

Installation of RedHawk™ 9.4 Gold on NVIDIA® Jetson AGX Orin series, Orin NX series, and Orin Nano series systems

Release Notes

November 20th, 2024

0898003-9.4-Orin



免責事項

本書に含まれる情報は予告なしに変更することがあります。Concurrent Real-Time, Incは本書から誤りを取り除くために努力していますが、存在し続ける可能性のある誤りに関するConcurrent Real-Time, Incの責務は、Concurrent Real-Time, Incに知らされている指摘された誤りを訂正する事です。

ライセンス

Concurrent Real-Time, Incの書面による同意なしに本書の複製を禁止します。同意によって複製された本書の如何なるコピーもConcurrent Real-Time, Incの著作権情報に含まれます。

商標に関する通知

Concurrent Real-Time, IncおよびそのロゴはConcurrent Real-Time, Incの登録商標です。その他すべてのConcurrent Real-Timeの製品名称はConcurrent Real-Timeの商標であり、同時にその他すべての製品名称は各々の所有者の商標または登録商標です。Linux®はLinux Mark Institute (LMI)のサブライセンスに準じて使用しています。

© 2026 Concurrent Real-Time, Inc - All Rights Reserved

コンカレント日本株式会社

〒111-0052 東京都台東区柳橋 2-19-6

柳橋ファーストビル4階

NOTE: 情報は告知なしに変更されます。また、本書は英文資料を意訳した内容となります。

1. 序文

本書では、Jetson AGX Orin, Orin NX, Orin NanoにRedHawk Linux™ 9.4 GoldのARM64バージョンをインストールする工程を解説します。本書内の指示は(Concurrent Real-TimeのRedHawkシステムへのソフトウェアのインストールに特化している)他の全てに取って代わるものです。

2. 必要条件

- ・ インターネット接続されたJetson AGX Orin, Orin NX, またはOrin Nano
- ・ インターネット接続されたUbuntu 22.04が動作しているx86_64ホスト・システム (ホスト・システムは最新のUbuntuパッケージ・アップデートに更新されている必要があります。)
- ・ Jetson AGX Orin, Orin NX, Orin Nano向けRedHawk 9.4 Goldの光メディア・ディスク

3. インストール

3.1. ホスト・システムに**SDK Manager**をダウンロード

Ubuntuが動作しているx86_64ホスト・システムに次のURLを介してNVIDIAのウェブサイトからSDK ManagerのVersion 2.2.0-12021をダウンロードして下さい：

<https://developer.nvidia.com/jetpack>

SDK ManagerのパッケージをダウンロードするにはNVIDIA SDK Managerのリンクをクリックして下さい (ログインが必要となる可能性があります)。

3.2. ダウンロードした**SDK Manager**をインストール

SDK Managerをインストールするにはターミナル・ウィンドウを開いて次のコマンドを実行して下さい：

```
$ cd ~/Downloads  
$ sudo apt install ./sdkmanager_2.2.0-12021_amd64.deb
```

3.3. ホスト・システムに**Orin**デバイスを接続

SDK ManagerがOrinを検出して正常にフラッシュできるようにするために、Orinをホスト・システムに接続する必要があります。2つの異なるUSBケーブルをOrinとホスト・システムに接続する必要があります。Orinに含まれるケーブルと資料を参照して下さい。

3.4. **Orin**デバイスをリカバリー・モードに移行

SDK Managerの最新バージョンは、Orinが最初にリカバリー・モードに移行する際に最適に動作します。AGX Orinについては、電源が投入され起動している場合、リカバリー・モードに入るために次のコマンドを実行して下さい：

```
reboot --force forced-recovery
```

あるいは、手動操作で特定のOrinデバイスをリカバリー・モードに移行するには、NVIDIAの資料を参照して下さい。リカバリー・モードで、lsusbをホストで実行すると次のデバイスのいずれかを特定するNVIDIA Corp.エントリを表示します：

0955:7023 (AGX Orin 64 GB)	0955:7323 (Orin NX 16 GB)	0955:7523 (Orin Nano 8 GB)
0955:7223 (AGX Orin 32 GB)	0955:7423 (Orin NX 8 GB)	0955:7623 (Orin Nano 4 GB)

これはOrinが適切にリカバリー・モードに入っていることを意味します。

3.5. SDK Managerを使ってOrinデバイスをフラッシュ

SDK Managerの利用を開始するには現在のユーザーで次のコマンドを実行して下さい：

```
$ sdkmanager
```

SDK Managerが要求する最初の操作は、ご自身の開発またはパートナー・アカウントのemailアドレスとパスワードを使ってnvidia.comのウェブサイトにログインすることです。適切な情報を入力しLOGINを押して進めて下さい。

NOTE

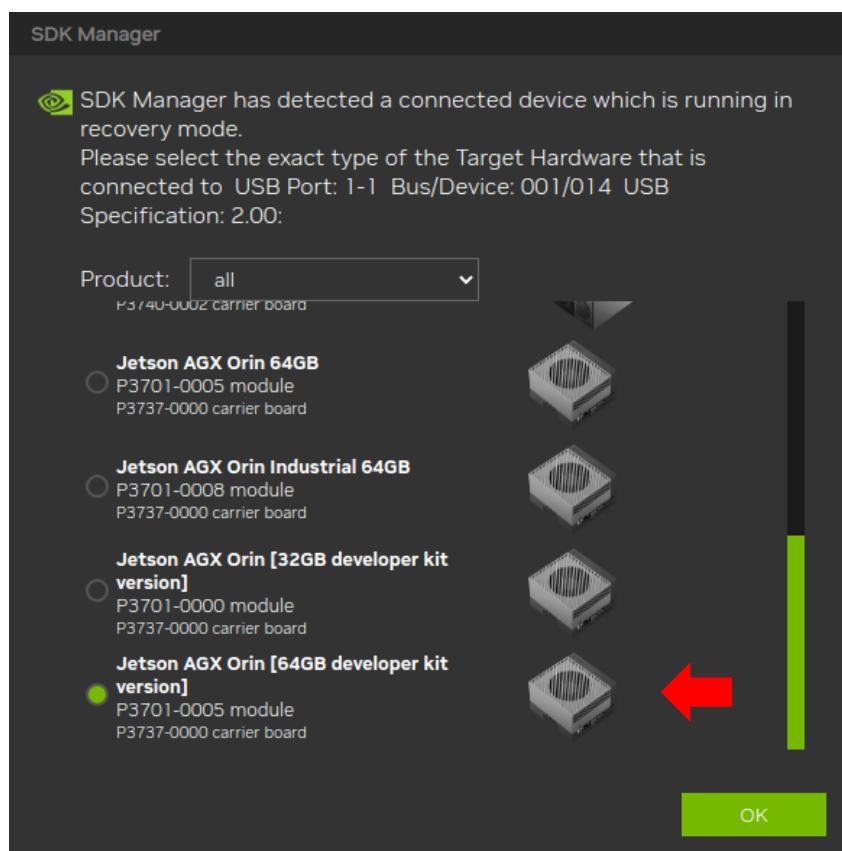
RedHawk 9.4はJetPack Version 6.1でのみ有効です。他のバージョンはRedHawk 9.4とは互換性がないため使用しないで下さい。

この時点で通常は*Jetson AGX Orin Developer Kit User Guide*の手順に加え、次のRedHawk固有の手順に従うことが可能です。

NOTE

*AGX Orin 64GB Developer Kit*のスクリーンショットを以下提供しますが、ここで説明するインストール過程は*Orin NX*および*Nano*デバイスにも適用されます。

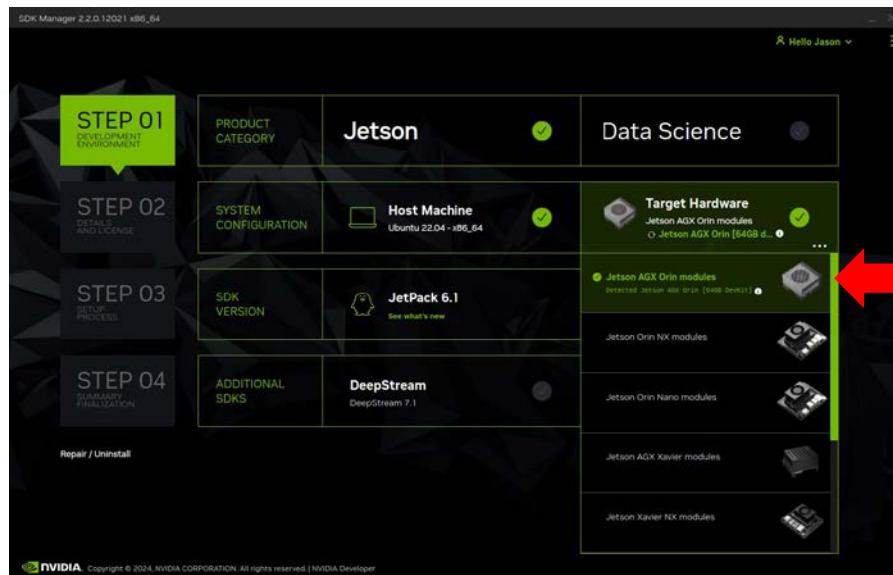
1. SDK Managerを最初に起動した際、この時点でOrinが接続され正常にリカバー・モードに移行している場合、以下のダイアログがすぐに画面に現れます：



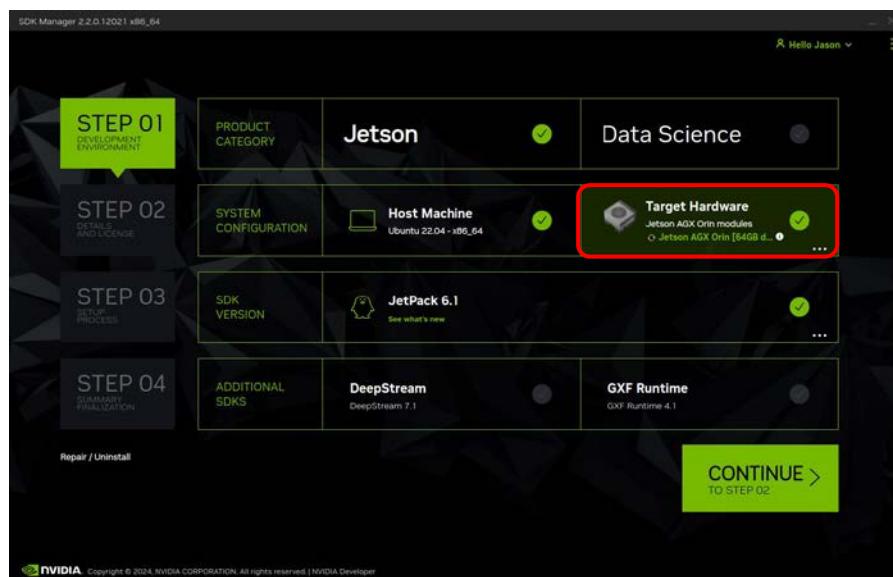
Jetson AGX Orin [64GB developer kit version]を選んでOKを押して下さい。

OrinがSDK Managerに検出されなかった場合、インストールするハードウェア・モデルを選択するには次項に進んで下さい。

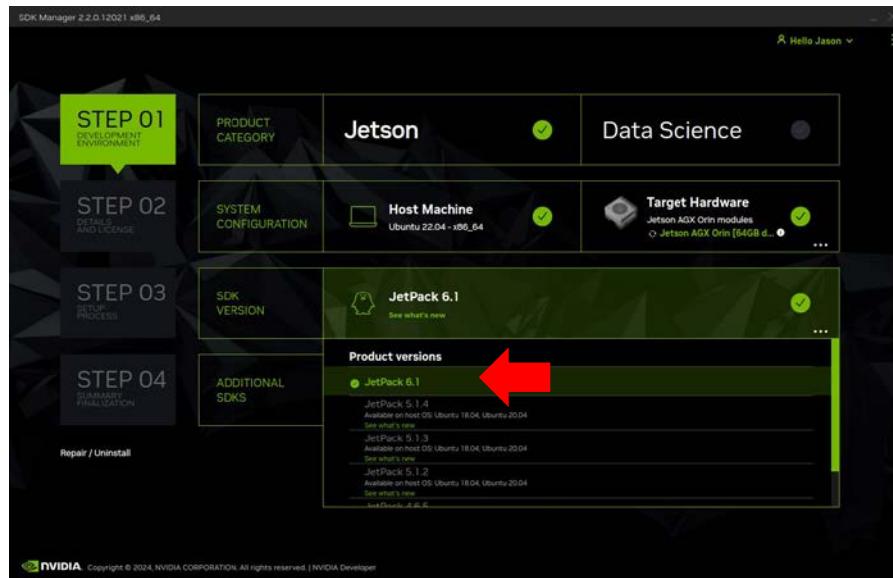
2. STEP 01 - DEVELOPMENT ENVIRONMENTの画面上で、次の画像に示すようにサポートされるハードウェア構成のリストを表示するにはTarget Hardware領域内の...アイコンをクリックして下さい：



3. 次の画像に示すようにターゲットが選択された状態にするにはJetson AGX Orin modulesターゲット・ハードウェア・オプションをクリックして下さい：



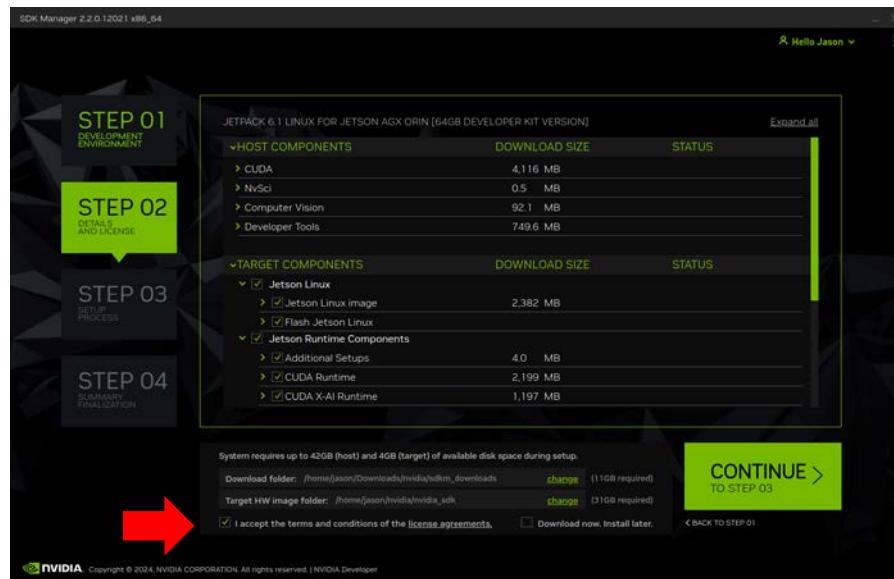
4. 次の画像に示すようにサポートされるJetPackのバージョンのリストを表示するにはJetPack領域内の...をクリックして下さい：



5. 次の画像に示すようにバージョンが選択された状態にするにはJetPack 6.1をクリックして下さい：



6. Continueを押してSTEP 02 – DETAILS AND LICENSE画面へ進み、次の画像に示すようにライセンスを承認するにはウィンドウ下部にあるI accept the terms and conditions of the license agreementsをクリックして下さい：

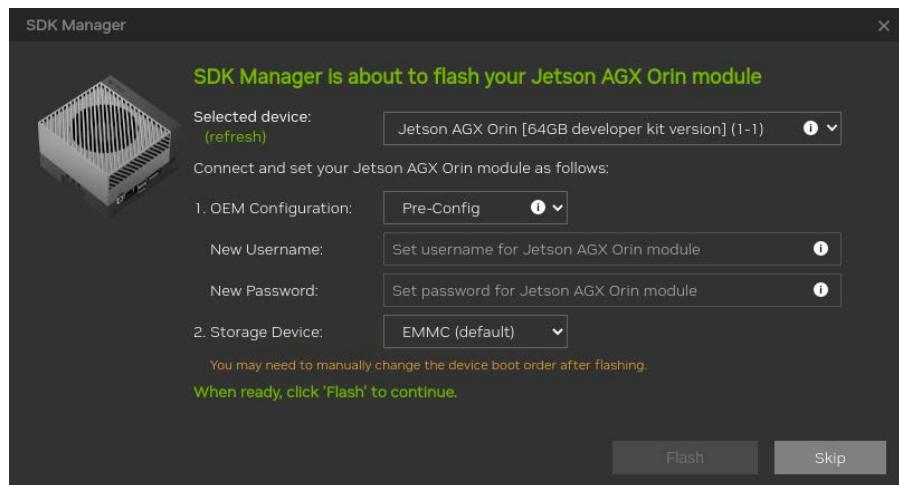


7. Continueを押してSTEP 03 – SETUP PROCESS画面へ進むと次に示すようにSDK ManagerがNVIDIAから全てのソフトウェア・コンポーネントのダウンロードを開始します：

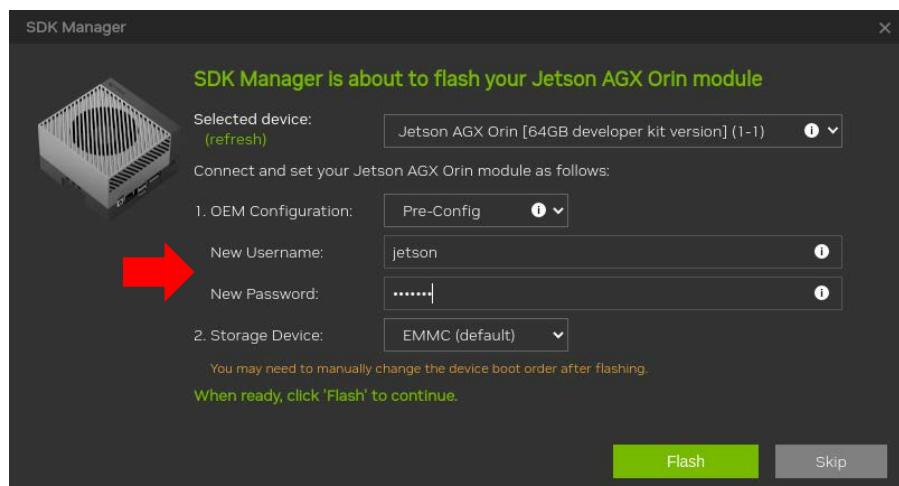


ネットワークの速度にもよりますが、ダウンロードは30分以上かかる可能性があります。ダウンロードおよびセットアップ中、詳細な出力ログはTERMINALタブの下で確認することができます。

8. ダウンロードとセットアップが完了するとフラッシュ・ダイアログが表示されます。



次のダイアログに示すようにOrinで生成されるデフォルト・ユーザー用に**New Username**と**New Password**の欄に入力して下さい：

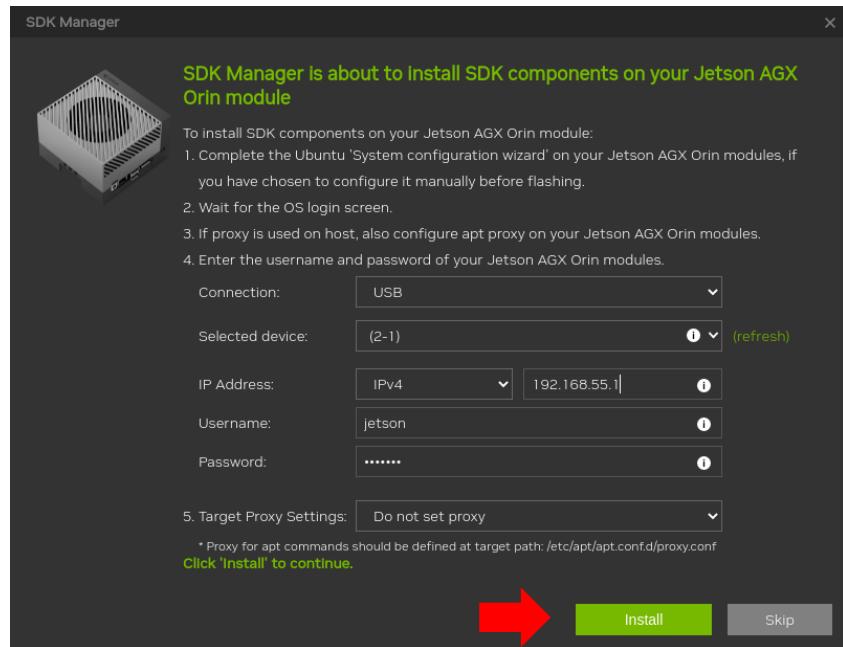


Orinのフラッシュを開始するには**Flash**をクリックして下さい。

NOTE

フラッシュは時折原因不明の失敗をする場合があります。フラッシュが失敗した場合、*Orin*を抜いた後にリカバリー・モードに移行している間に接続して下さい。また、*Orin*とホストを接続しているUSBケーブルを切断してから再接続して下さい。その後のフラッシュの試みは大抵成功します。

9. 次のダイアログに示すようにフラッシュ中、SDK Managerは再度UsernameとPasswordの欄への入力を促します：



これらの欄はデフォルトで前のステップで指定した値になっていますので、ソフトウェアのインストールを継続するには単にInstallをクリックして下さい。

10. フラッシュとソフトウェアのインストールが完了したら、自動的にSTEP 04 - SUMMARY FINALIZATION画面に進みます。



SDK Manaを終了するにはFinish and Exitをクリックして下さい。

3.6. 最適なパフォーマンスのためのNVIDIA電源モデルの設定

SDK ManagerがOrinへのUbuntuのインストールが完了した後、OrinがNVIDIA標準のカーネルで起動している際、フラッシュ時に選択したユーザー名とパスワードを使ってOrinにログインし、次のコマンドを実行して下さい：

```
$ sudo nvpmodel -m number

number : 3 for Orin AGX 64 GB (50 Watt)      3 for Orin AGX 32 GB (40 Watt)
          3 for Orin NX 16 GB (25 Watt)        3 for Orin NX 8 GB (20 Watt)
          0 for Orin Nano 8 GB (15 Watt)       0 for Orin Nano 4 GB (10 Watt)
```

次のような出力が表示されます：

```
NVPM WARN: Golden image context is already created
NVPM WARN: Reboot required for changing to this power mode: 3
NVPM WARN: DO YOU WANT TO REBOOT NOW? enter YES/yes to confirm:
```

yesと入力しENTERを押してOrinを再起動、再度ログインして次のコマンドを実行して下さい：

```
$ sudo nvpmodel -q
```

特定のOrinデバイスが上述の値に正しく設定されたことを確認して下さい。

Orinの電源モードが設定されたら、再度nvpmodelコマンドを使って明示的に変更されるまで設定は再起動しても持続されることに注意して下さい。

3.7. SOC Display Hand-Offモードを無効化(必須)

次項でインストールするRedHawkカーネルはSOC Display Hand-Off ModeをOrinのファームウェア設定内で完全に無効化(Neverに設定)する必要があります。Display Hand-Offを無効にするには本項の手順に従って下さい。

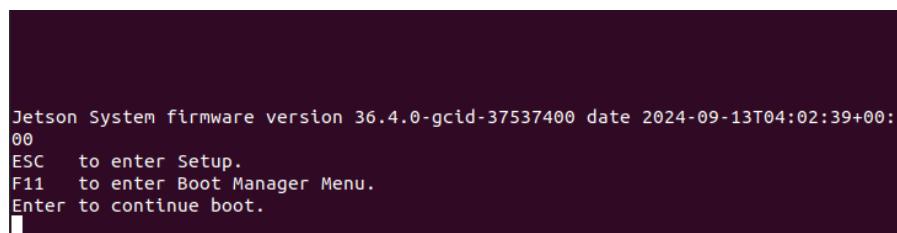
- ホスト・マシーンからOrinのシリアル・コンソールに接続して下さい。シリアル・コンソールにアクセスするのにminicomをインストールして利用することを選択した場合、次のコマンドを実行して下さい：

```
$ sudo minicom --color=on -b 115200 -D /dev/ttyACM1
```

NOTE

代わりにOrinに接続されたモニターとキーボードを使用することが可能です。

- Orinを再起動してシリアル・コンソールの下部に次のテキストが現れるまで待って下さい：

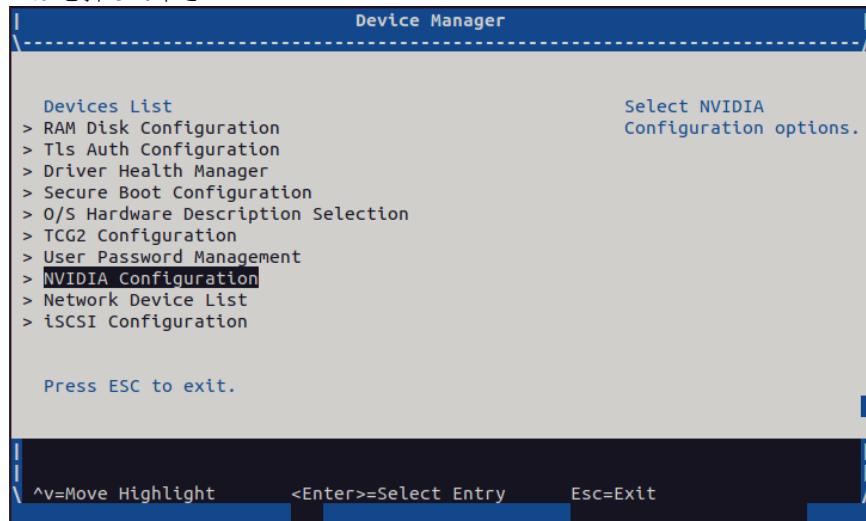


これが表れたら、ESCキーを押してファームウェア設定メニューに入って下さい。

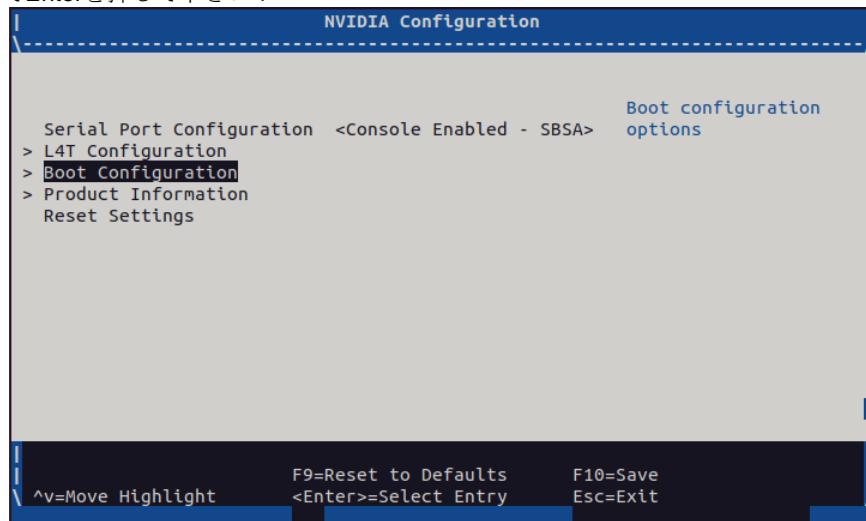
3. フームウェア設定メニューで、矢印キーで下に移動し(下に示す) Device Managerを選択してEnterを押して下さい：



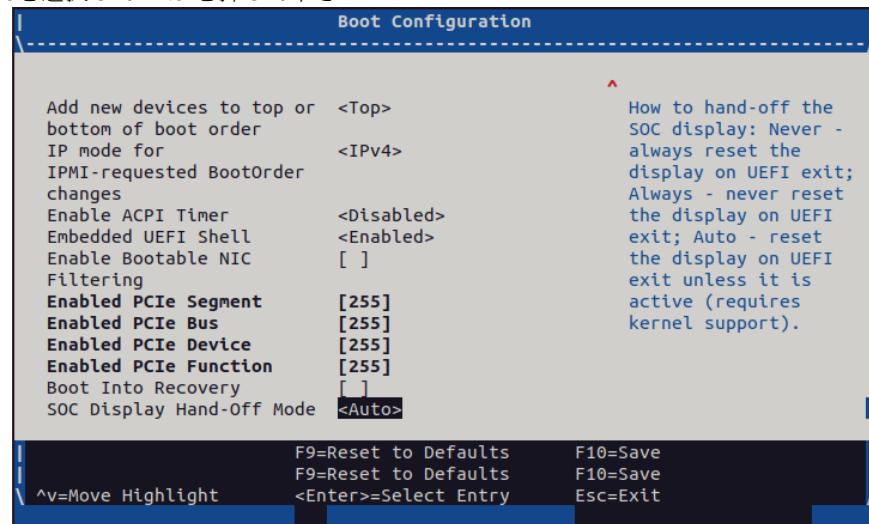
4. Device Managerメニューで、矢印キーで下に移動し(下に示す) NVIDIA Configurationを選択してEnterを押して下さい：



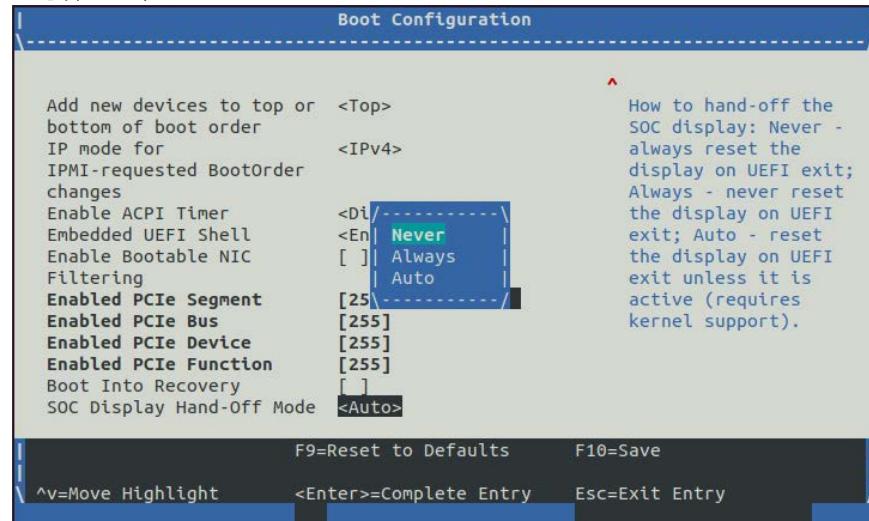
5. NVIDIA Configurationメニューで、矢印キーで下に移動し(下に示す) Boot Configurationを選択してEnterを押して下さい：



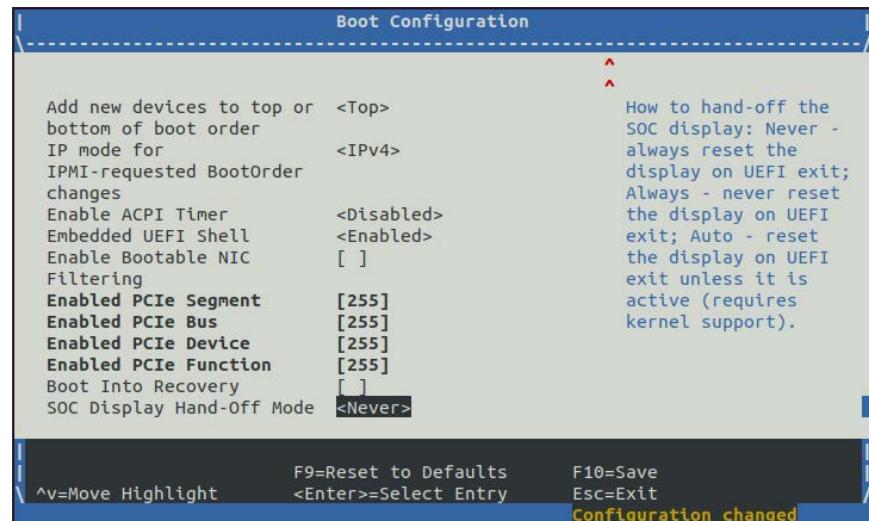
6. Boot Configurationメニューで、矢印キーで下に移動し(下に示す) SOC Display Hand-Off Modeを選択してEnterを押して下さい：



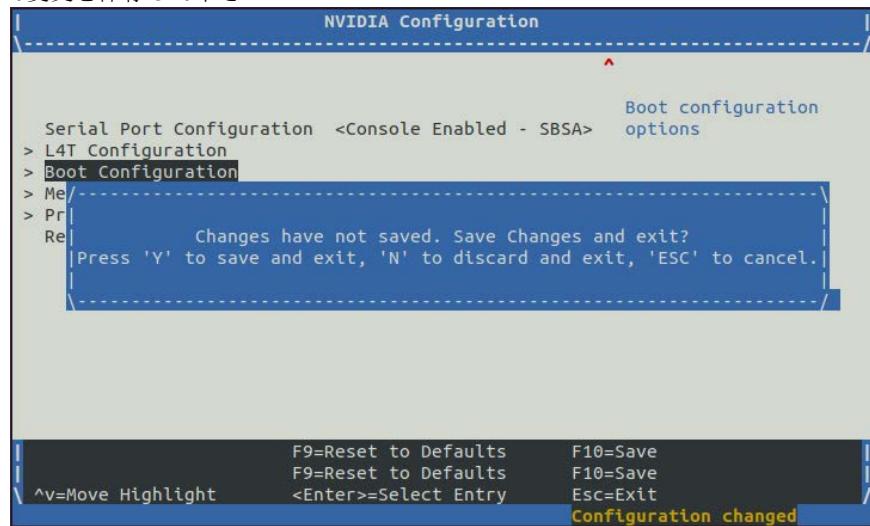
7. SOC Display Hand-Off Modeメニューで、矢印キーで下に移動し(下に示す) Neverを選択してEnterを押して下さい：



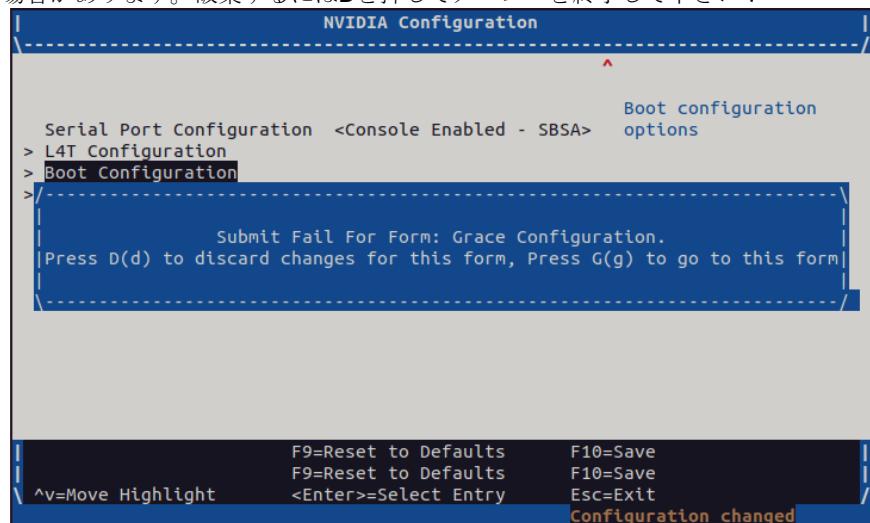
8. (下に示すように) Neverが選択されたことを確認し、ESCを押してメニューを終了して下さい：



9. ESCを2回押すと(下に示すように)変更を保存または破棄するかの入力が求められます、Yを押して変更を保存して下さい：



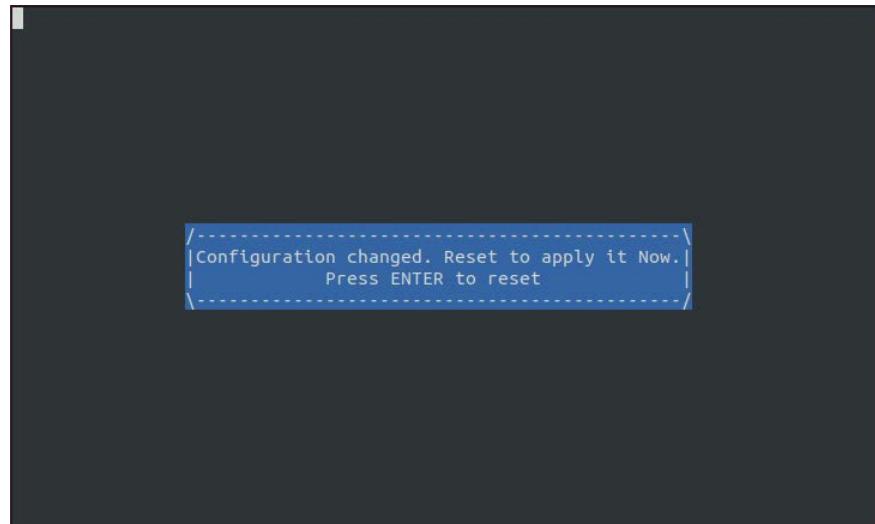
10. この時点でのGrace Configurationフォームへの変更に失敗したこと示す誤ったダイアログが現れる場合があります。破棄するにはDを押してメニューを終了して下さい：



11. 最上位のファームウェア・メニューへ戻るにはESCを再度押して矢印キーで下に移動し(下に示す) Continueを選択してEnterを押して下さい：



12. 最後に下に示すようにリセットすることを求められたらEnterを押してOrinを再起動して下さい：



3.8. OrinにRedHawkパッケージをコピー

NOTE

本項は、*Jetson AGX Orin, Orin NX, Orin Nano*用の*RedHawk 9.4 Gold*光学メディア・ディスクを利用するなどを前提としています。あるいはネットワークを介した*RedHawk*のインストールを望む場合、本書の最後にある付録A「Concurrent Real-Time リポジトリからRedHawkをインストール」を参照して下さい。インストールが終わったら、次の3.9項に進んでインストールを完了して下さい。

SDK ManagerがOrinへのUbuntuインストールを完了した後、*Jetson AGX Orin, Orin NX, Orin Nano*用の*RedHawk 9.4 Gold*光学メディア・ディスクをUbuntuホストの光ドライブへ挿入すると /media/\$USERディレクトリ(例： /media/jane/RedHawk-9.4-aarch64)以下に自動でマウントされるはずです。

マウントされたら、ディスクの全ての中身をOrinの/tmpディレクトリにコピーするため次のコマンドを実行して下さい。

```
$ scp -r /media/jane/Redhawk-9.4-aarch64 user:Orin-IP-Address:/tmp
```

NOTE

フラッシュ中に選択したユーザー名とパスワードを使用して下さい。

3.9. OrinにRedHawkパッケージをインストール

RedHawkパッケージをインストールするにはOrinに(ssh経由またはグラフィカル・コンソール経由のどちらかで)ログインして次のコマンドを実行して下さい：

```
$ cd /tmp/Redhawk-9.4-aarch64  
$ sudo ./install-redhawk
```

sudoコマンドが入力を指示したら現在のユーザーのパスワードを入力して下さい。

NOTE

パッケージがインストールされNVIDIAカーネル・モジュールがコンパイルされるので、インストールの完了に約10分かかります。

NOTE

以下を含むいくつかのエラーや警告がパッケージのインストール中に表示されます：

- *debconf: apt-utils* がインストールされていないのでパッケージ構成が遅延
- *dpkg: linux-libc-dev:arm64:* 依存関係の問題ですがともかく削除
- *Warning: fsck フック用root ファイルシステムの種類を確認できない*
- *update-alternatives: warning: 関連ファイルが存在しないため生成をスキップ*
- *Error: grub2構成ファイル「/boot/grub/grub.cfg」を読みない*
- *[Errno 2] 「/boot/grub/grub.cfg」 ファイルまたはディレクトリがない*

Jetsonプラットフォーム上のRedHawkの組み込みインストールには関係ありませんので、これらのメッセージは無視しても安全です。

この時点で、全てのRedHawkソフトウェアはOrinに正常にインストールされているはずですが、次項の追加手順はRedHawkカーネルを起動する前に必要となります。

3.10. RedHawkエントリをextlinux.confファイルに追加

RedHawkカーネル用の起動エントリを生成するためOrinで次の手順を実行して下さい：

1. /boot/extlinux/extlinux.confファイルのバックアップを生成して下さい。

```
$ cd /boot/extlinux  
$ sudo cp extlinux.conf extlinux.bak
```

2. お好みのエディタでextlinux.confファイルを開いてください。例、viを使う場合：

```
$ sudo vi extlinux.conf
```

3. 初期エントリの下にもう一つの同一エントリを生成するためprimary kernelエントリのテキスト行を全てコピーして下さい。例：

```
TIMEOUT 30  
DEFAULT primary  
  
MENU TITLE L4T boot options  
  
LABEL primary  
    MENU LABEL primary kernel  
    LINUX /boot/Image  
    INITRD /boot/initrd  
    APPEND ...many boot options...  
  
LABEL primary  
    MENU LABEL primary kernel  
    LINUX /boot/Image  
    INITRD /boot/initrd  
    APPEND ...many boot options...
```

4. 必要なRedHawkカーネルを指定するために最初のエントリをカスタマイズして下さい。例えば、RedHawkトレース・カーネル用のエントリを生成するにはLABEL, MENU LABEL, LINUX, INITRD, APPEND行を次のように変更して下さい：

```
LABEL redhawk-trace-graphics  
    MENU LABEL redhawk-trace kernel with graphics  
    LINUX /boot/vmlinuz-6.6.49-rt41-RedHawk-9.4-trace  
    INITRD /boot/initrd.img-6.6.49-rt41-RedHawk-9.4-trace
```

重要： 次のようにAPPEND行に対してnvidia_drm.fbdev=0と
nvidia_drm.modeset=0の起動オプションをRedHawkカーネルだけに忘れずに追加して
下さい：

```
APPEND ...many boot options... nvidia_drm.fbdev=0 nvidia_drm.modeset=0
```

NOTE

これらの追加の起動オプションはNVIDIA グラフィックスをRedHawk 9.4カーネルで動作させるために必要となります。これらのオプションは初期の起動メッセージを非表示にし、またランレベル3(グラフィック無しのマルチユーザー)への正常な遷移を妨げます。詳細および回避方法については4.1項の既知の問題を参照して下さい。

NOTE

最初のカーネル・エントリの変更は、最初のエントリのカスタマイズ中に誤字が入力された場合にNVIDIA カーネルはフォールバックとして起動されるようになります。本ガイドラインに従わないとOrinが起動できない状態になる、そして編集中に誤字が偶発的に取り込まれた場合は再フラッシュが必要となる可能性があります。

5. ファイルの先頭にあるDEFAULTエントリを変更してデフォルトのカーネルを新たに追加されたRedHawkカーネルに変更して下さい：

```
TIMEOUT 30
DEFAULT redhawk-trace-graphics
```

NOTE

Orinがシリアル・コンソールで設定された場合、本手順をスキップすることを選択して代わりにシリアル・コンソール上に起動メニューが表示された時に対話形式でどのカーネルを起動するかを決めることができます。

3.11. RedHawkカーネルのインストールを確認

再起動後、Orinは選択されたRedHawkカーネルを実行しているはずです。それを確認するには、次のコマンドを実行して下さい：

```
$ uname -r
```

次のように表示された出力が見れるはずです：

```
6.6.49-rt41-RedHawk-9.4-trace
```

インストール中に何かしらの問題に遭遇した場合、コンカレント日本の技術サポートサービス部(tech@concurrent-rt.co.jpまたは03-3864-5717)に連絡して下さい。

4. 既知の問題

特別な配慮が以下の分野で求められます。

4.1. NVIDIAフレーム・バッファの不一致と回避策

JetPack 6.1は、新しいLinuxカーネルと互換性のないフレーム・バッファを持つ古いNVIDIAグラフィックス・ドライバを提供します。NVIDIAはこの問題を認識しており、将来のJetPackアップデートでいずれこれを修正するはずですが、それまでは後述のとおりRedHawkは特別な回避策を必要とします。これらの回避策全てが、以前の3.7項に記載されたファームウェア設定でまずSOC Display Hand-Offモードを無効にする必要があります。

4.1.1. グラフィカル・モードで起動 (ランレベル5)

RedHawkでNVIDIAグラフィックスを動作させるには、下に示すようにextlinux.confエントリは追加の起動オプションnvidia_drm.fbdev=0とnvidia_drm.modeset=0が必要です：

```
LABEL redhawk-trace-graphics
    MENU LABEL redhawk-trace kernel with graphics
    LINUX /boot/vmlinuz-6.6.49-rt41-RedHawk-9.4-trace
    INITRD /boot/initrd.img-6.6.49-rt41-RedHawk-9.4-trace
    APPEND ...many boot options... nvidia_drm.fbdev=0 nvidia_drm.modeset=0
```

このグラフィカル・モードでは、最新のLinuxカーネルと互換性のないフレーム・バッファにテキストが表示されるためランレベル1またはランレベル3へ正常に遷移することが出来ません。

4.1.2. テキスト・モードで起動 (ランレベル3)

ランレベル3(グラフィックス無しのマルチユーザー)で正常に起動するには、下に示すようにextlinux.confエントリは追加の起動オプション「3」が必要です：

```
LABEL redhawk-trace-graphics
    MENU LABEL redhawk-trace kernel with graphics
    LINUX /boot/vmlinuz-6.6.49-rt41-RedHawk-9.4-trace
    INITRD /boot/initrd.img-6.6.49-rt41-RedHawk-9.4-trace
    APPEND ...many boot options... 3
```

このテキスト・モードでは、最新のLinuxカーネルと互換性のないフレーム・バッファにグラフィックスが表示されるためグラフィックスを起動するランレベル5へ正常に遷移することが出来ません。

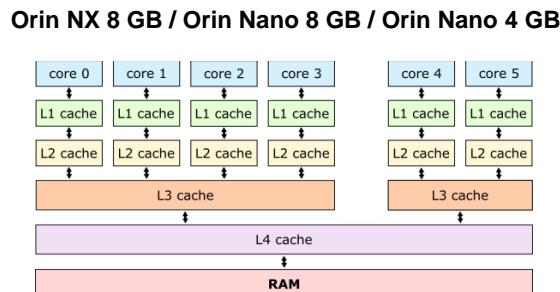
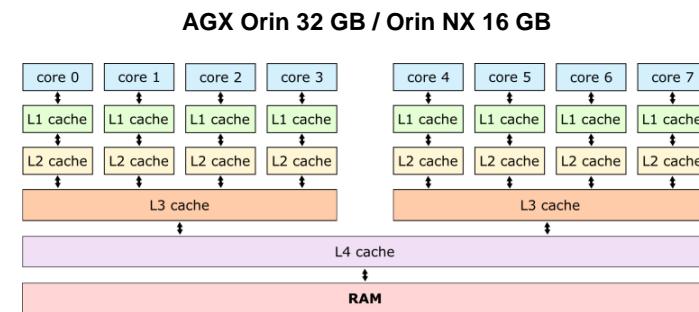
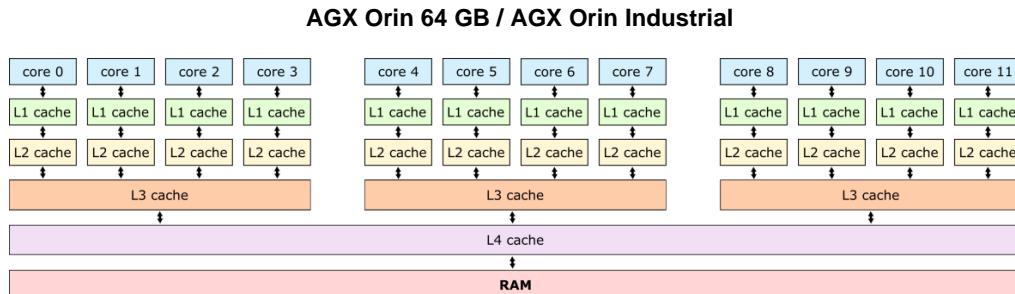
4.1.3. ハイブリッド・モードで起動 (全ランレベル、CUDA限定)

3つ目のハイブリッド・オプションは全ランレベル間での遷移をサポートしますが、本オプションではフレーム・バッファまたはグラフィックス表示を利用するCUDAアプリケーションは実行することは出来ません。しかしながら、非グラフィカルな作業のためにCUDAを利用するCUDAアプリケーションは本モードで正常に動作させることができますので、本モードは一部のアプリケーションには適している可能性があります。

本ハイブリッド・モードを正常に起動するには追加の起動オプションは必要なし、単に/etc/X11/xorg.confファイルを/etc/X11/xorg.conf.failsafeファイルのコピーと入れ替えて下さい。failsafeファイル内のXの構成は、グラフィックスがランレベルの遷移に沿ってRedHawkと連動することを可能にしますが、前述したCUDAの制限に影響します。

4.2. クラスタ内でのコアはシールドに干渉

全てのJetson AGX Orinシリーズ、Orin NXシリーズ、Orin Nanoシリーズのデバイスは次図で示すように4層のキャッシングを利用したクラスタの中にコアがグループ化されています：



各コアは自分のL1とL2キャッシングを持っていますが、L3キャッシングは同じクラスタ内で2個または4個が他のコアと常に共有しています。例えば、上の図全てにおいてコア0, 1, 2, 3は他のクラスタのL3キャッシングから分離されたL3キャッシングを常に共有しています。

これらの図のL4キャッシングは、GPUにとってこれはL3キャッシングですが、GPUもこのキャッシングを使用するために大抵は「システム・キャッシング」と呼ばれることに注意して下さい。

リアルタイム・アプリケーションのためにAGX Orin, Orin NX, Orin Nano上のコアをシールドする場合、クラスタ内の他のコアの動作が共有されたL3キャッシングを変更してリアルタイム・アプリケーションのデータミニズムや性能に影響を及ぼす可能性があります。最大限のリアルタイム性能を実現するには、同じクラスタ内の他の全てのコアも同様にシールドし、使用されていないコアはアイドルの状態にする必要があります。あるいは、同じクラスタ内の他のコアはリアルタイム性能を保護するためにオフラインにする一方で消費電力を減らすことも可能です。

4.3. クロック速度の最大化

NVIDIAはOrinデバイス上の個々のハードウェア・クロックの速度を制御するjetson_clocksユーティリティを提供します。本ユーティリティはいくつかのオプションを提供しており、rootユーザーとして実行する必要があります。現在のクロック速度を全て見るには--showオプションを付けてユーティリティを起動して下さい：

```
root@ubuntu:~# jetson_clocks --show 2>/dev/null
SOC family:tegra234 Machine:Jetson AGX Orin
Online CPUs: 0-11
cpu0: Online= Governor=schedutil MinFreq=729600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=1 c7=1
cpu1: Online= Governor=schedutil MinFreq=729600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=1 c7=1
cpu10: Online= Governor=schedutil MinFreq=729600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=1
c7=1
cpu11: Online= Governor=schedutil MinFreq=729600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=1
c7=1
cpu2: Online= Governor=schedutil MinFreq=729600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=1 c7=1
cpu3: Online= Governor=schedutil MinFreq=729600 MaxFreq=2201600 CurrentFreq=14208000 IdleStates: WFI=1 c7=1
cpu4: Online= Governor=schedutil MinFreq=729600 MaxFreq=2201600 CurrentFreq=17280000 IdleStates: WFI=1 c7=1
cpu5: Online= Governor=schedutil MinFreq=729600 MaxFreq=2201600 CurrentFreq=17280000 IdleStates: WFI=1 c7=1
cpu6: Online= Governor=schedutil MinFreq=729600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=1 c7=1
cpu7: Online= Governor=schedutil MinFreq=729600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=1 c7=1
cpu8: Online= Governor=schedutil MinFreq=729600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=1 c7=1
cpu9: Online= Governor=schedutil MinFreq=729600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=1 c7=1
GPU MinFreq=306000000 MaxFreq=1300500000 CurrentFreq=306000000
EMC MinFreq=204000000 MaxFreq=3199000000 CurrentFreq=2133000000 FreqOverride=0
DLA0_CORE: Online=1 MinFreq=0 MaxFreq=1600000000 CurrentFreq=1600000000
DLA0_FALCON: Online=1 MinFreq=0 MaxFreq=844800000 CurrentFreq=844800000
DLA1_CORE: Online=1 MinFreq=0 MaxFreq=1600000000 CurrentFreq=1600000000
DLA1_FALCON: Online=1 MinFreq=0 MaxFreq=844800000 CurrentFreq=844800000
PVA0_VPS0: Online=1 MinFreq=0 MaxFreq=1152000000 CurrentFreq=1152000000
PVA0_AXI: Online=1 MinFreq=0 MaxFreq=832000000 CurrentFreq=832000000
FAN Dynamic Speed control=active hwmon2_pwm=56
NV Power Mode: MAXN
root@ubuntu:~#
```

デフォルトで上記の一部の現在の周波数値は、ハードウェアでサポートされる最大周波数よりも著しく低いことに注意して下さい。

最高性能を得るには、オプションなしでjetson_clocksユーティリティを起動して全てのクロック速度を最大値に設定して下さい。全てのクロック速度が最大化されたことを確認するには、再度jetson_clocksユーティリティを--showオプションを付けて実行して下さい。例：

```
root@ubuntu:~# jetson_clocks
root@ubuntu:~# jetson_clocks --show 2>/dev/null
SOC family:tegra234 Machine:Jetson AGX Orin
Online CPUs: 0-11
cpu0: Online= Governor=schedutil MinFreq=2201600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=0 c7=0
cpu1: Online= Governor=schedutil MinFreq=2201600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=0 c7=0
cpu10: Online= Governor=schedutil MinFreq=2201600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=0 c7=0
cpu11: Online= Governor=schedutil MinFreq=2201600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=0 c7=0
cpu2: Online= Governor=schedutil MinFreq=2201600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=0 c7=0
cpu3: Online= Governor=schedutil MinFreq=2201600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=0 c7=0
cpu4: Online= Governor=schedutil MinFreq=2201600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=0 c7=0
cpu5: Online= Governor=schedutil MinFreq=2201600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=0 c7=0
cpu6: Online= Governor=schedutil MinFreq=2201600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=0 c7=0
cpu7: Online= Governor=schedutil MinFreq=2201600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=0 c7=0
cpu8: Online= Governor=schedutil MinFreq=2201600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=0 c7=0
cpu9: Online= Governor=schedutil MinFreq=2201600 MaxFreq=2201600 CurrentFreq=2201600 IdleStates: WFI=0 c7=0
GPU MinFreq=1300500000 MaxFreq=1300500000 CurrentFreq=1300500000
EMC MinFreq=204000000 MaxFreq=3199000000 CurrentFreq=3199000000 FreqOverride=1
DLA0_CORE: Online=1 MinFreq=0 MaxFreq=1600000000 CurrentFreq=1600000000
DLA0_FALCON: Online=1 MinFreq=0 MaxFreq=844800000 CurrentFreq=844800000
DLA1_CORE: Online=1 MinFreq=0 MaxFreq=1600000000 CurrentFreq=1600000000
DLA1_FALCON: Online=1 MinFreq=0 MaxFreq=844800000 CurrentFreq=844800000
PVA0_VPS0: Online=1 MinFreq=0 MaxFreq=1152000000 CurrentFreq=1152000000
PVA0_AXI: Online=1 MinFreq=0 MaxFreq=832000000 CurrentFreq=832000000
FAN Dynamic Speed control=active hwmon2_pwm=56
NV Power Mode: MAXN
root@ubuntu:~#
```

上記の現在の周波数値全てが最大速度に構成されたことに注意して下さい。

NOTE

nvpmodelとは異なり、jetson_clocksによるクロック速度の変更は再起動を超えて持続しません。従って、確実に全てのクロック速度を最大化するには再起動後も常にjetson_clocksを実行する必要があります。

4.4. 高負荷時にファンを最大化

デフォルトで、OrinのCPU冷却ファンの速度は常に適切な冷却を確保するために必要に応じて動的に調整されます。このデフォルトの動的なファン動作は、大抵の使用事例において十分過ぎるはずです。

しかしながら、長期に渡るベンチマークや負荷試験の期間中においては、急速な温度上昇を抑えるため試験開始前にCPU冷却速度を最大化することをConcurrentは推奨します。急速に上昇する温度は結果的にCPU速度を減速してしまう可能性があり、それはベンチマークにネガティブな影響を及ぼす可能性があります。

CPU冷却ファンの速度を最大化するには--fanオプションを付けてjetson_clocksを起動して下さい。

4.5. エントロピー収集デーモンがリアルタイムに干渉

Ubuntu 22.04は、エントロピーを収集し乱数発生を改善するために各種ハードウェア・リソース(例えば、キャッシング、分岐予測、メモリ変換テーブル)を継続的にサンプリングするHardware Volatile Entropy Gathering daemon(haveged)が有効です。

このデーモンの絶え間ないハードウェア調査活動は、シールドされたCPUコア上であってもリアルタイム性能にネガティブな影響を与える可能性がありますので、リアルタイムの作業を開始する前にrootユーザーで次のコマンドを実行してデーモンを停止する必要があります：

```
# systemctl stop haveged.service
```

各種Ubuntuの起動サービスは起動前に最小限レベルのエントロピーを収集する必要がありますので、エントロピー収集サービスは無効にしないで下さい。これはXサーバーを正常に初期化するために必要なサービスを含んでいます。しかしながら、Orinがグラフィックス表示を開始した後はリアルタイム性能を改善するために安全にサービスを停止することができます。

4.6. APTアップグレードはデフォルトのカーネルを変更

APTパッケージのアップグレードを実行するとデフォルトのカーネルとして非RedHawkカーネルが選択される可能性があります。APTパッケージのアップグレードを実行した後は /boot/extlinux/extlinux.confファイルを必ず再確認して下さい。

4.7. L4Tカーネル実行中にAPTアップグレードを実行

APTパッケージのアップグレードを実行する前にL4Tカーネル(非RedHawk)で再起動することを推奨します。RedHawkカーネル起動中にAPTパッケージのアップグレードを試みる場合、非L4Tカーネルによって混乱する不安定なインストール後スクリプトを実行する様々なNVIDIA提供のパッケージ(例、bootloader)が原因で失敗する可能性があります。

4.8. RedHawkカーネル起動中に無限ループ

前述の3.7項で記載されているファームウェア設定の変更、それに加え前述の3.10項に記載されている起動オプションがRedHawkカーネルの起動に確実に必要となります。ファームウェア設定の変更を怠ると起動中にRedHawkカーネルの無限ループに陥り、Jetsonがクラッシュしたように見えるようになります。一方で、次のようなメッセージがシリアル・コンソール上に断続的に表示されます：

```

[ 9.852776] CPU:0, Error: dce-fabric@0xde00000, irq=32
[ 9.859844] ****
[ 9.864784] CPU:0, Error:dce-fabric, Errmon:2
[ 9.869273] CBB registers returning all 1's which is invalid
[ 9.875141] CPU:0, Error: cbb-fabric@0x13a00000, irq=33
[ 9.880527] ****
...
[ 9.968482] WARNING: CPU: 0 PID: 0 at drivers/soc/tegra/cbb/tegra234-cbb.c:
[ 9.968573] CPU: 0 PID: 0 Comm: swapper/0 Tainted: G O
[ 9.968576] Hardware name: NVIDIA NVIDIA Jetson AGX Orin Developer Kit

```

これが発生した場合、Jetsonの電源を切断して3.7項及び3.10項に記載されている必要な変更を行ってください。

4.9. OrinプラットフォームでのNightViewの既知の問題

本書の公開時点で、次を含む現在調査中のOrinプラットフォームをサポートするNightViewについていくつかの既知の問題があります：

- CUDAを使用するプログラムをデバッグするとCUDAの初期化中に時々セグメンテーション違反(SIGSEGV)を引き起こす場合があります。
- CUDAを使用するプログラムでシングル・ステップ実行を行うと時々セグメンテーション違反(SIGSEGV)を引き起こす場合があります。

これらの問題は積極的に調査を行っており、NightViewの将来のリリースでは解決されているはずです。

4.10. カスタム・カーネルを構築するための新たな手順

RedHawk 9.4は、以前のRedHawkカーネルとは大幅に異なる新しいカスタム・カーネルを構築するための手順を利用します。Orinプラットフォーム用にカスタム・カーネルを構築するには次の手順に従って下さい。

NOTE

カスタム・カーネルの構築は相当量のストレージ空間が必要になりますので、より大きなNVME ドライブにインストールされたOrinを使用することを推奨します。

例えば、カーネル・トレース機能を無効にしてカスタム・カーネルを構築するには：

1. カーネルの構築に必要な追加のパッケージをインストールして下さい：

```
$ sudo apt install libssl-dev debhelper gawk qtbase5-dev
```

2. 標準カーネルをベースとするカスタム・カーネル構成を生成して下さい：

```
$ cd /usr/src/linux-6.6.49-RedHawk-9.4
$ sudo cp configs/arm64/standard .config
```

3. オプションでカーネル構成に更なるカスタマイズを行ってください：

```
$ sudo make xconfig
```

もしくはテキスト・モードを利用して下さい：

```
$ sudo make menuconfig
```

4. カスタム・カーネルを構築し、それをDebファイルにパッケージ化して下さい：

```
$ sudo make deb-pkg
```

NOTE

次を含む警告がカーネル構築中に表示される可能性があります：

- *warning: building a source package without cleaning*
- *warning: native package version may not have a revision*
- *warning: missing information for output field Standards-Version*

これらや他の警告は関連がありませんので、無視しても差し支えありません。

5. 新たに構築されたカスタム・カーネルのパッケージをインストールして下さい：

```
$ sudo apt install ./debian-pkgs/*custom_9.4*.deb
```

6. 重要：カスタム・カーネル向けNVIDIAのツリー外ドライバを全て構築しインストールして下さい：

```
$ sudo dkms install ccu-nvidia-oot/36.4.0 -k 6.6.49-rt41-RedHawk-9.4-custom
```

7. NVIDIAのツリー外ドライバを含めるためカスタム・カーネルのinitramfsを更新して下さい：

```
$ sudo update-initramfs -u -k 6.6.49-rt41-RedHawk-9.4-custom
```

カスタム・カーネルを構築しインストールした後、カスタム・カーネルのエントリを追加するためにextlinux.confファイルを編集する必要があります。例：

```
LABEL redhawk-custom
    MENU LABEL redhawk-custom
    LINUX /boot/vmlinuz-6.6.49-rt41-RedHawk-9.4-custom
    INITRD /boot/initrd.img-6.6.49-rt41-RedHawk-9.4-custom
    APPEND ...many boot options... nvidia_drm.fbdev=0 nvidia_drm.modeset=0
```

エントリを追加したら、extlinux.confのDEFAULTをredhawk-customに変更するか、起動メニューの初期に対話的にカスタム・カーネルを選択することができます。

A. Concurrent Real-TimeリポジトリからRedHawkをインストール

RedHawkはConcurrent Real-Timeのソフトウェア・リポジトリを利用してネットワークを介してインストールすることができます。これはリリースされた最新のアップデートがインストールされる利点がありますが、インストールを実行するにはいくつか事前の設定が必要になります。その設定を完了するには次の手順に従って下さい。

A.1. Concurrent Real-Time公開鍵のインストール

Concurrent Real-TimeはGPGで生成されたDSA鍵およびEIGamal鍵のペアを使ってAPTリポジトリに署名します。APTシステムがリポジトリの利用を許可できるようにConcurrent Real-Timeの公開鍵をインストールする必要があります。

ccur-public-keysファイルをダウンロードし、必要に応じてAPTの鍵リングにその鍵ファイルをインポートすることが可能です。Concurrent Real-Timeのリポジトリからソフトウェアにアクセスする前に鍵ファイルをインポートする必要があります。

次のコマンドで公開鍵をダウンロードし、インストールします。

```
wget -q http://redhawk.concurrent-rt.com/network/ccur-public-keys  
apt-key add ccur-public-keys
```

Concurrent Real-TimeのWebサイトからダウンロードした公開鍵の信頼性を確認したい場合は、Concurrentのソフトウェア・サポートから鍵のフィンガープリントを要求し、ダウンロードしたキーのフィンガープリントをそれらと比較して下さい。

次のコマンドを使ってダウンロードした鍵のフィンガープリントを取得することができます。

```
gpg --with-fingerprint ccur-public-keys
```

A.2. リポジトリ定義ファイルのインストール

リポジトリ定義ファイルはAPTインストールとシステムの更新でどのように特定のリポジトリにアクセスするのかを指示します。定義ファイルは/etc/apt/sources.list.d/ディレクトリの中に存在し、ファイル名称が.listで終わる(例、/etc/apt/sources.list.d/ccur.list)必要があります。

例えば、RedHawk 9.4 Gold Jetson AGX Orin製品のリポジトリ定義のために次の形式を使用して下さい：

```
deb [arch=arm64] http://redhawk.concurrent-rt.com/ubuntu/login/passwd/redhawk/orin 9.4.n core
```

例えば、次のエントリでは架空のログインとパスワードに置き換えておきます：

```
deb [arch=arm64] http://redhawk.concurrent-rt.com/ubuntu/L12345/xyzzy/redhawk/orin 9.4.n core
```

このマシーン用にNightStarツールを購入している場合、先に追加した行の下にNightStarツール用のリポジトリ定義も追加する必要があることに注意して下さい。例：

```
deb [arch=arm64] http://redhawk.concurrent-rt.com/ubuntu/L12345/xyzzy/nightstar 5.2 rt ubu22
```

リポジトリ定義ファイルを定位置に置いたらリポジトリの設定は完了です。ソフトウェアのリポジトリが正しく追加されたことを確認するにはrootで次のコマンドを実行して下さい：

```
sudo apt update
```

新たに追加したccur.listファイルに言及するエラーまたは警告が生成された場合、前述の全ての手順が記載されているとおりに全て完了していることを確認して下さい。

A.3. RedHawkパッケージのインストール

設定が完了したら、現在のシステムに最新のRedHawk 9.4パッケージをインストールするため、単に次のコマンドだけを実行して下さい：

```
apt install 'ccur-*'
```

このコマンドが完了したらRedHawkパッケージはインストールされます。RedHawkのインストールを終了するには、3.10項「RedHawkエントリをextlinux.confファイルに追加」に進んで下さい。

A.4. サポート

ご支援が必要な場合、コンカレント日本の技術サポートサービス部(03-3864-5717)まで連絡して下さい。技術サポートサービス部の営業時間は平日の9時から17時半となります。

コンカレント日本のWebサイト「<https://www.concurrent-rt.co.jp/company/>」を利用または「tech@concurrent-rt.co.jp」へ電子メールを送信することでいつでも支援のリクエストを申し込むことも可能です。