Introduction for Transparent Data Transmission between Mobile and Onboard device

V1.0.0

2015.06

目录

Introduction

In this chapter, we will give an overall introduction about the Transparent Data Transmission function between onboard device and mobile device. The main contents include the development purpose, method to use this function, and some sample codes.

Please pay attention that the upstream bandwidth from mobile to onboard device is around 1KB/s, while the downstream bandwidth from onboard to mobile is around 8KB/s

1 Development Purpose

DJI provides two kinds of API including Mobile API and Onboard API to developers, and to help them developing their own applications. Mobile API allows developers monitor and control the UAV from a mobile device running iOS or Android that is connected to the remote controller, while Onboard API allows developers to monitor and control the UAV from any system wired directly to the UAV's autopilot through the available serial (UART) interface.

Comparing with each other, developers do need to buy any other device when using Mobile API, meanwhile, it’s also easy for developers to monitor the flight status when running some applications. However, considering the fact that, on one hand the mobile device has a relatively low computing ability, and on the other hand the Mobile device communicates with UAV through wireless link, hence developers may not be able to control the UAV very real-timely or complicatedly. By contrast, when using the Onboard API, the computing device should be put on the UAV and communicates with it directly through serial port. This structure benefits developers to develop some complicate and robust applications. But due to the computing device has been put on the UAV, developers may not be able to monitor the status directly. When the program crashes, developers have to control the UAV manually.

Given the features of Onboard API and Mobile API, we developed the function of Transparent Data Transmission between mobile device and onboard device. In this way, developers can send data from mobile device to onboard device, and to control the program running on the onboard device. Meanwhile, onboard device can also send data to mobile device for the convenience for monitoring and so on.

To sum up, the Transparent Data Transmission function can serve as the bridge between Mobile API and Onboard API, which benefits developers to realize their own desire to the most.

2 Transparent Data Transmission (From Onboard Device to Mobile Device)

2.1 Onboard **Device** to Flight Control

The communication protocol between onboard device and flight control has been introduced on *onboard_SDK_introduction,* and this document will illustrate it in more detailed.

The specific communication protocol is introduced as below:

Command set: 0X00

Command ID: 0XFE

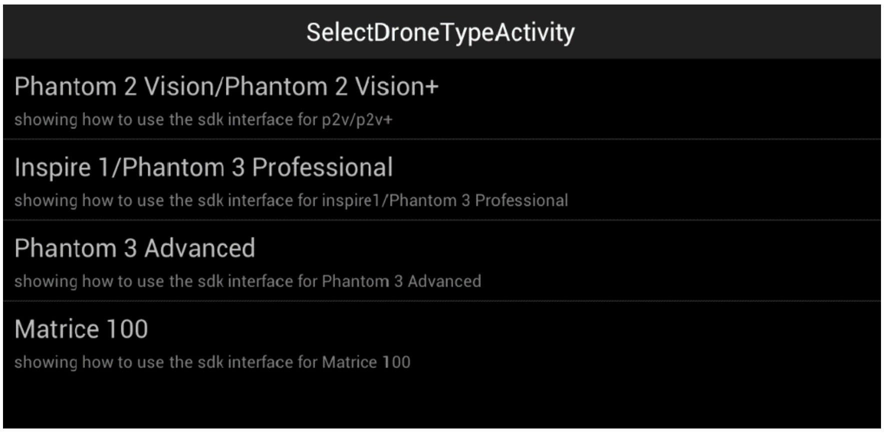
Sample code:

	Offset	*Size*	*Description*
Request data	**0**	1~100	Data needs to be sent to Mobile device
Return data	**0**	2	Return code 0 Success

```
char cmd_buf[10];
cmd_buf[0] = 0x00;
cmd_buf[1] = 0xFE;
memcpy(&cmd_buf[2], “Hello!”, 7);
Linklayer_Send(
                                 SESSION_MODE3,
                                 cmd_buf,
                                 9,
                                 0，
                                 200，
                                 3，
                                 0
);
```

2.2 Flight Control to Mobile Device

Developers can download corresponding applications on DJI website, or develop relative applications by themselves. In the document, we take the android system as an example. Users can enter the DJI-SDK-DEMO application (Fig .1), and monitor the UAV status through relative functions (Fig.2), then use the Transparent Data Transmission function on *Controller State* page (Fig.3). Please note that only Matrice 100 supports this function so far.



SelectDroneTypeActivity

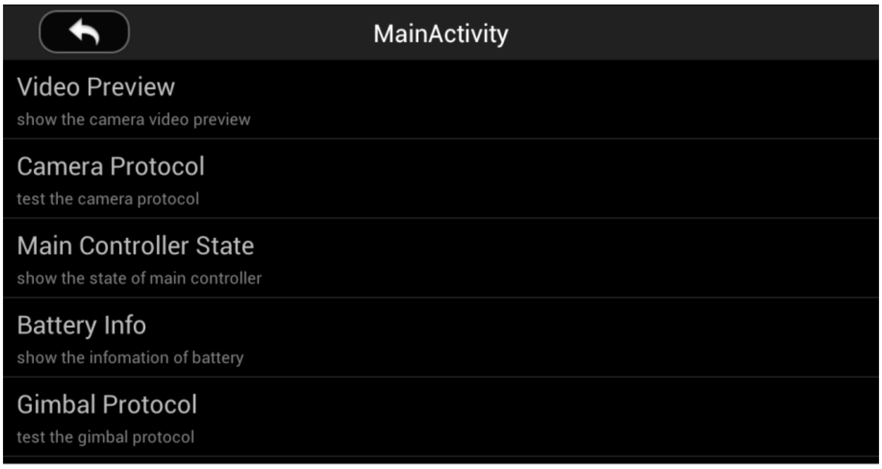
Phantom 2 Vision/Phantom 2 Vision+
showing how to use the sdk interface for p2v/p2v+

Inspire 1/Phantom 3 Professional
showing how to use the sdk interface for inspire1/Phantom 3 Professional

Phantom 3 Advanced
showing how to use the sdk interface for Phantom 3 Advanced

Matrice 100
showing how to use the sdk interface for Matrice 100

Fig.1 DJI-SDK-DEMO main interface



MainActivity

Video Preview
show the camera video preview

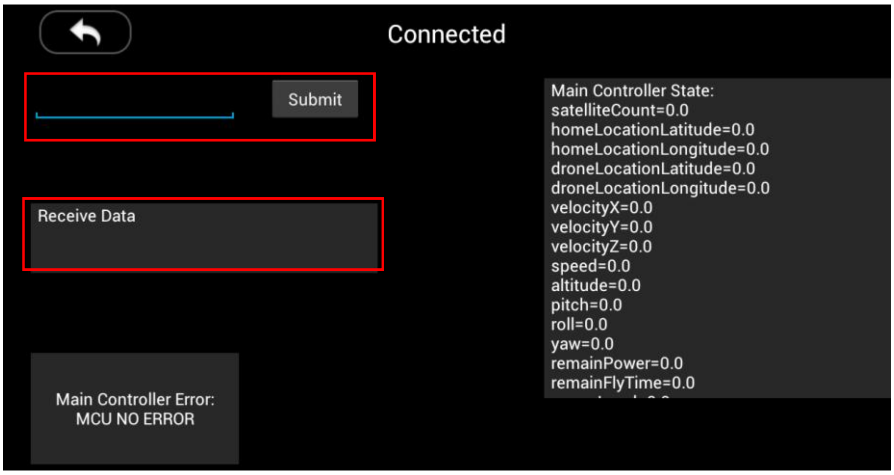
Camera Protocol
test the camera protocol

Main Controller State
show the state of main controller

Battery Info
show the infomation of battery

Gimbal Protocol
test the gimbal protocol

Fig.2 DJI-SDK-DEMO relative functions



Connected

Main Controller Error:
MCU NO ERROR

```
Main Controller State:
satelliteCount=0.0
homeLocationLatitude=0.0
homeLocationLongitude=0.0
droneLocationLatitude=0.0
droneLocationLongitude=0.0
velocityX=0.0
velocityY=0.0
velocityZ=0.0
speed=0.0
altitude=0.0
pitch=0.0
roll=0.0
yaw=0.0
remainPower=0.0
remainFlyTime=0.0
```

Fig.3 DJI-SDK-DEMO Transparent Data Transmission demo

The relative sample codes are shown as below:

1) iOS:

```
// Setting Delegation
inspireMC.mcDelegate = self;

//The legation function is called when receiving data
```

```
(void)mainController:(DJIMainController*)mc
didReceivedDataFromExternalDevice:(NSData*)data{
        //Here is the receiving data
        NSLog(@"%@",data);
}
```

2) Android:

```
//Receiving the data callback interface sent from flight controller
DJIMainControllerExternalDeviceRecvDataCallBack mExtDevReceiveDataCallBack = null;

//Instantiate callback interface
mExtDevReceiveDataCallBack = new DJIMainControllerExternalDeviceRecvDataCallBack() {
        @Override
        public void onResult(byte[] data)
        {
                //Here is the receiving data
        }
};
//Setting callback interface
DJIDrone.getDjiMC().setExternalDeviceRecvDataCallBack(mExtDevReceiveDataCallBack);
```

3 Transparent Data Transmission (From Mobile Device to Onboard Device)

3.1 Mobile Device to Flight Control

The development of Data Transparent Transmission function is according to corresponding API in DJI Mobile SDK.

The relative sample codes are shown as below:

1) iOS：

```
//Initialization. Create DJI Drone object according to relative UAV type.
DJIDrone* drone = [DJIDrone droneWithType:DJIDrone_Inspire];
//Obtain Main controller object from DJI Drone object.
DJIInspireMainController* inspireMC = (DJIInspireMainController*)drone.mainController;
//Start data connection.
[drone connectToDrone];

//Sending data. Please note that data size should be no larger than 100 bytes.
  NSData* data = [NSData dataWithByte:"…"];
// Sending data to peripheral and check the sending status through callback function.
```

```
[inspireMC sendDataToExternalDevice:data withResult:(^(DJIError* error)){
    if (error.errorCode == ERR_Successed)
    {
        //Data sent successfully.
    }
    Else if(error.errorCode == ERR_InvalidParam)
    {
        //Data size is null or larger than 100 bytes.
    }
    else
    {
        //Data sent failed.
    }
}];
```

2) Android:

```
//Data needs to be sent, please note the data size should be no larger than 100 bytes.
byte[] data = {0};
//Sending data to flight controller
DJIDrone.getDjiMC().sendDataToExternalDevice(data,new DJIExecuteResultCallback(){
    @Override
    public void onResult(DJIError result)
    {
        //result is the callback status after sending data:
        // 1 . result == DJIError.ERR_PARAM_IILEGAL ,   Data size is null or larger than 100 bytes.
        // 2 . result == DJIError.ERR_TIMEOUT ,         Data sent failed.
        // 3 . result == DJIError.RESULT_OK,            Data sent successfully.

    }
});
```

3.2 Flight Control to Onboard Device

The method to use this function is the same as previous, and the specific communication protocol is introduced as below:

Command set: 0X02

Command ID: 0X02

Sample code：

	Offset	*Size*	*Description*
Request data	**0**	1~100	User defined data
Return data	**0**	0	No return data