

Zao SDK for Jetson User's Guide

Revision 1.0-3

株式会社ソリトンシステムズ

2023/09/01

制約・特記事項

- 『Zao SDK』は株式会社ソリトンシステムズの著作物であり、これらにかかる著作権、その他権利は株式会社ソリトンシステムズに帰属します。
- その他記載されている会社名、製品名は各社の登録商標または商標です。
- 本製品の内容は製品改良のため予告無く変更する事があります。
- 本書に掲載している接続形態はあくまで参考例であり、すべての組み合わせを保証するものではありません。

株式会社ソリトンシステムズ STC サポートセンター
〒160-0022 東京都新宿区新宿 2-4-3
TEL: (03)5360-3850

目次

1. はじめに.....	5
2. 用語.....	6
3. Zao SDK とは.....	8
3.1. Zao SDK	8
3.2. Zao SDK for Jetson の機能.....	8
3.3. Zao SDK for Jetson の構造.....	10
4. Zao SDK for Jetson のセットアップ.....	11
4.1. JetPack のセットアップ.....	11
4.2. Zao SDK for Jetson セットアップ	11
4.3. ペアリング.....	12
5. 基本操作.....	13
5.1. 起動・終了手順	13
5.2. TUI 画面	13
5.3. シェルにログイン.....	15
5.4. Web カメラ.....	15
5.5. 田の字合成動作への切替.....	15
5.6. シリアルトンネリング.....	16
5.7. ネットワークインターフェース.....	17
5.8. GUI モードの場合の起動・終了手順.....	17
6. 構造.....	18
6.1. 全体構造.....	18
6.2. ZaoInfraCtrl プロセス	19
6.2.1. InfraCtrl.conf	19
6.3. ZaoRemoteEndpoint プロセス.....	20
6.4. ZaoVideoSource プロセス.....	21
6.5. ZaoAudioSource プロセス.....	22
6.6. ZaoAudioSink プロセス.....	22
6.7. ZaoSerialProxy プロセス.....	22
6.8. ZaoControl プロセス.....	22
7. カスタマイズ.....	24
8. アラート機能.....	25
9. ログファイル.....	26
10. Zao SDK for Jetson セットアップによる影響.....	27
10.1. Systemd サービスの追加・有効化・無効化.....	27
10.2. rsyslogd, logrotate 設定.....	27
10.3. udev ルール.....	27
10.4. アカウント・グループ.....	28
10.5. GUI 無効化、自動ログイン	28
10.6. カーネルモジュールの追加	28
10.7. ZaoVideoSource 等.....	28
11. 様々な手順.....	29
11.1. Wi-Fi USB ドングルの使用.....	29

11.1.1. SDK セットアップ時に GUI 環境での設定	29
11.1.2. text モードでの設定	29
11.1.3. その他.....	29
11.2. GUI 有効化	29
11.3. GUI 無効化	29
11.4. アンインストール	30
12. デフォルト設定値.....	31
12.1. 共通	31
12.2. Jetson Nano Developer Kit (Rev. B01)	31
12.3. Jetson Xavier NX Developer Kit	32
12.4. Jetson AGX Xavier Developer Kit.....	32
12.5. Aetina AX810.....	33
13. 更新履歴.....	35

1. はじめに

本文章が対象とする Zao SDK for Jetson は Version 1.0.x.x です。1.0.x.x を総じて Ver.1.0 と呼びます。

コマンド実行例において青文字が入力頂く部分を表しています。

関連資料を以下に示します。

- Zao SDK Quick Start Guide
- Zao SDK for Jetson / libzep API リファレンス
- zao-sdk.org ウェブサイト

2. 用語

表 2-1 に Zao SDK for Jetson に関わる用語を示します。

表 2-1 用語

用語	説明
SDK	Software Development Kit
Soliton MLU	ソリトンシステムズ製 Multi Link Unit。LTE モデム 4 個を内蔵
Zao Cloud View	Zao SDK for Jetson から送信された映像をクラウド経由で受信し表示する Windows アプリケーション
ZaoInfraCtrl	Zao SDK for Jetson において、USB-NIC, Soliton MLU 等の通信回線に関して制御を行うプロセス
ZaoRemoteEndpoint	Zao SDK for Jetson において、Zao Cloud との通信などを行う Zao SDK for Jetson の基盤プロセス
RASCOW2	ソリトンシステムズが開発したマルチリンクで映像等を伝送するためのアルゴリズム・通信プロトコル
ZaoVideoSource	Zao SDK for Jetson におけるプロセス種別の 1 つ。 映像を取り込み ZaoRemoteEndpoint へ供給します。
ZaoAudioSource	Zao SDK for Jetson におけるプロセス種別の 1 つ。 音声を取り込み ZaoRemoteEndpoint へ供給します。
ZaoAudioSink	Zao SDK for Jetson におけるプロセス種別の 1 つ。 ZaoRemoteEndpoint から送り返し音声データを受け取り、オーディオデバイスを駆動します。
ZaoSerialProxy	Zao SDK for Jetson におけるプロセス種別の 1 つ。 ZaoRemoteEndpoint とシリアルトンネルデータをやり取りし、シリアルデバイス等と仲立ちします。
ZaoControl	Zao SDK for Jetson におけるプロセス種別の 1 つ。 全体の動作指示や UI を司ります
TUI メイン画面	Zao SDK for Jetson においてデフォルトセットアップされる ZaoControl が表示する TUI (Text User Interface) 画面
netns	Linux の Network Namespace 機能、もしくは当該機能で作成された空間
default netns	netns を追加作成する前から存在する netns。 基本的にはプロセスは default netns 上で動作します
JetPack	Jetson 用の OS 環境等 (L4T や各種パッケージから成る)
Linux for Tegra (L4T)	Jetson 用の Linux
SoM	System-on-Module

用語	説明
NetworkManager	L4T 標準のネットワーク制御用デーモン・ユーティリティ
ModemManager	L4T 標準のモデム制御用デーモン・ユーティリティ

3. Zao SDK とは

3.1. Zao SDK

「Zao SDK」とは Zao SDK for Jetson、Zao Cloud を用いてビデオ、オーディオ、制御信号を安定して伝送するシステムです（図 3-1）。

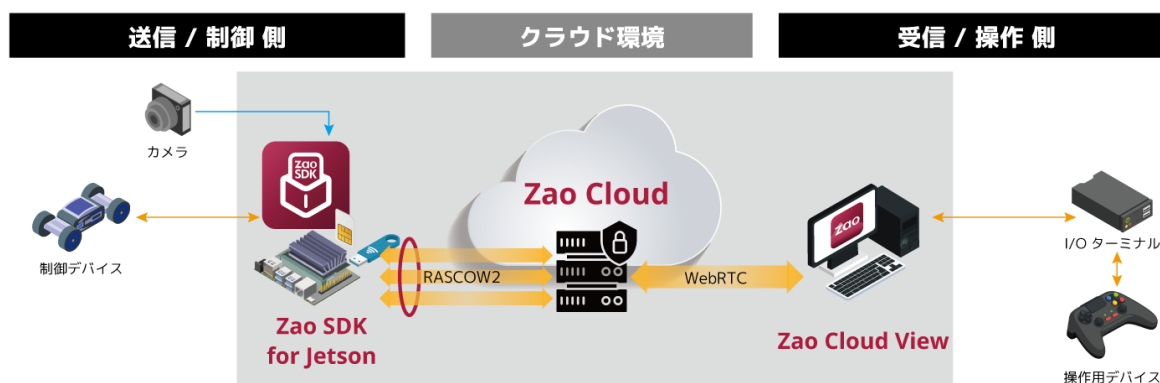


図 3-1 Zao SDK 全体像

3.2. Zao SDK for Jetson の機能

Zao SDK for Jetson は Zao SDK において Jetson を用いて映像送信・遠隔制御の端末を構築するための SDK です。

Zao SDK for Jetson の主な機能は以下のとおりです。

- Zao Cloud に接続
- RASCOW2 によるマルチリンク・低遅延の伝送
- 以下を送信可能
 - 映像 1 系統
 - 音声 2 系統（Ver.1.0 では非対応）
- Zao Cloud からの音声受信
- シリアル通信をトンネリング
- 回線状態に応じた映像ビットレート制御
- SDK ユーザによって各種カスタマイズが可能

Zao SDK for Jetson 1.0 の主な仕様を表 3-1 に示します。

表 3-1 Zao SDK for Jetson 1.0 の主な仕様

対応プラットフォーム	Jetson Nano Developer Kit B01 Jetson Xavier NX Developer Kit Jetson AGX Xavier Developer Kit 調整によって Jetson Nano SoM, Jetson Xavier NX SoM, Jetson AGX Xavier SoM を使用した環境でも使用可能
対応 JetPack	4.6.1 4.6.2 4.6.3
RASCOW2 回線数	最大 8
映像解像度・ フレームレート*2	最大 3,840x2,160 / 30fps 1,920x1,080 / 60fps 対応
対応モニター*3	HDMI モニタ (1,920x1,080 推奨)
対応 NIC*1	JetPack 環境で ethX (X は数字) と認識されるもの
対応カメラ*1	UVC 対応のウェブカメラ MIPI CSI-2, GMSL2 カメラ IP カメラ
対応オーディオ*1	Alsa で認識する USB Audio (ウェブカメラ内蔵マイク含む)
対応シリアルアダプタ*1	ftdi_sio ドライバにより ttyUSB として認識されるもの

*1 動作確認済みデバイスのメーカー・型番については、zao-sdk.org の FAQ を参照してください。SDK ユーザのカスタマイズにより、示したものの以外にも対応することが可能です。

*2 Jetson の HW 性能、Power Mode 設定、エンコード前の画像処理などによって変化します

*3 Jetson キャリアボードによります。

3.3. Zao SDK for Jetson の構造

Zao SDK for Jetson の構造について概説します。詳細は後述します。Zao SDK for Jetson は主に以下から構成されます（図 3-2）。

- ZaoRemoteEndpoint 等の実行バイナリ
- カスタマイズ可能部分
 - ◆ ビルド済み実行バイナリ
 - ◆ ソースファイル
- スクリプトファイル・設定ファイル等

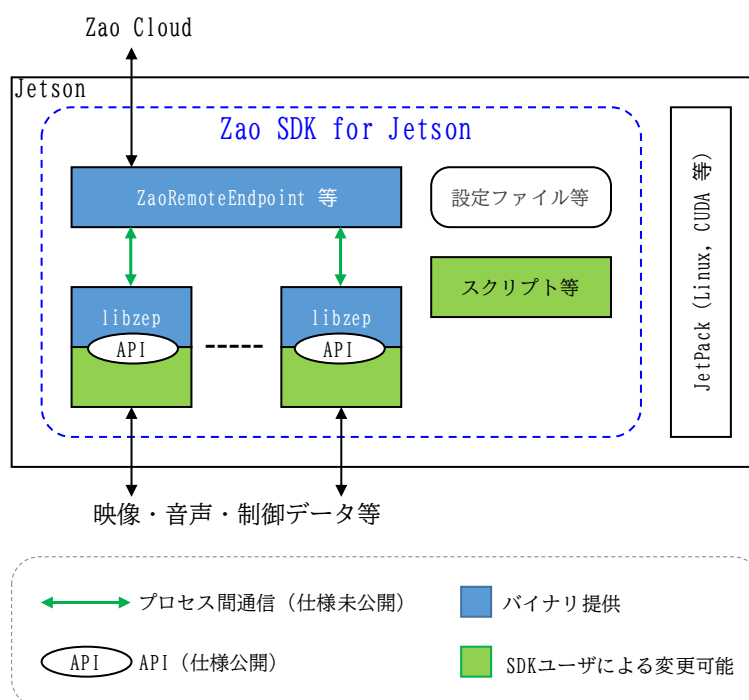


図 3-2 Zao SDK for Jetson の構造

ZaoRemoteEndpoint 等の実行バイナリは Zao Cloud との通信など動作の基盤となるプロセスです。これらは実行バイナリ形式で供給され、SDK ユーザによる変更はできませんが、設定ファイルで程度の動作調整が可能です。

それらプロセスとは別に映像・音声・制御データ等を扱うプロセスが複数存在し、これらは SDK ユーザによるカスタマイズが可能です。これらは libzep ライブラリをリンクし、SDK ユーザ開発側から libzep 公開 API にアクセスすることで Zao SDK for Jetson の機能を利用します。

Zao SDK for Jetson のデフォルト形態はカスタマイズ可能な範囲の一つの形として、ビルド済みバイナリ、example ソースコードで供給されます。SDK ユーザから例えば以下のようなカスタマイズが可能です。

- 一部の example ソースを変更しビルドし直してそのバイナリだけを置き換える
- SDK ユーザが開発したアプリケーションに libzep をリンクし Zao SDK for Jetson と接続する

加えて、Zao SDK for Jetson には Linux の udev, systemd, rsyslog 等と協調動作するための設定ファイルやスクリプトも含まれます。

4. Zao SDK for Jetson のセットアップ

本章では各 Jetson に Zao SDK for Jetson をセットアップする手順を説明します。

4.1. JetPack のセットアップ

各 Jetson プラットフォームの手順に従って JetPack 4.6.1, 4.6.2, 4.6.3 のいずれかをセットアップしてください。Nano Developer Kit の場合は Zao SDK Quick Start Guide も参考となります。

JetPack 初期設定の手順において選択する各項目についての制限等は以下となります。

- 言語、タイムゾーンに制約はありません
他アプリケーションを含めたトラブル防止には言語選択で English 選択を推奨します
- 初期アカウント名に zao, zaoauto を用いないでください
- 自動ログインは On を推奨します

4.2. Zao SDK for Jetson セットアップ

JetPack セットアップが完了したら、ターゲットにモニタ・キーボード・マウスを接続しログインしてください。

セットアップにはインターネットアクセスが必要となります。LAN 等を接続しインターネットにアクセスできる状態にしてください。この時点では Soliton MLU での接続は使用できないため、別途 LAN 等を準備下さい。

配布物（以降 zao_sdk_jetson_1.0.0.0.tar.xz として説明しますので、適宜バージョン等は読みかえてください）を USB メモリ等を用いてターゲット上のホームディレクトリにコピーしておいてください。

ターミナルを開き、以下のコマンドを順に実行してください。最初の sudo 実行にはパスワードが聞かれます。

```
$ sudo apt-get update
$ sudo apt-get -y install libqmi-utils socat dkms libcrypto++6 nano ¥
ca-certificates
$ tar xf zao_sdk_jetson_1.0.0.0.tar.xz
```

ターミナルで続いて以下のコマンドを実行してください。選択肢が表示されますので、適合するプラットフォーム番号、ZAO_CLOUD_URL のホスト名部分を入力してください。数分程度で処理が完了します。

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

Select a setup type > 2

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

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

これで SDK セットアップ手順は完了です。Jetson をシャットダウンしてください。

セットアップによって起動後の動作が text モードとなります。GUI モードに戻す手順については 11.2 を参照してください。

4.3. ペアリング

Zao Cloud に映像等を送信するにはペアリング手続きを完了させる必要があります。この手順は Zao SDK for Jetson を新規セットアップした初回のみ必要で、その後は Jetson の再起動や Zao SDK for Jetson の上書きインストールを経てもペアリング状態は維持されます。

Jetson に以下を接続し、起動してください。

- HDMI モニタ
- キーボード
- Web カメラ
- インターネット通信用回線 (Soliton MLU 可)

制御アプリ (ZaoControl) が自動起動し、図 4-1 のような画面表示となります。

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

図 4-1 TUI 画面 (Not Paired 表示)

この後の Zao Cloud とペアリングをする手順となります。具体的な手順については Quick Start Guide を参照ください。

5. 基本操作

本章では、セットアップ直後のデフォルト状態での使用について説明します。SDK ユーザによるカスタマイズが施されると動作・操作方法は本章説明の限りではなくなります。

詳細については後述しますがデフォルトの text モードでの画面の遷移を図 5-1 に示します。

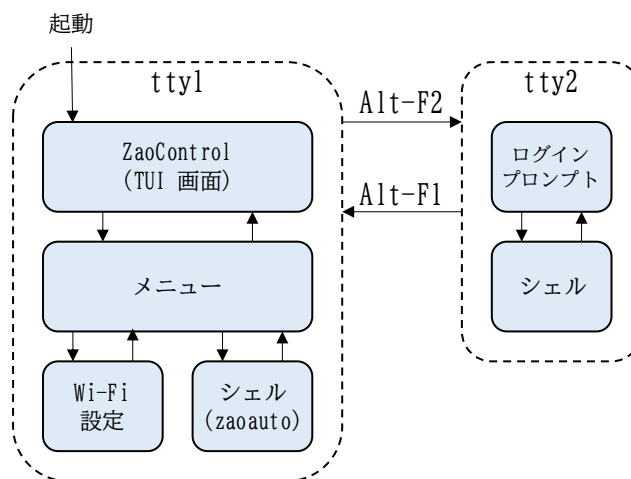


図 5-1 画面の遷移

5.1. 起動・終了手順

Zao SDK for Jetson はセットアップ後のデフォルトでは text モードでの動作となります。Jetson 起動後、自動的に TUI メイン画面が表示されます。

Jetson をリブート・PowerOff する場合は TUI 画面の System メニューから操作して下さい。PowerOff については OS のシャットダウンシーケンスが完了したのをボード上の LED 消灯などで確認してから AC アダプタを抜くなどしてください。手順を踏まずに電源断した場合、ファイルシステムが壊れる可能性があります。Jetson キャリアボード、ケースによっては電源ボタンが存在し、ボタン操作で自動的にシャットダウンシーケンスを開始するものもあります。

ssh リモートログイン可能な環境であれば、リモートログインしたシェルから `sudo poweroff` とコマンド実行しても PowerOff することが可能です。

5.2. TUI 画面

TUI 画面・操作について説明します。

キーボードの s キー、c キー押下でメニューが表示され各種操作が可能です (図 5-2)。メニュー選択時に画面下部の Tips エリアに説明が表示されます (図 5-3)。

```

System(S)          Connect(C)
  
```

図 5-2 TUI 画面 (Menu)

```
Tips
Pairing to cloud.
```

図 5-3 TUI 画面 (Tips)

Control エリアには Zao Cloud との接続状態が表示されます (図 5-4)。RASCOW2 エリアには RASCOW2 接続状態が表示されます (図 5-4)。

```
Control [ Connected ]
Room: DEFAULT

RASCOW2 Connected
OnAir
```

図 5-4 TUI 画面 (Control, RASCOW2)

映像送信中 (OnAir 状態) にのみ RASCOW2 接続が行われるため、待機中 (OffAir 状態) では、RASCOW2 は切断で Zao Control とは接続した状態となります。主な表示状態を表 5-1 に示します。

表 5-1 Control エリア・RASCOW2 エリアの主な表示

Control 表示	RASCOW2 表示	説明
Not Paired	Disconnected	未ペアリング状態
Stopped	Disconnected	Zao Cloud と切断状態
Network failed	Disconnected	Zao Cloud との接続エラー
Connecting	Disconnected	Zao Cloud と接続試行中
Connected	Disconnected	Zao Cloud とは接続済み OffAir 状態
	Connecting	Zao Cloud とは接続済み 映像送信開始中
	Connected	Zao Cloud とは接続済み OnAir 状態

RASCOW2 Lines エリアには RASCOW2 接続状態で各回線 (Line1~8) が Online か Offline が表示されます (図 5-5)。

```
RASCOW2 Lines
Line1: --- Line2: --- Line3: --- Line4: ---
Line5: On Line6: --- Line7: --- Line8: ---
```

図 5-5 TUI 画面 (RASCOW2 Lines)

Alert エリアにはアラート機能のデモとして、RASCOW2 Lines 状態がマルチリンクでなくなっていたら（つまり 1 回線だけが Online）アラート表示が出ます（図 5-6）。

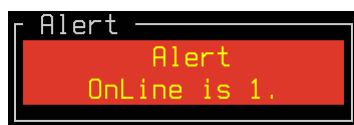


図 5-6 TUI 画面 (Alert)

Event Message エリアには、動作イベントについて直近 4 つのメッセージが表示されます（図 5-7）。



図 5-7 TUI 画面 (Event Messages)

メニューから Close を選択すると ZaoControl プロセスが終了しメニュー表示となります。メニューからは再び ZaoControl プロセスを起動する、Wi-Fi 設定をする、PowerOff する等が選択できます。Wi-Fi 設定については 11.1 を参照してください。

5.3. シェルにログイン

Jetson の Linux にログインして操作するには TUI 画面が表示されている状態で Alt-F2 を押下します。元々の tty1 表示から tty2 表示に切り替わりログインプロンプトが表示されますので、JetPack 初期設定で作成したアカウントでログインすることが可能です。Alt-F1 を押下すれば再び TUI 画面に戻ります。

5.4. Web カメラ

セットアップ後のデフォルトでは映像取り込みプロセスのサンプルの 1 つである ZaoVideoSourceWebcam プロセスが起動し、Web カメラ 1 個からの映像送信動作となります。

ZaoVideoSourceWebcam の実装からくる制約として、Web カメラを接続してから Jetson を起動するようにしてください。Jetson 起動後に Web カメラを接続しても認識しません。動作中の USB 挿抜にも対応していません。Web カメラを接続する USB コネクタ位置に制約はありません。これら制約は SDK ユーザ開発のプロセスを用いた場合には必ずしも当てはまりません。

ZaoVideoSourceWebcam デフォルトでは /dev/video0 にアクセスします。対象デバイスを変更する場合は設定ファイル /usr/local/etc/zao/VideoSourceWebcam.conf を書き換えたのち Jetson を再起動してください。

5.5. 田の字合成動作への切替

映像取り込みプロセスのサンプルとしては、ZaoVideoSourceWebcam の他に最大 4 個のカメラからの映像を田の字に面付し送信するサンプル ZaoVideoSource4cam もセットアップにより導入されます。

ZaoVideoSource4cam を動作させるには、ZaoVideoSourceWebcam を自動起動する部分を書き換え

て ZaoVideoSource4cam を自動起動するようにします。以下の手順となります。

- TUI 画面から Alt-F2 押下で tty2 に切り替えて JetPack 初期設定したアカウントでログイン
- 以下コマンドで必要なパッケージをインストール

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

- 以下コマンドでエディタを起動

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

- コメントアウト箇所を調整して以下のように編集

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

- ファイル保存してエディタ終了
- 以下コマンドでエディタを起動し設定ファイルを編集。設定ファイルの記述ルールについては libzep API リファレンスを参照してください

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

- Alt-F1 押下で TUI 画面に戻り、System メニューから再起動

5.6. シリアルトンネリング

シリアルトンネリングは Jetson と受信 PC の間で仮想的にシリアル接続する機能です (図 5-8)。

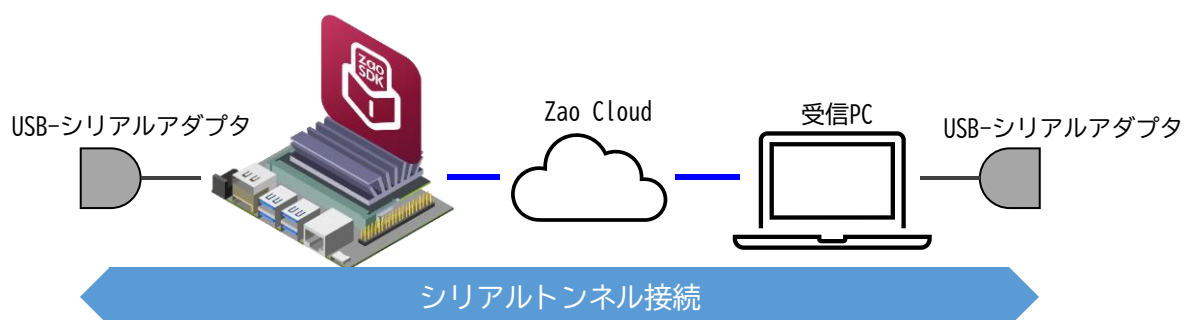


図 5-8 シリアルトンネル接続

セットアップ後のデフォルトではシリアルトンネル処理プロセスのサンプル ZaoSerialProxyTty プロセスが起動されており、USB-シリアルアダプタを端点とするシリアルトンネル動作が可能です。

USB-シリアルアダプタは Jetson 起動後に接続しても認識します。複数の USB-シリアルアダプタが存在した場合は、最初に認識されたものが対象デバイスとなります。ボーレート設定は Zao Cloud View 側の操作で行い

ます。

対応の USB-シリアルアダプタについては zao-sdk.org の FAQ を参照してください。

5.7. ネットワークインターフェース

内蔵 Ethernet は RASCOW2 Line5 として動作します。デフォルトでは DHCP 設定で動きます。内蔵 Ethernet は Linux の NetworkManager により制御され、シェルログインし nmcli コマンドや nmtui コマンドで設定することが可能です。

USB Wi-Fi ドングルを最大 1 個使用可能です。内蔵 Ethernet と同時に使用しても Linux 上の default netns のルーティング優先順位が高い方でだけ通信されるため RASCOW2 回線としては 1 本として動作します。Wi-Fi 設定方法については 11.1 を参照してください。

Soliton MLU を使用する場合は決まった USB コネクタに MLU を接続する必要があります。USB NIC, USB Modem についても、USB デバイスツリー上のどの位置に接続するかによって RASCOW2 回線との対応関係が変化します。対応関係は設定ファイル `/usr/local/etc/zao/InfraCtrl.conf` の記述内容によって決まります。デフォルト設定値、設定ファイルの詳細については後述の説明を参照してください。

5.8. GUI モードの場合の起動・終了手順

後述する手順で、GUI 有効化した場合についての起動・終了手順について説明します。

GUI モードの場合、ログイン後に Zao Cloud への通信が開始します。自動ログイン Off の場合は通信しない状態でログイン画面でのログイン待ちとなります。ログイン後、ターミナルで ZaoControl プロセスが自動実行され、Zao Cloud への通信が開始されます。

リブート・PowerOff については GUI 画面右上のメニューから操作してください。text モード同様に、手順を踏まずに電源断した場合はファイルシステムが壊れる可能性があります。

6. 構造

6.1. 全体構造

Zao SDK for Jetson の全体構成を図 6-1 に示します。いくつかのプロセスがプロセス間通信を行いながら協調して動作します。図中央の各プロセスは共通のプロセス間通信ライブラリ libzep を用いて基盤プロセスの ZaoRemoteEndpoint と通信します。libzep API は SDK ユーザから利用可能であり、SDK ユーザが開発したアプリで libzep をリンクし、API にアクセスすることで、独自のアプリからも Zao SDK for Jetson の機能を利用することが可能です。各プロセスについて後で説明しますが、プロセス種別と SDK ユーザによるカスタマイズ可能かを表 6-1 に示します。デフォルト形態としては、それぞれにプロセスを分けた実装としていますが、例えば 1 つのプロセスで映像・音声両方を扱うことも可能です。

ZaoInfraCtrl, ZaoRemoteEndpoint についても設定ファイルを編集することにより一定範囲で動作をカスタマイズすることが可能です。

Zao SDK for Jetson はこれらプロセス以外にも systemd, udev, rsyslogd などの Linux 上の仕組みも利用し動作します。

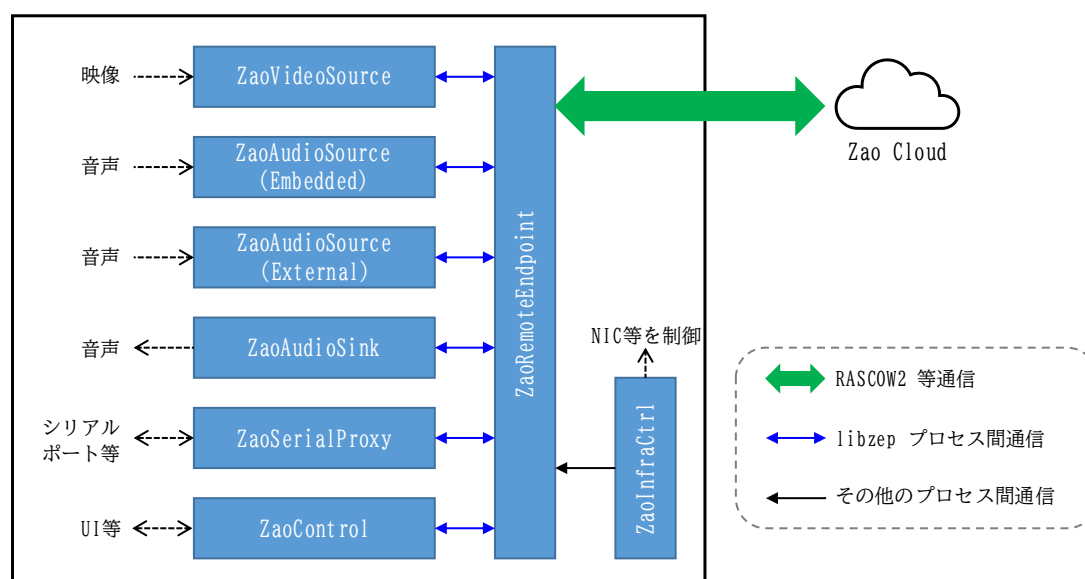


図 6-1 Zao SDK for Jetson 全体構成

Zao SDK for Jetson では音声 2 系統を送出可能ですが、それぞれを Zao 製品群の経緯から Embedded, External と呼んでいます。Zao SDK for Jetson においては Embedded と External は呼び方のみの差であり、それ以外の扱いの差はありません。Zao Cloud View 側の設定により、Zao Cloud への送信はどちらか 1 系統、もしくはどちらも送信しない動作となります。詳細については後の ZaoAudioSource の説明を参照してください。

表 6-1 Zao SDK for Jetson を構成するプロセス

プロセス種別	SDK ユーザによるカスタマイズ
ZaoInfraCtrl	不可
ZaoRemoteEndpoint	不可
ZaoVideoSource	可
ZaoAudioSource	可
ZaoAudioSink	可
ZaoSerialProxy	可
ZaoControl	可

6.2. ZaoInfraCtrl プロセス

ZaoInfraCtrl プロセスはマルチリンク動作のために、対象ネットワークインターフェースを制御するプロセスです。ビルド済みバイナリで供給されます。Zao SDK for Jetson ではマルチリンク動作に複数の netns を用いますが、ZaoInfraCtrl は設定ファイルに応じて、接続されたネットワークインターフェースを対応する netns へ移動させたり、DHCP/固定アドレス設定に基づいた制御を行います。ZaoInfraCtrl は USB-NIC や USB Modem が USB ツリー上のどの位置に接続されたかを判別し、対応する netns を決定します。

加えて、Soliton MLU に特有の制御を行います。

L4T の初期状態で有効化されている Linux の ModemManager とは競合するため、SDK セットアップ処理において ModemManager は無効化されます。

6.2.1. InfraCtrl.conf

ZaoInfraCtrl は設定ファイル (/usr/local/etc/zao/InfraCtrl.conf) に基づいて動作します。Zao SDK for Jetson をセットアップ時にデフォルトの設定ファイルが配置されます。デフォルト設定ファイルの設定内容については後述します。

設定ファイルには以下の設定を記述します。

- ネットワークインターフェースの接続位置と netns の対応関係
- 各 netns 用 Ethernet インターフェースの DHCP/固定アドレス設定
- Soliton MLU 制御の設定

netns 対応関係の部分について設定ファイル記述例（一部）を以下に示します。

```
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:
```

この記述では、Jetson Nano Developer Kit B01 において、USB コネクタに向かって左上に接続された MLU の LTE1、同コネクタに接続された USB2 の USB NIC、USB3 の USB NIC を netns:line1 に対応させています。

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

という書式になっており、*Num* は 1~8 の数値で netns:line1~line8 に対応します。*Name* は内部処理に用いる文字列です。*Path* は /sys/ 以下のパスを正規表現書式で記します。ZaoInfraCtrl はネットワークインターフェースが接続されたことを検知すると設定ファイル中の nwk_dev 記述の上から順に比較し一致した時点で対応する netns への移動を実行します。一致する記述が無かった場合には、特定 netns への移動は行われず、default netns に残ったまま Linux の NetworkManager の制御下になります。

以下に各 netns の固定アドレス設定に関する記述例を示します。

```
line_1_auto=0
line_1_addr=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
```

line_*N*auto (*N*=1~8)に 0 を指定することで固定アドレス設定となります。デフォルトは 1 (DHCP 設定)であり、記述が無い場合も DHCP となります。

固定アドレス設定の場合は記述例の様にアドレス、マスク等を記述します。

以下に Soliton MLU に関する設定記述例を示します。

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

mlu_wwan_*N*enable (*N*=1~4) に 1 を指定することで MLU の対応 WWAN を有効化します。デフォルトは 0 (無効) です。

6.3. ZaoRemoteEndpoint プロセス

ZaoRemoteEndpoint プロセスはクラウド側との通信 (制御用通信、RASCOW2 通信)、伝送状態の制御などを行うプロセスです。ビルド済みのバイナリで供給されます。図 6-2 のように、プロセス本体は default netns で動作していますが、各 netns 上の Socket も用いて通信を行うことでマルチリンク通信を行います。

SDK for Jetson の基本形としては、ZaoInfraCtrl がネットワークインターフェースを各 netns に配置し、各 netns からクラウドへ通信できる状態を作り出し、ZaoRemoteEndpoint がそれを利用してクラウド側と通信する形となります。全ての netns が通信可能である必要はなく、ZaoRemoteEndpoint は通信できる netns を判断し、そこから通信を行います。各 netns がクラウドと通信可能な状態にさえなれば、ZaoInfraCtrl の介在は必須ではありません。例えば、SDK ユーザ側で設定を施し、801.1Q Tagged VLAN を Jetson 内蔵 Ether I/F で収容し、各 VLAN を netns に対応付けることも可能です。

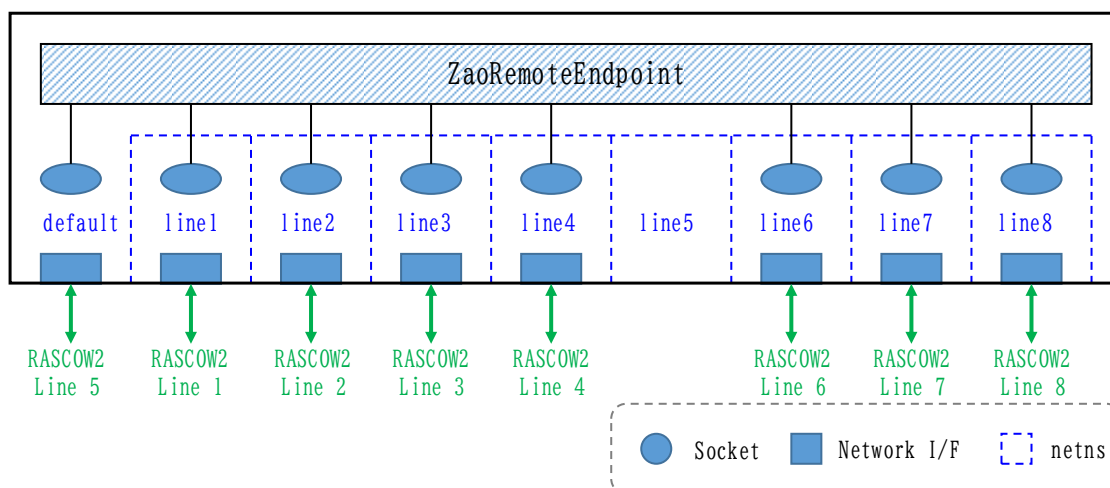


図 6-2 ZaoRemoteEndpoint と各 netns

表 6-2 に SDK セットアップでのデフォルトの RASCOW2 Line と netns の対応を示します。netns: line5 は用意してありますが、ZaoRemoteEndpoint には default netns を RASCOW2 Line 5 として扱うよう設定してあります。この設定は、/usr/local/etc/zao/RemoteEndpoint.conf で調整することが可能です。1 つの netns に複数のネットワークインターフェースが存在し得ますが、クラウドとの通信はその netns 内のルーティング設定にしたがって行われ、RASCOW2 回線としては 1 本にしかありません。RASCOW2 通信以外の制御用のクラウドとの通信は通信可能ないずれかの netns を通じて行われます。

表 6-2 デフォルト設定での RASCOW2 Line と netns 対応

RASCOW2 Line 番号	対応 netns 名
1	line1
2	line2
3	line3
4	line4
5	default
6	line6
7	line7
8	line8

6.4. ZaoVideoSource プロセス

ZaoVideoSource はプロセスの種別であり、送信する映像を ZaoRemoteEndpoint に供給するものです。Jetson 上で最大 1 個のプロセスが動作可能です。ビルド済みの ZaoVideoSourceWebcam, ZaoVideoSource4cam が供給され、セットアップ直後は ZaoVideoSourceWebcam が起動する形になっています。

加えて、サンプルソースが供給され、それを元に SDK ユーザによる開発が可能です。

ZaoVideoSource4cam を自動起動するよう設定変更する手順については後述します。

6.5. ZaoAudioSource プロセス

ZaoAudioSource はプロセスの種別であり、送信する音声を ZaoRemoteEndpoint に供給するものです。RASCOW2 では 2 系統の音声を送信可能であり、それぞれの系統を過去の Zao 製品群からの経緯により Embedded, External と呼んでいます。各系統に対応し最大 2 個の ZaoAudioSource プロセスを動作させることが可能です。ただし、Zao Cloud の制約から Zao Cloud View 上の設定で片方を選択し、そちらだけが有効化されデータ送信します。有効化されなかった側のプロセスは待機状態となります。音声を送信しない場合はプロセスを起動しなくても構いません。待機状態となる側もプロセスが存在していなくても構いません。

ビルド済みの ZaoAudioSourceAlsa が供給され、セットアップ直後は Embedded 音声に対応づいた形で ZaoAudioSourceAlsa が起動する形になっています。

加えて、サンプルソースが供給され、それを元に SDK ユーザによる開発が可能です。

6.6. ZaoAudioSink プロセス

ZaoAudioSink はプロセスの種別であり、Zao Cloud から送られてきた音声データを ZaoRemoteEndpoint 経由で受け取るものです。Jetson 上で最大 1 個のプロセスが動作可能です。ビルド済みの ZaoAudioSinkAlsa が供給されます。セットアップ直後は自動起動する設定にはなっていません。

ZaoAudioSinkAlsa はサンプルとしてソースも供給され、SDK ユーザによる開発が可能です。

6.7. ZaoSerialProxy プロセス

ZaoSerialProxy はプロセスの種別であり、シリアルトンネルのデータ入出力を行うプロセスです。Jetson 上で最大 1 個のプロセスが動作可能です。

セットアップによりビルド済みの ZaoSerialProxyTty が自動起動する形に設定されます。ZaoSerialProxyTty は ZaoSerialProxy の一例で、シリアルポート (/dev/ttyUSB) をシリアルトンネルの端点として動作するものです。

ZaoSerialProxyTty はサンプルとしてソースも供給され、それを元に SDK ユーザによる開発・カスタマイズが可能です。

6.8. ZaoControl プロセス

ZaoControl はプロセスの種別であり、ZaoRemoteEndpoint 動作の制御や UI の提供を担当するものです。ZaoControl プロセスは Zao SDK for Jetson 動作に必須です。

Zao Cloud とのペアリング操作や、起動後にクラウドと接続し映像送信を開始する動作や、ネットワークが切断された後に接続をリトライするなどの動作は ZaoControl からの指示で全体が動きます。

セットアップによりビルド済みの ZaoControlCloud が自動起動する形に設定されます。ZaoControlCloud は ZaoControl の一例で、TUI 画面を表示しながら動作するアプリです。

ZaoControlCloud はサンプルとしてソースも供給され、それを元に SDK ユーザによる開発・カスタマイズが

可能です。例えば、TUI 画面を持たずデーモンとして動作する形なども実現可能です。

7. カスタマイズ

Zao SDK for Jetson 配布物には SDK ユーザによるカスタマイズのためのサンプルとして ZaoVideoSourceWebcam 等のソースファイルが含まれています。以下のプロセスのソースがまとめて examples プロジェクトに含まれています。

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

examples は以下のコマンド操作で Jetson 上でビルドすることが可能です。

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

このプロジェクトを元に SDK ユーザが独自のプロセスを作成し、Zao SDK for Jetson セットアップ時に配置されたバイナリファイルと置き換えることで、動作をカスタマイズすることが可能です。

ライブラリの API 仕様や、examples プロジェクトについては libzep API リファレンスを参照してください。

8. アラート機能

ZaoControl プロセスはプロセス間通信を通じて、RASCOW2 通信状態など得ることができます。この情報を用いて、様々な動作をさせることが可能です。

Zao SDK for Jetson 1.0 で利用可能な情報は以下です。

- RASCOW2 各回線の Online/Offline 状態

今後のバージョンアップで映像エンコード状態（ビットレート等）、通信状態（RTT、推定帯域等）なども取得・利用可能となる見込みです。

Zao SDK for Jetson デフォルト状態では ZaoControl プロセスで Online 状態の回線が 1 本（マルチリンク状態から崩れている）場合に TUI 画面に Alert 表示を出す例を実装してあります。カスタマイズ機能を用いて SDK ユーザが独自の ZaoControl プロセスを構築することで、特定状況で連携機器に通知したり、GPIO を制御するようなことが可能です。

9. ログファイル

ログについては `/var/log/zao-rascow2.log` (バイナリ)、`/var/log/zao-remote-endpoint.log`、`/var/log/zao.log` にログファイルが出力されます。

各ログの流れを図 9-1 に示します。ZaoRemoteEndpoint は直接ログファイルを `zao-rascow2.log`、`zao-remote-endpoint.log` に出力します。`zao.log` は `rsyslog` によって出力されます。ZaoInfraCtrl は `syslog` でログ情報を出力し、`rsyslog` が受け取る形になります。ZaoVideoSourceWebcam 等のカスタイズ可能なプロセス群はログを標準出力に出し、`systemd` 経由で `rsyslog` に渡され、`zao.log` に集約されていく形となります。

`rsyslog` は LOCAL0 ファシリティのログを `zao.log` に出力するよう設定ファイルが SDK セットアップ時に配置されます。

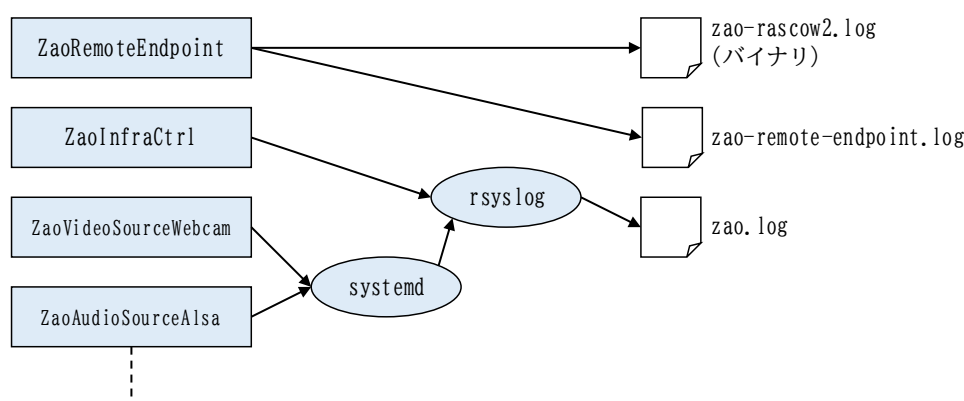


図 9-1 ログの流れ

`zao-rascow2.log`、`zao-remote-endpoint.log` は ZaoRemoteEndpoint プロセスがローテーションを行います。起動時、もしくはファイル容量が一定以上になったタイミングでローテーションが行われます。

`zao.log` は `/etc/cron.daily/logrotate` で日毎に実行される `logrotate` によってローテーションされます。

10.Zao SDK for Jetson セットアップによる影響

SDK セットアップによって Jetson 上の Linux 環境に与える影響について説明します。

10.1.Systemd サービスの追加・有効化・無効化

以下の systemd サービスを無効化します。

- ModemManager

Linux の ModemManager は ZaoInfraCtrl による MLU 制御と干渉するため、セットアップによって無効化されます。

以下の systemd サービスを追加します。

- 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

これらは /usr/local/lib/systemd/system/ にファイル配置されます。

zao-audio-source-external、zao-audio-sink 以外はセットアップ時に有効化され、OS 起動時に自動実行されます。

zao-infra-netns、zao-remote-endpoint、zao-video-source は起動必須です。zao-infra は ZaoInfraCtrl プロセスを用いない場合には無効化可能です。zao-audio-*、zao-serial-proxy の各サービスは必要に応じて有効化・無効化して使用することが可能です。

10.2. rsyslogd, logrotate 設定

/etc/rsyslog.d/, /etc/logrotate.d/ それぞれに設定ファイルを追加します。

これら設定により LOCAL0 ファシリティで syslog に出力されたログは /var/log/zao.log に記録され、日毎にローテートされるようになります。

10.3. udev ルール

セットアップによって /etc/udev/rules.d/ にいくつかのルールファイルが配置されます。主には以下を担うためのルールです。

- MLU の制御
- 一部の USB モデムの制御
- シリアルトンネル用 USB-シリアルデバイスの認識

10.4. アカウント・グループ

アカウント zao, zaoauto を作成します。グループ zao を作成します。

10.5. GUI 無効化、自動ログイン

セットアップによって GUI は無効化され、text モードで起動するようになります。加えて tty1 でアカウント zaoauto に自動ログインする設定を施します。アカウント zaoauto のログインシェルは ZaoControl プロセスを起動するスクリプトとなっているため、OS が text モードで起動し、ZaoControl が起動する動きとなります。

10.6. カーネルモジュールの追加

セットアップによって MLU に対応した qmi_wwan カーネルモジュールが配置されます。

10.7. ZaoVideoSource 等

セットアップによってビルド済みの ZaoVideoSource プロセス等が配置されます。表 10-1 に配置されるプロセスを示します。

表 10-1 セットアップで導入されるプロセス

プロセス種別	プロセス名	対応 service	自動起動
ZaoVideoSource	ZaoVideoSourceWebcam	zao-video-source	○*
ZaoVideoSource	ZaoVideoSource4cam	zao-video-source	×*
ZaoAudioSource	ZaoAudioSourceAlsa	zao-audio-source-embedded	○
		zao-audio-source-external	×
ZaoAudioSink	ZaoAudioSinkAlsa	zao-audio-sink	×
ZaoSerialProxy	ZaoSerialProxyTty	zao-serial-proxy	○
ZaoControl	ZaoControlCloud	無し	○*

- * ZaoVideoSourceWebCam, ZaoVideoSource4cam は zao-video-source.service からどちらかを起動する形であり、デフォルトでは ZaoVideoSourceWebcam が起動するようになっています。
- * ZaoControlCloud は systemd の service としてではなく、アカウント zaoauto の自動ログインで起動されます。

11. 様々な手順

11.1. Wi-Fi USB ドングルの使用

Jetson に対応 Wi-Fi USB ドングルを接続し、以下のいずれかの方法で Wi-Fi 接続に関して設定することで、RASCOW2 Line 5 として Wi-Fi を使用可能になります。

11.1.1. SDK セットアップ時に GUI 環境での設定

OS 設定から Wi-Fi 設定を施すと設定保存され、SDK セットアップ後も有効となります。

11.1.2. text モードでの設定

TUI 画面で System - Close Window で ZaoControl プロセスを終了させます。その後、メニューが表示されますので、5. Wi-Fi Setting を選択します。Linux の NetworkManager の TUI が起動されるので、Activate a connection から Wi-Fi 設定を行います。設定後、Quit すると、再びメニューが表示されます。1. Re-launch ZaoControl を選択すると ZaoControl が起動し、Wi-Fi 接続も含めて Zao Cloud へ接続しようとしています。

11.1.3. その他

他にも GUI を有効にしてから設定ダイアログで設定する方法、nmcli コマンドで設定する方法など、Linux の NetworkManager によって Wi-Fi 接続される状況にできれば RASCOW2 Line5 として動きます。

11.2. GUI 有効化

TUI メイン画面状態から Alt-F2 押下すると、画面が切り替わりログインプロンプトが現れます。JetPack セットアップ時に設定したアカウントでログインし、以下コマンドを実行すると再起動し GUI が立ち上がります。

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

11.3. GUI 無効化

ターミナルから以下コマンドを実行し、再起動すると text モードに戻ります。

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

11.4. アンインストール

Zao SDK for Jetson 1.0.x.x の段階でアンインストール手順は整備されていません。

sudo 可能なアカウント（JetPack 初期設定で作成したアカウント等）でログインし以下のコマンドを実行することで Zao SDK for Jetson の影響をほぼ無い状態にできます。

```
$ 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. デフォルト設定値

セットアップスクリプトがプリセット対応しているプラットフォームについてのデフォルト設定値について説明します。

12.1. 共通

プラットフォーム選択によらず共通に設定される項目は以下です。

- Soliton MLU 使用のための設定ファイルが配置されます
- ZaoInfraCtrl 制御下に入る USB NIC のアドレス設定は DHCP です
- ZaoVideoSourceWebcam の自動起動
 - ◆ 対象デバイス /dev/video0
- ZaoSerialProxyTtty の自動起動
 - ◆ 対象デバイス /dev/ttyZAO0
/dev/ttyZAO0 は udev との連携で最初に認識した ftdi_sio ドライバ由来の ttyUSB へのシンボリックリンクとなります

12.2. Jetson Nano Developer Kit (Rev. B01)

Jetson Nano Developer Kit の場合、デフォルト設定では図 12-1 の USB コネクタに Soliton MLU, USB NIC 等を接続する形となります。USB コネクタに各デバイスを接続した場合の RASCOW2 Line との対応を図 12-2 に示します。この USB Port 以外に Soliton MLU を接続しても動作しません。USB NIC, Modem については図示以外の位置に接続すると内蔵 Ether と横並びに Linux の NetworkManager による制御下になり RASCOW2 Line5 に対応する形となります。

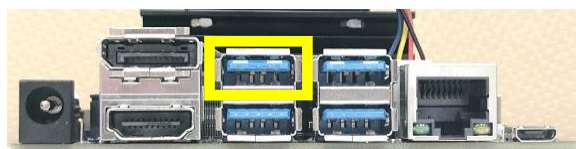


図 12-1 Soliton MLU 等を接続する USB コネクタ
(Jetson Nano Developer Kit の場合)

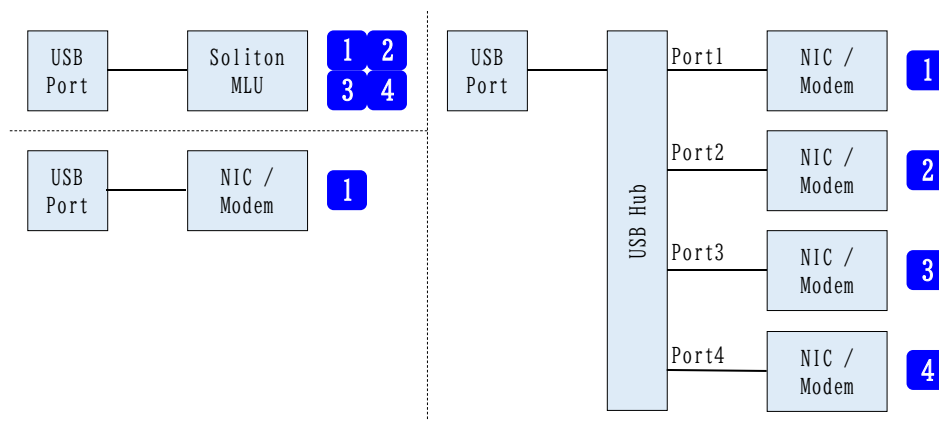


図 12-2 デバイスの接続と RASCOW2 Line との対応関係
(Jetson Nano Developer Kit の場合)

内蔵 Ether は RASCOW2 Line5 に対応します。USB Wi-Fi ドングルは接続位置に制約はありません。どの位置（USB Hub 経由も含む）に接続しても RASCOW2 Line5 に対応します。

12.3. Jetson Xavier NX Developer Kit

Jetson Xavier NX Developer Kit の場合、デフォルト設定では図 12-3 の USB コネクタに Soliton MLU, USB NIC 等を接続する形となります。USB コネクタに各デバイスを接続した場合の RASCOW2 Line との対応を図 12-4 に示します。この USB Port 以外に Soliton MLU を接続しても動作しません。USB NIC, Modem については図示以外の位置に接続すると内蔵 Ether と横並びに Linux の NetworkManager による制御下になり RASCOW2 Line5 に対応する形となります。

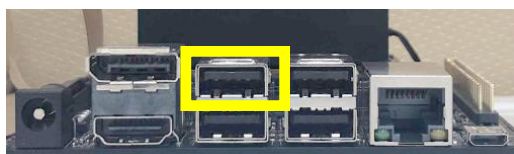


図 12-3 Soliton MLU 等を接続する USB コネクタ
(Jetson Xavier NX Developer Kit の場合)

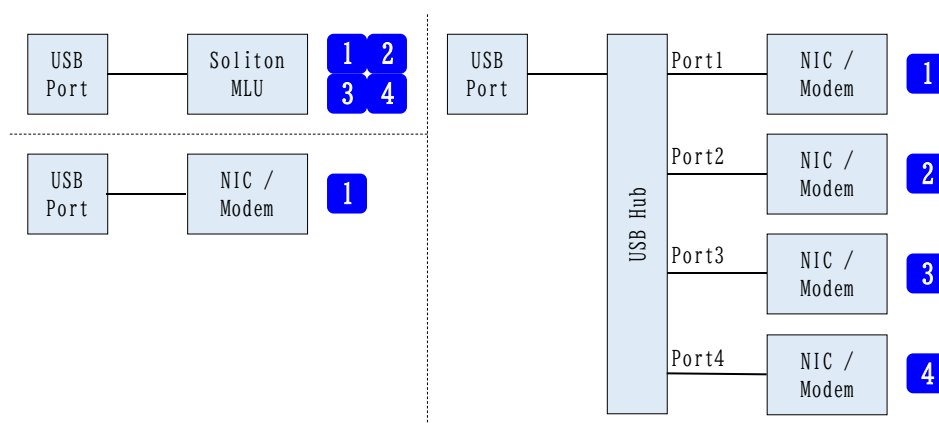


図 12-4 デバイスの接続と RASCOW2 Line との対応関係
(Jetson Xavier NX Developer Kit の場合)

12.4. Jetson AGX Xavier Developer Kit

Jetson AGX Xavier Developer Kit の場合、デフォルト設定では図 12-5 の USB Type-A コネクタに Soliton MLU, USB NIC 等を接続する形となります。USB コネクタに各デバイスを接続した場合の RASCOW2 Line との対応を図 12-6 に示します。この USB Port 以外に Soliton MLU を接続しても動作しません。USB NIC, Modem については図示以外の位置に接続すると内蔵 Ether と横並びに Linux の NetworkManager による制御下になり RASCOW2 Line5 に対応する形となります。

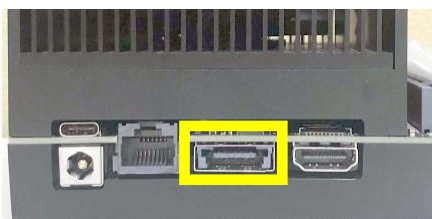


図 12-5 Soliton MLU 等を接続する USB Type-A コネクタ
(Jetson AGX Xavier Developer Kit の場合)

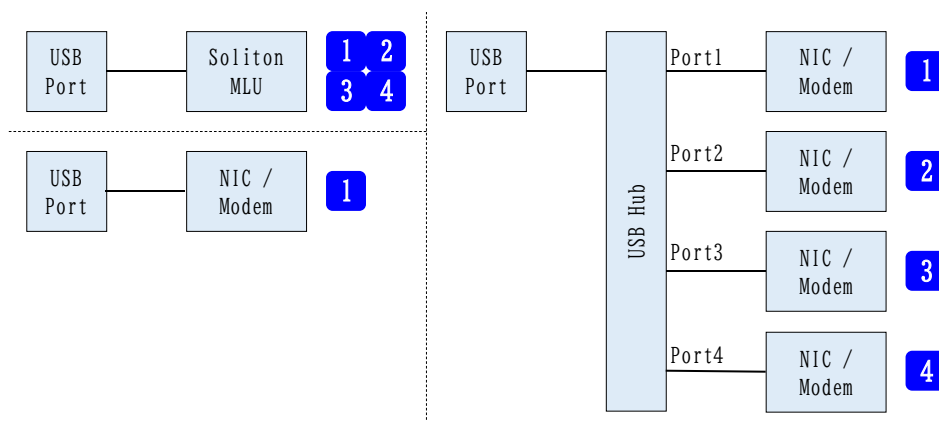


図 12-6 デバイスの接続と RASCOW2 Line との対応関係
(Jetson AGX Xavier Developer Kit の場合)

12.5. Aetina AX810

Aetina AX810 の場合、デフォルト設定では図 12-7 の USB コネクタから USB Hub を経由して Soliton MLU, USB NIC 等を接続する形となります。USB コネクタに各デバイスを接続した場合の RASCOW2 Line との対応を図 12-8 に示します。異なる位置に Soliton MLU を接続しても動作しません。USB NIC, Modem については図示以外の位置に接続すると内蔵 Ether と横並びに Linux の NetworkManager による制御下になり RASCOW2 Line5 に対応する形となります。

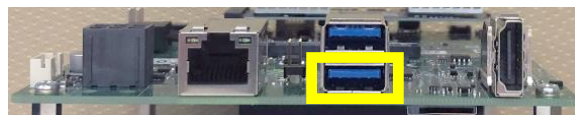


図 12-7 Soliton MLU 等を接続する USB コネクタ
(Aetina AX810 の場合)

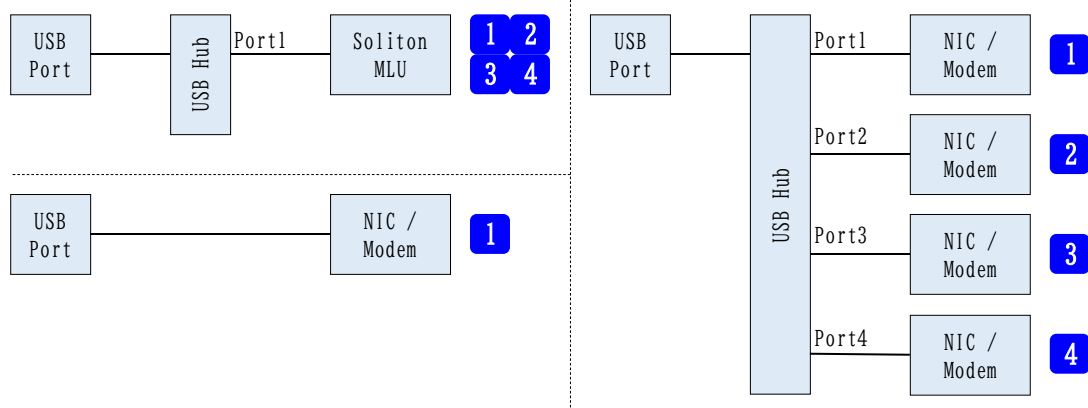


図 12-8 デバイスの接続と RASCOW2 Line との対応関係
(Aetina AX810 の場合)

13. 更新履歷

Rev.	Date	更新內容
1.0-1	2023-05-08	初版
1.0-2	2023-05-12	誤字修正
1.0-3	2023-09-01	誤字修正