件的基本信息后，发送确认回读信息给下位机，下位机将开始发送文件内容给上位机。
7. 如果上位机在接收文件内容出错，那么上位机可以发送错误码给下位机，终止下位机的文件内容传输。
8. 如果下位机在发送完文件内容后，未接收到上位机的错误码反馈，那么下位机会发送文件结束传输请求给上位机，上位机检查文件传输结果后，将检查信息反馈给下位机。
9. 至此完成一个文件的一次回读操作。
10. 数据包定义

## 文件下发

1. 文件开始传输请求

#pragma pack(1)

//结构定义

typedef struct HFileStartAsk

{

huint16 len; ///< 命令包长度

huint16 cmd; ///< 命令值

hint8 md5[MD5\_LENGHT + 1]; ///< 文件md5值

huint64 size; ///< 文件大小

hint16 type; ///< 文件类型

hint8 name[2]; ///< 文件名

} HFileStartAsk;

#pragma pack()

/\*\* 字段说明

\* 1. Len = 2 + 2 + 33 + 8 + 2 + nameLen;

\* 2. md5表示待下发文件的md5值, md5值中的a-f字符使用小写字母, 例如: "239a1b2d06d959aca2676bd0103e60e0"

\* 3. size表示文件大小 (单位: 字节)

\* 4. type取值范围{0(图片), 1(视频), 2(字体), 3(固件), 4(基本配置参数), 5(FPG配置参数)}

\* 5. name表示文件名, 在节目编辑、文件管理使用时, 用来定位文件, 因此用户需要保证该文件名的唯一性.文件名末尾需要添加’\0’.

\*/

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 长度(Len) | 命令(Cmd) | MD5值 | 文件大小 | 文件类型 | 文件名 |
| 字节数 | 2字节 | 2字节 | 33字节 | 8字节 | 2字节 | 2-256字节 |
| 取值 | Len | kFileStartAsk | md5 | size | type | name |

1. 文件开始传输应答

#pragma pack(1)

//结构定义

typedef struct HFileStartAnswer

{

huint16 len; ///< 命令包长度

huint16 cmd; ///< 命令值

hint16 status; ///< 错误码

huint64 existSize; ///< 文件在下位机存储的大小

} HFileStartAnswer;

#pragma pack()

/\*\* 字段说明

\* 1. status表示反馈的错误码, 参考<<SDK协议流程.docx>> 2.3错误码定义小结

\* 2. 如果status为非kSuccess时, 上位机应检查错误码并中断传输

\* 3. 如果status为kSuccess时, 此时existSize生效

\* 4. existSize表示文件在下位机存储的大小; 如果文件不存在时, 该值为0

\*/

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 长度(Len) | 命令(Cmd) | 错误码 | 已存在大小 |
| 字节数 | 2字节 | 2字节 | 2字节 | 8字节 |
| 取值 | 8 | kFileStartAnswer | status | existSize |

1. 文件内容传输请求

#pragma pack(1)

//结构定义

typedef struct HFileContentAsk

{

huint16 len; ///< 命令包长度

huint16 cmd; ///< 命令值

hint8 data[1]; ///< 文件内容

} HFileContentAsk;

#pragma pack()

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 长度(Len) | 命令(Cmd) | 文件内容 |
| 字节数 | 2字节 | 2字节 | 1-(9\*1024-4)字节 |
| 取值 | Len | kFileContentAsk | data |

1. 文件结束传输请求

#pragma pack(1)

//结构定义

typedef struct HFileEndAsk

{

huint16 len; ///< 命令包长度

huint16 cmd; ///< 命令值

} HFileEndAsk;

#pragma pack()

|  |  |  |
| --- | --- | --- |
| 字段 | 长度(Len) | 命令(Cmd) |
| 字节数 | 2字节 | 2字节 |
| 取值 | 4 | kFileEndAsk |

1. 文件结束传输应答

#pragma pack(1)

//结构定义

typedef struct HFileEndAsk

{

huint16 len; ///< 命令包长度

huint16 cmd; ///< 命令值

huint16 status; ///< 错误码

} HFileEndAsk;

#pragma pack()

/\*\* 字段说明

\* 1. status表示反馈的错误码, 参考<<SDK协议流程.docx>> 2.3错误码定义小结

\*/

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 长度(Len) | 命令(Cmd) | 错误码 |
| 字节数 | 2字节 | 2字节 | 2字节 |
| 取值 | 6 | kFileEndAnswer | status |

## 文件回读

1. 回读文件传输请求

#pragma pack(1)

//结构定义

typedef struct HReadFileAsk

{

huint16 len; ///< 命令包长度

huint16 cmd; ///< 命令值

hint8 name[2]; ///< 文件名

} HReadFileAsk;

#pragma pack()

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 长度(Len) | 命令(Cmd) | 文件名 |
| 字节数 | 2字节 | 2字节 | 2-256字节 |
| 取值 | Len | kReadFileAsk | name |

1. 文件基本信息应答

#pragma pack(1)

//结构定义

typedef struct HReadFileAnswer

{

huint16 len; ///< 命令包长度

huint16 cmd; ///< 命令值

hint16 status; ///< 错误码

hint8 md5[MD5\_LENGHT + 1]; ///< 文件md5值

huint64 size; ///< 文件大小

hint16 type; ///< 文件类型

} HReadFileAnswer;

#pragma pack()

/\*\* 字段说明

\* 1. status表示错误码, 参考<<SDK协议流程.docx>> 2.3错误码定义小结

\* 2. md5表示待下发文件的md5值, md5值中的a-f字符使用小写字母, 例如: "239a1b2d06d959aca2676bd0103e60e0"

\* 3. size表示文件大小 (单位: 字节)

\* 4. type取值范围{0(图片), 1(视频), 2(字体), 3(固件), 4(基本配置参数), 5(FPG配置参数)}

\*/

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 长度(Len) | 命令(Cmd) | 错误码 | MD5值 | 文件大小 | 文件类型 |
| 字节数 | 2字节 | 2字节 | 2字节 | 33字节 | 8字节 | 2字节 |
| 取值 | 49 | kReadFileAnswer | status | md5 | size | type |

1. 文件基本信息确认(参考2.1文件下发中的“文件开始传输应答”)
2. 文件内容传输请求(参考2.1文件下发中的“文件内容传输请求”)
3. 文件结束传输请求(参考2.1文件下发中的“文件结束传输请求”)
4. 文件结束传输应答(参考2.1文件下发中的“文件结束传输应答”)

## 回读文件列表

1. 回读文件列表请求(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=**"GetFiles"**/>

</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. 回读文件列表反馈(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=**"GetFiles"** result=**"##result"**>

<files>

<file size=**"##value"** existSize=**"##value"** md5=**"##value"** type=**"##value"** name=**"##value"**/>

</files>

</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. files标签可包含多个file子节点

6. file.size: 文件的实际大小 (单位字节)

7. file.existSize: 文件在下位机存储的大小 (单位字节)

8. 如果file.existSize 小于 file.size时表示该文件未传输完成

9. 如果file.existSize 等于 file.size时表示该文件已完成传输

10.如果file.existSize 大于 file.size时表示该文件出错了, 用户应该调用删除接口将该文件删除并重传

11.file.md5: 文件的md5值. 例如: "ce3002778cb9e5e973a9301f81aef64e"

12.file.type: 文件类型, 取值范围{"image"(图片), "video"(视频), "font"(字体), "fireware"(固件), "setting"(基本固件参数), "fpga"(FPGA配置参数)}

-->

## 删除文件

1. 删除文件请求(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=**"DeleteFiles"**>

<files>

<file name=**"##value"**/>

</files>

</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. files标签可包含多个file子节点

3. file.name: 文件名

-->

1. 删除文件反馈(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=**"DeleteFiles"** result=**"##result"**>

<files>

<file name=**"##value"** result=**"##result"**/>

</files>

</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"GetBootLogo"

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. 当指定被删除的文件均正确删除后, files无子节点, 否者则存在子节点点并携带对应的出错信息

6. files标签可包含多个file子节点

7. file.name: 文件名

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

-->