DJI Onboard API Sample for Linux

Version	Date	Remarks
V1.0.0	2015-05	Created

This document introduce the general features of the DJI Onboard API C++ sample based on Linux OS. Compile and run this sample in the Linux console to gain basic control of the aircraft such as take-off, landing and go home, etc.

Development Environment

Operating System：Ubuntu 12.04

Directory Structure

Directory structure for *DJI_Onboard_API_Cmdline_Sample* is listed as below:

Directory	Description
src	Source code files directory
cmake	Temporary and Makefile files directory.
output	Output directory for executable files
doc	Documents

List of key functions

- Serial Port Configuration

```
int Pro_Hw_Setup(const char *device,int baudrate)
```

Usage：Setup and activate serial port in the Linux OS

Parameters： *device* denotes the file name of the serial device . *baudrate* denotes transmission rate of the serial port.

Return Value：0 as Success and -1 as Failed.

- Initialization

```
int DJI_Pro_Test_Setup(void)
```

Function Feature：Initialize various variables including app ID, app key and serial port.
Parameters：Void.
Return Value：0 as Success and -1 as Failed.

- API Activation

```
void DJI_Onboard_API_Activation(void)
```

Function Feature： Activate the DJI Onboard API.

Parameters： Void.

Return Value： Void.

- Gain or Release Control of the Aircraft

```
void DJI_Onboard_API_Control(unsigned char arg)
```

Function Feature： Obtain or release control of the aircraft after DJI Onboard API is activated.

Parameters： 1 to obtain the control of the aircraft, 2 to release the control of the aircraft.

Return Value： Void.

- Take-off Function

```
void DJI_Onboard_API_Takeoff(void)
```

Function Feature： Request for take-off.

Parameters： Void.

Return Value： Void.

- Landing Function

```
void DJI_Onboard_ API_Landing(void)
```

Function Feature： Request for landing.

Parameters： Void.

Return Value： Void.

- Go home Function

```
void DJI_ API_Request_Gohome(void)
```

Function Feature： Initiate go home process. Aircraft will go home and land

Parameters： Void.

Return Value： Void.

Configuration

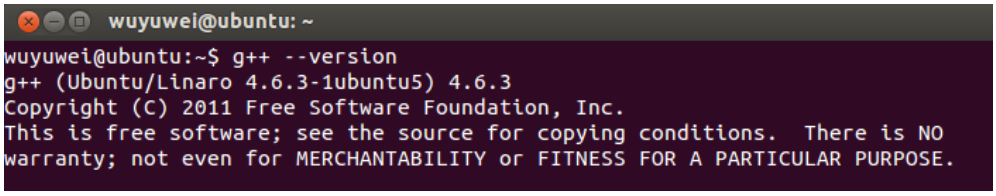
Developers must obtain the app ID, API level and encryption key before compiling the source files. The screenshot listed below shows where these values should be assigned in the *DJI_Pro_Test_Setup* function as well as the baudrate and the serial device name.

```
int DJI_Pro_Test_Setup(void)
{
    int ret;

    activation_msg.app_id = 10086;
    activation_msg.app_api_level = 2;
    activation_msg.app_ver = 1;
    memcpy(activation_msg.app_bundle_id,"12345678901234567890123", 32);
    key = "5837313ef98f1f7f1c50eebb0b06363d523a369289e042c4d00b66d8e49337a7";

    ret = Pro_Hw_Setup("/dev/ttyUSB0",230400);
    if(ret < 0)
        return ret;
    Pro_Link_Setup();
    App_Recv_Set_Hook(App_Recv_Req_Data);
    App_Set_Table(set_handler_tab, cmd_handler_tab);
    CmdStartThread();
    Start_Simple_Task_Thread();

    return 0;
}
```

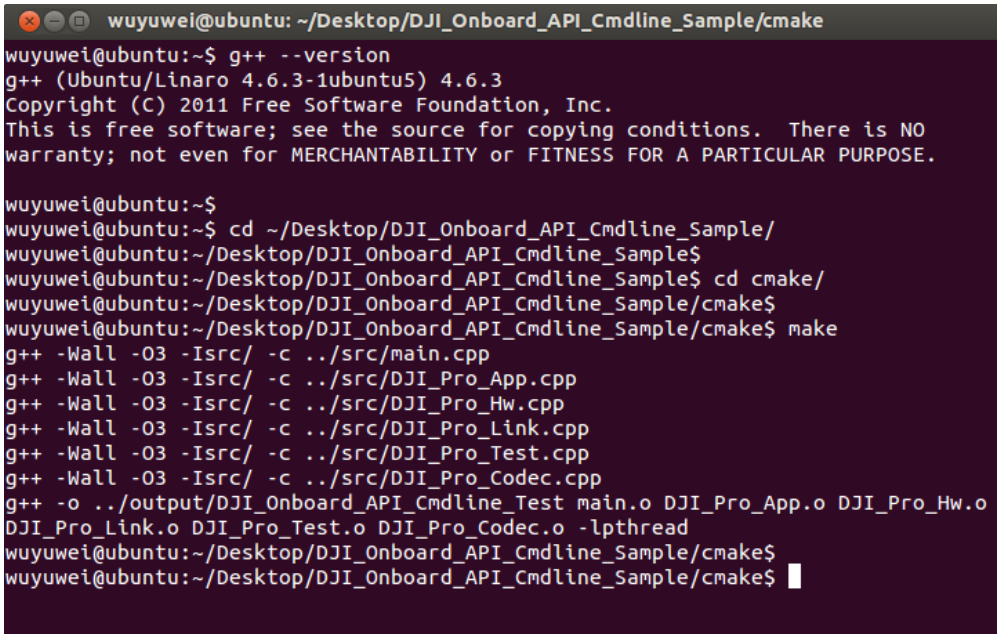

Developers must ensure the baudrate set by the *DJI_Pro_Test_Setup* is consistent with the one of aircraft.

Compile

The following process will guide you to compile the sample code based on the Ubuntu 12.04 distribution. Open the Linux terminal, input `g++ --version` to check whether the g++ compiler is installed or not. If yes, the terminal looks like as screenshot below.



Goto *DJI_Onboard_API_Cmdline_Sample\cmake* directory, input `make` to compile the source code.



You may locate the Linux executable file in the *DJI_Onboard_API_Cmdline_Sample/output* directory.

Run

You may locate the *DJI_Onboard_API_Cmdline_Test* file in the *DJI_Onboard_API_Cmdline_Sample/output* directory after the sample code is compiled with success.

Check the sample version in the Linux terminal using the command below:

```
./DJI_Onboard_API_Cmdline_Test  -v
```

The sample version is displayed as follow:

```
DJI Onboard API Cmdline Test,Ver 1.x.x
```

Ensure that the current account has access privilege to the serial device. Assume that the serial device is named as "/dev/ttyUSB0", use the following command to gain access privilege for the serial device.

```
sudo chmod 777 /dev/ttyUSB0
```

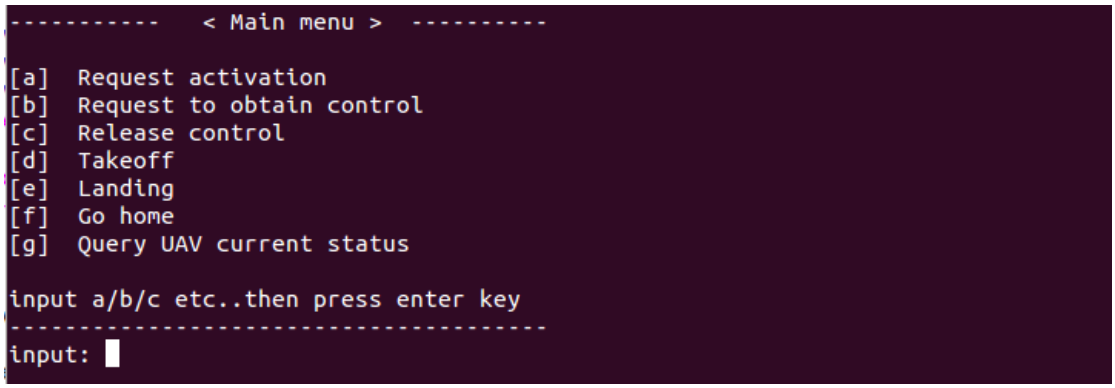
Using the following command to launch the testing program.

```
./DJI_Onboard_API_Cmdline_Test
```

The following menu option will be displayed:

Main Menu Options

The Main menu is displayed as follow, the remaining section will introduce the features of the each options.



Guideline for Controlling

Connect the aircraft to the PC with a serial cable. Set the aircraft in API mode by switching the flight mode using remote controller.

- Query UAV current status

  Input ‘g’ to enquire current status of the aircraft.

```
-----------   < Main menu >  ----------

[a]  Request activation
[b]  Request to obtain control
[c]  Release control
[d]  Takeoff
[e]  Landing
[f]  Go home
[g]  Query UAV current status

input a/b/c etc..then press enter key
---------------------------------------
input: g
--  Current status info: --
Activation status:[unknown]
Battery capacity:[50%]
Control device:[RC]
```

The status information includes API activate status, remaining battery level and controlling device of the aircraft.

- Request Activate

  Input ‘a’ to activate API. The status message of ACTIAVTE_SUCEESS will be displayed if the API is activated with success.

```
-----------   < Main menu >  ----------

[a]  Request activation
[b]  Request to obtain control
[c]  Release control
[d]  Takeoff
[e]  Landing
[f]  Go home
[g]  Query UAV current status

input a/b/c etc..then press enter key
---------------------------------------
input: Pro_Link_Recv_Hook:Recv Session 2 ACK
Sdk_ack_cmd0_callback,sequence_number=0,session_id=2,data_len=2
[ACTIVATION] Activation result: ACTIVATION_SUCCESS
[ACTIVATION] set key DJI-DEMO AES256 KEY-lala-haha-MA
```

- Request for Control

  Input ‘b’ to gain control of the aircraft.

```
-----------   < Main menu >  ----------

[a]  Request activation
[b]  Request to obtain control
[c]  Release control
[d]  Takeoff
[e]  Landing
[f]  Go home
[g]  Query UAV current status

input a/b/c etc..then press enter key
---------------------------------------
input: Pro_Link_Recv_Hook:Recv Session 1 ACK
call sdk_ack_nav_open_close_callback
Recv ACK,sequence_number=13,session_id=1,data_len=2
```

You may query the aircraft status information when you obtained the control of the aircraft.

```
-----------   < Main menu >  ----------

[a]  Request activation
[b]  Request to obtain control
[c]  Release control
[d]  Takeoff
[e]  Landing
[f]  Go home
[g]  Query UAV current status

input a/b/c etc..then press enter key
---------------------------------------
input: g
--  Current status info: --
Activation status:[Activation pass]
Battery capacity:[48%]
Control device:[third party onboard device]
```

The screenshot above shows that the API has been activated and the aircraft is controlled by the third party onboard device.

- Aircraft Control

  Input either‘d’ or ‘e’ or ‘f’ to initiate take-off, landing and go home process respectively

```
-----------   < Main menu >  ----------

[a]  Request activation
[b]  Request to obtain control
[c]  Release control
[d]  Takeoff
[e]  Landing
[f]  Go home
[g]  Query UAV current status

input a/b/c etc..then press enter key
---------------------------------------
input: [DEBUG] in send
[DEBUG] send req cmd ok
Pro_Link_Recv_Hook:Recv Session 2 ACK
Sdk_ack_cmd0_callback,sequence_number=5,session_id=2,data_len=2
[DEBUG] CMD_RECIEVE
[DEBUG] send req status ok
Pro_Link_Recv_Hook:Recv Session 2 ACK
Sdk_ack_cmd0_callback,sequence_number=6,session_id=2,data_len=2
[DEBUG] recv ack1 status ok
[DEBUG] recv_ack 0x5
random test Cmd result: STATUS_CMD_EXE_SUCCESS
```