Wisenet Open Platform v4.02

Troubleshooting

v4.02

2022-12-16

Copyright

Trademark

Restriction

Disclaimer

Hanwha Techwin Co., Ltd. has made every effort to ensure the completeness and accuracy of this document, but makes no guarantees regarding the information contained herein. All responsibility for proper and safe use of the information in this document lies with users. Hanwha Techwin Co., Ltd. may revise or update this document without prior notice.

Contact Information

HANWHA TECHWIN Co., LTD.

Hanwhatechwin R&D Center, 701, Sampyeong-dong,
Bundang-gu, Seongnam-si, Gyeonggi-do, Korea, 463-400

TEL: +82-70-7147-8740~60   FAX: +82-31-8018-3745

http://step.hanwha-security.com

HANWHA TECHWIN AMERICA Inc.

100 Challenger Road Ridgefield Park, New Jersey, 07660
U.S.A.

HANWHA TECHWIN EUROPE LTD.

2nd Floor, No. 5 The Heights, Brooklands, Weybridge,
Surrey, KT13 0NY, U.K.

Hanwha Techwin

Table of Contents

Troubleshooting

1. Storing Files on a Flash Memory

You may read and write files under the following path:

```
/mnt/opensdk/apps/<AppName>
```

If you delete an application, the corresponding information stored in the flash memory will be removed as well. If you reboot the camera only without deleting the application, stored files won’t be removed but will be available. For information to be saved even after the application is deleted, refer to “Nand Flash Space” under Chapter 6 Others in the SDK API document.

2. Storing Files on a SD Card

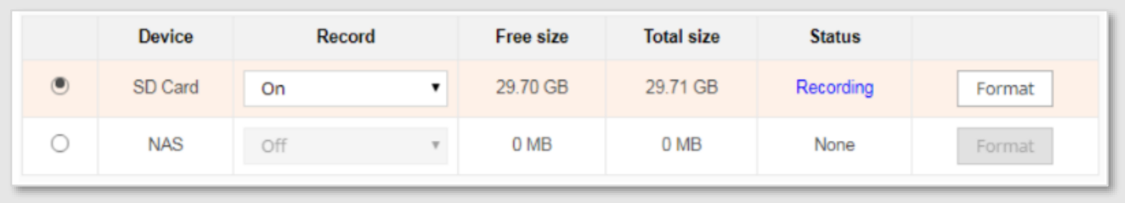
Use the following API to check the path for the mounted SD card.

```
Const INT8_N *opensdk_getSDcardStoragePath();
```

For any of the following cases, NULL will be returned.

- No SD card is mounted
- No access right available to the SD card in “IPCameraManufest.xml”

If NULL is returned despite not applicable to cases above, check if the SD card is activated. To activate the SD card from the camera web viewer, select on the menu Basic > Event > Storage, select [SD Card] from the [Storage action setup] list, and choose [On] under the [Record] column.



	Device	Record	Free size	Total size	Status	
●	SD Card	On ▾	29.70 GB	29.71 GB	Recording	Format
○	NAS	Off ▾	0 MB	0 MB	None	Format

The space is limited to 80% of the size of the SD card. If the SD card recording is enabled, the space will be shared with a network camera main application.

3. Implementing FTP/Email/JPEG Encoding Function

Regarding FTP/Email/JPEG encoding, you should receive the event after calling send_event().

The behavior scenario is as below:

Open Platform Application   Apphelper

send_event() ------------------------------------> processing

recv_event() <------------------------------------ processing result

Event type is OPENSDK_EVENT_STATUS.

TASK_STATUS.TaskName will be OPENSDK_FTP_FILE_UPLOAD for FTP, OPENSDK_JPEG_ENCODE for

JPEG Encoding.

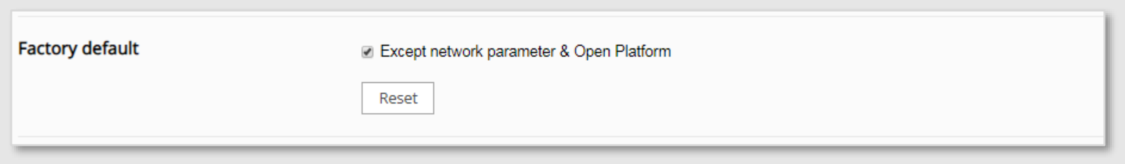
In the case of JPEG encoding, it takes up to 100 ms. If you request it without waiting for the result

notification, it causes memory leaks.

4. Preserving Applications after a Factory Default

During factory default initialization, you may preserve open applications installed and their settings but reset other settings.

If you don’t want to remove open applications during factory default initialization, select on the menu System > Upgrade/Restart from the web viewer, and select [Except network parameter & open platform] under [Factory default].



Factory default

☑ Except network parameter & Open Platform

If you physically press the reset button on the camera, all settings, including network settings and open platform information, will be initialized to the factory default.

5. Connecting Docker and Ubuntu by using a shared folder

If you create an app by loading a Docker image on Wisenet Open Platform (WOP) 4.01 or above and create a *.cap file (WOP app file) or source code for user OS, use the following commands:

Example 1

In the example, the folder that you want to share on a current directory is 'pwd' and the shared folder is 'mnt'.

```
$> docker run --rm -it -v=`pwd`:/mnt wop:4.01
```

```
$> docker run --rm -it --volume="$PWD:/mnt" wop:4.01
```

Example 2

If you want to display debug_message, add the options below.

```
$> docker run -p 8080:8080 --rm -it --volume="$PWD:/mnt" wop:4.01
```

Example 3

In the example, VirtualBoxMountExample is the OpenApp Folder that is currently being created.

```
root@-VirtualBox:/home/VirtualBoxShared/# docker run --rm -it --
volume="$PWD:/mnt" wop:4.01
root@0994f5fa31d7:/# cd /mnt/
root@0994f5fa31d7:/mnt# ls
VirtualBoxMountExample
root@0994f5fa31d7:/mnt# cd VirtualBoxMountExample/
root@0994f5fa31d7:/mnt/VirtualBoxMountExample# ls
IPCameraManifest.xml  Makefile  bin  html  inc  res  src
root@0994f5fa31d7:/mnt/VirtualBoxMountExample# touch test.ttt
root@0994f5fa31d7:/mnt/VirtualBoxMountExample# exit
exit
root@-VirtualBox:/home/VirtualBoxShared# ls
VirtualBoxMountExample
root@-VirtualBox:/home/VirtualBoxShared# cd VirtualBoxMountExample/
root@-VirtualBox:/home/VirtualBoxShared/VirtualBoxMountExample# ls
IPCameraManifest.xml  Makefile  bin  html  inc  res  src  test.ttt
```

Reference

OpenPlatform_v4.01_Programming_Guide.pdf > 2. Developing an Application > 2.2 How to Run Docker Images, Step 4

6. Sample application location

When creating an app by loading a Docker image on Wisenet Open Platform (WOP) 4.01 or above, a test application was created when performing verification in Hanwha Techwin. This is a sample application.

On this page, we are going to describe its source code location and how to create a build.

SThe sample application is a test application used to create an app by loading a Docker image on Wisenet Open Platform (WOP) 4.01 or above. The build location and build method of the sample application is as below.

Sample application location

```
$> /opt/opensdk/opensdk-4.01/SampleApplication
```

Example

```
oot@20090071-L03:/opt/opensdk/opensdk-4.01/SampleApplication$ ls
test_PTZ  test_Storage  test_log  test_sunapi_api  wn7
root@20090071-L03:/opt/opensdk/opensdk-4.01/SampleApplication$ cd wn7/
root@20090071-L03:/opt/opensdk/opensdk-4.01/SampleApplication/wn7$ ls
DynamicCPU_MemoryChange  PeopleCounter  RawImageTest  SNTest  ServerPushMJPEG
test_Email  test_FTP
root@20090071-L03:/opt/opensdk/opensdk-4.01/SampleApplication/wn7/$ cd
test_FTP
root@20090071-L03:/opt/opensdk/opensdk-4.01/SampleApplication/wn7/test_FTP#
make clean;make;opensdk_packager
rm -f bin/*   src/buffermanager_wn7.o  src/test_FTP_wn7.o
src/Opensdk_FTP_wn7.o *.cap
```

7. Connecting to a Docker container in a remote server by using Visual Studio Code in Windows 10

To connect to a docker container on a remote server running Windows 10 in Wisenet Open

Platform (WOP) 4.01, follow the steps below:

Requirements

- Windows10
- Windows Terminal (download from Microsoft Store)
- Container created by running a Docker image on a remote server

How to connect

1 Install WSL and Docker Desktop for windows

Step 1. After executing Windows terminal as an administrator, enter the following commands in order and reboot.

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

Step 2. Download the Ubuntu distribution version from the Microsoft Store, install, and run it.

Step 3. Enter `wsl –l` in the Windows terminal to confirm that Ubuntu Linux is installed.

Step 4. To update the wsl2 Linux kernel, refer to https://docs.microsoft.com/ko-kr/windows/wsl/install-win10#step-4---download-the-linux-kernel-update-package.

Step 5. Enter `wsl –l –v` in the Windows terminal to check the current WSL version.

Step 6. `Enter wsl –set-default-version 2 to change the default to wsl2.`

Step 7. Download Docker Desktop for Windows from the following link, install, and run it: https://hub.docker.com/editions/community/docker-ce-desktop-windows.

Step 8. After running, check the option to use the WSL2 based engine in the general tab of the settings page, check the Enable integration with my default WSL distro option in the resources tab, and activate Ubuntu-20.04.

2 SSH-remote connection in Visual Studio Code

Step 1. Install the remote development extension in Visual Studio Code

Step 2. Execute the command below on the Ubuntu server where the container is running.

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install ssh
sudo apt-get install openssh-server
sudo nano /etc/ssh/sshd_config
sudo service ssh status
sudo service ssh start
sudo ufw enable
sudo ufw allow 22
sudo ufw reload
```

Step 3. SSH-remote connection in Visual Studio Code

```
sc config ssh-agent start=auto
net start ssh-agent
```

Step 4. Run PowerShell on Windows and enter the command below.

```
ssh-keygen -t rsa -b 4096
```

Step 5. After entering Get-Content ..ssh\id_rsa.pub, save the output to /root/.ssh/authorized_keys on the Ubuntu server and change the permission by chmod 644 /root/.ssh/authorized_keys.

Step 6. After pressing F1 in Visual Studio Code and clicking Remote-SSH: Connect to Host, run Configure SSH Hosts to open the config file and enter the IdentityFile information under the server to be used as below.

```
Host 192.168.38.80
  HostName 192.168.38.80
  User simba
  IdentityFile ~/ .ssh/id_rsa
```

Step 7. Click Remote-SSH: Connect to Host in Visual Studio Code to connect to the Ubuntu server.

Step 8. In Visual Studio Code, press F1 and click Preferences: Open Settings (JSON) to open the settings.json file and enter the following contents at the bottom.

```
"docker.host":"ssh://simba@192.168.38.80"
```

Step 9. Enter sudo usermod –aG docker on the Ubuntu server.

```
sudo usermod –aG docker simba
```

Step 10. Press F1 in Visual Studio Code and click Remote-Containers: Attach to Running Container to display a list of containers that are running on the Ubuntu server. Choose the one you want and wait.

Step 11. If a segmentation fault occurs during project build (make clean; make), refer to 9. WOP Docker Malfunction (page 9) to disable the WSL2 option.

8. Port forwarding for multi-SOC cameras

Port number 8080 used by cameras with a single SOC should not be used with multiple SOCs.

Channel	Old	New	Output
CH1	http://192.168.38.188:8080/dashboard/	http://192.168.38.188:10005/	Connection OK
CH2	http://192.168.38.188:8080/dashboard/	http://192.168.38.188:10006/	Connection OK
CH3	http://192.168.38.188:8080/dashboard/	http://192.168.38.188:10007/	Connection OK
CH4	http://192.168.38.188:8080/dashboard/	http://192.168.38.188:10008/	Connection OK

9. WOP Docker Malfunction

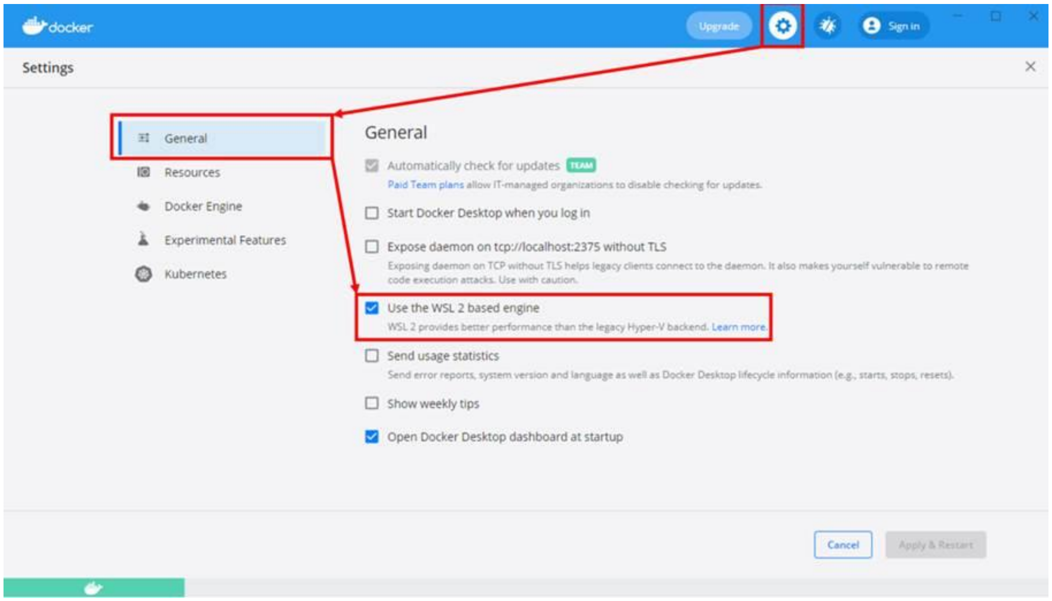
When developing an application using Docker on Wisenet Open Platform (WOP) 4.01 or above, seq.fault may occur if the WSL2 option is on. We recommend turning off the WSL2 option in Docker. To disable the WSL2 option, follow the steps below:

Disabling the WSL2 option

Step 1. Click the Docker settings button.

Step 2. Uncheck 'Use the WSL 2 based engine' in the General menu.

Step 3. Restart Docker.



General

☑ Automatically check for updates TEAM
Paid Team plans allow IT-managed organizations to disable checking for updates.

☐ Start Docker Desktop when you log in

☐ Expose daemon on tcp://localhost:2375 without TLS
Exposing daemon on TCP without TLS helps legacy clients connect to the daemon. It also makes yourself vulnerable to remote code execution attacks. Use with caution.

☑ Use the WSL 2 based engine
WSL 2 provides better performance than the legacy Hyper-V backend. Learn more.

☐ Send usage statistics
Send error reports, system version and language as well as Docker Desktop lifecycle information (e.g., starts, stops, resets).

☐ Show weekly tips

☑ Open Docker Desktop dashboard at startup

Running error example

```
$docker load -i wop_4.01.tar
$docker run --rm -it wop:4.01 /bin/bash
$docker ps 로 container 확인함(다른 terminal에서)
# cd /opt/opensdk/opensdk-4.01/SampleApplication/PeopleCounter/
# make
Building file: src/mjpegServer.cpp
Invoking: Cross G++ Compiler
/opt/opensdk//opensdk-4.01/armv7-wn7-linux-gnueabihf/bin/armv7-wn7-linux-gnueabihf-g++ -O2 -g -Wall -fmessage-length=0 -c -o 'src/mjpegServer_wn7.o' 'src/mjpegServer.cpp' -I/opt/opensdk//opensdk-4.01/armv7-wn7-linux-gnueabihf/bin/../armv7-wn7-linux-gnueabihf/sysroot/usr/include/ -I/opt/opensdk//opensdk-4.01/common/inc/ -Iinc/
make: *** [Makefile:153: src/mjpegServer_wn7.o] Segmentation fault
```

10. Port forwarding for Remote Debug Viewer

If Remote Debug Viewer does not work normally, check whether the Remote Debug Viewer port is open.

Setting port forwarding

From Docker

When creating a container, use the “-p” option to port forward the host port.

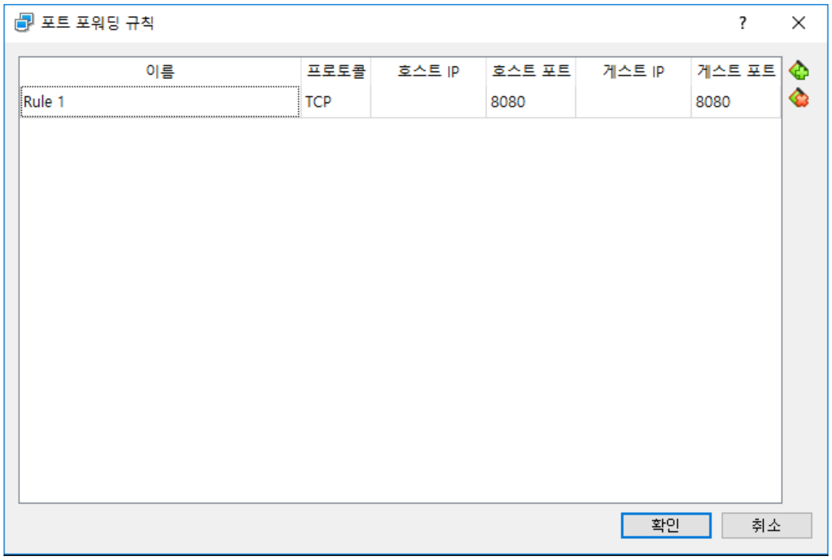
Ex) When setting the receiving port to 8080

```
# docker run -p 8080:8080 --rm -it --volume="$PWD:/mnt" wop:4.02
```

From VirtualBox

You need to port forward on network settings from VirtualBox.

1. Settings > Network > Advanced > Port Forwarding



포트 포워딩 규칙

이름	프로토콜	호스트 IP	호스트 포트	게스트 IP	게스트 포트
Rule 1	TCP		8080		8080

2. Enter the following commands on the Linux shell.

```
root@wisenet-VirtualBox:/home/wisenet/Window/SampleApplication_cv2x/test_SDmount#
 RemoteDebugViewer -i 192.168.38.185 -p 80 -u admin -w 5tkatjd! -d 8080 -a test_SDmount
flag_uname 1 flag_password 1
../src/main.cpp:125 Control is here. URL is http://192.168.38.185:80/stw-cgi/opensdk.cgi?msubmenu=debug&action=set&AppID=test_SDmount&Enable=True&Port=8080
../src/main.cpp:135 Control is here. camera credentials are admin:5tkatjd!
curl response: OK
../src/main.cpp:75 Control is here. Debug mode is set to ON
 Data: test_SDmount one_shot

 Data: test_SDmount : one_shot
 Data: test_SDmount opensdk_startSDcardUse err : 2
```

3. On the test app, the code should be entered as follows:

Ex) File

```
/opt/opensdk/opensdk-4.01/SampleApplication/test_SDmount/src/test_SDmount.cpp
```

```
void one_shot(void)
{
    debug_message("one_shot\n");
    ...

    err = OPENSDK::RECORD::opensdk_startSDcardUse(sd_number,type);
    debug_message("opensdk_startSDcardUse err : %d", static_cast<int>(err));
    ...
```

11. The number of threads running

Basically, 3 main threads run at the same time. One is for encoded video, one for raw video, and one for the other events (FD, MD, VA, and so on).

Each thread sends data. If your code stays in recv_event() while processing raw video, you cannot receive raw video events anymore. Encoded video events can still be received.

12. Viewable area of encoding video and raw video

Encoding video and raw video do not have the same viewable area. They have different viewable areas.

For example, SNB-5004 (1.3MP camera) captures video with a 5:4 ratio, SNB-6004 (2MP camera) with 16.9, and SNB-7004 (3MP camera) with 4:3.

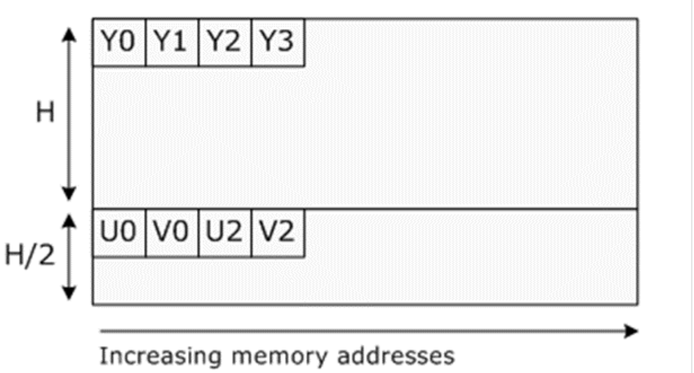
In order to create different resolutions from the original image, in the case of encoding, we just remove a part of the viewable range. For raw video, we do scaling.

13. Viewing the YUV420 video format

Y and UV are separated and concatenated in the memory.

The Y plane consists of serial Y data, from Y0 to Yn-1, and UV is interleaved in the plane, as U0 V0 U1 V1.

Please refer to the following (H stands for height):



14. Logs of the Open Platform Web interface

There is no support for web-related logs on opensdk.

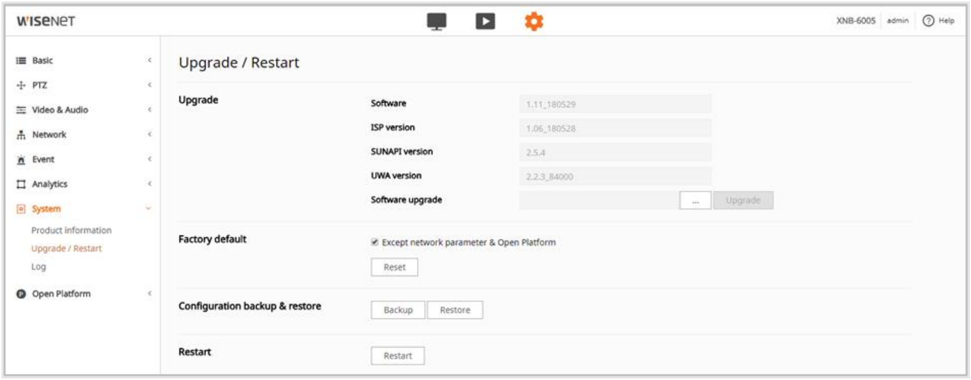
The web developer of the camera development team also debugs using the logs provided by the browser.

15. Backing up the camera settings

To back up the current settings

Step 1. In the camera web viewer, click [Backup].

Step 2. A backup file named “model name Config.bin” will be created.



Upgrade / Restart

Upgrade

Software 1.11_180529

ISP version 1.06_180528

SUNAPI version 2.5.4

UWA version 2.2.3_84000

Software upgrade

Factory default

☑ Except network parameter & Open Platform

Configuration backup & restore

Restart

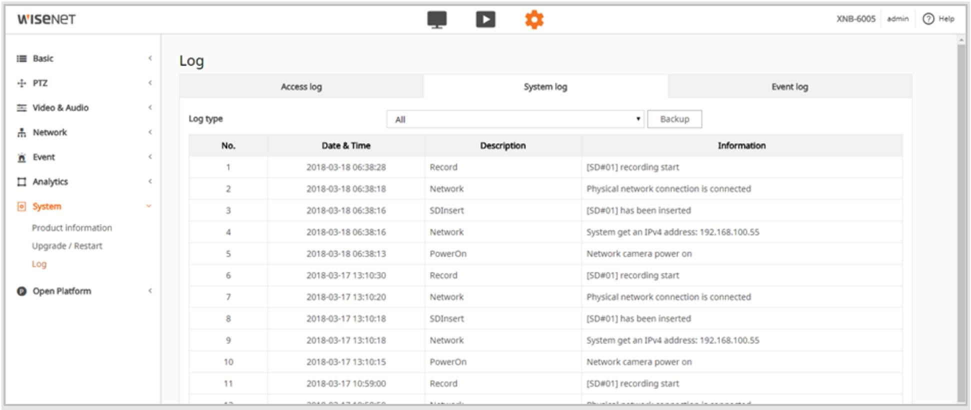
16. How to check log files

You can check the system logs or event logs.

Step 1. From the Setup menu of the camera web viewer, select the [System] tab

Step 2. Click [Log].

Step 3. Click on the [Backup] button to save all the log data for the currently selected mode in the "time stamp value created by camera in modelname-mode-camera.txt" file in the download folder of the browser



Log

Log type: All

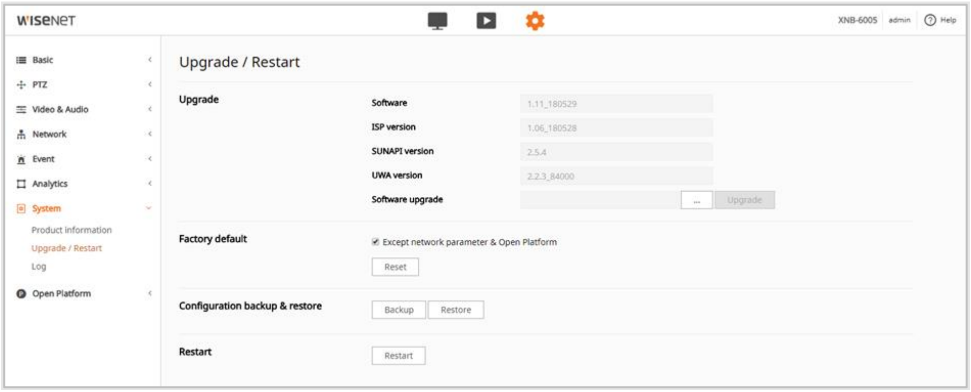
No.	Date & Time	Description	Information
1	2018-03-18 06:38:28	Record	[SD#01] recording start
2	2018-03-18 06:38:18	Network	Physical network connection is connected
3	2018-03-18 06:38:16	SDInsert	[SD#01] has been inserted
4	2018-03-18 06:38:16	Network	System get an IPv4 address: 192.168.100.55
5	2018-03-18 06:38:13	PowerOn	Network camera power on
6	2018-03-17 13:10:30	Record	[SD#01] recording start
7	2018-03-17 13:10:20	Network	Physical network connection is connected
8	2018-03-17 13:10:18	SDInsert	[SD#01] has been inserted
9	2018-03-17 13:10:18	Network	System get an IPv4 address: 192.168.100.55
10	2018-03-17 13:10:15	PowerOn	Network camera power on
11	2018-03-17 10:59:00	Record	[SD#01] recording start

17. Checking the camera firmware version

The supported features are listed below.

Step 1. From the Setup menu of the camera web viewer, select the [System] tab

Step 2. Click [Upgrade / Restart].



Upgrade / Restart

Upgrade
Software: 1.11_180529
ISP version: 1.06_180528
SUNAPI version: 2.5.4
UWA version: 2.2.3_84000
Software upgrade

Factory default
☑ Except network parameter & Open Platform

Configuration backup & restore

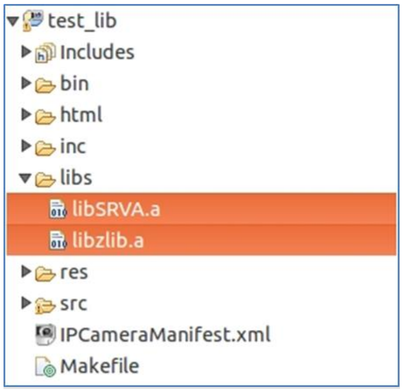
Restart

18. Adding a library file in Open Platform

To add a library file in Open Platform,

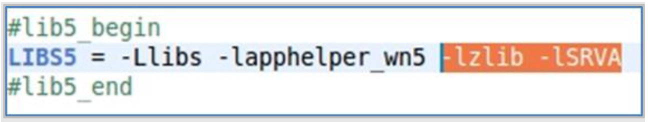
Step 1. Build the library you want to use with the Hanwha openplatform tool chain.
Since the space provided for the Open SDK is about 18 MB, you need to build it with only the necessary parts.

Step 2. Add the library file to any folder in your application (default: / libs)



```
▼ test_lib
  ▶ Includes
  ▶ bin
  ▶ html
  ▶ inc
  ▼ libs
      libSRVA.a
      libzlib.a
  ▶ res
  ▶ src
    IPCameraManifest.xml
    Makefile
```

Step 3. Add library-specific options to the Makefile



```
#lib5_begin
LIBS5 = -Llibs -lapphelper_wn5 -lzlib -lSRVA
#lib5_end
```