Reeplayer Camera

Software Installation Manual

(V1)

Change Log

12/28/2021

- Initial version of the software installation manual

Contents

1. Introduction

This document is the software installation manual for the Reeplayer camera system.

2. Hardware system

1. Jetson Xavier NX

The camera software system was developed and tested on the Jetson Xavier NX developer kit from Nvidia, but also targets the Jetson Xavier NX module on third-party carrier boards. The major differences for third-party carrier boards include:

(1) The carrier board may not support booting from a micro SD card, so the operating system needs to be installed in the on module eMMC storage, or NVMe SSD.

(2) The WiFi module that comes from the carrier board may work in a different mode than the WiFi module on the Nvidia developer kit. So the settings for dual mode WiFi (WiFi client + AP) need to be further tested. (3) The settings of dual CSI cameras with custom driver for IMX477 sensor need to be further tested. The camera software was designed to work on a custom IMX477 driver that supports 4032x3040 resolution (which is not supported by the official driver from L4T and JetPack).

2. Dual CSI cameras

The camera software system was developed and tested on a pair of cameras that were equipped with an IMX477 sensor and a low distortion lens. Other sensors and lens may not work.

3. Software installation

The final production of the camera system will use a simple way to install and configure the software system on the hardware, for example, directly copy or extract the entire software system from a backup system or an installation package.

But currently, during the development and testing, the camera software system needs to be installed step by step, as is listed in this section.

3.1. Install system software

The system software refers to the L4T system with SDKs and tools from Nvidia that is necessary to support the Jetson hardware features, including the GPU and dedicated media processing units. Current camera software was developed and tested on JetPack 4.6 (L4T 34.6.1).

The camera software will be tested and upgraded to the “latest” L4T system when necessary to utilize the new features provided by the newer system software.

1. **Choose a install method**

Nivia provides two ways to install the system software on the Jetson hardware:

(1) using the JetPack micro SD card image to extract the entire system on a micro SD card.

(2) using the SDK Manager to install the system software to the eMMC storage on the Jetson module or the external MVMe SSD. Depending on the hardware system, e.g. the developer kit from Nvidia or Jetson module on a third-party carrier board, one of the above two ways should be used.

2. **Username of the system**

A user name will be created during the installation. The camera software was designed to work with arbitrary usernames and passwords on the native system. But for consistency and convenience of the future maintenance, it is recommended to create a default user with the same name for all camera devices, for example “camera” or “reeplayer”. Current software development and testing is done on a system with “reeplayer” as the username. But arbitrary usernames will be tested for future versions.

3. **Following the official guides**

To install the system on a micro SD card, please follow the instructions in the link below to (1) write the image to the micro SD card and (2) setup and first boot the system.

https://developer.nvidia.com/embedded/learn/get-started-jetson-xavier-nx-devkit

To install the system to eMMC storage with SDK manager, please follow the instructions in the link below to (1) install the L4T system and necessary components with SDK manager (2) setup and first boot the system.

https://docs.nvidia.com/sdk-manager/install-with-sdkm-jetson/index.html

If the system is a Jetson module with a third-party carrier board, please refer to the guide from the manufacturer of the carrier board to complete the installation. Usually the installation will be based on the above methods but with extra steps to inject or install the board specific drivers into the official system from Nvidia.

4. **Trim and clean the system (optional)**

After the installation following the official guides, the L4T system will be installed with a desktop environment and a bunch of software tools, most of which is not necessary to the camera

software. For the final production, the system will be reconfigured (with trim and clean) to a minimal version to save the resources.

Currently, a simple clean of the system could be done to remove unnecessary software, by running the script “system_clean.sh” in a terminal. The “system_clean.sh” may be distributed with this guide, or found in the “camera-production” package.

3.2. Install driver for camera sensors

The camera software works with a pair of IMX477 camera sensors. From JetPack 4.6 (L4T 34.6.1), the IMX477 sensor is officially supported by the system. But the system-provided driver does not support the resolution (4032x3040@30) used by the camera software, so a custom driver needs to be installed and configured.

1. **Install the custom driver**

Please find the latest driver released in the link below:

https://github.com/ArduCAM/MIPI_Camera/releases

Download and run the “insall_full.sh” script in a terminal as below to install the driver “imx477”:

```
Install_full.sh -m imx477
```

This script will download the driver information file and binary driver files to the same directory and complete the installation. For a backup, it is recommended to run the script in a specific location like below:

```
mkdir -p ~/drivers/imx477
cd ~/drivers/imx477
./insall_full.sh -m imx477
```

2. **Configure for dual CSI cameras**

From JetPack 4.6 (L4T 34.6.1), a hardware I/O configuration is necessary to support the dual CSI cameras.

The official L4T system provided a configuration script in “/opt/nvidia/jetson-io/jetson-io.py”. IT SHOULD BE IGNORED. Instead, the script provided by the custom camera driver in “/opt/arducam/jetson-io/jetson-io.py” should be used to complete the configuration as below:

```
cd /opt/arducam/jetson-io
python jetson-io.py
```

The “jetson-io.py” provided by Arducam provides the same interface as the “jetson-io.py” provided by the official L4T system. So please follow the same instructions in the link below:

https://docs.nvidia.com/jetson/l4t/index.html#page/Tegra%20Linux%20Driver%20Package%20Development%20Guide/hw_setup_jetson_io.html#

Find “Configuring the CSI Connector” and configure it to use “Camera IMX477 Dual”, following the instructions.

3. **Verify the camera’s working mode**

After rebooting the system, verify the camera's working mode using v4l2 tools.

First, install the tool in a terminal as below:

```
sudo apt-get update && sudo apt-get install v4l-utils
```

Then, run the following command:

```
v4l2-ctl --list-devices --list-formats-ext
```

The output should look as below:

```
---------------------------------------------------------------------
vi-output, imx477 9-001a (platform:15c10000.vi:0):
        /dev/video0

vi-output, imx477 10-001a (platform:15c10000.vi:2):
        /dev/video1

ioctl: VIDIOC_ENUM_FMT
        Index       : 0
        Type        : Video Capture
        Pixel Format: 'RG10'
        Name        : 10-bit Bayer RGRG/GBGB
                Size: Discrete 4032x3040
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 3840x2160
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 1920x1080
                        Interval: Discrete 0.017s (60.000 fps)
---------------------------------------------------------------------
```

It indicates that two cameras (/dev/video0 and /dev/video1) are found, and the resolution “4032x3040” is in the formats list.

4. **Test video capture from cameras (optional)**

The script “jetson_preview.sh” or “pairwise_preview.sh” could be found with this guide or found in the “camera-production” package, which can be used to verify the video capture from the cameras.

Running the script in terminal with “-h” will display the supported options. The important options include:

The “-i” is used to choose the camera device (0 or 1).

The “-m” is used to choose the sensor formats, e.g. “3040p” for resolution “4032x3040”.

The “-r” is used to control the rotation of the image.

The “-s” is used to downscale the image displayed on screen, e.g. “-s 0.25” with “-m 3040p” will display the video in a “1008x760” window on screen.

3.3. Install camera software

Please note that the installation steps here are only for the testing version. The production installation will be different, at least in two ways:

(1) the following installation is completed on the L4T desktop environment and then set the system working in console mode after installation, while the production installation will be done on a console only system.

(2) The production system will be set to work in dual WiFi mode (WiFi client + AP), so the client devices (that runs the administration web pages or the camera app) connect to the camera system through the AP set on the camera device, while the following steps set the camera system working in WiFi client mode. Both the camera device and the client device connect to the same WiFi router (or same local network).

1. **Prepare for camera software installation**

The camera software mostly works in docker containers, so there is not much necessary to install on the native system. A script “system_install.sh” could be found with this guide or in the “camera-production” package. Running the script in a terminal will complete the necessary configuration of the system and install the necessary software tools.

The script will create some default folders for the camera software, including:

- The “/opt/reeplayer/” will be the default location to install the camera software.
- The “~/camera_data” will be the default location for recorded video files.
- The “~/camera_log” will be the default location for software running logs.

After running the script, please restart the system to apply the changes for docker.

2. **Install the camera software**

First, download the latest camera software package from the link below:

https://github.com/maoxuli/reeplayer_update

Then, install the software in a terminal as below (for example, for “camera-0.5.3.0.tar.gz”):

```
cp ~/downloads/camera-0.5.3.0.tar.gz /opt/reeplayer
cd /opt/reeaplayer
tar -xzvf camera-0.5.3.0.tar.gz
cd camera-0.5.3.0
./install.sh
```

Finally, manually start the camera software once to download and install the docker images. This step is optional. It could be done automatically on the next restart of the system. But because the downloading of docker images may take several minutes, it is better to run explicitly in a terminal so that we can see the detailed status (including possible errors).

```
cd /opt/reeplayer/camera/scripts
./camera.sh run
```

After the initial installation, the camera software can upgrade to a newer version by either automatically updating, or manually updating from the administrative web pages or the camera app.

3. **Connect camera device to local network**

The testing will be done by connecting both the camera system and the client software (the administrative web pages or the camera app) to the same local network, or the same WiFi router. It is optional to connect the camera to the router by ethernet cable or WiFi.

Use the “ifconfig” command in a terminal to see the IP address of the camera system for the ethernet interface or the WiFi interface, as below:

```
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.1.109  netmask 255.255.255.0  broadcast 192.168.1.255
        ether 48:b0:2d:07:8c:d0  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 37

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.218  netmask 255.255.255.0  broadcast 192.168.1.255
```

```
inet6 fe80::2f85:6fef:1954:226b  prefixlen 64  scopeid 0x20<link>
ether 70:66:55:b1:32:13  txqueuelen 1000  (Ethernet)
RX packets 4778  bytes 886337 (886.3 KB)
RX errors 0  dropped 0  overruns 0  frame 0
TX packets 2199  bytes 328748 (328.7 KB)
TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

If the “Connect automatically” is set in the settings of the WiFi connection (should be set by default), the WiFi will be connected automatically and the assigned IP address for the camera device should be kept unchanged (usually) on the next startup. In the production system, the IP address for the camera may be set to a static one, but it is optional for the testing system.

4. **Set the camera system working in console mode**

The desktop environment is not necessary for the camera software. So the camera system will be set to work in console mode after the software installation, with the command below in a terminal:

```
sudo systemctl set-default multi-user
```

On the next reboot, the system will start into console mode by default. The monitor and the keyboard/mouse are also not necessary to be connected. The status of the camera system can be monitored only by the administrative web pages or the camera app.

If there is a necessity to run the system in graphical mode temporarily for installation or settings, start the system with a monitor and keyboard/mouse connected, run the below command in a terminal:

```
sudo systemctl start gdm3.service
```

3.4. Administrative web pages

The camera software supports some built-in administrative web pages to set up the system and monitor the status. Once the camera software is installed following the above steps, and the system is restarted, the camera software will be launched by default, the administrative web pages are accessible too.

1. **URL for camera web pages**

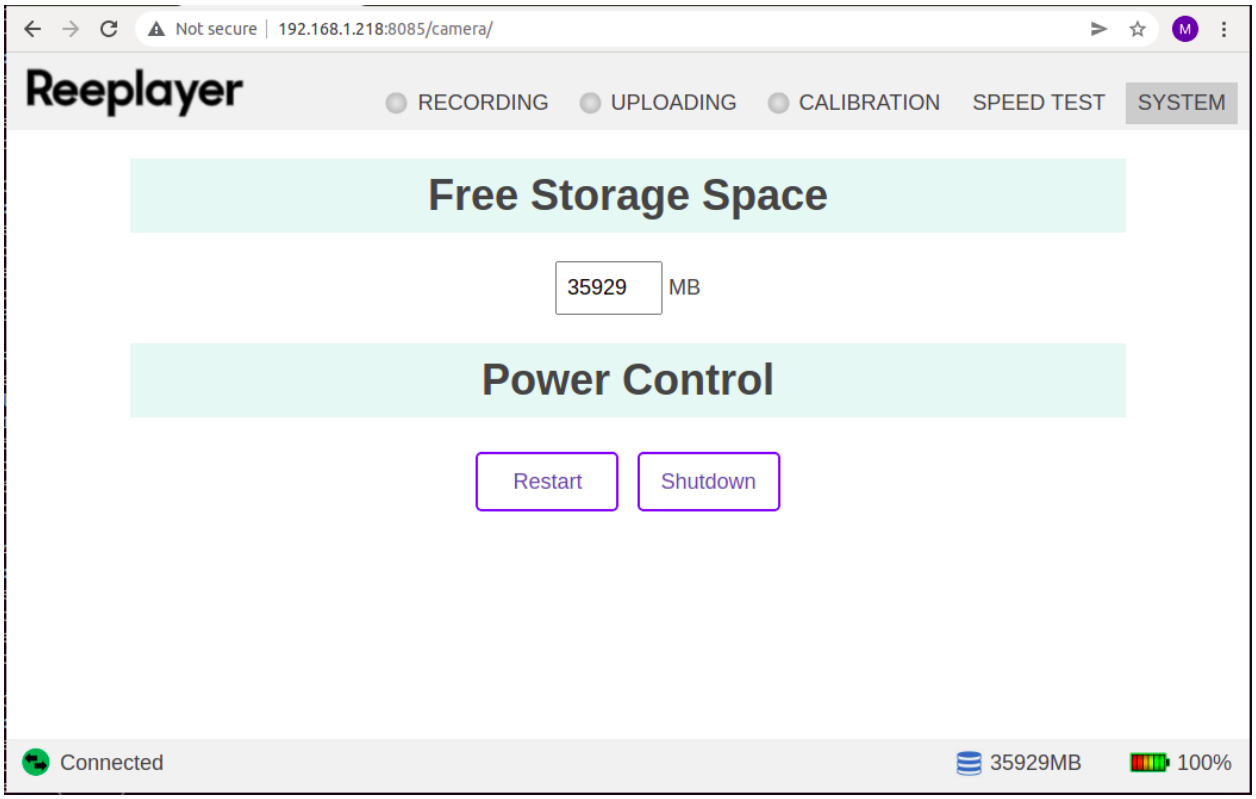
To use the administrative web pages, we need to know the IP address of the camera system. For example in the above step from the “ifconfig” command in terminal window, we know the WiFi IP is “192.168.1.218”, then we can access the camera web pages with below url, in a web browser:

http://192.168.1.218:8085/camera

Currently, the web pages are developed and tested on a PC/laptop that has a big screen with resolution higher than 1280x720. It can also work well on iPad or other tablet with similar resolutions. For smartphones, the functions of the web pages are normal but the layout of the web pages has more or less issues. So it is recommended to do the tests with a laptop or an iPad/tablet.

2. Restart and shutdown the system

In the “System” page, there are button to “restart” ro “shutdown” the system, as below:

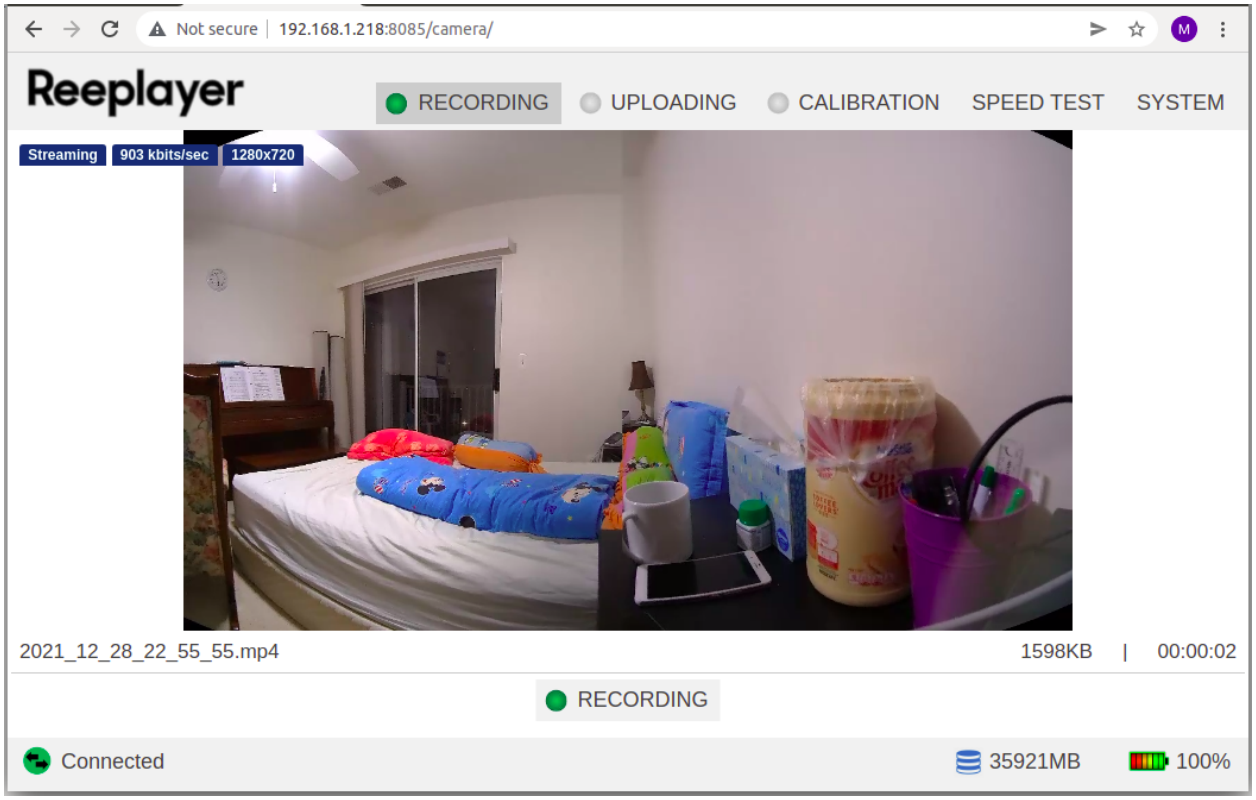


The “System” page is designed to configure the system and monitor the system status. Now there is only “free storage space” available as an example.

Some common status of the system will be displayed acrossing the web pages, for example, the status of “recording”, “uploading” and “calibration” will be displayed with a small icon indicator on the top menu items. The “connection”, “free storage space”, and “battery level” are displayed in the bottom status bar.

3. Start and stop video recording

In the “Recording” page, the live video from cameras is previewed. The recording can be “start” and “stop” by the button below the image, as below:



Please note that the camera in the screenshot is not calibrated so the images from two cameras are not aligned well.

In the “Recording” page, the video file name, the byte size of the video file, and the duration of the video file will be displayed, when the recording is ongoing.

The resolution for the video preview and the bitrate of the video streaming is also displayed on the live video. Please note that the resolution displayed is only for the previewed video. The resolution of the recorded video is pre-set to 4K by default (could be configured in future versions).

The live preview with video streaming is still not working smoothly in the current version. When switching the web pages to the “Recording” page, the video streaming is started slowly. it will succeed in several to tens of seconds, so please wait patiently.

4. Other web pages

The other web pages, including “Uploading” and “Calibration” pages, are still in debugging, so disabled in this testing version.