

Zao SDK for Jetson User's Guide

Revision 1.0-2

Soliton Systems
2023/05/12

Restrictions and special notes

- Zao SDK” is a copyrighted work of Soliton Systems Corporation, and the copyright and other rights to it belong to Soliton Systems Corporation.
- All other company and product names are registered trademarks or trademarks of their respective companies.
- The contents of this product are subject to change without notice for product improvement.
- The connection configurations shown in this document are examples for reference only, and not all combinations are guaranteed.

STC Support Center, Soliton Systems, Inc.
2-4-3 Shinjuku, Shinjuku-ku, Tokyo 160-0022, Japan
TEL: (03)5360-3850

Table of Contents

1.	Introduction.....	6
2.	Terminology.....	7
3.	What is Zao SDK?.....	9
3.1.	Zao SDK.....	9
3.2.	Zao SDK for Jetson Features.....	9
3.3.	Structure of Zao SDK for Jetson.....	11
4.	Setup of Zao SDK for Jetson.....	12
4.1.	JetPack Setup.....	12
4.2.	Zao SDK for Jetson Setup.....	12
4.3.	Pairing.....	13
5.	Basic Operation.....	14
5.1.	Startup and Shutdown Procedures.....	14
5.2.	TUI Screen.....	14
5.3.	Log in to Shell.....	16
5.4.	Web camera.....	16
5.5.	Switching to multiple video composition operation.....	16
5.6.	Serial tunneling.....	17
5.7.	Network Interface.....	18
5.8.	Startup and shutdown procedures in case of GUI mode.....	18
6.	Structure.....	19
6.1.	Overall Structure.....	19

6.2.	ZaoInfraCtrl Process.....	20
6.2.1.	InfraCtrl.conf	20
6.3.	ZaoRemoteEndpoint Process	21
6.4.	ZaoVideoSource Process	22
6.5.	ZaoAudioSource Process	23
6.6.	ZaoAudioSink Process.....	23
6.7.	ZaoSerialProxy Process.....	23
6.8.	ZaoControl Process	23
7.	Customize	25
8.	Alert function.....	26
9.	Log file.....	27
10.	Impact of Zao SDK for Jetson Setup.....	28
10.1.	Adding, enabling, and disabling system services	28
10.2.	rsyslogd, logrotate configuration.....	28
10.3.	udev rules.....	28
10.4.	Account Groups.....	29
10.5.	GUI disable, auto-login.....	29
10.6.	Adding Kernel Modules	29
10.7.	ZaoVideoSource, etc.....	29
11.	Various Procedures	31
11.1.	Using the Wi-Fi USB Dongle	31
11.1.1.	Configuration in the GUI environment during SDK setup.....	31

11.1.2.	Setting in text mode.....	31
11.1.3.	Other.....	31
11.2.	GUI Enable.....	31
11.3.	GUI Disable.....	31
11.4.	Uninstallation.....	32
12.	Default setting value.....	33
12.1.	Common.....	33
12.2.	Jetson Nano Developer Kit (Rev. B01).....	33
12.3.	Jetson Xavier NX Developer Kit.....	34
12.4.	Jetson AGX Xavier Developer Kit.....	35
12.5.	Aetina AX810.....	36
13.	Update history.....	37

1. Introduction.

The Zao SDK for Jetson covered in this document is Version 1.0.x.x. 1.0.x.x is generally referred to as Ver. 1.0.

In the command execution example, blue letters indicate the part of the command to be input.

Related documents are listed below.

- Zao SDK Quick Start Guide
- Zao SDK for Jetson / libzep API Reference
- zao-sdk.org website

2. terminology

Table 2- shows terms related to Zao SDK for Jetson.

Table 2- Terms

terminology	Description.
SDK	Software Development Kit
Soliton MLU	Multi Link Unit from Soliton Systems with 4 LTE modems built in
Zao Cloud View	Windows application that receives and displays video sent from Zao SDK for Jetson via cloud
ZaoInfraCtrl	Process to control communication lines for USB-NIC, Soliton MLU, etc. in Zao SDK for Jetson
ZaoRemoteEndpoint	Zao SDK for Jetson's underlying processes for communication with Zao Cloud in Zao SDK for Jetson
RASCOW2	Algorithms and communication protocols developed by Soliton Systems for transmitting video and other data over a multilink.
ZaoVideoSource	One of the process types in Zao SDK for Jetson. It captures video and supplies it to ZaoRemoteEndpoint.
ZaoAudioSource	One of the process types in Zao SDK for Jetson. Captures audio and feeds it to ZaoRemoteEndpoint.
ZaoAudioSink	One of the process types in Zao SDK for Jetson. It receives audio data sent back from ZaoRemoteEndpoint and drives an audio device.
ZaoSerialProxy	One of the process types in Zao SDK for Jetson. It exchanges serial tunnel data with ZaoRemoteEndpoint and acts as a go-between for serial devices and other devices.
ZaoControl	One of the process types in Zao SDK for Jetson. Controls overall operation instructions and UI
TUI Main Screen	TUI (Text User Interface) screen displayed by ZaoControl, the default setup in Zao SDK for Jetson
netns	Linux Network Namespace feature, or a space created with this feature
default netns	netns that existed before netns was additionally created. Basically, the process runs on default netns
JetPack	OS environment for Jetson, etc. (consisting of L4T and various packages)

terminology	Description.
Linux for Tegra (L4T)	Linux for Jetson
SoM	System-on-Module
NetworkManager	L4T standard daemon utility for network control
ModemManager	L4T standard daemon utility for modem control

3. What is Zao SDK?

3.1. Zao SDK

Zao SDK™ is a system for stable transmission of video, audio, and control signals using Zao SDK for Jetson and Zao Cloud (Figure 3- Figure 31).

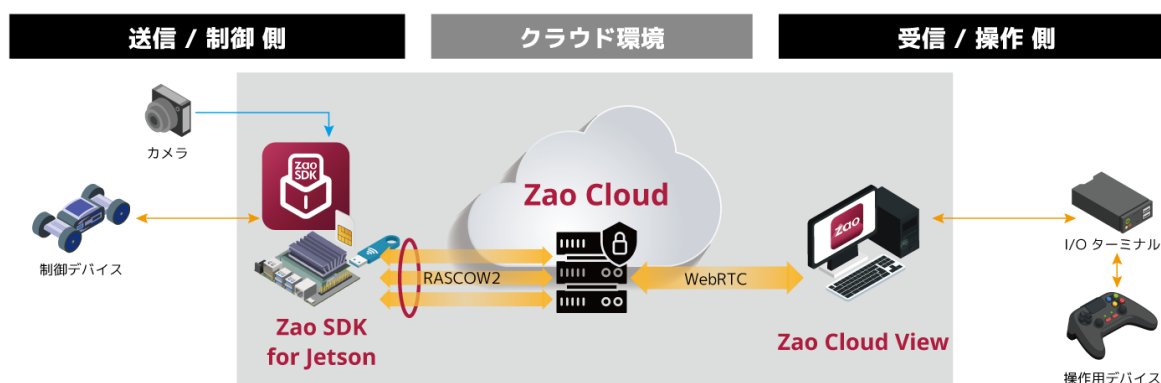


Figure 3- Zao SDK Overall view

3.2. Zao SDK for Jetson Features

Zao SDK for Jetson is an SDK for building video transmission and remote control terminals using Jetson in Zao SDK.

The main features of Zao SDK for Jetson are as follows

- Connect to Zao Cloud
- Multi-link and low latency transmission by RASCOW2
- The following can be sent
 - Video 1 system
 - 2 audio systems (not supported in Ver.1.0)
- Receive audio from Zao Cloud
- Serial tunneling communications.
- Video bit rate control based online conditions.
- Various customizations can be made by SDK users.

The main specifications of Zao SDK for Jetson 1.0 are shown in Table 3- shows the main specifications of Zao SDK for Jetson 1.0.

Table 3- Main specifications of Zao SDK for Jetson 1.0

Supported Platforms	Jetson Nano Developer Kit B01 Jetson Xavier NX Developer Kit Jetson AGX Xavier Developer Kit Can be used in environments with Jetson Nano SoM, Jetson Xavier NX SoM, and Jetson AGX Xavier SoM with adjustments
Supported JetPack	4.6.1 4.6.2 4.6.3
RASCOW2 Number of lines	Max. 8
Video resolution Frame rate*2	Up to 3,840x2,160 / 30fps 1,920x1,080 / 60fps
Supported Monitors*3	HDMI monitor (1,920x1,080 recommended)
Supported NIC*1	Recognized as ethX (where X is a number) in the JetPack environment
Supported cameras*1	Webcams with UVC support MIPI CSI-2 and GMSL2 cameras IP cameras
Supported audio*1	USB Audio recognized by Alsa (including microphone with built-in webcam)
Supported serial adapters*1	Recognized as ttyUSB by the ftdi_sio driver

*1 Please refer to the FAQ on zao-sdk.org for the manufacturers and model numbers of the devices that have been tested for compatibility.

*2 Varies depending on Jetson HW performance, Power Mode setting, image processing before encoding, etc.

*3 Depends on the Jetson Carrier Board.

3.3. Structure of Zao SDK for Jetson

This section outlines the structure of Zao SDK for Jetson. Zao SDK for Jetson consists mainly of the following (Figure 3- Figure 32).

- Executable binaries such as ZaoRemoteEndpoint
- Customizable parts
 - ◆ Pre-built executable binaries
 - ◆ source file
- Script files, configuration files, etc.

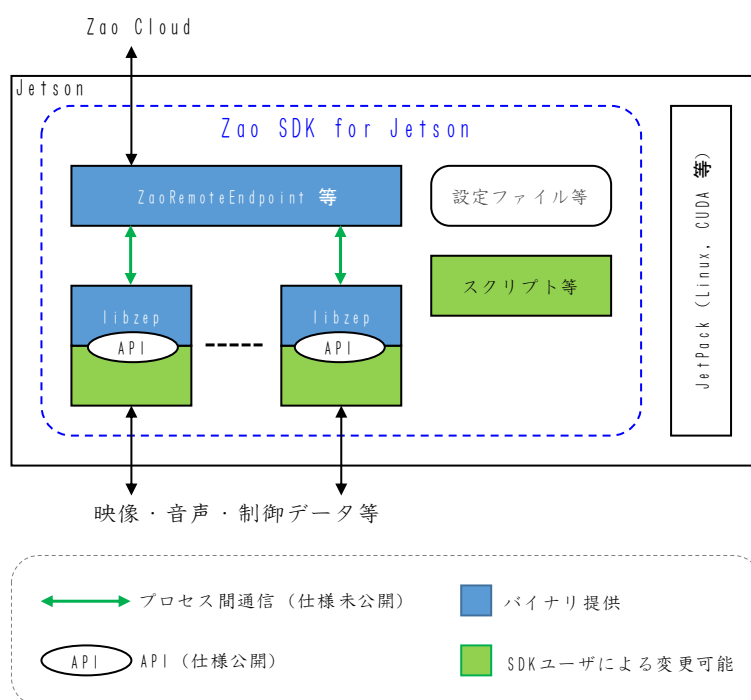


Figure 3- Structure of Zao SDK for Jetson

Executable binaries such as ZaoRemoteEndpoint are the processes that form the basis of operations such as communication with Zao Cloud. These are supplied in executable binary form and cannot be modified by SDK users, but their behavior can be adjusted to some extent in configuration files.

Apart from these processes, there are several processes that handle video, audio, control data, etc., which can be customized by SDK users. These processes are linked to the libzep library and access the libzep public API from the SDK user development side to use Zao SDK for Jetson functions.

The default form of Zao SDK for Jetson is supplied in pre-built binary, example source code as a form of customizable scope, allowing for example the following customizations from SDK users

- Change some example sources, rebuild, and replace only the binaries.
- Link libzep to applications developed by SDK users and connect to Zao SDK for Jetson

In addition, Zao SDK for Jetson includes configuration files and scripts to cooperate with Linux udev, systemd, rsyslog, etc.

4. Setting up Zao SDK for Jetson

This chapter describes the procedure for setting up Zao SDK for Jetson on each Jetson.

4.1. JetPack Setup

Follow the instructions for each Jetson platform to set up either JetPack 4.6.1, 4.6.2, or 4.6.3. For the Nano Developer Kit, the Zao SDK Quick Start Guide is also helpful.

The restrictions for each item you select in the JetPack initial setup procedure are as follows

- No language or time zone restrictions
English is recommended for language selection to prevent problems with other applications.
- Do not use zao or zaoauto in the initial account name
- Auto-login is recommended On

4.2. Zao SDK for Jetson Setup

After JetPack setup is complete, connect a monitor, keyboard, and mouse to the target and log in.

Internet access is required for setup. At this point, the Soliton MLU connection cannot be used, so please prepare a separate LAN, etc.

Copy the distribution (hereinafter referred to as `zao_sdk_jetson_1.0.0.tar.xz`, so please read the version accordingly) to your home directory on the target using a USB stick or similar device.

Open a terminal and execute the following commands in this order The first time you run `sudo`, you will be asked for your password.

```
$ sudo apt-get update
$ sudo apt-get -y install libqmi-utils socat dkms libcrypto++6 nano ¥
ca-certificates
$ tar xf zao_sdk_jetson_1.0.0.tar.xz
```

In the terminal, execute the following commands. Enter the appropriate platform number and the host name portion of the `ZAO_CLOUD_URL`. The process will take a few minutes.

```
$ sudo zao_sdk_jetson_1.0.0.0/setup_zao_sdk_jetson.sh
-----
Preset: Jetson AGX Xavier Developer Kit
Preset: Jetson Nano Developer Kit
Preset: Jetson Xavier NX Developer Kit
Preset: Aetina AX810
0. Custom

Select a setup type > 2
```

```
Input hostname of ZAO_CLOUD_URL > cvp-00000000-00v
```

```
Wait a minute ...
```

```
Done.
```

This completes the SDK setup procedure; shut down Jetson.

For the procedure to change back to GUI mode, please refer to the following section.11.2 for instructions on how to return to GUI mode.

4.3. pairing

Pairing procedures must be completed before video and other data can be sent to Zao Cloud. This procedure is required only for the first time after a new setup of Zao SDK for Jetson, and the pairing status will be maintained after rebooting Jetson or overwriting Zao SDK for Jetson installation.

Connect the following to Jetson and start it up

- HDMI Monitor
- keyboard
- Web camera
- Internet communication line (Soliton MLU available)

The control application (ZaoControl) starts automatically, Figure 4-1 The control application (ZaoControl) will automatically start and the screen display will look like Figure 41.

```
System(S)          Connect(C)
2023-03-10 17:06:53 +0900, SUID:
Control
[ Not Paired ]
```

Figure 4-1 TUI screen (Not Paired display)

This is the procedure for pairing with Zao Cloud. Please refer to the Quick Start Guide for specific instructions.

5. basic operation

This chapter describes the use of the default state immediately after setup; if any customization is made by the SDK user, the operation and operation methods are not limited to those described in this chapter.

The details are described below, but the screen transition in the default text mode is shown in the following figure. Figure 5- Figure 51 shows the screen transition in default text mode.

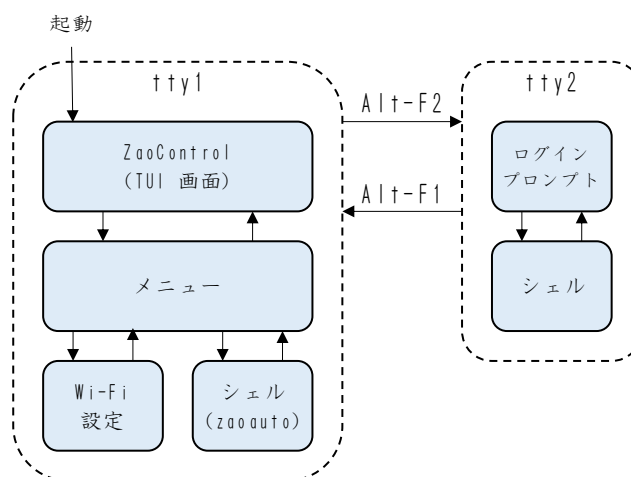


Figure 5- Screen Transitions

5.1. Startup and Shutdown Procedures

Zao SDK for Jetson operates in text mode by default after setup, and the TUI main screen is automatically displayed after Jetson is launched.

To reboot or PowerOff Jetson, use the System menu on the TUI screen, and check that the OS shutdown sequence is complete by checking that the LED on the board is off before unplugging the AC adapter. Some Jetson carrier boards and cases have a power button that automatically initiates the shutdown sequence at the touch of a button.

If the environment allows ssh remote login, you can also execute the command `sudo poweroff` from a remotely logged-in shell to turn off power.

5.2. TUI Screen

This section describes the TUI screen and operations.

Pressing the `s` or `c` key on the keyboard displays a menu for various operations (Figure 52). Figure 5- Figure 52). When a menu item is selected, an explanation is displayed in the Tips area at the bottom of the screen (Figure 5- Figure 53).



Figure 5- TUI Screen (Menu)



Figure 5- TUI Screen (Tips)

The Control area displays the status of the connection to Zao Cloud (Figure 54).Figure 5- The RASCOW2 area displays the RASCOW2 connection status (Figure 5- Figure 54).



Figure 5- TUI Screen (Control, RASCOW2)

RASCOW2 connection is made only during video transmission (OnAir state), so during standby (OffAir state), RASCOW2 is disconnected and connected to Zao Control. The main display states areTable 5- Table 51 shows the main display states.

Table 5- Main indications for Control area and RASCOW2 area

Control Display	RASCOW2 Display	Description.
Not Paired	Disconnected	unpaired state
Stopped	Disconnected	Zao Cloud and disconnected state
Network failed	Disconnected	Connection error with Zao Cloud
Connecting	Disconnected	Trying to connect to Zao Cloud
Connected	Disconnected	Connected to Zao Cloud OffAir Status
	Connecting	Connected to Zao Cloud Video transmission is in progress.
	Connected	Connected to Zao Cloud OnAir Status

The RASCOW2 Lines area displays whether each line (Lines 1-8) is Online or Offline while RASCOW2 is connected (Figure 5- Figure 55).

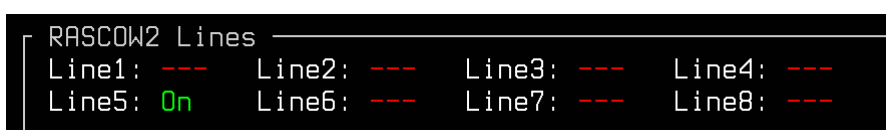


Figure 5- TUI Screen (RASCOW2 Lines)

In the Alert area, as a demonstration of the alert function, if the RASCOW2 Lines state is no longer multilink (i.e., only one line is Online), an alert will appear (Figure 56).Figure 5- Figure 56).

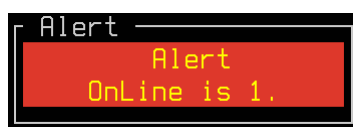


Figure 5- TUI Screen (Alert)

The Event Message area displays the last four messages about the operating event (Figure 57).Figure 5- Figure 57).



Figure 5- TUI Screen (Event Messages)

Selecting Close from the menu will exit the ZaoControl process and display the menu. The menu allows you to start the ZaoControl process again, configure Wi-Fi settings, PowerOff, etc. For details on Wi-Fi settings, see Section 11.1.11.1 See 11.1 for details on Wi-Fi settings.

5.3. Log in to Shell

To log in and operate Jetson's Linux, press Alt-F2 while the TUI screen is displayed. Press Alt-F1 to return to the TUI screen.

5.4. Web camera

By default after setup, the ZaoVideoSourceWebcam process, one of the sample video capture processes, is launched, and the video transmission operation is performed from a single webcam.

As a limitation of the ZaoVideoSourceWebcam implementation, please connect the webcam before starting Jetson; Jetson will not recognize the webcam if it is connected after starting Jetson. There are no restrictions on the position of the USB connector to connect the webcam. These restrictions do not necessarily apply when using the SDK user development process.

ZaoVideoSourceWebcam accesses /dev/video0 by default. To change the target device, rewrite the configuration file /usr/local/etc/zao/VideoSourceWebcam.conf and restart Jetson.

5.5. Switching to multiple video composition operation

In addition to the ZaoVideoSourceWebcam, the setup also includes the ZaoVideoSource4cam, a sample of the video capture process, which sends images from up to four cameras in an aligned pattern.

To make ZaoVideoSource4cam work, rewrite the part that auto-starts ZaoVideoSourceWebcam to auto-start ZaoVideoSource4cam. The procedure is as follows

- From the TUI screen, press Alt-F2 to switch to tty2 and log in with the account you initially set up for JetPack.
- Install the necessary packages with the following command

```
$ sudo apt-get install -y cuda-toolkit-10-2
```

- Start the editor with the following command

```
$ sudo nano /usr/local/lib/systemd/system/zao-video-source.service
```

- Edit as follows, adjusting the commented-out part

```
#ExecStart=/usr/local/bin/ZaoVideoSourceWebcam (abbreviated)
ExecStart=/usr/local/bin/ZaoVideoSource4cam (abbreviated)
```

- Save file and exit editor
- Start the editor with the following command to edit the configuration file. See the libzep API reference for the rules for writing configuration files.

```
$ sudo nano /usr/local/etc/zao/VideoSource4cam.conf
```

- Press Alt-F1 to return to the TUI screen and restart from the System menu

5.6. serial tunneling

Serial tunneling is a virtual serial connection between Jetson and the receiving PC (Figure 5-).

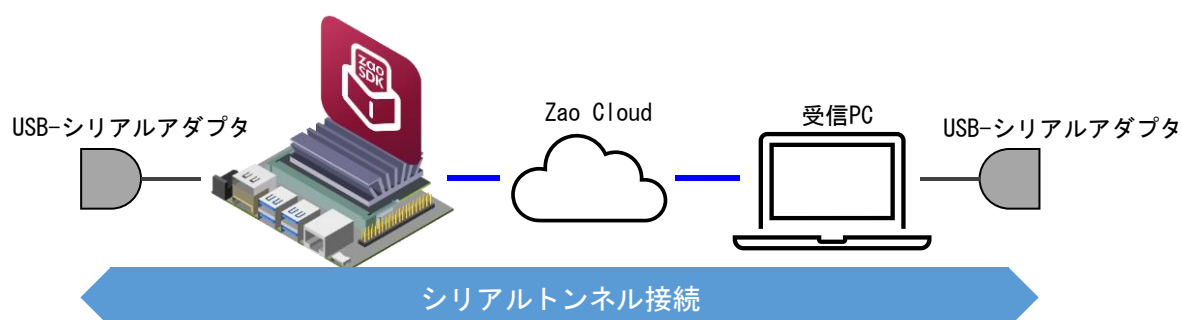


Figure 5- Serial Tunnel Connection

After setup, the sample ZaoSerialProxyTty process, a serial tunneling process, is invoked by default to enable serial tunneling operation with the USB-to-serial adapter as the end point.

USB–serial adapters are recognized even if they are connected after Jetson has been started. If multiple USB–serial adapters exist, the first one recognized will be the target device. Baud rate setting is done by Zao Cloud View.

See the FAQ on zao-sdk.org for supported USB–to–serial adapters.

5.7. network interface

The built–in Ethernet operates as RASCOW2 Line5. By default, it works with DHCP settings. The built–in Ethernet is controlled by Linux NetworkManager, which can be configured using `nmcli` and `nmtui` commands with a shell login.

Up to one USB Wi–Fi dongle can be used. For details on how to set up Wi–Fi, refer to the following section.11.1 See section 11.1 for details on Wi–Fi configuration.

When using the Soliton MLU, the MLU must be connected to a specific USB connector, and the correspondence between the USB NIC and the USB Modem and the RASCOW2 line varies depending on where they are connected in the USB device tree. The mapping is determined by the contents of the configuration file `/usr/local/etc/zao/InfraCtrl.conf`. Please refer to the description below for the default values and details of the configuration file.

5.8. Startup and shutdown procedures in case of GUI mode

The procedure below describes the startup and shutdown procedures when GUI is enabled.

In GUI mode, communication to Zao Cloud starts after login. If auto–login is off, no communication is performed and the user waits for login on the login screen. After login, the ZaoControl process is automatically executed in the terminal and communication to Zao Cloud is started.

For reboot and PowerOff, use the menu in the upper right corner of the GUI screen. as in text mode, file system corruption may occur if power is turned off without following the procedure.

6. structure

6.1. Overall Structure

The overall structure of the Zao SDK for Jetson is shown in Figure 6- Several processes cooperate with each other through inter-process communication. Each process in the center of the figure communicates with the underlying ZaoRemoteEndpoint using a common inter-process communication library, libzep. libzep API is available to SDK users, who can link libzep in their apps and access the API. Zao SDK for Jetson functionality can also be used from their own apps by linking libzep in the app developed by the SDK user and accessing the API. Each process will be explained later, but the process type and whether it can be customized by the SDK user are shown in Table 6-1. Table 6- shows the process types and whether they can be customized by the SDK user. The default configuration is to implement separate processes for each, but it is possible to handle both video and audio in a single process, for example.

ZaoInfraCtrl and ZaoRemoteEndpoint can also be customized to a certain extent by editing their configuration files.

In addition to these processes, Zao SDK for Jetson also utilizes Linux mechanisms such as systemd, udev, and rsyslogd.

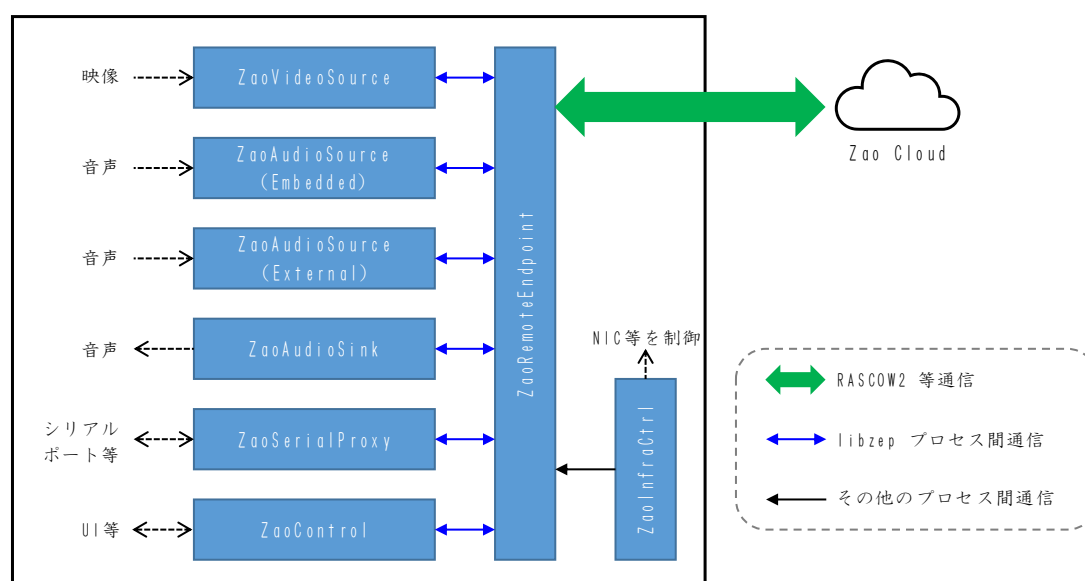


Figure 6- Zao SDK for Jetson Overall Configuration

Zao SDK for Jetson can send two audio streams, which are called Embedded and External because of the history of Zao products. Depending on the settings in Zao Cloud View, either one or neither of them will be sent to Zao Cloud. See the description of ZaoAudioSource below for details.

Table 6- Process to configure Zao SDK for Jetson

process type	Customization by SDK Users
ZaoInfraCtrl	not allowed
ZaoRemoteEndpoint	not allowed
ZaoVideoSource	acceptable
ZaoAudioSource	acceptable
ZaoAudioSink	acceptable
ZaoSerialProxy	acceptable
ZaoControl	acceptable

6.2. ZaoInfraCtrl Process

The ZaoInfraCtrl process controls the target network interface for multilink operation. Zao SDK for Jetson uses multiple netns for multilink operation, and ZaoInfraCtrl can move connected network interfaces to the corresponding netns or control them based on DHCP/fixed address configuration, depending on the configuration file. ZaoInfraCtrl determines where in the USB tree a USB-NIC or USB Modem is connected and determines the corresponding netns.

In addition, it provides controls specific to the Soliton MLU.

ModemManager is disabled during the SDK setup process because it conflicts with Linux's ModemManager, which is enabled by default in L4T.

6.2.1. InfraCtrl.conf

ZaoInfraCtrl operates based on a configuration file (/usr/local/etc/zao/InfraCtrl.conf). The settings in the default configuration file are described below.

The configuration file contains the following settings

- Correspondence between network interface connection locations and netns
- DHCP/fixed address settings for Ethernet interfaces for each netns
- Soliton MLU Control Configuration

The following is an example of a configuration file description (in part) for the netns correspondence part.

```
nwk_dev=1,MLU-LTE1,70090000¥.xusb/usb1/1-2/1-2¥.1/1-2¥.1¥.1/1-2¥.1¥.1:1¥. [25]/net/wwan
nwk_dev=1,USB1,70090000¥.xusb/usb1/1-2/1-2¥.1/1-2¥.1¥.1/1-2¥.1¥.1:
nwk_dev=1,USB1,70090000¥.xusb/usb2/2-1/2-1¥.1/2-1¥.1¥.1/2-1¥.1¥.1:.
```

In this description, netns:line1 corresponds to LTE1 on the MLU connected to the upper left toward the USB connector, USB NIC on USB2 connected to the same connector, and USB NIC on USB3 on the Jetson Nano Developer Kit B01.

```
nwk_dev=Num,Name,Path
```

The *name* is a string used for internal processing, and *Path* is a regular expression that describes a path under /sys/. When ZaoInfraCtrl detects that a network interface is connected, it compares the nwk_dev descriptions in the configuration file from the top to the bottom and moves to the corresponding netns when a match is found. If there is no matching entry, the move to the specific netns is not performed, and the network interface remains in the default netns and is under the control of Linux NetworkManager.

The following is an example description of fixed address settings for each netns.

```
line_1_auto=0
line_1_address=192.168.1.2
line_1_mask=255.255.255.0
line_1_gw=192.168.1.1
line_1_dns=192.168.1.1
```

Specifying 0 for line_N_auto (N=1-8) specifies a fixed address setting. The default is 1 (DHCP setting).

In the case of a fixed address setup, the address, mask, etc. are described as shown in the description example.

The following is a sample configuration description for Soliton MLU.

```
mlu_wwan_1_enable=1
mlu_wwan_2_enable=1
mlu_wwan_3_enable=1
mlu_wwan_4_enable=1
```

Specify 1 for mlu_wwan_N_enable (N=1-4) to enable the MLU supported WWAN. Default is 0 (disabled).

6.3. ZaoRemoteEndpoint Process

The ZaoRemoteEndpoint process is responsible for communication with the cloud side (control communication, RASCOW2 communication) and control of transmission status. It is supplied as a pre-built binary. As shown in Figure 62, the process itself is running on default netns, but multi-link communication is performed by communicating using Socket on each netns as well.

The basic form of SDK for Jetson is that ZaoInfraCtrl places network interfaces on each netns, creates a state in which each netns can communicate with the cloud, and ZaoRemoteEndpoint uses this state to communicate with the cloud side. ZaoRemoteEndpoint determines which netns can communicate and communicates with those netns. As long as each netns is ready to communicate with the cloud, ZaoInfraCtrl intervention is not required. For example, the SDK user can configure the 801.1Q Tagged VLAN with Jetson's built-in Ether I/F and map each VLAN to netns.

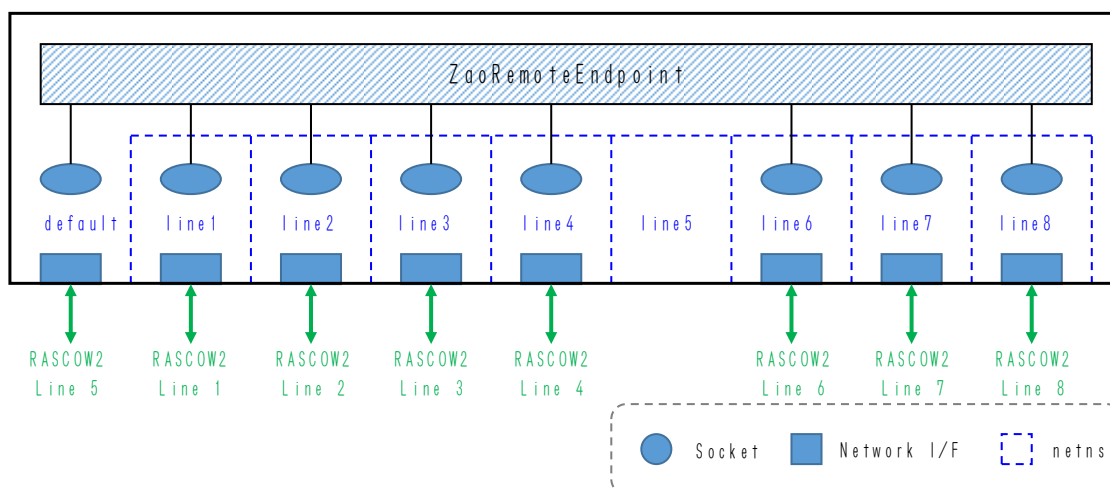


Figure 6- ZaoRemoteEndpoint and each netns

Table 6- shows the default RASCOW2 Line and netns correspondence in the SDK setup. netns: line5 is provided, but ZaoRemoteEndpoint is configured to treat default netns as RASCOW2 Line 5. This setting can be adjusted in /usr/local/etc/zao/RemoteEndpoint.conf. There can be multiple network interfaces in one netns, but communication with the cloud is done according to the routing settings in that netns, Communication with the cloud for control purposes other than RASCOW2 communication is done through whichever netns are available for communication.

Table 6- RASCOW2 Line and netns support with default settings

RASCOW2 Line No.	Supported netns Name
1	line1
2	line2
3	line3
4	line4
5	default
6	line6
7	line7
8	line8

6.4. ZaoVideoSource Process

ZaoVideoSource is the type of process that supplies the ZaoRemoteEndpoint with the video to be transmitted; up to one process can run on Jetson. A pre-built ZaoVideoSourceWebcam and ZaoVideoSource4cam are supplied, and ZaoVideoSourceWebcam is started immediately after setup.

In addition, sample source is provided and can be used for development by SDK users.

The procedure for changing the settings so that ZaoVideoSource4cam starts automatically is described

below.

6.5. ZaoAudioSource Process

ZaoAudioSource is a process type that provides the audio to be transmitted to the ZaoRemoteEndpoint, and RASCOW2 can transmit two audio systems, each of which is called Embedded and External according to the history of the past Zao products. Up to two ZaoAudioSource processes can run for each system. However, due to the limitations of Zao Cloud, only one of them will be enabled and send data when one of them is selected in the Zao Cloud View settings. The process on the non-enabled side will be in a waiting state. If you do not want to send voice, you do not need to start the process. The process on the side that is put into standby state does not have to exist.

A pre-built ZaoAudioSourceAlsa is supplied, and immediately after setup, ZaoAudioSourceAlsa is launched in a manner corresponding to Embedded Audio.

In addition, sample source is provided and can be used for development by SDK users.

6.6. ZaoAudioSink Process

ZaoAudioSink is a type of process that receives audio data from Zao Cloud via ZaoRemoteEndpoint. A pre-built ZaoAudioSinkAlsa is supplied. It is not configured to start automatically immediately after setup.

ZaoAudioSinkAlsa is also available as sample source and can be developed by SDK users.

6.7. ZaoSerialProxy Process

ZaoSerialProxy is a type of process that performs data input/output for serial tunnels; up to one process can run on Jetson.

The setup will set up a pre-built ZaoSerialProxyTty to start automatically, which is an example of a ZaoSerialProxy that uses the serial port (/dev/ttyUSB) as the endpoint for serial tunnels.

ZaoSerialProxyTty source code is also provided as a sample, which can be developed and customized by SDK users.

6.8. ZaoControl Process

ZaoControl is a type of process responsible for controlling ZaoRemoteEndpoint operation and providing UI.

Pairing with Zao Cloud, connecting to the cloud and starting video transmission after startup, and retrying to connect after the network is disconnected are all performed by ZaoControl.

ZaoControlCloud is an example of ZaoControl, an application that runs while displaying a TUI screen.

ZaoControlCloud source code is also provided as a sample, which can be developed and customized by SDK users. For example, it can operate as a daemon without a TUI screen.

7. customize

The Zao SDK for Jetson distribution includes source files for ZaoVideoSourceWebcam and others as examples for customization by SDK users. The sources for the following processes are included in the examples project.

- ZaoVideoSourceWebcam
- ZaoVideoSource4cam
- ZaoAudioSourceAlsa
- ZaoAudioSinkAlsa
- ZaoSerialProxyTty
- ZaoControlCloud

examples can be built on Jetson with the following command operation.

```
$ sudo apt-get -y install cmake libncurses5-dev cuda-toolkit-10-2 ¥  
nvidia-l4t-jetson-multimedia-api  
(Abbreviation)  
$ tar xf zao_sdk_jetson_x.x.x/examples/zao_sdk_jetson_examples_x.x.x.tar.xz  
$ mkdir zao_sdk_jetson_examples_x.x.x/build  
$ cd zao_sdk_jetson_examples_x.x.x.x/build  
$ cmake .  
(Abbreviation)  
$ make  
(Abbreviation)
```

SDK users can customize the behavior by creating their own processes based on this project and replacing them with the binary files deployed during Zao SDK for Jetson setup.

See the libzep API reference for library API specifications and examples projects.

8. alert function

ZaoControl processes can obtain information such as RASCOW2 communication status through inter-process communication. This information can be used to perform various operations.

The following information is available in Zao SDK for Jetson 1.0.

- RASCOW2 Online/Offline status of each line

Future version upgrades are expected to enable acquisition and use of video encoding status (bit rate, etc.) and communication status (RTT, estimated bandwidth, etc.).

Zao SDK for Jetson By default, the ZaoControl process displays an alert on the TUI screen when one line is in the Online state (broken from the multilink state). By using the customization function, SDK users can build their own ZaoControl process to notify linked devices or control GPIO in specific situations.

9. logfile

For logging, log files are output to `/var/log/zao-rascow2.log` (binary), `/var/log/zao-remote-endpoint.log`, and `/var/log/zao.log`.

The flow of each log is shown in Figure 9- ZaoRemoteEndpoint directly outputs log files to `zao-rascow2.log` and `zao-remote-endpoint.log`. `zao.log` is output by `rsyslog`. Customizable processes, such as `ZaoVideoSourceWebcam`, output logs to standard output, which are passed to `rsyslog` via `systemd` and aggregated in `zao.log`.

A configuration file is placed during SDK setup so that `rsyslog` outputs logs for the LOCAL0 facility to `zao.log`.

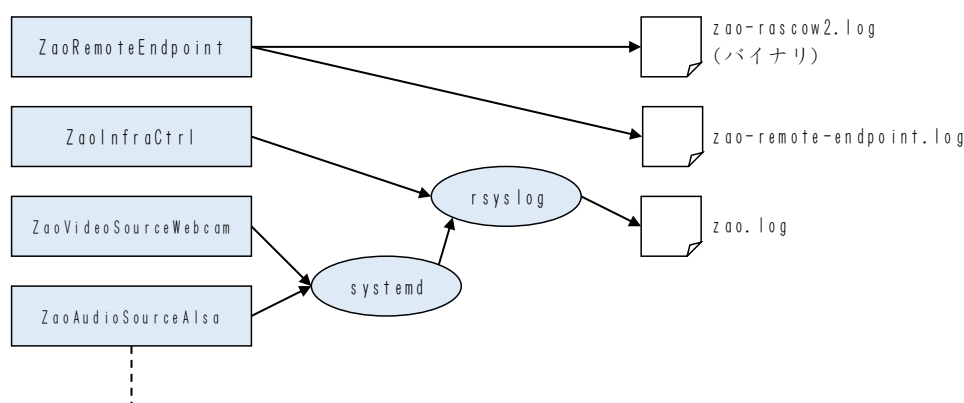


Figure 9- Log flow

`zao-rascow2.log` and `zao-remote-endpoint.log` are rotated by the `ZaoRemoteEndpoint` process. Rotation is performed at startup or when the file size exceeds a certain level.

`zao.log` is rotated by `logrotate` executed daily in `/etc/cron.daily/logrotate`.

10. Impact of Zao SDK for Jetson setup

This section describes how the SDK setup affects the Linux environment on Jetson.

10.1. Adding, enabling, and disabling Systemd services

Disable the following systemd services

- ModemManager

ModemManager on Linux is disabled by setup because it interferes with MLU control by ZaoInfraCtrl.

Add the following systemd services

- zao-infra-netns
- zao-infra
- zao-remote-endpoint
- zao-video-source
- zao-audio-source-embedded
- zao-audio-source-external
- zao-audio-sink
- zao-serial-proxy

These are placed in `/usr/local/lib/systemd/system/`.

All except `zao-audio-source-external` and `zao-audio-sink` are enabled at setup and automatically executed at OS startup.

`zao-infra-netns`, `zao-remote-endpoint`, and `zao-video-source` must be started. `zao-infra` can be disabled if the `ZaoInfraCtrl` process is not used. `zao-audio-*` and `zao-serial-proxy` services can be enabled and disabled as needed.

10.2. rsyslogd, logrotate configuration

Add configuration files to `/etc/rsyslog.d/` and `/etc/logrotate.d/` respectively.

With these settings, logs output to syslog by the LOCAL0 facility are recorded in `/var/log/zao.log` and rotated on a daily basis.

10.3. udev rules

The setup places several rule files in `/etc/udev/rules.d/`. The main purpose of these rules is to

- MLU Control
- Control of some USB modems

- Recognition of USB-Serial devices for serial tunnels

10.4. account group

Create accounts zao and zaoauto. Create group zao.

10.5. GUI disable, auto-login

The setup will disable the GUI and start in text mode. In addition, tty1 is configured to automatically log in to account zaoauto. The login shell for account zaoauto is a script that starts the ZaoControl process, so the OS will start in text mode and ZaoControl will start.

10.6. Adding the kernel module

The setup places the qmi_wwan kernel module for MLU.

10.7. ZaoVideoSource, etc.

The setup places pre-built ZaoVideoSource processes, etc. Table 10- shows the processes that are deployed.

Table 10- Processes introduced in the setup

process type	process name	Supported services	Auto-activation
ZaoVideoSource	ZaoVideoSourceWebcam	zao-video-source	** used to indicate a non-negative integer
ZaoVideoSource	ZaoVideoSource4cam	zao-video-source	×*
ZaoAudioSource	ZaoAudioSourceAlsa	zao-audio-source-embedded	period
		zao-audio-source-external	×x-mark (used to indicate an incorrect answer in a test, etc.)
ZaoAudioSink	ZaoAudioSinkAlsa	zao-audio-sink	×x-mark (used to indicate an incorrect answer in a test, etc.)

ZaoSerialProxy	ZaoSerialProxyTty	zao-serial-proxy	period
ZaoControl	ZaoControlCloud	without	** used to indicate a non-negative integer

- * ZaoVideoSourceWebCam and ZaoVideoSource4cam are invoked by either from zao-video-source.service, with ZaoVideoSoruceWebcam invoked by default.
- * ZaoConrtolCloud is started with an automatic login for the account zaoauto, not as a systemd service.

11. Various Procedures

11.1. Using the Wi-Fi USB Dongle

Wi-Fi is enabled for RASCOW2 Line 5 by connecting a compatible Wi-Fi USB dongle to Jetson and configuring the Wi-Fi connection in one of the following ways

11.1.1. Configuration in GUI environment during SDK setup

The Wi-Fi settings are saved in the OS settings and will remain valid after the SDK is set up.

11.1.2. Setting in text mode

On the TUI screen, terminate the ZaoControl process by selecting System - Close Window. The Linux NetworkManager TUI will be launched, and the Wi-Fi settings will be made by clicking "Activate a connection". Select "Re-launch ZaoControl" to start ZaoControl and try to connect to Zao Cloud including Wi-Fi connection.

11.1.3. Other

Other methods, such as enabling the GUI and then configuring in the configuration dialog, or using the nmcli command, will work as RASCOW2 Line5 if the Wi-Fi connection can be made by the Linux NetworkManager.

11.2. GUI Enable

From the main TUI screen, press Alt-F2 to switch the screen to the login prompt, and then login with the account you set up during JetPack setup.

```
$ sudo systemctl set-default graphical.target
$ sudo reboot
```

11.3. GUI Disable

Execute the following command from the terminal and reboot to return to text mode.

```
$ sudo systemctl set-default multi-user.target
```

11.4. uninstall

There is no uninstall procedure in place as of Zao SDK for Jetson 1.0.x.x.

You can make Zao SDK for Jetson almost completely unaffected by the SDK by logging in with a sudo-enabled account (such as the account you created in the initial JetPack configuration) and executing the following command.

```
$ sudo systemctl disable zao-infra-netns.service
$ sudo systemctl disable zao-infra.service
$ sudo systemctl disable zao-remote-endpoint.service
$ sudo systemctl disable zao-video-source.service
$ sudo systemctl disable zao-audio-source-embedded.service
$ sudo systemctl disable zao-serial-proxy.service
$ sudo mv /etc/systemd/system/getty@tty1.service.d/override.conf ¥
/etc/systemd/system/getty@tty1.service.d/override.conf.bak
```


12. Default setting value

This section describes the default setting values for the platforms for which the setup scripts support presets.

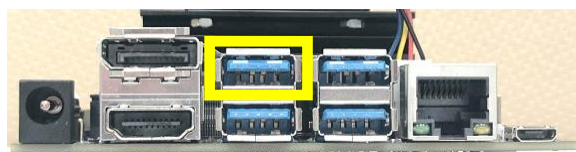
12.1. be common

The following items are commonly set regardless of platform selection.

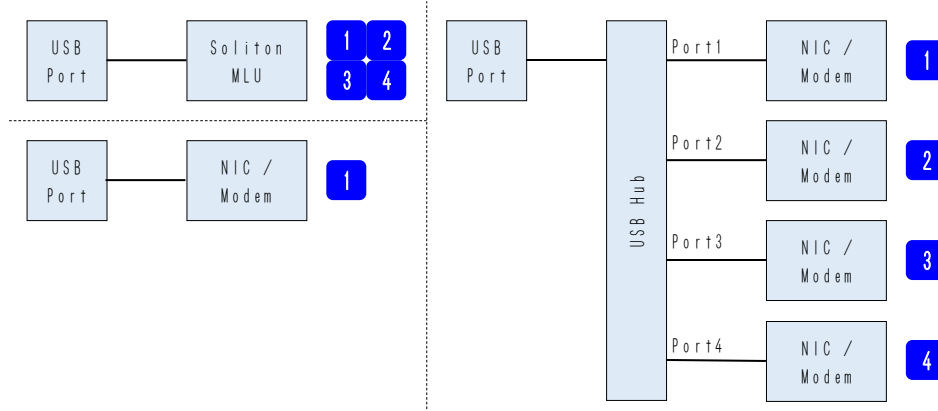
- The configuration file for using Soliton MLU will be located
- Addressing of USB NICs coming under ZaoInfraCtrl control is DHCP
- Automatic startup of ZaoVideoSourceWebcam
 - ◆ Target device `/dev/video0`
- Automatic startup of ZaoSerialProxyTtty
 - ◆ Target device `/dev/ttyZAO0`
`/dev/ttyZAO0` is a symbolic link to `ttyUSB` derived from the `ftdi_sio` driver, which was first recognized in conjunction with `udev`

12.2. Jetson Nano Developer Kit (Rev. B01)

For the Jetson Nano Developer Kit, the default settings areFigure 12- Figure 122 shows the correspondence between the RASCOW2 Line and the Soliton MLU, USB NIC, etc. when each device is connected to the USB connector.Figure 12- Figure 122 shows the correspondence with RASCOW2 Line when each device is connected to the USB connector. If Soliton MLU is connected to any other USB port, it will not work. USB NIC and Modem will be controlled by Linux NetworkManager and RASCOW2 Line5 will be supported if they are connected to any other position than shown in the figure.



**Figure 12- USB connector to connect Soliton MLU etc.
(for Jetson Nano Developer Kit)**

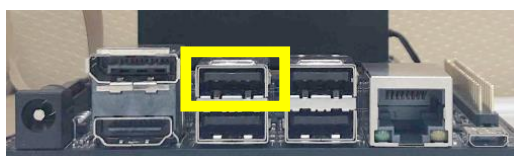


**Figure 12- Device connections and correspondence to RASCOW2 Line
(for Jetson Nano Developer Kit)**

The built-in Ethernet supports RASCOW2 Line5, and the USB Wi-Fi dongle can be connected in any position. The USB Wi-Fi dongle is not restricted to any connection position, and supports RASCOW2 Line5 no matter where it is connected (including via the USB Hub).

12.3. Jetson Xavier NX Developer Kit

For the Jetson Xavier NX Developer Kit, the default settings areFigure 12- Figure 124 shows the correspondence between the RASCOW2 Line and the Soliton MLU, USB NIC, etc. when each device is connected to the USB connector.Figure 12- Figure 124 shows the correspondence with RASCOW2 Line when each device is connected to the USB connector. If Soliton MLU is connected to any other USB port, it will not work. USB NIC and Modem will be controlled by Linux NetworkManager and RASCOW2 Line5 will be supported if they are connected to other locations than shown in the figure.



**Figure 12- USB connector to connect Soliton MLU, etc.
(for Jetson Xavier NX Developer Kit)**

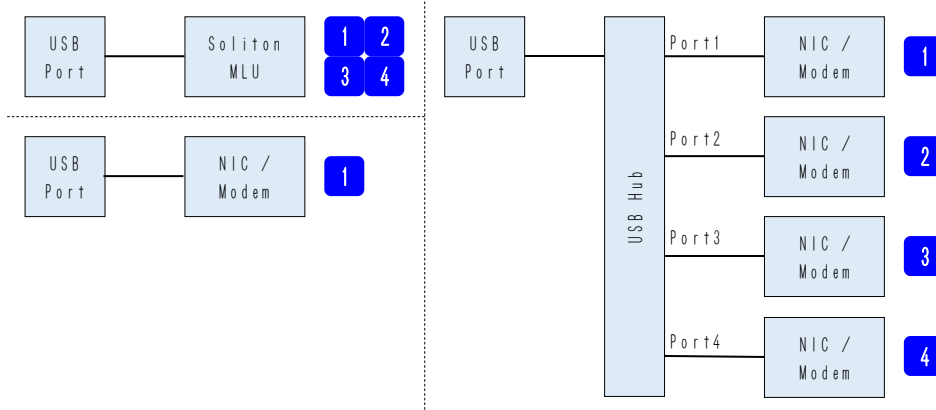


Figure 12- Device connections and correspondence to RASCOW2 Line (for Jetson Xavier NX Developer Kit)

12.4. Jetson AGX Xavier Developer Kit

For the Jetson AGX Xavier Developer Kit, the default settings are shown in Figure 12- Figure 126. Figure 126 shows the correspondence between the RASCOW2 Line and the Soliton MLU, USB NIC, etc. when each device is connected to the USB connector. Figure 12- Figure 126 shows the correspondence with the RASCOW2 Line when each device is connected to the USB connector. If Soliton MLU is connected to any other USB port, it will not work. USB NIC and Modem will be controlled by Linux NetworkManager and RASCOW2 Line 5 will be supported if they are connected to any other position than shown in the figure.

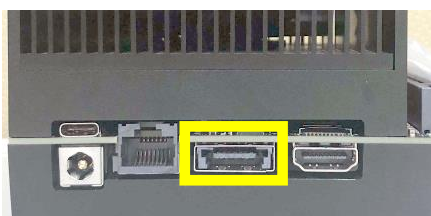


Figure 12- USB Type-A connector to connect Soliton MLU, etc. (for Jetson AGX Xavier Developer Kit)

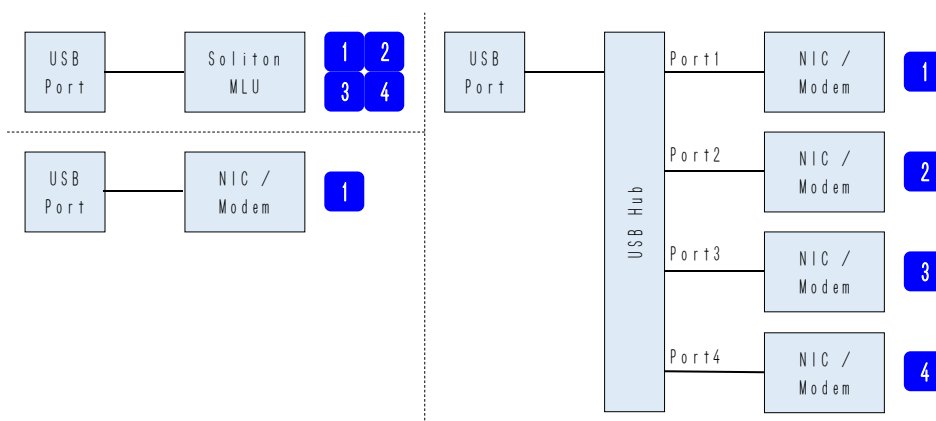
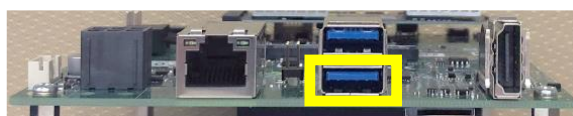


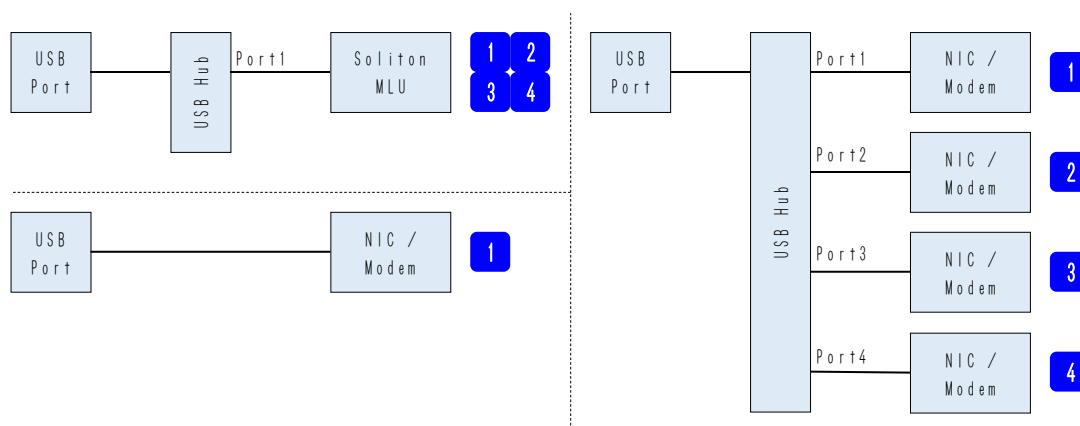
Figure 12- Device connections and correspondence to RASCOW2 Line (for Jetson AGX Xavier Developer Kit)

12.5. Aetina AX810

For the Aetina AX810, the default setting is Figure 12- Figure 128 shows the correspondence between the USB connector and the RASCOW2 Line when each device is connected to the USB connector. Figure 12- Figure 128 shows the correspondence with the RASCOW2 Line when each device is connected to the USB connector. If Soliton MLU is connected to a different position, it will not work. USB NIC and Modem are controlled by Linux NetworkManager alongside the built-in Ethernet when connected to a position other than the one shown in the figure, and will correspond to RASCOW2 Line5.



**Figure 12- USB connector to connect Soliton MLU etc.
(in the case of Aetina AX810)**



**Figure 12- Device connections and their correspondence to RASCOW2 Line
(For Aetina AX810)**

13. Update History

Rev.	Date	Update details
1.0-1	2023-05-08	first edition
1.0-2	2023-05-12	typographical error correction