DoubleCamera设计说明书

拟 写： 日期：

审评人： 日期：

批 准： 日期：

文件控制

**变更记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 修改人 | 版本 | 更改说明 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**审阅**

|  |  |  |
| --- | --- | --- |
| 日期 | 审阅者 | 意见 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

目录

[DoubleCamera设计说明书 1](#_Toc471480401)

[1．引言 5](#_Toc471480402)

[1.1 编写目的 5](#_Toc471480403)

[1.2 背景 5](#_Toc471480404)

[1.3 参考资料 5](#_Toc471480405)

[1.4 术语说明 5](#_Toc471480406)

[2. 设计概述 6](#_Toc471480407)

[2.1任务和目标 6](#_Toc471480408)

[2.2设计要求 6](#_Toc471480409)

[3.需求规定 7](#_Toc471480410)

[3.1界面要求 7](#_Toc471480411)

[3.2 功能要求 8](#_Toc471480412)

[3.3 通信协议 10](#_Toc471480413)

[4.运行环境规定 11](#_Toc471480414)

[4.1设备 11](#_Toc471480415)

[4.2 接口 11](#_Toc471480416)

[5.代码设计 12](#_Toc471480417)

[5.1 主要技术 12](#_Toc471480418)

[5.2设计难点 12](#_Toc471480419)

[5.2软件模块 12](#_Toc471480420)

[5.3 代码流程 15](#_Toc471480421)

[5.4 主要方法和参数 18](#_Toc471480422)

[6．实施计划 23](#_Toc471480423)

[6.1 限制 23](#_Toc471480424)

[6.2实施内容和进度安排 23](#_Toc471480425)

[6.3实施条件和措施 24](#_Toc471480426)

[7．测试计划 25](#_Toc471480427)

[7.1 测试方案 25](#_Toc471480428)

[7.2 测试结果 25](#_Toc471480429)

[7.3 修改内容 28](#_Toc471480430)

[8.验收标准 28](#_Toc471480431)

# 1．引言

## 1.1 编写目的

本文档主要对DoubleCamera项目做一些详细的说明和讲解，使开发人员能够明白本项目的设计思路和步骤，以便于日后的修改和深入拓展。

文档主要包括：APP中实现的功能、控制指令、代码的设计模块、主要应用技术、初步的测试、性能需求等。

## 1.2 背景

DoubleCamera，主要用于娱乐和无人机航拍，适用于12岁以上人群。无人机上的摄像头提供WIFI热点，通过该热点建立起无人机和智能手机的连接通道，打开手机上安装的APP，获取WIFI摄像头的视频影像等信息的同时，也可对无人机的飞行进行操控。WIFI摄像头含有两个服务器，一个是用来传输视频的视频服务器（RTSP协议）；另一个是用来传输命令的UDP协议。

WIFI摄像头的SSID：以Skycam开头，例如：Skycam\_ab-9f-f4。

密码：12345678

开发人员：王磊、宋伟

项目的委托单位和开发单位：武汉新联空间信息技术有限公司

## 1.3 参考资料

1. RTSP协议
2. UDP协议
3. FFMPEG库

## 1.4 术语说明

# 2. 设计概述

## 2.1任务和目标

根据需求文档，在规定时间内完成APP的开发，要求做到功能完善，操作界面简洁明了，易于操作。对于控制指令的发送，影像的拍照、录像，画面的延时，是否有马赛克等方面需做好测试记录。

## 2.2设计要求

主要包括：

1. 界面要求；

APP有四个摄像头模式：广角摄像头；非广角摄像头；主屏广角，副屏非广角；主屏非广角，副屏广角。

主屏幕显示的画面为720P。

1. 功能需求；

在线OTA升级、摄像头版本查询；影像的拍照、录像等；通过摇杆控制无人机的飞行。

1. 性能需求。

摄像头模式切换时，等待约3s，进入下一模式；画面长时间静止时，能够正常发送和接收数据；画面延时在500ms以内；影像无马赛克和破图现象，距离过远时影像会卡死，但在距离靠近时，影像恢复。

# 3.需求规定

## C:\Users\Administrator\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Word\Screenshot_2016-12-26-15-48-30.jpeg3.1界面要求

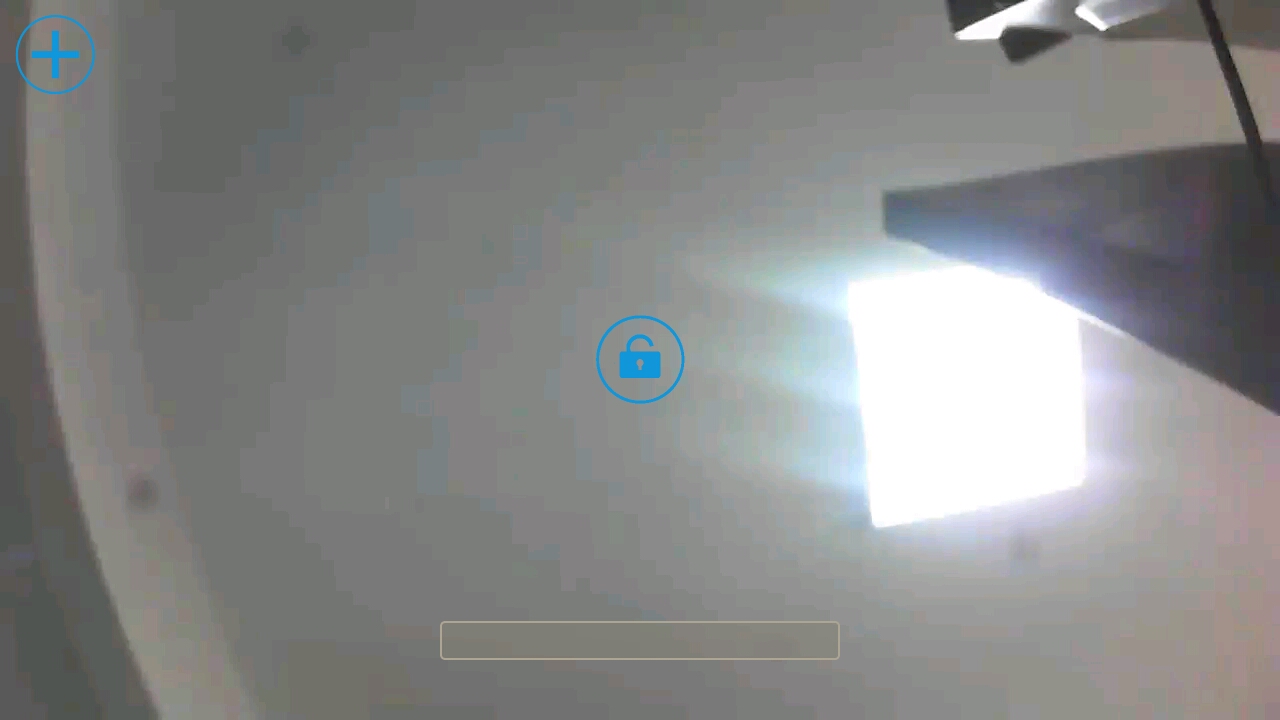
图3.1

图3.2

图3.3

图3.1为刚进入APP时的界面，在WIFI未连接或连接的不是WIFI摄像头的热点时，点击START按钮不会进入图3.2，而是会弹出网络设置Dialog，点击“确定”进入Android系统的WLAN设置界面，连接正确WIFI后点击“返回”，退出到图3.1。

点击START按钮，进入图3.2，图3.2背景为摄像头的实时影像，左上角按钮为收放按钮， 可以对菜单栏和模式选择栏进行显示和隐藏操作（如图3.3）。

左右摇杆分别控制飞机的油门和方向，默认为隐藏，当手指接触屏幕时摇杆显示，手指离开屏幕时摇杆隐藏。底部长方形半透明框用来接收摄像头返回的数据。当屏幕超过60s未被触碰时，所有按钮会隐藏，触控屏幕，按钮显示。

## 3.2 功能要求

|  |  |  |
| --- | --- | --- |
| /Users/zhushuai/.Trash/屏幕快照 2016-12-13 下午5.08.40.png | 帮助 | 点击后进入操作讲解界面。 |
| **/Users/zhushuai/Desktop/屏幕快照 2016-12-13 下午5.17.13.png** | 设置 | 点击后进入设置界面：   1. 设置网络－点击后到手机的网络设置中，选择WIFI摄像头的WIFI； 2. 无头模式－开启或关闭无头模式； 3. 影像翻转－设置手机影像180度翻转； 4. 右手操作－切换手机油门于左摇杆或右摇杆； 5. 版本号查询－查询硬件的版本参数；   （6）OTA版本升级－一键升级摄像头版本。 |
| /Users/zhushuai/Desktop/屏幕快照 2016-12-13 下午5.19.55.png | 文件夹 | 查看摄像头拍摄的照片和录制的视频。 |
| /Users/zhushuai/Desktop/屏幕快照 2016-12-13 下午5.22.13.png | 模式选择 | 选择影像传输的模式，UDP模式和TCP模式，用于测试哪种模式下影像传输的效果更好。默认为UDP |
| D:\workspace1\DoubleCamera20161222\res\drawable-hdpi\menu00_ov.png | 返回 | 返回到主界面（图3.1）。 |
| D:\workspace1\DoubleCamera20161222\res\drawable-hdpi\menu01_ov.png | 拍照 | 获取手机端当前的视频影像，保存到手机或SD卡中。 |
| D:\workspace1\DoubleCamera20161222\res\drawable-hdpi\menu02_ov.png | 手机录像 | 录制手机端收到的视频，保存到手机中。 |
| D:\workspace1\DoubleCamera20161222\res\drawable-hdpi\menu03_ov.png | SD卡录像 | 录制手机端收到的视频，保存到SD卡中。 |
| **D:\workspace1\DoubleCamera20161222\res\drawable-hdpi\menu04_ov.png** | 文件 | 查看摄像头拍摄的照片和录制的视频。 |
| **D:\workspace1\DoubleCamera20161222\res\drawable-hdpi\menu08_ov.png** | 微调 | 显示或隐藏微调 |
| **D:\workspace1\DoubleCamera20161222\res\drawable-hdpi\menu10_ov.png** | 设置 | 1. 一键平衡 2. 紧急停止 3. 定高模式－定高模式的开启或关闭，若开启，中间则放一个解锁按钮； 4. 无头模式－开启或关闭无头模式； 5. 影像翻转－设置手机影像翻转180度； 6. 操作模式—设置左手或右手操作 7. 存储位置—设置存储在SD卡或手机中 |
| **C:\Users\Administrator\Desktop\rocker_bg1.png** | 摇杆 | 按住中间的按钮，可上下左右随意移动；在摇杆上方显示相对的坐标，并将坐标数据发送给WIFI摄像头。  摇杆1-油门＋方向舵  摇杆2-副翼（侧飞）＋升降舵（前后） |
| **C:\Users\Administrator\Desktop\trimbar_bg1.png** | 微调 | 细微调整遥控数据，按两端的按钮可进行微调，按一次其调整值为+/-2，并会发出按键声音，最高限制值为输出值的50%。主要是来调节偏航微调，俯仰微调，侧滚微调。如果按钮调整至中间时，则会发出归零声音。 |

## 3.3 通信协议

串口发送数据的格式（串口波特率19200，一个起始位，一个停止位），APP每40ms发送一个数据包，每个包为8个字节。

|  |  |  |
| --- | --- | --- |
| BYTE[0] | 数据头 | 固定为0x66。 |
| BYTE[1] | 侧滚 roll  AIL副翼 | 数据范围0x00-0xff，中间值0x7f，最左边为0x00，最右为0xff。 |
| BYTE[2] | 俯仰 pitch  ELE 升降舵 | 数据范围0x00-0xff，中间值0x7f，后最大为0x00，前最大为0xff。 |
| BYTE[3] | 油门 throttle | 数据范围0x00-0xff，中间值0x7f，最下为0x00，最上为0xff。 |
| BYTE[4] | 偏航 yaw  方向舵 | 数据范围0x00-0xff，中间值0x7f，最左边为0x00，最右边0xff。 |
| BYTE[5] | 标志位 | Bit0－保留位；  Bit1－一键起飞（油门解锁），先置1，1s后置0；  Bit2－一键着陆，先置1，1s后置0；  Bit3－飞行器紧急停止，先置1，1s后置0；  Bit4－定高模式开启或关闭；  Bit5－无头模式的开启或关闭；  Bit6－一键翻转，点击后置1，当方向键超过一半时置0；  Bit7－一键平衡，先置1，1s后置0，油门为0时才能送出平衡值。 |
| BYTE[6] |  | （BYTE[1]^BYTE[2]^BYTE[3]^BYTE[4]^BYTE[5]）&0xff |
| BYTE[7] | 帧尾 | 固定为0x99 |

# 4.运行环境规定

## 4.1设备

通过UART命令实现数据透传。APP发送的遥控数据，WIFI摄像头接受后，不做任何处理，直接透过UART接口发送给飞控板。

## 4.2 接口

手机和WIFI摄像头连接通过http的请求连接；控制命令的发送要求有返回值，改为UDP发送。

# 5.代码设计

## 5.1 主要技术

1. 手机APP和WIFI摄像头的连接；
2. 手机APP给WIFI摄像头的数据发送以及接受接受返回数据；
3. 通过手机给WIFI摄像头上传.bin文件实现摄像头升级；
4. FFmpeg解码实现影像的实时解码和绘制。

## 5.2设计难点

1. 摄像头利用WIFI传输数据，距离越远，出现画面卡顿或卡死的几率越大。而当画面卡死后，当摄像头向手机靠拢时，画面恢复且画面延时不超过500ms。
2. FFmpeg库针对安卓不同型号手机和不同系统版本的优化适配问题。

## 5.2软件模块

该APP按功能划分，大致可分文两个模块，控制命令模块（需要实现的飞控相关功能）和视频图传模块（主要和FFmpeg库的编解码有关）。控制命令和视频图传之间不存在耦合关系，两个模块各自独立，这样可提高代码的可读性、维护性和方便测试。

1. 视频图传

（1）720P画面的显示

获取控制权限，得到返回值sAuthcode

值

利用sAuthcode的值拼接720p的URL

发送720P的请求命令，返回值为1表示请求成功

解码器获取摄像头的path，获取图像数据解码

实时显示解码视频

（2）拍照

调用RtspVideoView中绘图线程DrawThread，将当前流信息绘制的画面保存在指定路径。

（3）录像

录像分为两种，一种为手机录像，另一种为调用无人机上自带的SD卡录像。

手机录像，直接调用FFmpeg中的IsRecord（），从视频流中读出相应数据流packet，然后将packet进行解码，存储在指定手机内存的指定路径。

SD卡录像，开启子线程，向摄像头发送http请求，摄像头接收请求后，会将相关视频数据保存在指定SD卡路径。

（4）FFmpeg解码流程

从文件中提取流信息av\_find\_stream\_info()

打开文件：av\_open\_input\_

file()

注册所有容器格式和CODEC：av\_regist\_all()

打开编解码器：avcodec\_open()

查找对应的解码器：avcodec\_find\_decoder()

穷举所有的流，查找其中种类为CODEC\_TYPE\_VIDEO

为解码帧分配内存：avcodec\_alloc\_frame()

不停地从码流中提取出帧数据：av\_read\_frame()

判断帧的类型，对于视频帧调用：avcodec\_decode\_video()

关闭输入文件：av\_close\_input\_file()

解码完成后，释放解码器：avcodec\_close()

（5）版本号查询

APP设定版本号查询的方式为根据给定的url（http://192.168.100.1/information），向双摄像头发送http请求，摄像头的返回值为该固件的版本信息。当手机连接上摄像头的WIFI热点后，点击查询版本号信息，开启查询版本号的子线程，利用handler更新UI，通过Dialog显示出固件的版本。

（6）OTA升级

双摄像头为一个简单的服务器，当有OTA升级需求且手机连接上摄像头的WIFI热点时，开启子线程，利用FTP协议，将内置于开发项目中的升级资源文件，上传到服务器，并采用进度条的方式显示上传文件的进度。资源上传完成，自动开启新的子线程，通过发送http请求（url为http://192.168.100.1:80/server.command?command=get\_privilege）， 获取摄像头的权限，再利用获取权限的返回值sAuthcode，发送OTA升级的http请求，其url为http://192.168.100.1:80/server.command?command=ota2&authcode=sAuthcode，摄像头接收到http请求后，进行固件升级。升级过程中，摄像头的LED指示灯会进行300ms的闪烁，升级成功，摄像头自动重启。

（7）影像翻转

影像翻转功能同版本号查询类似，也是开启子线程，通过http发送翻转url（http://192.168.100.1:80/server.command?command=set\_flip\_mirror&value=value&authcode=sAuthcode），其中value值为0表示正常画面，1表示180°翻转，2表示镜面翻转，3表示180°同时镜面翻转；sAuthcode为获取权限的返回值。发送http请求后，手机接收到的画面就发生翻转。目前仅小镜头可以翻转，大镜头可能是因为硬件问题无法翻转。

1. 控制命令

（1）控制命令

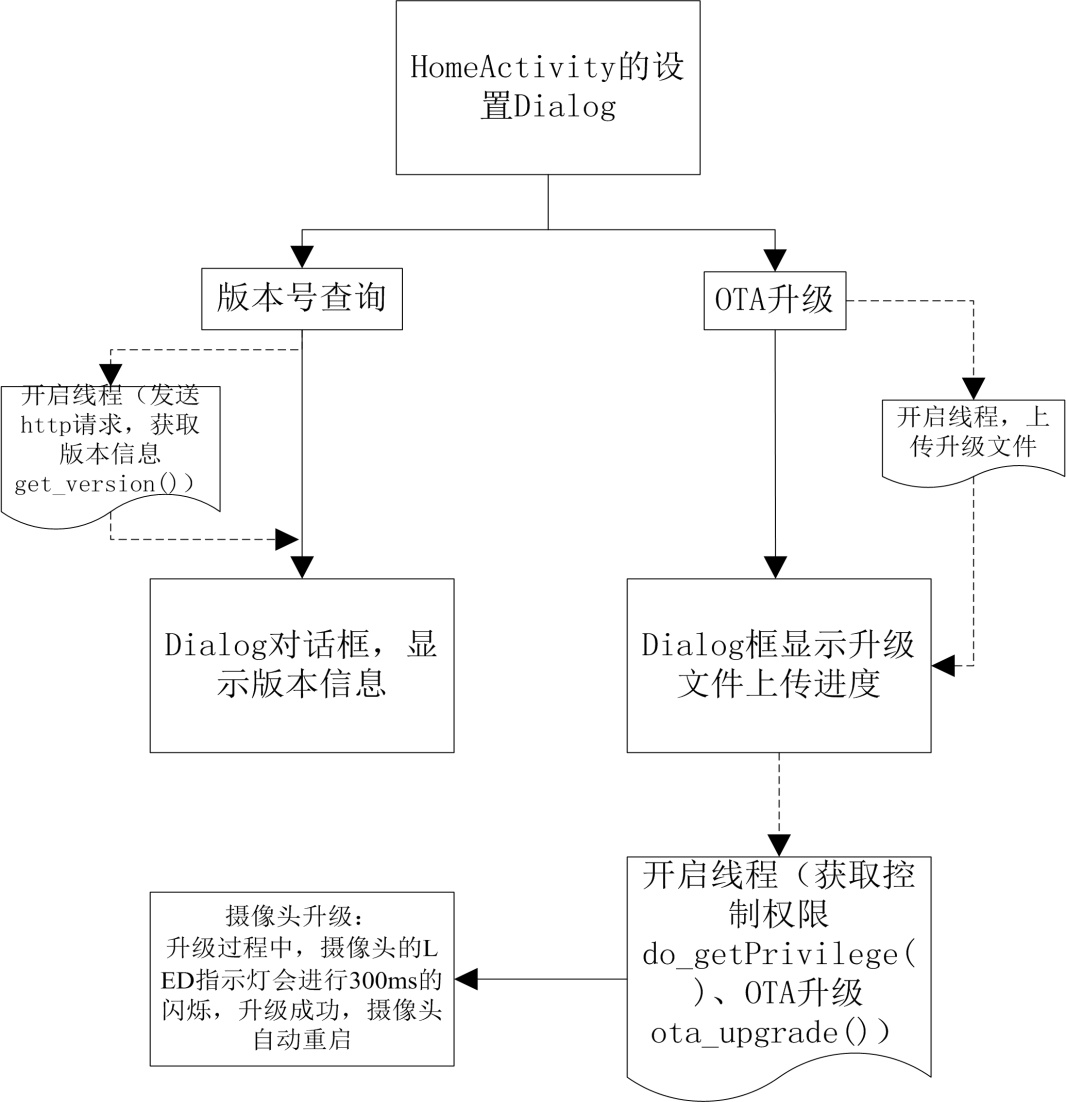
控制命令为本APP核心功能之一，可控制无人机的飞行方向、油门等，根据自定义飞行器控制协议，APP不断的同无人机发送数据包，数据包中包含有长度为8的byte[]，数组中每个位置的值用0到255的十六进制数表达，根据命令的不同，不同位置负责不同功能。如byte[0]为数据头，固定为0x66；byte[1]为副翼，中间值0x80，左边最大为00，右边最大为0xff，即00--0x80--0xff线性变换；byte[2]为升降舵，中间值0x80，前最大为0xff，后最大为0x00，即0x00--0x80--0xff线性变化；byte[3]表示油门00为最小，0xff为最大，油门平时置于底部，值为0x00，0x00--0x80--0xff线性变化；byte[4]表示方向中间值0x80，左转最大0x00，右转最大0xff，0x00--0x80--0xff线性变化；byte[5]表示标志位，bit0：保留；bit1:一键启动(自动起飞并定高）；bit2:一键着陆；bit3:飞行器紧急停止； bit4:保留 ；bit5:无头模式；bit6:一键翻滚；bit7:一键平衡；其余bit 保留；byte[6] =(BYTE[1]^BYTE[2]^BYTE[3]^BYTE[4]^BYTE[5])&0xff ；byte[7]为数据尾，固定为0x99。

（2）微调

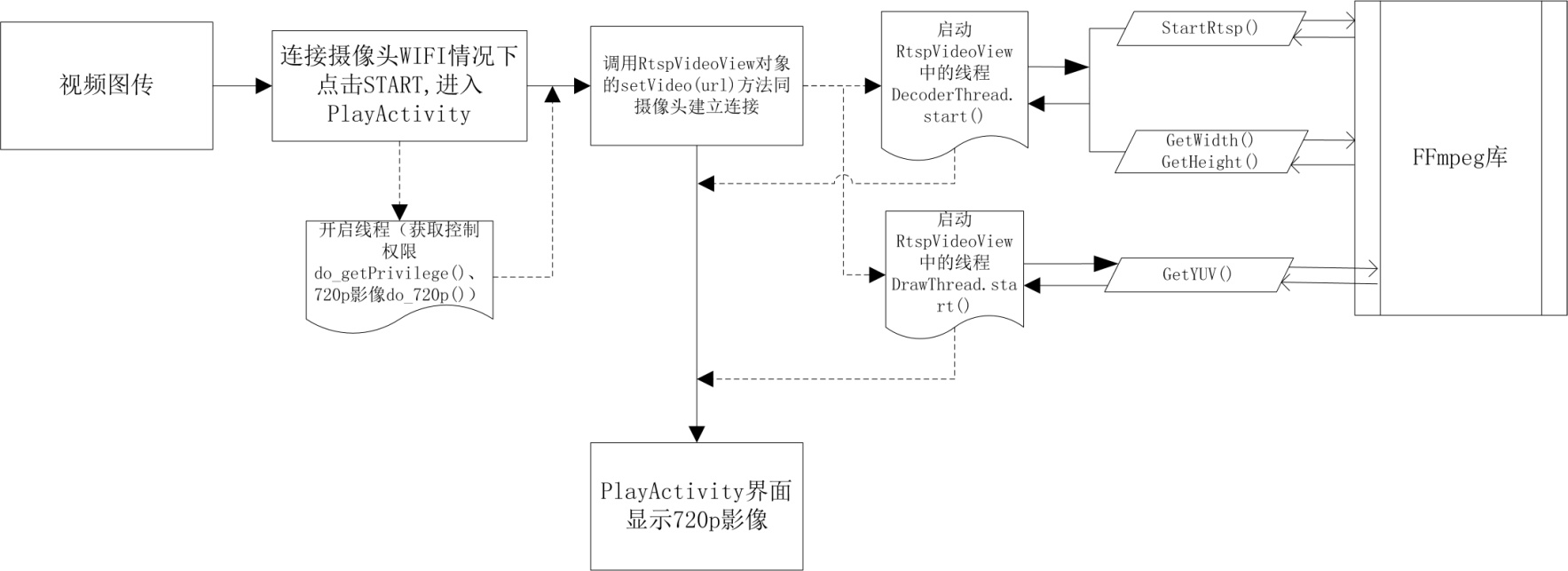
因手机界面操控无人机的屏幕和控制手感的局限性，加入微调功能，能够对无人机有更加精准的控制。微调功能主要是微调控无人机的方向、侧滚和俯仰情况。

## C:\Users\Administrator\Desktop\DoubleCamera流程图\DoubleCamera流程图1_主流程图.jpg5.3 代码流程

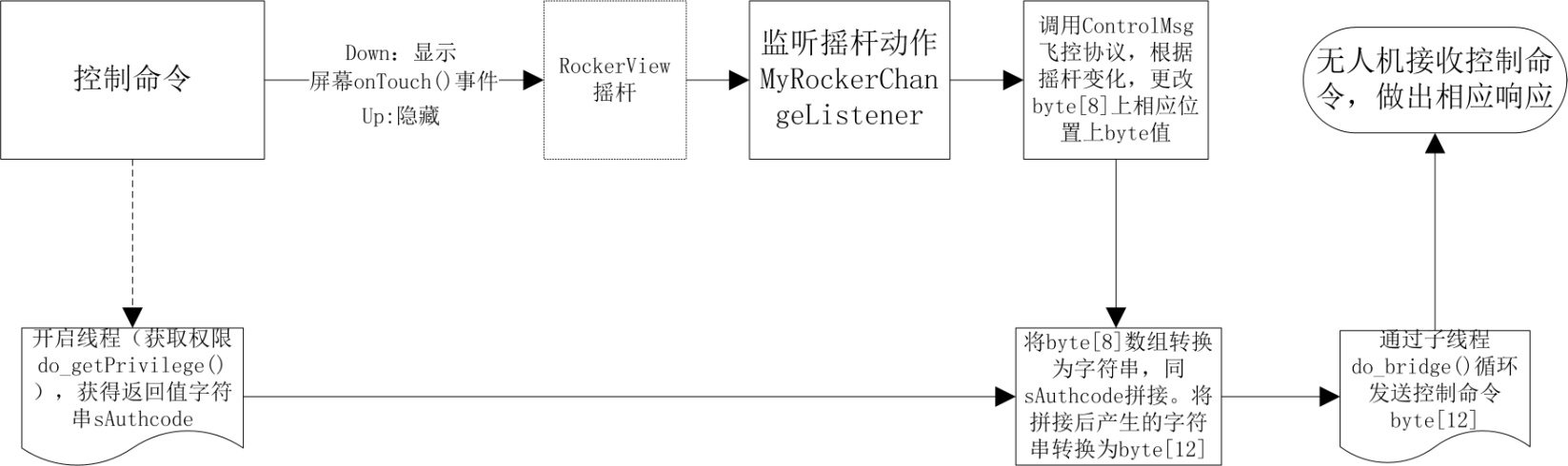
DoubleCamera主流程图

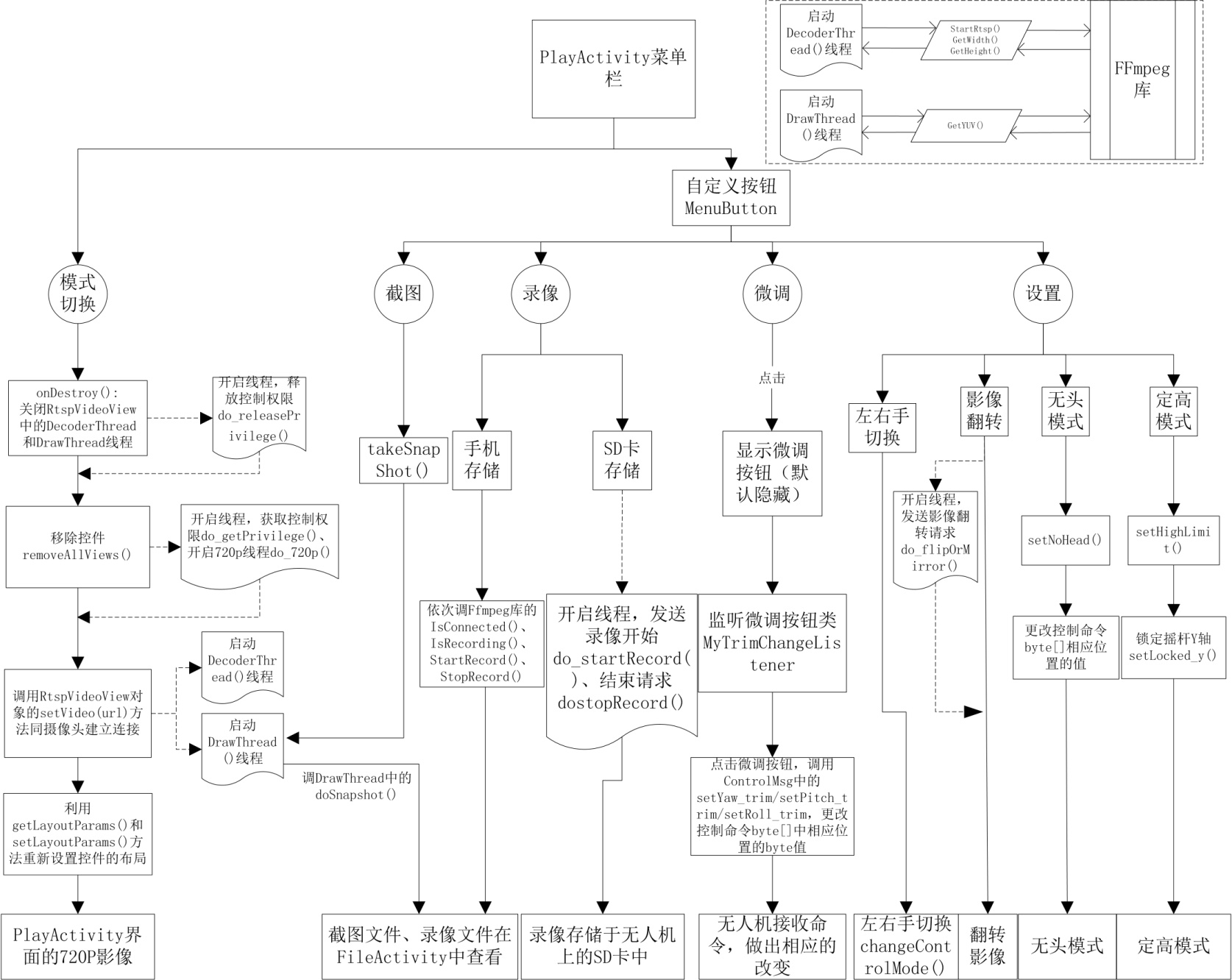


DoubleCamera版本信息查询、OTA升级流程图



DoubleCamera\_视频图传流程图

DoubleCamera控制命令流程图



DoubleCamera菜单栏流程图

## 5.4 主要方法和参数

列举代码中用到的主要方法和参数，并注明该方法和相应参数在代码中所起到的作用。以类为单位，其列举如下：

1. HomeActivity中主要涉及到的方法和参数有：
2. handler()，涉及到线程的开启，并将线程的返回值用于更新UI，例如在HomeActivity中查询版本号信息时，开启的版本信息线程，就是通过Bundle传值，利用handler()拿到值，并在Dialog中显示出来。
3. isWifiConnected()，用于检测网络状态（是否连接的是wifi），获取连接的wifi信息，查看wifi名称是否以“Skycam”和“TANK”开头，如果是，则返回true，表示手机所连接的是摄像头的wifi热点。
4. setNetwork()，当网络未连接，点击START时，调用此方法，该方法是打开Android系统的wifi设置界面。
5. pathIsExist()，主要用于检查截图或录像文件的保存路径是否存在，若不存在则创建路径。
6. FileActivity，主要是针对查看截图和录像文件，主要用到的方法有：
7. refresh\_listView()，每当有新的截图或录像文件生成时，调用此方法，刷新listView的列表。
8. openFile()，用于打开文件，当点击文件时，通过Android系统自带的图片管理器或视频打开jpg和mp4文件。
9. shareFile()，调用Android系统自带的分享功能，将文件分享。
10. onContextItemSelected(MenuItem item)，在创建的菜单长按触发被选中时，触发此方法。

item表示被选中的项，本APP中共包含打开、分享、重命名和删除四个项。通过item.getItemId()指向其对应的选项。

1. PlayActivity
2. onTouch(MotionEvent event)，本方法主要作用为触屏事件发生时，摇杆显示并可操作，触屏事件结束后，摇杆消失；同时固定触屏区域，防止触控事件同菜单栏发生触屏冲突。利用此方法使APP界面更加简洁，也有利于回传视频的展示，防止摇杆遮挡视频。

event参数表示触屏事件，共有ACTION\_DOWN、ACTION\_UP、ACTION\_MOVE等事件，分别表示按下、抬起和滑动。根据event的值不同表示不同的触屏状态。

1. changeRecordLocation()，改变存储位置。APP中有两种存储方式，手机存储和SD卡存储，调用此方法可自由选择存储方式。
2. changeTextLanguage(int mLanguage)，语言切换，APP的语言切换是随Android系统语言改变而改变，更改系统语言，打开APP，会自动调用此方法检测系统语言，同时APP语言也随之做出相应改变。目前，语言切换仅支持简体中文、繁体中文和英文三种。

mLanguage为整型变量，当检测到手机系统语言时，mLanguage会发生改变，对应的引用AppUtil中的负责相应语言的String[][]中的下标也会更改，进而改变显示的语言。如当检测为简体中文时，mLanguage=0；检测繁体中文时，mLanguage=1；检测英文时，mLanguage=2等。

1. MyTrimChangeListener，PlayActivity的内部类，用于实现TrimView类中的TrimChangeListener接口。主要用于设置微调按钮的作用和调整的幅度。
2. handler()，主要接收子线程发送的数据，并用此数据配合主线程更新UI。这里的handler主要用于处理录像信息和清除发送的控制命令标志位、处理PlayActivity开启的HttpThread中多线程反馈的信息，如获取权限返回的消息、释放控制权限返回的消息、发送控制数据返回的消息、启动录像返回的消息等。
3. setRecordTime(MenuButton btn, long startTime)，设置录像时间的方法，当录像功能开启，亦调用此方法，来动态显示录像时间的变化。

btn表示按钮对象，当录像开始后，调用对象的setText方法，可将动态变化的时间显示在按钮上。

startTime，为录像开始的时间，用当前时间减去录像开始时间就可获得录像时长。

1. AppUtil，APP工具类
2. getCurrentTime()，调用系统当前时间，并用年月日时分秒的形式表达。
3. getFilePath()、getImagePath()、getVideoPath()，获取文件路径、获取截图和录像保存的路径。
4. writeSetupParameterToFile()，本方法用于保存对APP的部分参数设置后的数据，以便于下一次进入APP时，不需要重新设置。主要保存如速度选择、无头模式、左右手控置、存储位置、影像翻转等参数设置。
5. setSetupParameter(int item)，用于读取writeSetupParameterToFile()保存的数据，每次打开APP时会自动调用此方法。

item为设置保存和读取的选项，可以是一个item也可以是多个item。

1. FTP，主要用于OTA升级时，向双摄像头控件发送升级文件的类。
2. uploadingSingle(InputStream inputStream,String fileName, UploadProgress-

Listener listener)，用带进度的方式上传单个文件的方法。用于将升级文件上传给控件。

inputStream，表示上传给控件的输入流。

fileName，表示上传的文件名称。

listener，监听器，上传进度监听对象，本身为预留的一个接口。

1. uploadBeforeOperate(String remotePath，UploadProgressListener listener)，初始化相关参数并设置上传模式、创建文件夹等，用于上传文件之前。

remotePath，表示FTP目录。

listener，监听器，上传进度监听对象，本身为预留的一个接口。

1. openConnect()，打开FTP服务，通过ip、端口号等将APP同FTP服务器建立连接登录服务器并设置相关信息，如中文转码、被动模式、二进制文件支持等。
2. closeConnect()，关闭FTP服务。
3. HttpThread，处理HTTP通信的线程类，线程中的run方法均写于此处。只需要调用该类对象的start方法就可开启线程。
4. do\_getPrivilege()，获取控制权限，通过http发送请求至控件，并获取返回值sAuthcode。一般进入PlayActivity后在其他线程前需开启此线程。

sAuthcode，获得双摄像头的控制权限后的返回值。720p、OTA、do\_bridge、do\_flipOrMiror等发送的url都需要和sAuthcode拼接。

1. do\_720p\_1()、do\_720p\_2()，获取720p影像的方法，置于线程中并开启后，可向控件发送720p请求，若返回值为1，则表示请求成功，可以拿到720p影像。
2. ota\_upgrade()，用于OTA升级。当升级文件通过FTP传至控件后，开启包含此方法的线程，则向控件发送http升级请求，控件接收该请求后则开始升级。
3. do\_releasePrivilege()，释放控制权限，一般在退出PlayActivity或模式切换时执行。
4. do\_bridge()，发送控制命令，用tcp方法循环向控件发送控制命令，并接收控件端返回的数据。
5. get\_version()，获取控件版本信息。发送http请求至控件，控件的返回值为版本信息。
6. do\_flipOrMirror(int value)，影像翻转，在设置栏中的影像翻转选项，目前仅支持小镜头的翻转，发送的url中包含一个value值，value为0表示正常画面，1表示180°翻转，2表示镜面翻转，3表示180°同时镜面翻转。目前设置中的影像翻转仅写入了180°翻转。
7. RockerView，自定义摇杆类，继承自View。
8. RockerView()，构造方法，用于加载自定义摇杆的图片，设置宽高，中心位置等。
9. onDraw()，根据设置，开始绘制摇杆。
10. setUseSensor()，调用系统的重力传感器，并将摇杆的中心点随着重力感应的进行而改变相应位置。
11. setLocked\_y()，锁定Y轴，根据position的位置，来确定中心点的Y轴位置。position为1时，中心点位于底部；position为2时，中心点位于中间；position为3时，中心点位于顶部。
12. onSingleTapUp()，此方法的作用为当释放按键时，摇杆中心点位置恢复为初始位置。
13. RtspVideoView，自定义控件，用于在APP中展示视频回传的画面。初始画面默认为640x368。
14. RtspVideoView，构造方法，设置startBitmap引用的图片，在刚进入PlayActivity中，回传画面未显示时调用startBitmap。回传画面显示后，startBitmap消失。
15. DecoderThread，解码线程，调用FFmpeg库中StartRtsp，开启解码流程，并获取视频流的宽和高。
16. DrawThread，绘制线程，其作用是调用FFmpeg库中的GetYUV,将FFmpeg中解码的视频流绘制展现出来。
17. setVideo(String url, Handler handler)，调用RtspVideoView对象的setVideo方法，将摄像头的url值传入DecoderThread中（调用FFmpeg库中StartRtsp需要此url）。

url，为摄像头控件的路径，用于连接手机APP和摄像头。

1. videoIsRecording()，通过该方法的返回值IsRecording，调用FFmpeg库中的IsRecording(int index)，开始录像。

index，为摄像头开启时显示的下标，双摄像头的下标分别为1和2。

1. getBitmap(int w, int h)，根据视频流获取的byte[] yuv\_pixel在DecorderThread线程中创建一个带有特定宽度、高度和颜色格式的位图。

w和h表示画面的宽和高，其值分别等于GetWidth(index)和GetHeight(index)。

1. 带native的方法，native关键字的一种，用native修饰的方法为java调用非java（本APP中指C语言）代码的接口，这些方法由非java的语言实现，主要是调用的FFmpeg库中相应的方法。
2. SetupListViewAdapter，自定义adapter类，继承自BaseAdapter。主要用于自定义的Dialog，这些Dialog用于APP中设置功能。
3. SetupListViewAdapter()，构造函数，含有两个该构造函数，通过参数不同，加以区分，分别用于HomeActivity中的设置和PlayActivity中的设置。通过ArrayList<SetupItem>将设置中的各项加入Dialog中，通过内部类SetupItem来固定每项设置的name、setup0、setup1，name表示该设置选项的名称，setup0和setup1表示该选项的选择项，如开和关等。
4. SetupItem()，SetupListViewAdapter的内部类，表示每个设置项。
5. HelpActivity，APP中帮助的Activity，利用ViewPager加载帮助图片。
6. judgeLanguage()，判断当前Android系统的语言环境，并根据系统语言选择加载相应语言的帮助图片，目前帮助图片仅有简体中文和英文。
7. initDots()，初始化并设置小圆点。在HelpActivity中布局中根据帮助图片的数量设置一排与帮助图片数量相同的小圆点，该小圆点与帮助图片一一对应，当滑动到哪一位置的图片时，该位置上的小圆点由白色变成灰色，便于用户根据小圆点查看当前滑动帮助图片的位置。
8. setCurDot(int position)，设置底部小圆点的选中状态，颜色变为灰色。

position，表示新的图片被选中时传递过来的位值，相应位置的小圆点就会发生改变。

1. TrimView，自定义微调控件。
2. initView()，加载微调布局并初始化微调相关控件。微调按钮共两种排列方式，水平和竖直，通过isHorz判断水平或竖直情况加载相应微调布局。
3. onClick()，点击事件，通过点击事件来控制微调按钮每次点击移动的幅度。
4. TrimBar，自定义微调控件。
5. TrimBar()，构造方法，绘制微调按钮，根据微调排列的不同绘制水平和竖直方向的mBmpTrimBar\_bg和mBmpTrim\_point。并获取微调控件实际的宽和高。
6. onDraw(Canvas canvas)，利用自绘控件的方式绘制微调。通过Rect确定微调按钮的上下左右位置。

canvas，Canvas类的对象，表示画布对象。可在上面进行属性设置、绘制等操作。

1. MyApplication，全局变量设计的基本类，继承Application，主要通过该类声明全局变量Context。本项目的自定义控件较多，而控件中需用到Context对象，利用Application生命周期贯穿APP的开始和停止的优点，创建统一的Context对象，供其他类调用。

# 6．实施计划

## 6.1 限制

说明在APP开发时的局限性，包括：（1）APP实际开发过程中和理想设计的差距；（2）外部环境因素对开发的限制。

开发DoubleCamera中遇到的一些问题在此说明：

1. 关于影像翻转问题。软件需求说明上给了影像翻转的url，也就是需要影像翻转时，开启子线程，利用http发送翻转请求，控件收到请求后回传的影像为翻转后的影像。但实际开发过程中发现，仅仅是小摄像头的影像发生了翻转，大摄像头无法翻转，后经反馈，控件大镜头的翻转程序未做，故大镜头无法翻转。虽然现在利用rotate()方法使大镜头发生物理性翻转，但测试发现，虽然回传影像可翻转，但截图、录像等获取的文件仍为倒置状态。
2. 关于720p影像问题。大小两个镜头默认的为640\*368影像，软件需求说明上给了720p的url，因需要得到720p影像，故通过HomeActivity的START按钮进入PlayActivity和镜头模式切换时，均会开启子线程，通过http发送720p请求。当该接收该请求的返回值为1时，主线程的RtspVideoView对象通过setVideo(url)使APP和摄像头联系起来就可得到视频影像。但实际测试中发现从向控件发送720p请求，到获取该请求的返回值需要大概3.5s左右的时间。若在3.5s之前开始setVideo(url)方法，因720p线程未完成，得到的影像为640\*368；而在3.5s后开始setVideo(url)，则可以正常得到720p影像。故在开启720p请求的子线程后，将主线程休眠了3500ms。但是，安卓开发的思想是应该尽量去避免这种主线程的耗时操作的，当主线程耗时超过5s，就会产生ANR。
3. 因条件限制等原因，没有将双镜头控件装入无人机中去空地测试。关于视频图传方面，第七章所得的测试数据均为测试者通过手机APP观察影像，一人手持控件以行走的方式远离和靠近测试者来获得的距离和图像效果，且手持控件的高度几乎与人正常情况下手持手机的高度一致，并不能模拟出真实无人机遥控环境下影像回传的情景。因此第七章的测试数据仅有参考价值，并不是真实情况的数据。关于控制命令方面测试，也只是通过Log日志打印和利用串口收发数据的形式测试正常，无法模拟无人机真实环境下对操控命令的反馈情况。

## 6.2实施内容和进度安排

DoubleCamera是在其他单摄像头的基础上开发，其中许多代码和功能借鉴于单摄像头的APP，在次基础上大致做了如下修改：

1. 增加模式选择。本APP的控件拥有两个摄像头，广角和非广角，不同于单摄的单一镜头，需增加两个镜头的切换。现做了四个模式的切换：广角镜头；非广角镜头；广角镜头为主屏幕，非广角为副屏幕；非广角为主屏幕，广角为副屏幕。
2. 720P影像显示。双摄控件默认影像为640\*368，需要得到720p影像就要在APP和摄像头建立连接前给双摄发送720p影像的http请求，否则得到影像仍为368.
3. OTA升级。通过APP将升级文件上传至双摄，完成控件升级。
4. 发送控制命令。在单摄像头APP中，利用http发送控制命令，而在本APP中，根据软件需求，改为TCP方式发送控制命令，并能接受控件的返回值，并在界面显示。
5. 功能的修改和优化。拍照和录像的画面根据模式选择的不同而不同；为提高人机交互，提升视觉效果，将摇杆默认隐藏，在触屏情况下显示，手指离开屏幕则隐藏；根据软件需求，对其他方面进行修改。
6. 视频图传和控制命令的测试。
7. 修改测试出现的BUG和对代码的维护优化。

## 6.3实施条件和措施

APP开发过程中需要的技术支持和设备支持，APP运行所需的条件，避免外部条件对APP的运行起冲突。

# 7．测试计划

## 7.1 测试方案

该APP主要是用手机实现对无人机的操控和实时获取WIFI摄像头回传的影像，需确保手机能够操控飞机的正常飞行，手机能够显示正常的影像信息。所以需要测试的项目有：

1. 手机控制无人机的最远飞行距离；
2. 手机画面的延时大小；
3. 手机近距离显示画面是否出现的马赛克、破图、卡顿等现象；
4. 手机远距离显示画面出现的马赛克、破图、卡顿及画面回传的最远距离；
5. 拍照、录像功能是否正常使用；
6. APP中文字是否随Android系统语言切换而切换。

针对以上需求，制定测试方案：

1. 手机安装APP后，更改手机的系统语言，进入APP，查看APP语言是否切换；
2. 将摄像头对准秒表，手机图像显示出秒表的时间，用另外的手机进行拍照，将秒表和显示秒表的画面都拍摄到，对应拍摄照片上的时间，用手机图像上秒表的时间减去实际秒表上的时间，即可得到画面的延迟，反复的多次测试，求的平均值，即为视频图传的延时时间；
3. 手机连上WIFI摄像头，在近距离情况下测试回传画面是否出现马赛克、破图和卡顿等现象，同时测试拍照、录像功能是否正常；
4. 分别在0m、5m、10m、15m ……位置分别进行测试，在不同距离下观察手机显示的画面是否出现破图、马赛克、卡顿和延时情况；同时测试在相应距离下手机是否能够正常控制无人机的起降、方向、油门等功能。

## 7.2 测试结果

对应需要测试的内容，对APP做好相应的测试，并纪录好测试的结果，方便日后对比。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DoubleCamera 安卓版本测试（室外） | | | | | | | | | |
|
| 测试日期 | 测试 环境 | 测试 距离 | 测试 项目 |  | 手机型号1 | 手机型号2 | 测试人员 | 记录人员 | 备注 |
|
|  |  |  | 720P影像 |  | LG G4 Android 5.1 | 华为 P9 Android 6.0 | 王磊、邹宇 |  |  |
| 2016.12.22 | 羽毛球场 | 5m | 图像效果 | 流畅性 | ○ | ○ |  |  |  |
| 马赛克 | ○ | ○ |  |  |  |
| 破图 | ○ | ○ |  |  |  |
| 卡顿 | ○ | ○ |  |  |  |
|  |  | 10m | 图像效果 | 流畅性 | ○ | ○ |  |  |  |
| 马赛克 | ○ | ○ |  |  |  |
| 破图 | ○ | ○ |  |  |  |
| 卡顿 | ○ | ○ |  |  |  |
|  |  | 15m | 图像效果 | 流畅性 | ○ | ○ |  |  |  |
| 马赛克 | ○ | ○ |  |  |  |
| 破图 | ○ | ○ |  |  |  |
| 卡顿 | ○ | ○ |  |  |  |
|  |  | 20m | 图像效果 | 流畅性 | ○ | ○ |  |  |  |
| 马赛克 | × | ○ |  |  |  |
| 破图 | ○ | × |  |  |  |
| 卡顿 | ○ | × |  |  |  |
|  |  | 25m | 图像效果 | 流畅性 | ○ | × |  |  |  |
| 马赛克 | ○ | × |  |  |  |
| 破图 | × |  |  |  |  |
| 卡顿 | × | × |  |  |  |
|  |  | 30m | 图像效果 | 流畅性 | × |  |  |  |  |
| 马赛克 | ○ |  |  |  |  |
| 破图 | × |  |  |  |  |
| 卡顿 | × |  |  |  |  |
|  | 最大距离 | |  | | 最大距离30m左右，到25米处开始出现卡顿，随后恢复，随着距离增大，开始出现轻微马赛克和明显卡顿破图现象，至30m完全卡死。 | 最大距离25m左右，在20m处出现一次卡顿，20-25m破图卡顿渐渐严重，并无法恢复正常影像，至25m完全卡死 |  |  | 室内近距离测试华为荣耀4X（Android 4.4.4）时，双摄轻微移动就导致大面积破图和长时间卡顿，几乎无法正常工作，延时1200ms，可能720p影像对手机性能要求较高，故4X未参与室外测试 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DoubleCamera室内近距离延时测试(毫秒) | | | | |
|
| 次数 | LG G4 | | 华为 P9 | |
|  | 大镜头 | 小镜头 | 大镜头 | 小镜头 |
| 1 | 301 | 315 | 258 | 215 |
| 2 | 258 | 182 | 301 | 215 |
| 3 | 345 | 258 | 265 | 199 |
| 4 | 258 | 215 | 258 | 306 |
| 5 | 197 | 111 | 215 | 172 |
| 6 | 266 | 203 | 258 | 219 |
| 7 | 258 | 215 | 344 | 215 |
| 8 | 301 | 258 | 258 | 172 |
| 9 | 215 | 172 | 301 | 218 |
| 平均 | 267 | 215 | 274 | 215 |

## 7.3 修改内容

根据测试结果和预期达到的效果对比，修改不足的地方，并做好相应的纪录。

# 8.验收标准

明确APP需要达到的要求，对应APP的测试结果，看是否达到客户要求。