

蝴蝶飞舞播放操作步骤

Step1:

进入目录

```
root@cymx-hosts:/mnt/hgfs/workspace/P2P/Release_PPCCS_1.5.5/Sample/Linux/PPCCS# ls
Makefile  PPCCS_Client  PPCCS_Client.cpp  PPCCS_Device  PPCCS_Device.cpp
```

使用 vim 或者使用 NotePad++ 打开 PPCCS_Device.cpp 在

PPCCS_Device.cpp 宏定义处打开 #define AVSTREAM_TEST

(1)、使用 vim 打开 PPCCS_Device.cpp

```
20 #include <windows.h>
21 #include <stdio.h>
22 #include <direct.h>
23 #endif
24
25
26 #include "../..//Include/PPCCS/PPCCS_API.h"
27 #include "../..//Include/AVSTREAM_IO_Proto.h"
28
29
30 //#define CONNECT_TEST
31 //#define RW_TEST
32 //#define FT_TEST
33 //#define RW_TEST1
34 #define AVSTREAM_TEST
35 //#define MULTI_CHANNEL_TEST
36 //#define RW_TEST2
37 //#define RW_TEST3
38
"PPCCS_Device.cpp" 1179L, 32236C
```

(2)、使用 NotePad++ 打开 PPCCS_Device.cpp

```
// #define CONNECT_TEST
// #define RW_TEST
// #define FT_TEST
// #define RW_TEST1
#define AVSTREAM_TEST
// #define MULTI_CHANNEL_TEST
// #define RW_TEST2
// #define RW_TEST3
```

Step2: 在 ubuntu 终端下 make 编译 PPCS_Device.cpp

```
root@cymx-hosts:/mnt/hgfs/workspace/P2P/Release_PPCS_1.5.5/Sample/Linux/PPCS# make
g++ -O2 -Wall -DLINUX ./PPCS_Client.cpp ../../Lib/Linux/x86/libPPCS_API.so -o PPCS_Client -s -lpthread
g++ -O2 -Wall -DLINUX ./PPCS_Device.cpp ../../Lib/Linux/x86/libPPCS_API.so -o PPCS_Device -s -lpthread
./PPCS_Device.cpp: In function 'void* myThreadGetDataFromFile(void*)':
```

Step3: 执行 PPCS_Device 程序

```
root@cymx-hosts:/mnt/hgfs/workspace/P2P/Release_PPCS_1.5.5/Sample/Linux/PPCS# ./PPCS_Device
Usage: ./PPCS_Device DID APILicense [InitString]
DID: This is Device DID: ABCD-123456-ABCDEF
APILicense: If you set the CRCKey, This parameter must be input  APILicense:CRCKey
InitString: The InitString, if this parameter is empty, program default InitString will be used.
Example:
./PPCS_Device ABCD-123456-ABCDEF ABCDEF
./PPCS_Device ABCD-123456-ABCDEF ABCDEF EBGAIEBIKHJJGFJKEOGCFAEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPB
GOLKIKLKAJMKFD00FMOBECEJIMM
./PPCS_Device ABCD-123456-ABCDEF ABCDEF:ABC123
./PPCS_Device ABCD-123456-ABCDEF ABCDEF:ABC123 EBGAIEBIKHJJGFJKEOGCFAEPHPMAHONDGJFPBKCPAJJMLFKBD
BAGCJPBGOLKIKLKAJMKFD00FMOBECEJIMM
root@cymx-hosts:/mnt/hgfs/workspace/P2P/Release_PPCS_1.5.5/Sample/Linux/PPCS# |
```

会打印一堆提示说明，这些参数可以在技术白皮书《PPC Technical white paper-中文.pdf》中获得更详细的说明，参数格式参考上图中的 Example 其中：

DID：为设备的 DID 号，由命令行传入。

APILicense：此参数要看客户在部署 P2P 服务器时是否设置了 CRCKey，若客户部署 P2P 服务器设置了 CRCKey 则，需要在 APILicense 后面添加 CRCKey，此参数格式： APILicense:CRCKey

InitString：这是 P2P 服务器 IP 地址的加密字符串，每位客户在部署 P2P 服务器时，都会为其服务器的 IP 加密成一串字符串。

然后输入 DID APILicense InitString 执行，因为尚云的 P2P 服务器设置了 CRCKey，所以在命令行中会看到参数格式为 EMFYCB:EasyView 如下图：

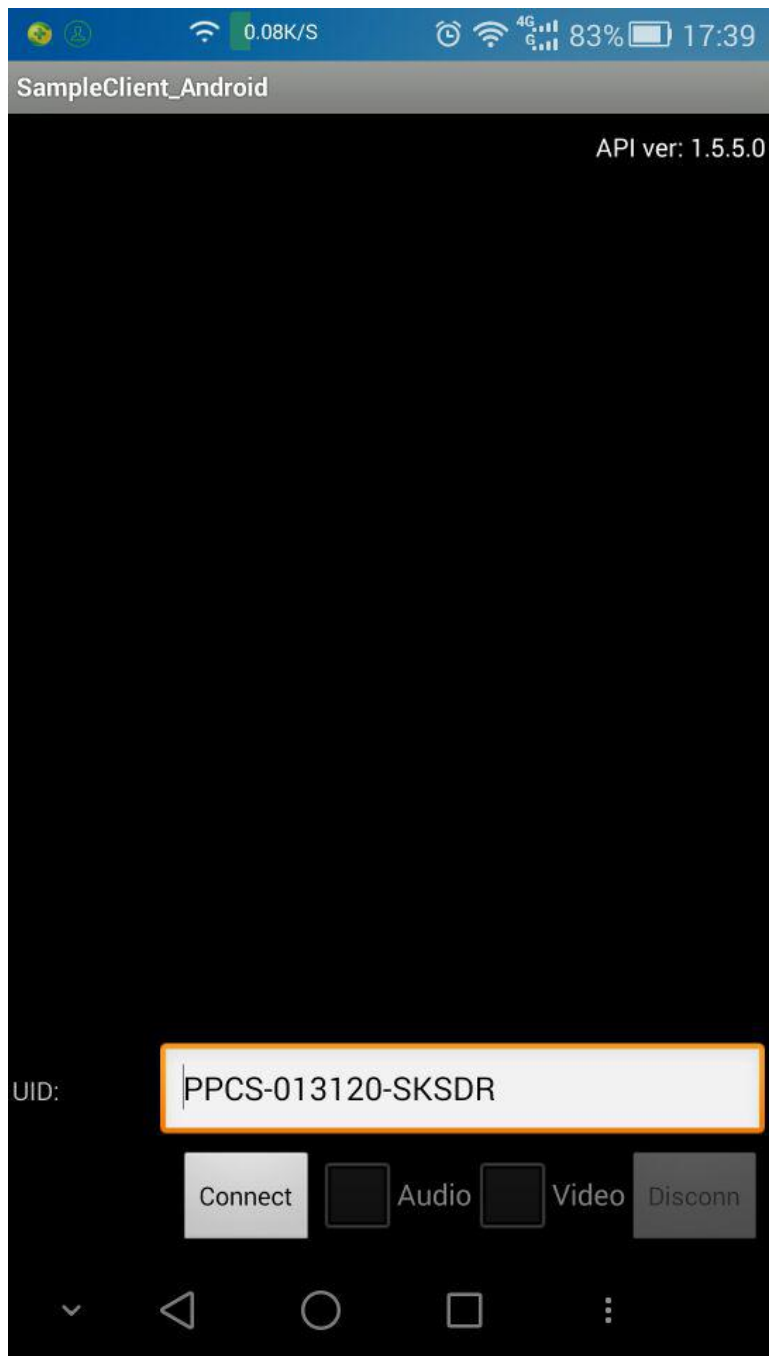
```
root@cymx-hosts:/mnt/hgfs/workspace/P2P/Release_PPCS_1.5.5/Sample/Linux/PPCS# ./PPCS_Device PPCS-013120-S
KSDR EMFYCB:EasyView EBGAEIBIKHJJGFJKEOGCFAEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMKFD00FM0BECEJIMM
PPCS_API Version: 1.5.5.0
PPCS_NetworkDetect() ret = 0
----- NetInfo: -----
Internet Reachable : YES
P2P Server IP resolved : YES
P2P Server Hello Ack : YES
Local NAT Type : Port-Restricted Cone
My Wan IP : 61.141.178.189
My Lan IP : 192.168.56.21
```

Step4: 安装手机客户端 Apk

进入安卓目录，看到 Sample_Client_AndroidActivity.apk，将其安装到手机。

« P2P ▶ Release_PPCS_1.5.5 ▶ Sample ▶ Android ▶ PPCS ▶ bin		
名称		修改日期
classes		2016-3-2 09:45
dexedLibs		2016-3-2 09:45
res		2016-3-2 09:45
AndroidManifest.xml		2016-2-24 17:51
classes.dex		2016-3-2 09:43
jarlist.cache		2016-3-2 09:06
R.txt		2016-3-2 09:06
resources.ap_		2016-3-2 09:19
Sample_Client_AndroidActivity.apk		2016-3-2 09:43

Step5: 点击打开 app，会看到黑乎乎一片，确保设备端已经运行起来，然后在框框内输入设备的 DID 号，点击 Connect.等待一小会。



当连接成功之后，勾选 Audio 和 Video,就可以看到漂亮的蝴蝶飞舞的视频画面和动听的音乐啦。。。见下图

SampleClient_Android

API ver: 1.5.5.0



美柯视频-tinck

P2P, Num:1

UID:

PPCS-013120-SKSDR

Connect



Audio



Video

Disconn



如果连接不上，看看返回错误值是多少，错误值可在头文件中查找相应说明。

↑

« Data (D:) » workspace » P2P » Release_PPCS_1.5.5 » Include » PPCS

名称	修改日期	类型
PPCS_API.h	2014-2-18 23:53	H 文
PPCS_Error.h	2014-3-28 09:54	H 文
PPCS_Type.h	2013-7-15 11:52	H 文

司的位置

```
1 #ifndef _PPCS_ERROR__INC_H_
2 #define _PPCS_ERROR__INC_H_
3
4 #define ERROR_PPCS_SUCCESSFUL 0
5 #define ERROR_PPCS_NOT_INITIALIZED -1
6 #define ERROR_PPCS_ALREADY_INITIALIZED -2
7 #define ERROR_PPCS_TIME_OUT -3
8 #define ERROR_PPCS_INVALID_ID -4
9 #define ERROR_PPCS_INVALID_PARAMETER -5
10 #define ERROR_PPCS_DEVICE_NOT_ONLINE -6
11 #define ERROR_PPCS_FAIL_TO_RESOLVE_NAME -7
12 #define ERROR_PPCS_INVALID_PREFIX -8
13 #define ERROR_PPCS_ID_OUT_OF_DATE -9
14 #define ERROR_PPCS_NO_RELAY_SERVER_AVAILABLE -10
15 #define ERROR_PPCS_INVALID_SESSION_HANDLE -11
16 #define ERROR_PPCS_SESSION_CLOSED_REMOTE -12
17 #define ERROR_PPCS_SESSION_CLOSED_TIMEOUT -13
18 #define ERROR_PPCS_SESSION_CLOSED_CALLED -14
19 #define ERROR_PPCS_REMOTE_SITE_BUFFER_FULL -15
20 #define ERROR_PPCS_USER_LISTEN_BREAK -16
21 #define ERROR_PPCS_MAX_SESSION -17
22 #define ERROR_PPCS_UDP_PORT_BIND_FAILED -18
23 #define ERROR_PPCS_USER_CONNECT_BREAK -19
24 #define ERROR_PPCS_SESSION_CLOSED_INSUFFICIENT_MEMORY -20
25 #define ERROR_PPCS_INVALID_API_LICENSE -21
26 #define ERROR_PPCS_FAIL_TO_CREATE_THREAD -22
27
28 #endif // _PPCS_ERROR__INC_H_
```

如有不明之处，请联系尚云互联技术支持，谢谢！