

Zao SDK for Jetson User's Guide

Revision 1.2-1

株式会社ソリトンシステムズ

2023/10/26

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.....	12
4.1. JetPack のセットアップ.....	12
4.2. Zao SDK for Jetson セットアップ	12
4.3. ペアリング.....	13
5. 基本操作.....	15
5.1. 起動・終了手順	15
5.2. TUI 画面	15
5.3. TUI 画面を SSH ターミナルから操作.....	17
5.4. シェルにログイン.....	17
5.5. Web カメラ	17
5.6. 映像ソースの変更	18
5.7. シリアルトンネリング.....	18
5.8. ネットワークインターフェース.....	18
5.9. GNSS レシーバ.....	19
5.10. GUI モードの場合の起動・終了手順	19
6. 設定ユーティリティ.....	20
6.1. 設定ユーティリティの起動手順	20
6.2. Soliton-MLU の設定	20
6.3. GNSS レシーバの設定	21
6.4. text mode / GUI mode の切り替え.....	21
6.5. ZaoVideoSource プロセスの選択.....	21
6.6. ZaoVideoSourceWebcam の設定.....	21
6.7. ZaoVideoSourceArgus の設定	22
6.8. Audio Input の設定	22
6.9. Audio Output の設定	22
6.10. ZaoSerialProxyTty の設定.....	22
6.11. アンインストール	23
7. 構造.....	24
7.1. 全体構造.....	24
7.2. ZaoInfraCtrl プロセス	26
7.2.1. InfraCtrl.conf	26
7.3. ZaoRemoteEndpoint プロセス.....	27
7.4. ZaoVideoSource プロセス	28
7.5. ZaoAudioSource プロセス.....	29
7.6. ZaoAudioSink プロセス.....	29
7.7. ZaoSerialProxy プロセス.....	29

7.8. ZaoControl プロセス.....	29
7.8.1. text モードでの ZaoControlCloud 起動.....	30
7.8.2. GUI モードでの ZaoControlCloud 起動.....	30
7.9. ZaoAlert プロセス.....	30
8. カスタマイズ.....	31
9. アラート機能.....	32
9.1. Alert.toml.....	33
10. GNSS 受信機対応.....	35
11. ログファイル.....	36
11.1. ログアップロード機能.....	36
12. Zao SDK for Jetson セットアップによる影響.....	37
12.1. Systemd サービスの追加・有効化・無効化.....	37
12.2. rsyslogd, logrotate 設定.....	37
12.3. udev ルール.....	38
12.4. アカウント・グループ.....	38
12.5. GUI 無効化、自動ログイン.....	38
12.6. カーネルモジュールの追加.....	38
12.7. ZaoVideoSource 等.....	39
13. ROM 化.....	40
13.1. ROM 化と SDK 動作.....	40
13.2. Nano Developer Kit での ROM 化.....	40
13.2.1. ROM 化手順.....	41
13.2.2. ROM 化解除/再 ROM 化.....	41
13.2.3. 補足.....	42
14. デフォルト設定値.....	43
14.1. 共通.....	43
14.2. Jetson Nano Developer Kit (Rev. B01).....	43
14.3. Jetson Xavier NX Developer Kit.....	44
14.4. Jetson AGX Xavier Developer Kit.....	45
15. Zao SDK for Jetson 過去のバージョンからのアップデート.....	46
15.1. Zao SDK for Jetson 1.0 から 1.1 への変更点.....	46
15.2. Zao SDK for Jetson 1.1 から 1.2 への変更点.....	46
15.3. Zao SDK for Jetson 1.0 からのアップデート手順.....	46
15.4. Zao SDK for Jetson 1.1 からのアップデート手順.....	47
16. 様々な手順.....	48
16.1. Wi-Fi USB ドングルの使用.....	48
16.1.1. SDK セットアップ時に GUI 環境での設定.....	48
16.1.2. text モードでの設定.....	48
16.1.3. その他.....	48
17. 更新履歴.....	49

1. はじめに

本文章が対象とする Zao SDK for Jetson は Version 1.2.x.x です。1.2.x.x を総じて 1.2 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 Cookbook
- Zao SDK for Jetson / libzao-endpoint 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
Zao SDK	Zao SDK for Jetson、Zao Cloud、Zao Cloud View 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全体
Zao SDK for Jetson	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
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。
ZaoAlert	Zao SDK for Jetson におけるプロセス種別の 1 つ。アラート機能を実現するビルド済みアプリ名でもあります。
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

用語	説明
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
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
設定ユーティリティ	Zao SDK for Jetson の各種設定を行うユーティリティ コマンド名 ZaoSetting
libzep	Zao SDK for Jetson 1.0 までの API 公開しているライブラリ名
libzao-endpoint	Zao SDK for Jetson 1.1 以降での API 公開しているライブラリ名
apps	Zao SDK for Jetson にビルド済みバイナリ・ソースを同梱しているアプリケーション
examples	Zao SDK for Jetson に AS-IS サンプルとしてソースのみ同梱しているアプリケーション (1.2 時点では 0 個)
tmux	Terminal multiplexer の一種

3. Zao SDK とは

3.1. Zao SDK

「Zao SDK」とは Zao SDK for Jetson、Zao Cloud、Zao Cloud View 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）。

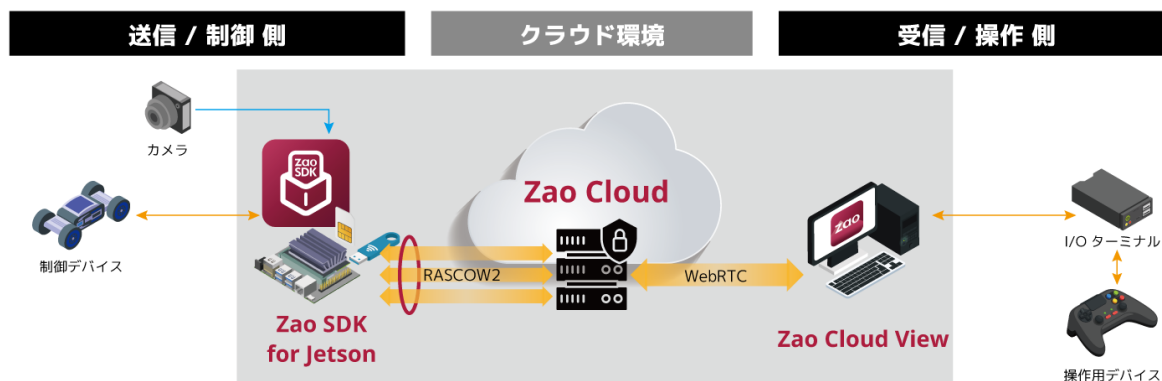


図 3-1 Zao SDK 全体像

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（Zao Cloud View で受信できるのは選択した 1 系統のみ）
- 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
- GNSS レシーバからの位置情報を Zao Cloud へ送信
- SDK ユーザによって各種カスタマイズが可能
- アラート機能により特定イベントでスクリプト等の実行

Zao SDK for Jetson 1.2 の主な仕様を表 3-1 に示します。

表 3-1 Zao SDK for Jetson 1.2 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	4.6.1~4.6.4 (4.6.4 推奨)
RASCOW2 回線数	最大 8
映像解像度・ フレームレート*2	最大 3,840×2,160 / 30fps 1,920×1,080 / 60fps 対応
対応モニター*3	HDMI モニタ (1,920×1,080 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・キャプチャアダプタ 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
対応 GNSS レシーバ	GlobalSat BU-353N5

*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
- 動作補助のスク립トファイル・設定ファイル等
- 設定ユーティリティ

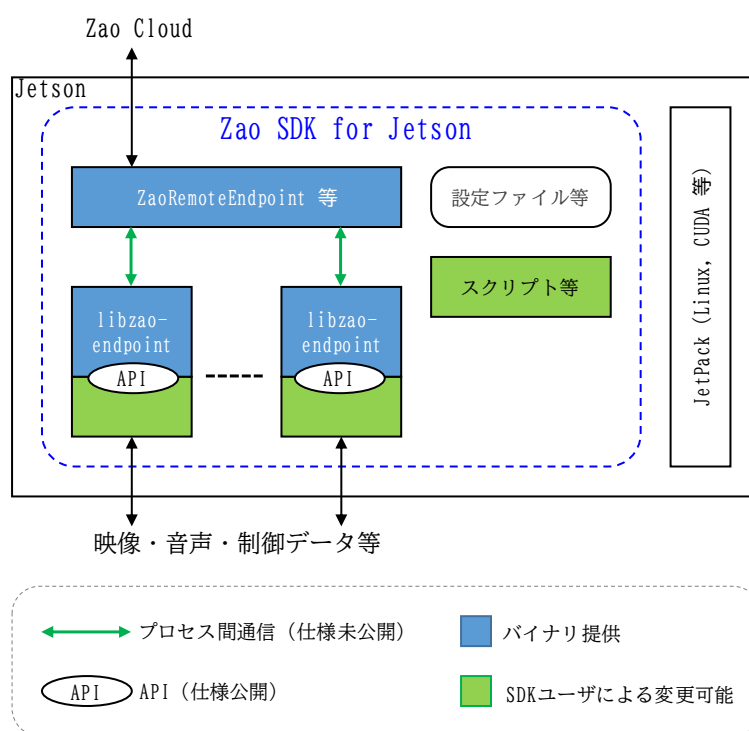


図 3-2 Zao SDK for Jetson の構造

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、設定ファイルを通じて一定範囲で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 libzao-endpoint ライブラリをリンクし、SDK ユーザ開発側から公開 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 セットアップ後のデフォルト状態ではソリトンシステム製のビルド済みのアプリケーションが自動起動する形になっています。これらアプリケーション (apps) のソースコードも同時に提供され、SDK ユーザはカスタマイズすることが可能です。例えば以下のようなカスタマイズが可能です。

- 一部の apps 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 libzao-endpoint 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、設定ユーティリティも含まれます。

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。過去のバージョンがセットアップ済みの環境からの 1.2 へのアップデートについては 15 章を参照してください。

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 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 初期設定は GUI を用いずシリアルコンソールからも可能ですが（一部のキャリアボードを除く）、本章では GUI での初期設定を想定して説明しています。本章で説明していませんが、シリアルコンソールで JetPack 初期設定を行った後、SSH リモートログインで Zao SDK for Jetson のセットアップを行うことも可能です。

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

可能ならば Zao SDK for Jetson セットアップの前に JetPack をアップデートすることを推奨します。apt を用いてアップデートすることが可能です。キャリアボードによっては、apt によるアップデートを行うと機能に支障が出る場合がありますので注意してください。

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、Jetson にモニタ・キーボード・マウスを接続し JetPack 初期設定で入力したアカウントで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。内蔵 Ether、Wi-Fi 等から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、別途有線 LAN 等を準備下さい。

配布物（以降 zao_sdk_jetson_1.2.0.0.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。

```
$ tar xf zao_sdk_jetson_1.2.0.0.tar.xz
$ sudo zao_sdk_jetson_1.2.0.0/setup_zao_sdk_jetson.sh
Setup of Zao SDK for Jetson 1.2.0.0

Wait a minute ...
```

数分程度の処理の後、ZAO_CLOUD_URL の入力を求められます（図 4-1）。初期状態でホスト名が example になっている部分以外は完成していますので、契約者情報に記されている ZAO_CLOUD_URL のホスト名に修正してから OK 選択して進めてください。すぐに処理完了し、コマンドプロンプトへ戻ってきます。

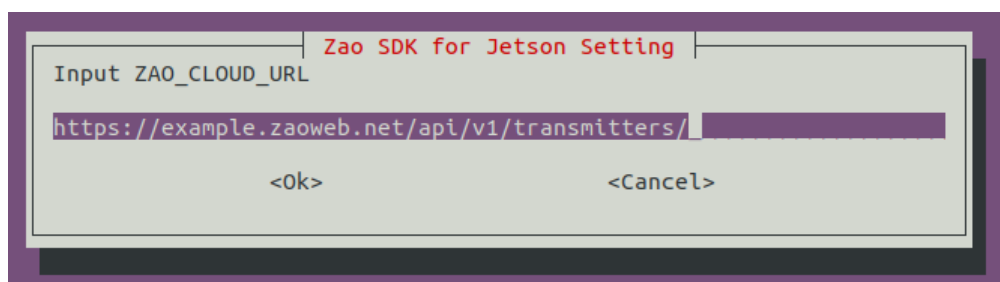


図 4-1 ZAO_CLOUD_URL 入力画面

これで Zao SDK for Jetson 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 Linux 起動が text モードとなります。GUI モードに戻す手順については 6.4 節を参照してください。

セットアップがエラーに終わった場合、同梱しているアンインストールスクリプトを実行することで中途なファイル配置、設定を除去することが可能です。エラー原因を解消した後、再びセットアップスクリプトを実行してください。以前のバージョンに戻す場合についても同梱のアンインストール、もしくは後述の設定ユーティリティからのアンインストールを行ってから再度セットアップを行ってください。

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

インターネット通信用回線に Soliton-MLU を使用する場合は先に設定ユーティリティを使って Soliton-MLU 有効化を行う必要があります (6.2 節参照)。

Jetson が起動すると、制御アプリ (ZaoControlCloud) が自動起動し、図 4-2 のような画面表示となります。

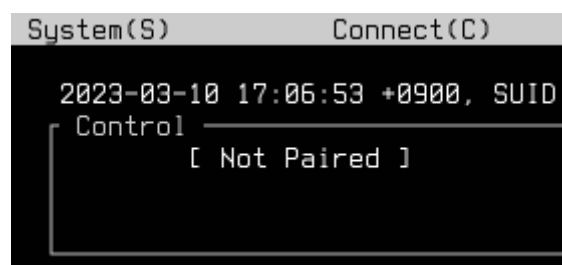


図 4-2 TUI 画面 (Not Paired 表示)

キーボード c キーを押下し Connect メニューを表示し、Pairing を選択します。Zao Cloud との通信がで

きていればペアリング番号（6桁の数値）が表示されます。Zao Cloud View 側のペアリング操作でこの数値を入力してください。ペアリング番号はペアリング手続きの度に变化するワンタイムの値で、取得してから 15 分間有効です。正常にペアリングが完了すれば Jetson の表示は Connected に变化します。ペアリング操作に関しては Quick Start Guide も参照ください。

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。

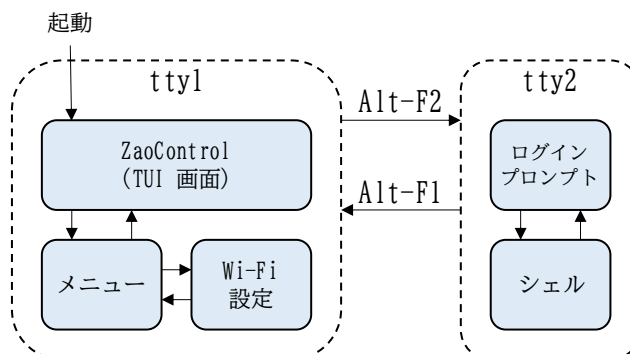


図 5-1 画面の遷移

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・シャットダウンする場合は TUI 画面の System メニューから操作して下さい。Shutdown 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シャットダウン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)。



図 5-2 TUI 画面 (Menu)

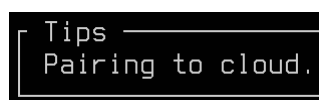


図 5-3 TUI 画面 (Tips)

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)。



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 Cloud 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)。

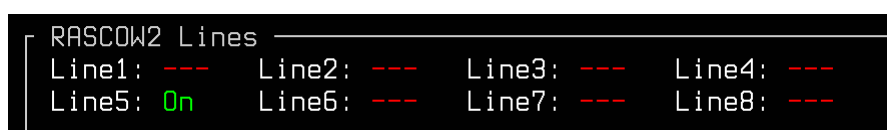


図 5-5 TUI 画面 (RASCOW2 Lines)

RASCOW2 Lines RTT エリアには RASCOW2 接続中の回線の RTT (Round Trip Time) が表示されます (図 5-6)。



図 5-6 TUI 画面 (RASCOW2 Lines RTT)

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)。

```
Event Messages
Connected to Zao Cloud.
Request to OnAir to Zao Cloud.
OnAir to Zao Cloud.
Received Room Group List.
```

図 5-7 TUI 画面 (Event Messages)

メニューから Close を選択すると ZaoControlCloud 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 ZaoControlCloud プロセスを起動する、Wi-Fi 設定をする、シャットダウンする等が選択できます。Wi-Fi 設定については 16.1 節を参照してください。

5.3. TUI 画面を SSH ターミナルから操作

TUI 画面を表示している ZaoControlCloud プロセスは tmux 上で動作しています。そのため ssh リモート接続等の別ターミナルから tmux にアタッチすることで同じ画面を共有・操作することが可能です。

当該 tmux にアタッチするには以下のコマンドを実行します。

```
$ tmux -S /tmp/tmux-ZaoControl attach
```

5.4. 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。

5.5. 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アプリケーションの 1 つである ZaoVideoSourceWebcam 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、動作中に USB 挿抜すると動作が停止します。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 はデフォルトでは以下の設定で動作します。

- デバイス /dev/video0
- 解像度 1,920 × 1,080
- フレームレート 30 fps
- ピクセルフォーマット Motion JPEG

後述の設定ユーティリティを用いて設定を変更することが可能です。

5.6. 映像ソースの変更

デフォルトの Web カメラ 1 画面伝送の他にも Zao SDK for Jetson 初期状態で MIPI カメラ 1 画面伝送、田の字面付合成の伝送について個別のアプリケーションのバイナリ・ソースが提供されています。それらは別々のバイナリになっており、Web カメラ 1 画面伝送は ZaoVideoSourceWebcam、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アプリは ZaoVideoSource4cam、MIPI カメラ 1 画面伝送は ZaoVideoSourceArgus と名付けられています。

これらアプリケーションの実行は排他的であり、どのアプリケーションを自動起動するかは後述の設定ユーティリティで設定が可能です。

ZaoVideoSource4cam を動作させるには、設定ユーティリティでの起動プロセス選択設定に加えて、設定ファイルを整える必要があります。設定ファイルの記述ルールについては libzao-endpoint API リファレンスを参照してください。

ZaoVideoSourceArgus を動作させるための設定は設定ユーティリティから調整することが可能です。

5.7. 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)。

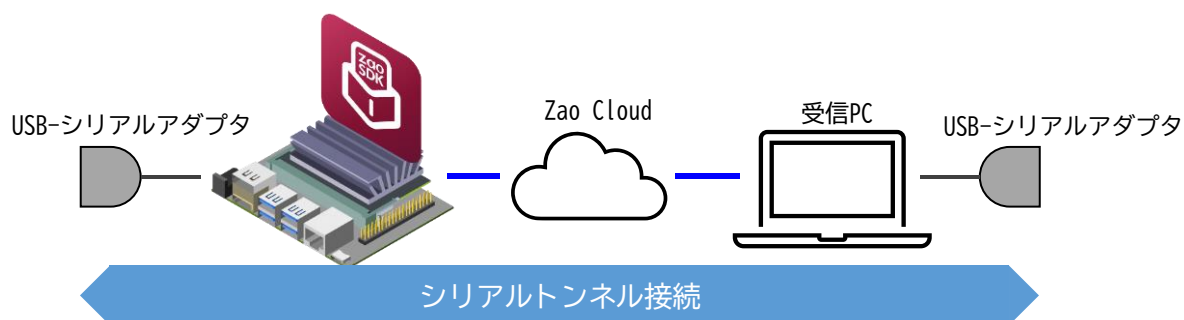


図 5-8 シリアルトンネル接続

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を担当する 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。デフォルトでは複数の 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。対象デバイスを設定ユーティリティで指定することが可能です。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。

5.8. ネットワークインターフェース

内蔵 Ethernet は RASCOW2 通信においては 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 16.1 を参照してください。

Soliton-MLU を使用する場合は前もって設定ユーティリティで導入作業が必要です。Soliton-MLU については導入作業時と同じ USB コネクタに Soliton-MLU を接続する必要があります。デフォルトでは RASCOW2 Line1 ~ 4 に対応します。

USB NIC, USB Modem は 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 `/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。

5.9. GNSS レシーバ

対応の GNSS レシーバを接続し、設定ユーティリティで導入作業を行っておくと、GNSS レシーバで得た位置情報を定期的にクラウドへ送信するようになります。

GNSS レシーバを接続する USB コネクタ位置に制限はありませんが、導入設定時の選択によって特定個体のみをレシーバとして認識するか切り替わります。Jetson 起動後に接続しても認識します。GNSS レシーバは通電してから測位が完了し位置情報を出力し始めるまでに数分~数十分かかる場合があります。

5.10. 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、自動ログインが Off でログイン画面が出ている状態でもバックグラウンドで ZaoControlCloud プロセスが動作し Zao Cloud への通信は開始します。ログイン後は、アプリケーション自動実行の仕組みで Gnome Terminal に TUI 画面が表示され操作可能です。ZaoControlCloud プロセスはバックグラウンドで動作する tmux 上で動作しており、Terminal は tmux にアタッチする形で画面を表示します。

リブート・シャットダウン 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. 設定ユーティリティ

設定ユーティリティを用いて以下の操作や設定変更を行うことが可能です。Zao SDK for Jetson アンインストール以外の操作は、設定ファイルの書き換えが行われ、再起動以降で効果が反映されます。

- Soliton-MLU の導入・無効化
- GNSS レシーバの導入・無効化
- text mode / GUI mode の切り替え
- ZaoVideoSource プロセスの選択
- ZaoVideoSourceWebcam の設定
- ZaoVideoSourceArgus の設定
- Audio Input / Output の設定
- SerialProxyTtty の設定
- Zao SDK for Jetson アンインストール

以下については設定ユーティリティからの設定には対応していません。

- USB NIC と RASCOW2 Line 対応付け
- ZaoVideoSource4cam の設定

6.1. 設定ユーティリティの起動手順

設定ユーティリティは `sudo` 実行する必要があります。`sudo` 実行可能なアカウント（基本的には JetPack 初期設定で作成したアカウント）でログインしたターミナルから以下のように実行することで設定ユーティリティが起動します。

```
$ sudo ZaoSetting
```

設定ユーティリティは TUI アプリとなっており、カーソルキー、TAB キー等を用いて階層メニューを進みながら設定変更等を行います。

ターミナルに到達するにはいくつかの方法があります。以下に一例を挙げます。

- (text モードの場合) Alt-F2 で `tty2` に移動してログイン
- (GUI モードの場合) Terminal アプリを起動
- SSH リモートログイン
- シリアルコンソール（画面が崩れる可能性があり非推奨）

6.2. Soliton-MLU の設定

設定ユーティリティからは Soliton-MLU の導入、無効化が行えます。設定項目はトップメニュー Network 以下にあります。

Soliton-MLU の導入には APN 等を書き込み済みの Soliton-MLU 本体が必要です。導入処理には Linux カーネルモジュールの追加を含みますが、一部のキャリアボードでは BSP 構造との干渉により Soliton-

MLU 導入は行えません。

Soliton-MLU は導入時と同じ USB コネクタ (Soliton-MLU に至る USB Hub の接続経路も含む) に接続しないと認識し制御開始となりません。

無効化を行うと導入時に追加されたカーネルモジュールの削除は行われませんが、設定ファイルが初期化され、Soliton-MLU に反応しなくなります。

6.3. GNSS レシーバの設定

設定ユーティリティからは GNSS レシーバの導入、無効化が行えます。設定項目はトップメニュー GNSS 以下にあります。

GNSS レシーバ導入では、対応 GNSS レシーバ BU-353N5 を認識できるように Linux カーネルモジュールの追加や関連スクリプト・設定ファイルの配置が実行されます。

通常は使用する GNSS レシーバ個体を用いて「Enable with a specific device」から GNSS レシーバ導入をしてください。その場合には、シリアル番号で判別し、その個体を接続した場合のみレシーバとして認識するようになります。

「Enable with any devices」から導入した場合は、個体によらずレシーバと認識します。USB Vendor ID, Product ID の一致する USB-シリアルアダプタやそれを内包する機器を接続した場合に誤認識する可能性があります。

無効化を行うと追加されたカーネルモジュールの削除は行われませんが、設定ファイルが初期化され、当該デバイスを接続しても GNSS レシーバとしては認識しなくなります。

6.4. text mode / GUI mode の切り替え

設定ユーティリティで Linux の起動モードを text モード / GUI モードと切り替えることが可能です。設定項目はトップメニュー Control 以下にあります。

いずれに設定した場合も ZaoControlCloud が自動起動し、Jetson の起動後、自動的にクラウドへの伝送が開始されます。

6.5. ZaoVideoSource プロセスの選択

設定ユーティリティでは ZaoVideoSource プロセスとしてどのアプリケーションを自動起動するか設定することが可能です。設定項目はトップメニュー Video 以下の Select ZaoVideoSource process になります。

Zao SDK for Jetson に含まれている ZaoVideiSourceWebcam, ZaoVideoSource4cam, ZaoVideoSourceArgus いずれのプロセスを自動起動させるかを選択可能です。

SDK ユーザが独自に開発した ZaoVideoSource プロセスを使用する場合は (Disable all) を選択し、設定ユーティリティ外でプロセスの自動起動設定を行ってください。

6.6. ZaoVideoSourceWebcam の設定

設定ユーティリティでは ZaoVideoSourceWebcam のデバイス、解像度、フレームレート、ピクセルフォーマット

トについて設定することが可能です。設定項目はトップメニュー Video 以下の Configure VideoSourceWebcam 以下に配置されています。

6.7. ZaoVideoSourceArgus の設定

設定ユーティリティでは ZaoVideoSourceArgus のデバイス、解像度、フレームレートについて設定することが可能です。設定項目はトップメニュー Video 以下の Configure VideoSourceArgus 以下に配置されます。

6.8. Audio Input の設定

設定ユーティリティでは、オーディオ入力の有効/無効、デバイス指定を設定することができます。設定項目はトップメニュー Audio 以下の Audio Input 以下に配置されます。

設定ユーティリティにおける Audio Input とは ZaoAudioSourceAlsa を Embedded 音声として動作させる部分に相当します。デフォルトでは Enable、デバイス設定は plughw:CARD=2 となっています。

ZaoAudioSourceAlsa が Enable に設定されて起動していても、Zao Cloud View 側のオーディオ設定が Embedded になっていなければ音声は伝送されません。

デバイス選択は、その際に認識されているデバイス一覧からの選択となりますので、設定時に選択するオーディオデバイスを接続しておく必要があります。デバイス一覧に表示される tegrasndt210ref は Jetson 内蔵のオーディオデバイスです。設定ユーティリティでは存在する具体的なデバイス一覧から選択する形となりますので、デフォルト状態の plughw:CARD=2 に戻すことはできません。設定ユーティリティを用いず sudo 権限で起動したテキストエディタで /etc/zao/AudioSourceAlsa.conf を編集することで plughw:CARD=2 等も設定することが可能です。

6.9. Audio Output の設定

設定ユーティリティでオーディオ出力の有効/無効、デバイス指定を設定することができます。設定項目はトップメニュー Audio 以下の Audio Output 以下に配置されます。

設定ユーティリティにおける Audio Output とは ZaoAudioSinkAlsa に関する設定となります。デフォルトは Disable になっています。

デバイス選択は、その際に認識されているデバイス一覧からの選択となりますので、設定時に選択するオーディオデバイスを接続しておく必要があります。デバイス一覧に表示される tegrahda は HDMI に重畳する音声出力で、texasndt210ref は Jetson 内蔵のオーディオデバイスです。

6.10. ZaoSerialProxyTty の設定

設定ユーティリティで ZaoSerialProxyTty の自動起動の有効/無効、デバイスの設定が行えます。

デフォルトではそれぞれ Enable, /dev/ttyZAO0 となっています。/dev/ttyZAO0 は Zao SDK for Jetson によって配置される udev 連携スクリプトが作り出すシンボリックリンクです。対象デバイスの接続を検知しそのデバイスパスに向けたシンボリックリンクが作成されます（例えば /dev/ttyUSB5 に向けたシンボリックリンク）。

6.11. アンインストール

設定ユーティリティから Zao SDK for Jetson のアンインストールが可能です。アンインストールを実行すると設定ファイルも削除されます。

7. 構造

7.1. 全体構造

Zao SDK for Jetson の全体構成を

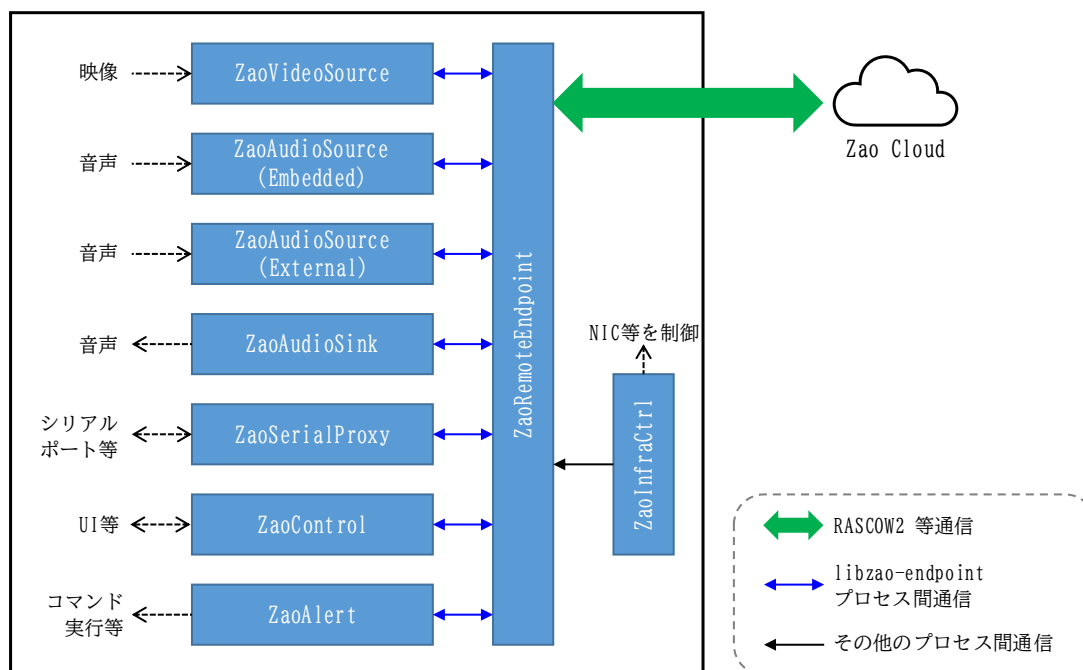


図 7-1 に示します。いくつかのプロセスがプロセス間通信を行いながら協調して動作します。図中央の各プロセスは共通のプロセス間通信ライブラリ libzao-endpoint を用いて基盤プロセスの ZaoRemoteEndpoint と通信します。libzao-endpoint API は SDK ユーザから利用可能であり、SDK ユーザが開発したアプリから API にアクセスすることで、独自のアプリからも Zao SDK for Jetson の機能を利用することが可能です。各プロセスについて後で説明しますが、プロセス種別と SDK ユーザによるカスタマイズ可能かを表 7-1 に示します。デ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。

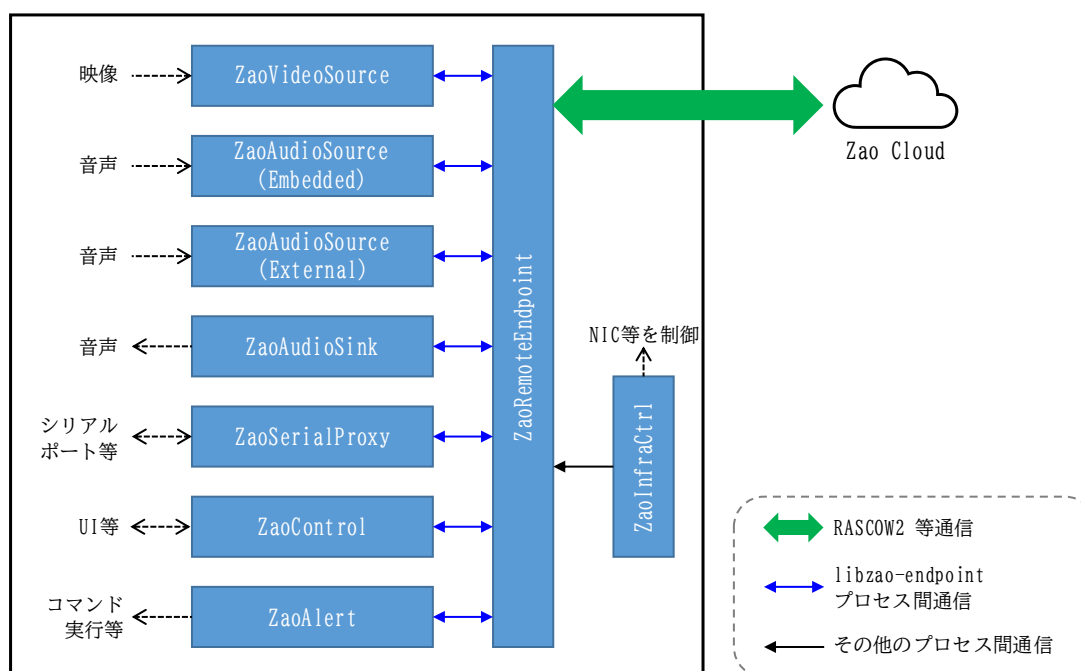


図 7-1 Zao SDK for Jetson 全体構成

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 7-1 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
ZaoAlert	可

7.2. 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、Soliton-MLU 有効化処理において ModemManager は無効化されます。

7.2.1. InfraCtrl.conf

ZaoInfraCtrl は設定ファイル (/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,1
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 Soliton-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[,isMLU]
```

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。[,isMLU] は必須ではない項目で Soliton-MLU に関する設定の場合は ,1 と追記します。

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_1_Nauto (*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_1_Nenable (*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。

7.3. 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 7-2 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 で收容し、各 VLAN を netns に対応付けることも可能です。

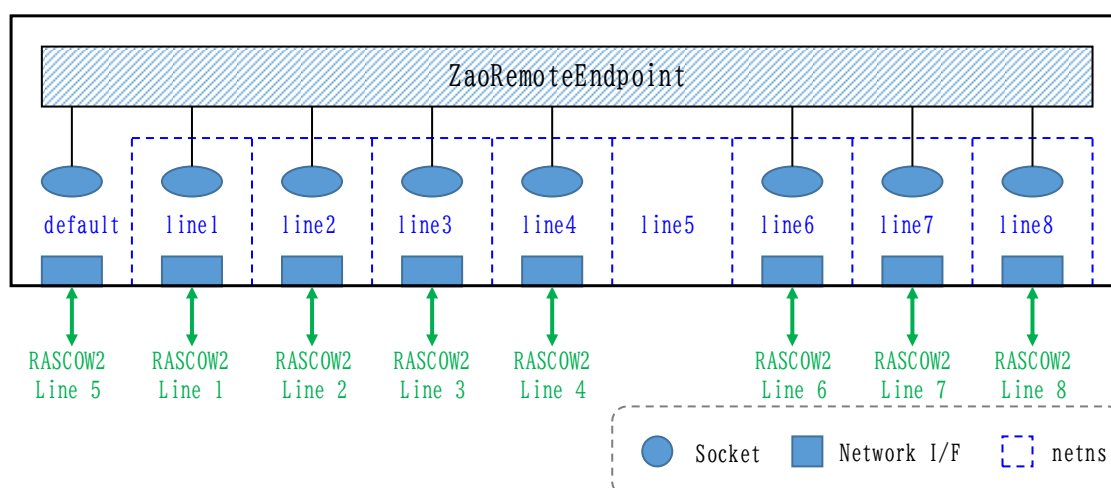


図 7-2 ZaoRemoteEndpoint と各 netns

表 7-2 に Zao SDK for Jetson 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、/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 7-2 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

7.4. 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, ZaoVideoSourceArgus 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、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。

7.5. 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、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。

7.6. 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 はソースも供給され、SDK ユーザによる開発が可能です。

7.7. 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*N*) 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。

ZaoSerialProxyTty のデフォルト設定ではターゲットデバイスが /dev/ttyZAO0 となっています。同時にインストールされる udev 設定によって ttyZAO0 は /dev/ttyUSB0 や /dev/ttyUSB1 へのシンボリックリンクとなる構造になっています。

7.8. 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。

7.8.1. text モードでの ZaoControlCloud 起動

Linux が text モードで起動する場合は、tty1 でアカウント zaoauto が自動ログインし、ログインシェルとして `/home/zao/autologin.sh` が実行されます。autologin.sh では tmux 下で `/home/zaoauto/session.sh` を起動し、この中で ZaoControlCloud が実行される形になっています。

ZaoControlCloud を起動しているシェルスクリプト session.sh が tmux 下で動作しているため、他のターミナルから tmux にアタッチすることで画面を共有することが可能です。

7.8.2. GUI モードでの ZaoControlCloud 起動

Linux が GUI モードで起動する場合は text モード時の tty1 での zaoauto 自動ログインは動作しません。systemd の zao-control-cloud.service が起動し、その中でアカウント zao 権限で tmux とその下で ZaoControlCloud を実行させる形になっています。systemd サービスとして起動しているため、GUI 自動ログイン設定が Off でログイン画面で止まってもバックグラウンドでは ZaoControlCloud が動作しており Zao Cloud への通信は開始されます。

GUI ログイン後に Gnome Terminal に ZaoControlCloud の画面がでてきますが、プロセスは systemd サービスとしてログイン前から起動しており、ログイン後の画面は tmux にアタッチして共有された画面です。

ZaoControlCloud はアカウント zao 権限で動作しているため、ZaoControlCloud System メニュー以下の Shutdown, Reboot は権限不足で動作しません。GUI 画面右上のボタンから Shutdown, Reboot 操作を行ってください（もしくは権限のある ssh リモートログインのシェル上で `sudo reboot` 実行するなど）。

7.9. ZaoAlert プロセス

ZaoAlert はプロセスの種別でもありビルド済みで提供されるアプリ名でもあります。ZaoAlert プロセスはアラート機能を実現するものです。この種別のプロセスは libzao-endpoint の API を通じて動作状態などの情報を取得することが可能です。Jetson 上で複数個のプロセスを実行することが可能です。

セットアップによりビルド済みアプリ ZaoAlert が自動起動する形に設定されます。

ZaoAlert アプリはソースも供給され、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。ZaoAlert 種別プロセスは複数同時実行も可能なため、ビルド済みアプリ ZaoAlert に加えて、独自の ZaoAlert 種別プロセスを同時に実行することも可能です。

ビルド済みアプリ ZaoAlert によって実現されるアラート機能については 9 章で説明します。

8. 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に apps のソースファイルが含まれています。以下のプロセスが apps ソースに含まれています。

- ZaoVideoSourceWebcam
- ZaoVideoSource4cam
- ZaoVideoSourceArgus
- ZaoAudioSourceAlsa
- ZaoAudioSinkAlsa
- ZaoSerialProxyTty
- ZaoControlCloud
- ZaoAlert

apps ソースを Jetson 上でビルドするには以下のコマンド操作でソースの展開と準備を行います。準備はインターネットへアクセスできる状態で実行する必要があります。

```
$ tar xf zao_sdk_jetson_x.x.x.x/apps/apps_src.tar.xz
$ sudo apps/prepare.sh
(略)
$
```

準備後は目的のアプリケーション毎にビルドを行います。例として ZaoSerialProxyTty の場合、以下のようになります。他アプリケーションも同様の形となります。

```
$ cd apps/serial-proxy-tty
$ mkdir build
$ cd build
$ cmake ..
$ make
$ sudo make install
```

最後の make install でビルドしたバイナリが /usr/local/bin/ へ配置されますので、Zao SDK for Jetson セットアップ時に配置された /usr/bin/ のバイナリを自動起動している部分のパスを調整し /usr/local/bin/ のバイナリを自動起動するよう調整することで動作をカスタマイズ可能です。

ライブラリの API 仕様や、apps 各プログラムについては libzao-endpoint API リファレンスを参照してください。

9. アラート機能

アラート機能を用いることで、Zao SDK for Jetson の動作が特定条件を満たした時に指定したコマンドを実行することが可能です。これは ZaoAlert プロセスによって実現されます。

アラートは「アラートカテゴリ」とそれに属する「アラートイベント」によって定義されます。イベントが発火した際に指定したコマンドが実行されます。ZaoAlert に実装されているカテゴリとイベントを表 9-1 に示します。

表 9-1 アラートカテゴリとアラートイベント

アラートカテゴリ	アラートイベント	説明
streaming	started	ストリーミングの開始
	stopped	ストリーミングの停止
	changed	ストリーミング状態の変化
line.state	active	いずれかの RASCOW2 回線が接続
	inactive	いずれかの RASCOW2 回線が切断
	changed	いずれかの RASCOW2 回線状態が変化
line.count	over	RASCOW2 接続状態の回線数が上限以上
	under	RASCOW2 接続状態の回線数が下限未満
line.rtt	over	いずれかの RASCOW2 回線の RTT が上限以上
	under	いずれかの RASCOW2 回線の RTT が下限未満
video	down	ストリーミング中で一定期間映像データが供給されなかった

一部のアラートカテゴリはユーザによって設定可能なパラメータを持ちます。パラメータを表 9-2 に示します。

表 9-2 アラートカテゴリのパラメータ

アラートカテゴリ	パラメータ	必須	説明
streaming	(無し)		
line.state	(無し)		
line.count	high	必須	上限値 (整数、1~8)
	low	必須	下限値 (整数、1~8、(下限値) ≤ (上限値))
line.rtt	high	必須	上限値 (整数、60~10,000)
	low	必須	下限値 (整数、10~1,000、(下限値+50) ≤ (上限値))
video	timeout	任意	タイムアウト時間 (整数、5~600、デフォルト値 30)

上限・下限をパラメータに持つカテゴリはイベントの発火がヒステリシスを考慮した動作となります。

各アラートイベントが発火した時に実行するコマンド文字列中で特定のパターンが置換されます。置換の対応を表 9-3 に示します。

表 9-3 アラートイベントのコマンド実行時の置換パターン

アラートカテゴリ	パターン	置換文字列
(任意)	%%	% に置換される
streaming	%v	変化後の状態 (started または stopped)
line.state	%v	変化後の状態 (active または inactive)
	%i	回線番号 (1~8)
line.count	%v	active な回線数 (0~8)
line.rtt	%v	RTT 値 [ms]
	%i	回線番号 (1~8)
Video	(無し)	

注: 上記以外の %x (x は任意文字) や % 単体はエラーとなります。

9.1. Alert.toml

ZaoAlert は TOML 形式の設定ファイルを読み込んで動作します。デフォルトは /etc/zao/Alert.toml を読み込みますが、コマンドラインオプションで異なるパスを指定することも可能です。

設定ファイルのサンプルを以下に示します。

```
# # から行末はコメントになります

# すべてのアラートイベントは以下のキーを指定可能
# - enabled
# - command
# - cwd
# アラートによってはアラートカテゴリ階層でパラメータ等を設定します

# テーブル名に (アラートカテゴリ).(イベント) を指定します
[streaming.started]

# false を指定すると無効扱いでコマンド起床されません
# (未指定時のデフォルト値は true)
enabled = false

# 文字列を渡すと、トリガ条件成立時にコマンドが起床されます
# (/bin/sh -c "<文字列>" 扱いで引数が解釈されます)
command = "/usr/local/bin/ZaoAlertExample.sh streaming.started"

# カレントディレクトリの指定が可能
# (未指定時は、ZaoAlert プロセスのカレントディレクトリを引き継ぎます)
cwd = "/tmp"

[streaming.stopped]
# 文字列の配列を渡すと、配列の要素がそれぞれ引数として解釈されます
# (引数内にスペースを含む場合等に有用)
command = ["/usr/local/bin/ZaoAlertExample.sh", "streaming stopped"]

[streaming.changed]
```

```
# 文字列中の %* 表記は、アラート毎のルールに従い置換されます
command = "/usr/local/bin/ZaoAlertExample.sh %v"

# アラートによってはパラメータの設定が可能です
# パラメータは (アラートカテゴリ) の階層で記述します
[line.count]
high = 2
low = 2

# コマンドは (アラートカテゴリ).(イベント) の階層になります
[line.count.over]
command = "/usr/local/bin/ZaoAlertExample.sh line.count.over %v"

[line.count.under]
command = "/usr/local/bin/ZaoAlertExample.sh line.count.under %v"
```

TOML 形式については <https://toml.io/ja/> を参照してください。

TOML のテーブルには、[アラートカテゴリ] ([line.rtt] 等) と [アラートカテゴリ.アラートイベント] ([line.rtt.over] 等) が指定可能です。

[アラートカテゴリ] のテーブルにはカテゴリに応じたパラメータをキーに記す事ができます。

[アラートカテゴリ.アラートイベント] のテーブルには以下のキーを記す事ができます。

- enabled false または true。記述は任意でデフォルトは true。false の場合はこのイベントは発火しません。
- command コマンド文字列、記述は必須。イベントが発火した時に実行されるコマンドを指定します。
- cwd 実行ディレクトリ指定。記述は任意でデフォルトは ZaoAlert プロセスのカレントディレクトリが引き継がれます。

10.GNSS 受信機対応

Zao SDK for Jetson 1.1 では以下の GNSS レシーバを使用し、位置情報を Zao Cloud へ逐次送信することが可能です。

- GlobalSat BU-353N5

GNSS レシーバを使用するには設定ユーティリティでの有効化操作が必要です。詳細は 6.3 節を参照してください。有効化後は当該 GNSS レシーバを接続 USB コネクタ位置に制限はありません。Jetson 起動後にレシーバ接続したり、挿抜してもレシーバは認識されます。

GNSS レシーバは原理上、通電開始してから位置情報が得られ始めるまで数分～数十分かかる場合があります。電波状態が悪い環境では位置情報が得られないことがあります。

11. 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 11-1 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 Zao SDK for Jetson セットアップ時に配置されます。

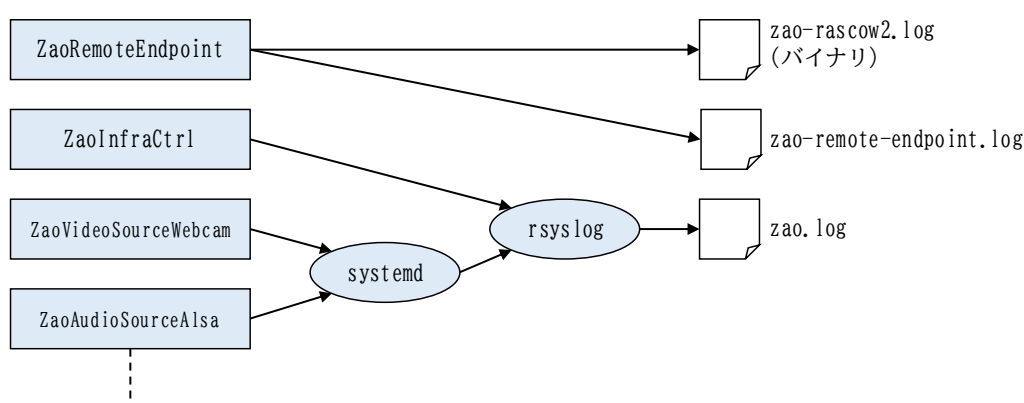


図 11-1 ログの流れ

zao-rascow2.log, zao-remote-endpoint.log は ZaoRemoteEndpoint プロセスによってローテートが実施されます。zao-rascow2.log について、Jetson 起動タイミング、またはファイルサイズが約 64 MiB 超過したタイミングでローテートされ、zao-rascow2.1.log, …, zao-rascow2.9.log まで保持されます。zao-remote-endpoint.log についても同様です。Zao SDK for Jetson 1.2 時点で、ファイルサイズしきい値、保持ファイル数に変更できません。

zao.log は /etc/cron.daily/logrotate で毎日に実行される logrotate によってローテーションされます。設定ファイル /etc/logrotate.d/zao によって動作を調整することが可能です。

上記ログ以外に、セットアップや設定変更に関しては /var/log/zao-setting.log に記録されます。

11.1. ログアップロード機能

Zao SDK for Jetson 1.1 からログアップロード機能に対応しました。Zao Cloud View 側でアップロード指示 (一定以上のユーザ権限が必要です) を行うと、Jetson からログファイルが Zao Cloud サーバへアップロードされます。アップロード中は回線容量を消費するため、映像等伝送に影響する可能性があります。

12.Zao SDK for Jetson セットアップによる影響

Zao SDK for Jetson 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。

12.1. Systemd サービスの追加・有効化・無効化

設定ユーティリティで Soliton-MLU を導入する際に、ModemManger サービスを無効化します。ZaoInfraCtrl による MLU 制御と干渉するためです。

以下の systemd サービスを追加します。

- zao-infra-ctrl
- zao-remote-endpoint
- zao-video-source-webcam
- zao-video-source-4cam
- zao-video-source-argus
- zao-audio-source-alsa-embedded
- zao-audio-source-alsa-external
- zao-audio-sink-alsa
- zao-serial-proxy-tty
- zao-control-cloud
- zao-alert

これらは /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, zao-alert 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。

12.2. 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。

12.3. 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。

- Soliton-MLU の制御
- シリアルトンネル用 USB-シリアルデバイスの認識
- GNSS レシーバの認識

12.4. 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。

12.5. 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。

12.6. カーネルモジュールの追加

設定ユーティリティから Soliton-MLU 有効化を行った際に QMI_WWAN カーネルモジュールが追加されます。

設定ユーティリティから GNSS レシーバ有効化を行った際に PL2303 カーネルモジュールが追加されます。

12.7. 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 12-1 に配置されるプロセスを示します。

表 12-1 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-webcam	○*
ZaoVideoSource	ZaoVideoSource4cam	zao-video-source-4cam	×*
ZaoVideoSource	ZaoVideoSourceArgus	zao-video-source-argus	×*
ZaoAudioSource	ZaoAudioSourceAlsa	zao-audio-source-alsa-embedded	○
		zao-audio-source-alsa-external	×
ZaoAudioSink	ZaoAudioSinkAlsa	zao-audio-sink-alsa	×
ZaoSerialProxy	ZaoSerialProxyTty	zao-serial-proxy-tty	○
ZaoControl	ZaoControlCloud	zao-control-cloud	○*
ZaoAlert	ZaoAlert	zao-alert	○

- * 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。
- * zao-control-cloud は GUI モード時のみ自動起動されます。Linux が text モードで起動する場合には zaoauto アカウントの自動ログインを通じて ZaoControlCloud が起動されます。

13.ROM 化

Zao SDK for Jetson では Nano Developer Kit での ROM 化を提供しています。ROM 化するとログファイル等はシャットダウンで揮発しますが、シャットダウン処理を経ず電源断することが可能となります（ROM 化していない状態で突然電源断するとファイルシステムが破損する可能性があります）。

Nano Developer Kit 以外のキャリアボードについてはボード毎の対応になります。ストレージ構成、ブートローダ、Linux カーネル、デバイスツリーなどを考慮した対応が必要となります。

本章では ROM 化に関連する Zao SDK for Jetson 動作内容の説明と、Nano Developer kit での ROM 化の関連手順について説明します。

13.1. ROM 化と SDK 動作

Zao SDK for Jetson セットアップ・設定変更・ペアリング操作は ROM 化していない状態で行ってください。セットアップでは /etc/, /lib/, /usr/, /var/ への書き込みが生じます。ペアリングでは /var/lib/zao/ への書き込みが生じます。設定ユーティリティは操作内容によりますが主には /etc/ への書き込みが生じます。これらの書き込みが揮発しない状態で実行する必要があります。

ペアリング等が完了し、伝送動作のみ行う段階では、Zao SDK for Jetson からは以下への書き込みが生じますので、/etc 等が ReadOnly では正しく動作しません。Overlay FS で構成され書き込み可能であれば Jetson 再起動で揮発しても伝送動作には問題ありません。この場合は Jetson の電源断でログファイルは失われます。

- /etc/netns/
- /var/log/
- /run/ZaoInfraCtrl/
- /run/zao/

13.2. Nano Developer Kit での ROM 化

本節では Zao SDK for Jetson が提供する Nano Developer Kit での ROM 化の操作について説明します。NVIDIA から提供されている JetPack 4.6.1 の SD card image から起動した状態での動作確認を行っていますが、apt を通じたアップデートを経るなどで状態が変化していると ROM 化が正しく動作せず、うまく起動できなくなる可能性がありますのでご注意ください。

13.2.1. ROM 化手順

ROM 化は設定の調整・ペアリングを完了した後に行うようにしてください。ROM 化状態では、ログは Jetson 電源断で揮発します。

ROM 化するには以下の手順を実行してください。

- tty2 等に sudo 可能アカウントでログイン
- sudo /usr/share/zao/make_initrd_ro.sh 実行
これにより /boot/initrd.ro が生成・配置されます。
- sudo 権限で /boot/extlinux/extlinux.conf を以下のように編集します
(sudo vim /boot/extlinux/extlinux.conf 等)
青字が変更・追加部分です。

```

TIMEOUT 30
DEFAULT readonly

MENU TITLE L4T boot options

LABEL primary
    MENU LABEL primary kernel
    LINUX /boot/Image
    INITRD /boot/initrd
    APPEND ${cbootargs} quiet root=/dev/mmcblk0p1 rw rootwait rootfstype=ext4
console=ttyS0,115200n8 console=tty0 fbcon=map:0 net.ifnames=0
(略)

LABEL readonly
    MENU LABEL readonly
    LINUX /boot/Image
    INITRD /boot/initrd.ro
    APPEND ${cbootargs} quiet root=/dev/mmcblk0p1 rw rootwait rootfstype=ext4
console=ttyS0,115200n8 console=tty0 fbcon=map:0 net.ifnames=0

```

- ルートディレクトリに /not_overlay ファイルが存在する場合は消しておきます
- 再起動します

13.2.2. ROM 化解除/再 ROM 化

ROM 化を解除する場合は ROM 化状態で設定ユーティリティを起動してください。設定ユーティリティが ROM 化状態を認識し、ROM 化解除して再起動するか尋ねてきます。そのまま進むと再起動が始まり、ROM 化解除した状態で起動します。

再び ROM 化する場合には sudo 権限でルートディレクトリの /not_overlay ファイルを消してから再起動してください。

13.2.3. 補足

Ubuntu のパッケージアップデートで Linux kernel, ブートローダ等がアップデートされた場合、ROM 化が解除される場合があります。その際は、再 ROM 化手順ではなく、再度 ROM 化手順からやり直してください。

うまく起動しない場合、デバッグコンソールを用いることで手掛かりを得たり、リカバリーできる可能性があります。デバッグコンソールは Nano Developer Kit の場合、J50 のピンに出ています。詳しくは Nano Developer kit の関連ドキュメントを参照ください。

14. デフォルト設定値

セットアップスクリプトは一部のキャリアボードを認識して、デフォルト設定を変化させます。本章ではキャリアボード毎のデフォルト設定値について説明します。

14.1. 共通

キャリアボードによらず共通に設定される項目は以下です。

- 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
- ZaoControlCloud の自動起動
 - ◆ Linux を text モード起動し、ttyl にアカウント zaoauto が自動ログイン、tmux 下で ZaoControlCloud が実行される形です。
- ZaoAlert の自動起動
 - ◆ 各アラートイベントは disable 状態

以降に説明されていないキャリアボードについては USB NIC 等と RASCOW2 Line との対応についての設定は導入されません。前述の説明に基づいて SDK ユーザが設定を施すことにより、対応した USB NIC を指定の RASCOW2 Line に対応付けてマルチリンク接続を実現する形となります。

14.2. Jetson Nano Developer Kit (Rev. B01)

Jetson Nano Developer Kit の場合、デフォルト設定では図 14-1 の USB コネクタに 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 14-2 に示します。USB NIC を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。設定ファイルを調整することによってこれ以外の対応関係で動作させることも可能です。

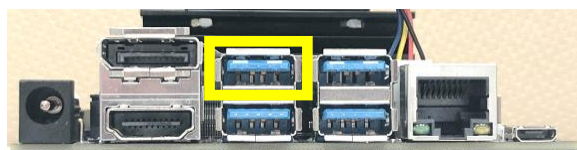


図 14-1 USB NIC 等を接続する USB コネクタ
(Jetson Nano Developer Kit の場合)

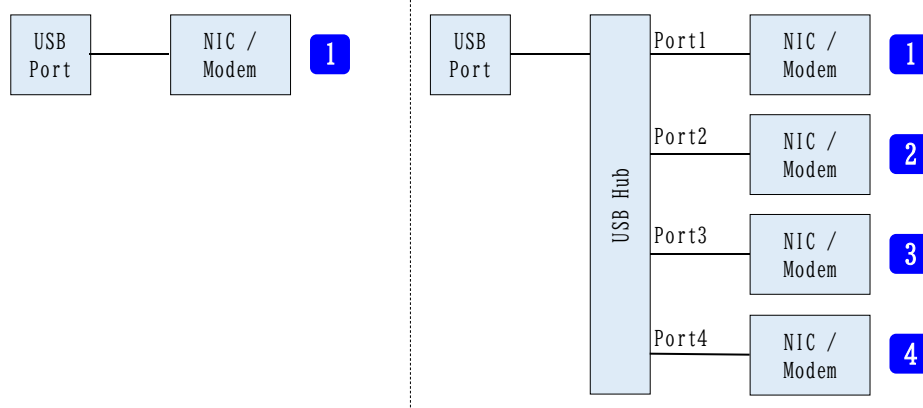


図 14-2 デバイスの接続と RASCOW2 Line との対応関係
(Jetson Nano Developer Kit の場合)

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。

14.3. Jetson Xavier NX Developer Kit

Jetson Xavier NX Developer Kit の場合、デフォルト設定では図 14-3 の USB コネクタに 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 14-4 に示します。USB NIC を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。設定ファイルを調整することによってこれ以外の対応関係で動作させることも可能です。

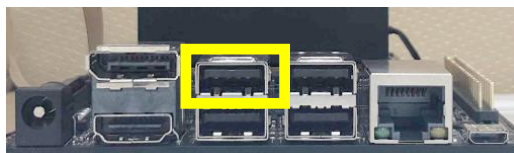


図 14-3 Soliton-MLU 等を接続する USB コネクタ
(Jetson Xavier NX Developer Kit の場合)

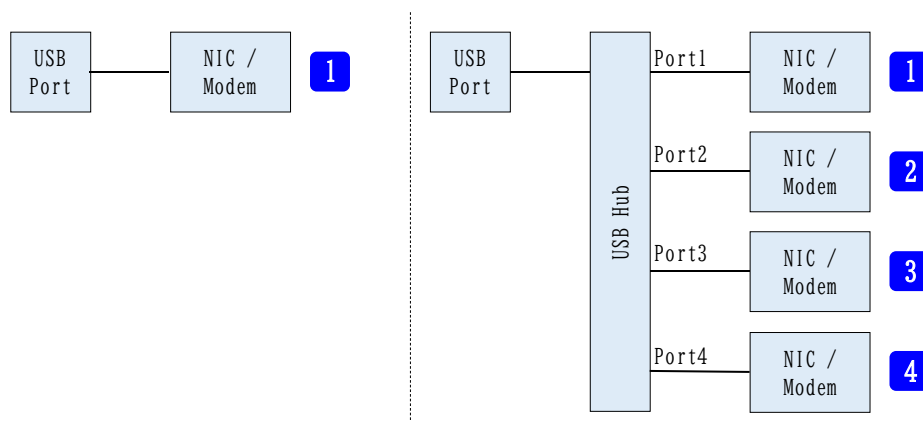


図 14-4 デバイスの接続と RASCOW2 Line との対応関係
(Jetson Xavier NX Developer Kit の場合)

14.4. Jetson AGX Xavier Developer Kit

Jetson AGX Xavier Developer Kit の場合、デフォルト設定では図 14-5 の USB Type-A コネクタに 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 14-6 に示します。USB NIC を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。設定ファイルを調整することによってこれ以外の対応関係で動作させることも可能です。



図 14-5 Soliton-MLU 等を接続する USB Type-A コネクタ
(Jetson AGX Xavier Developer Kit の場合)

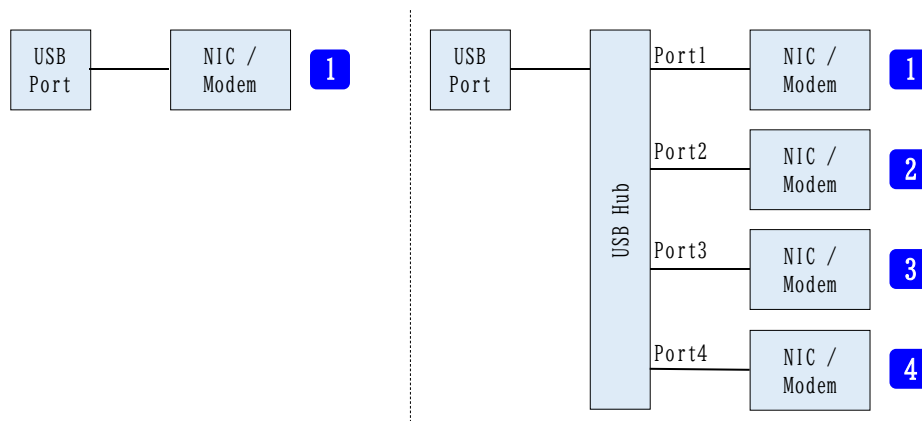


図 14-6 デバイスの接続と RASCOW2 Line との対応関係
(Jetson AGX Xavier Developer Kit の場合)

15.Zao SDK for Jetson 過去のバージョンからのアップデート

15.1. Zao SDK for Jetson 1.0 から 1.1 への変更点

Zao SDK for Jetson 1.1.x.x を新規セットアップした際の 1.0.x.x との違いを以下に列挙します。

- 事前に apt パッケージ導入を手動で行う必要がなくなりました
- セットアップスクリプト実行中にインターネットにアクセスできる必要があります
- Soliton-MLU に関するファイルはデフォルトでは導入されなくなりました
- セットアップ時にプラットフォーム選択が無く、一部のキャリアボードを自動認識する形になりました。
Aetina AX810 は自動認識されませんので、AX810 でマルチリンク接続を行うにはユーザにて設定ファイルを調整頂く必要があります。
- セットアップ時の ZAO_CLOUD_URL の入力方法が変わりました
- ZaoControlCloud を tmux 下で起動する形になります

15.2. Zao SDK for Jetson 1.1 から 1.2 への変更点

Zao SDK for Jetson 1.2.x.x を新規セットアップした際の 1.1.x.x との違いを以下に列挙します。

- ZaoAlert と関連ファイルが追加配置されます

15.3. Zao SDK for Jetson 1.0 からのアップデート手順

Zao SDK for Jetson 1.0.x.x が動作している Jetson を 1.2.x.x にアップデートする手順等を説明します。

1.2 へのアップデートは新規セットアップと同じ手順 (sudo で setup_zao_sdk_jetson.sh 実行) となりますが、下記に注意点を挙げます。

- ZAO_CLOUD_URL は設定済みの値が引き継がれ、新規セットアップのように入力を求められません
- Soliton-MLU に関する設定は引き継がれません。
- 自動起動する ZaoVideoSource プロセスが ZaoVideoSourceWebcam になります
- ZaoSerialProxyTty の自動起動が有効になります
- Linux 起動が text モードになります
- その他の主な設定値、ペアリング済みか否かは引き継がれます

ZAO_CLOUD_URL の変更が必要な場合は、スクリプト実行後に設定ユーティリティで変更してください。

Soliton-MLU 使用するためには、前述の手順で設定ユーティリティを用いて有効化手順を実行してください。上記セットアップスクリプト実行直後に再起動挟まずに設定ユーティリティを使用可能です。

アップデートで自動起動するプロセスは 1.2 のデフォルトと同じになり、ZaoVideoSource としては ZaoVideoSourceWebcam が起動し、シリアルトンネルに関わる ZaoSerialProxyTty が自動起動する形となります。設定ユーティリティ、もしくは手動での設定ファイル調整で自動起動するプロセスを調整してください。

アップデート後は Linux 起動が text モードになりますので、GUI に変更する場合は前述の手順を実行してください。

15.4. Zao SDK for Jetson 1.1 からのアップデート手順

Zao SDK for Jetson 1.1.x.x が動作している Jetson を 1.2.x.x にアップデートするには、新規セットアップと同様に `sudo` で `setup_zao_sdk_jetson.sh` を実行してください。

新規セットアップ同様にインターネットにアクセスできる状態で実行する必要があります。

既存の設定ファイル、ペアリング状況は引き継がれます。設定ファイルに関しては `/etc/zao/Alert.toml` が追加で配置されます。

自動起動するプロセスは新規インストールの場合と同様になります。Zao SDK for Jetson 1.1 をデフォルトと異なる設定で動作させていた場合などは再度設定をしてください。

16. 様々な手順

16.1. 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。

16.1.1. 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、Zao SDK for Jetson セットアップ後も有効となります。

16.1.2. 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 が起動されますので、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。

16.1.3. 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。

17.更新履歴

Rev.	Date	更新内容
1.0-1	2023-05-08	初版
1.0-2	2023-05-12	誤字修正
1.1-1	2023-08-18	1.1 対象に更新 ・ 設定ユーティリティについて追加 ・ GNSS レシーバ対応について追加 ・ ROM 化について追加 ・ 1.0 との差、アップデートについて追加 ・ その他、細部のアップデートを反映
1.1-2	2023-09-01	7.2.1 節の誤字修正 (誤) line_1_address (正) line_1_addr
1.2-1	2023-10-26	1.2 対象に更新 ・ アラート機能について更新 ・ 1.1 との差、アップデートについて追記 ログについて追記 誤字修正