

Toradex TEZI使用マニュアル BSP5用

本マニュアルは岡本無線電機株式会社が独自作成したものでありメーカーが保証した内容ではありません。万が一本マニュアルに間違いがあり、事故が生じたとしても岡本無線電機株式会社は一切の責任を問われないものとさせていただきます。

本マニュアルについて

本マニュアルはトラデックスのCPUモジュールにOSイメージを書き込むために使用するTEZI(Toradex Easy Installer)の使用方法について記述しています。

本マニュアルに記載されていないTEZIに関する情報も多々あります。詳しくはToradexのWEBサイトをご参照ください。

参考:

<https://developer.toradex.com/easy-installer>

https://developer-archives.toradex.com/knowledge-base/load-toradex-easy-installer#Download_Toradex_Easy_Installer

1. 実行環境

本マニュアルの実行環境は下記です。

OS: Windows 10 21H2

TEZI: 5.7.1+build.13

CPUモジュール: Verdin iMX8M Plus Quad 4GB Wi-Fi / Bluetooth IT V1.1A

キャリアボード: キャリアボード: Verdin開発ボード Rev 1.1C + アクセサリーキット

BSP: Linux Reference Minimal 5 LTS Downstream SoC vendor based kernel Wayland/XWayland graphics back-end supported but not included 5.7.0+build.20

(LTSとついていますがVerdin-iMX8M Plusに対してはLTS版ではありません。)

本マニュアルとは異なるモジュールや評価ボード以外のキャリアボードを使われても大雑把には同じ操作となります。

2. 注意点

TEZIはパスワードなしでシリアルコンソールを使用できますまたセキュリティ皆無のVNCが入っています。本番環境では安全なネットワークに接続してください。

TEZIは多くのオープンソースで作成されています。TEZIファイルシステムの/usr/share/common-licenses/配下にあるライセンス規約をご参照ください。/usr/share/common-licenses/license.manifestにライセンスの概要があります。

開発環境(パソコン)と実行環境(モジュール)の違いをわかりやすくするためにコマンドの表記の前に下記をつけています。

開発環境(パソコン)上で入力するコマンド:[Ubuntu]\$

実行環境(モジュール)上で入力するコマンド(Linux) : [Module]#

実行環境(モジュール)上で入力するコマンド(U-Boot): [U-Boot]#

実行環境(モジュール)上で入力するコマンド(TEZI): [Tezi]#

TEZIについて

TEZI(Toradex Easy Installer)はToradexのSOMの内部ROMにOSイメージを書き込むためのツールです。TEZIはLinux + アプリケーションで出来ています。SOM上で動作するものですのでパソコンなどがなくてもモジュール単体で書き込むことが可能です。SDやUSBメモリなどに格納したOSイメージや任意のサーバーからダウンロードしたものを書き込むこともできます。

TEZIが起動するとGUIが立ち上がり数クリックするだけで書き込みできるため誰でも利用することができます。設定次第では電源を入れるだけで書き込みを自動で行うことも可能です。

SOMは出荷時にTEZIが書き込まれます。そのため量産工程で出荷時にインストールされているTEZIを利用して最終製品用のOSイメージを書き込みを行うこともできます。

The screenshot displays the TEZI GUI interface. At the top, there is a menu bar with icons and labels: Install (i), Refresh (r), Feeds (f), RNDIS (n), Erase Flash (c), Exit (Esc), and License. Below the menu is a list of OS images, each with an icon, a title, version, date, and URL, and a download icon on the right. The images listed are:

- Toradex Embedded Linux Demo with LXDE (Stable) 2.7, 2018-01-04, http://tezi.toradex.com/Colibri-iMX7_LXDE-Image_2.7/
- Toradex Embedded Linux Demo with LXDE 2.8b2.97, 2018-03-31, http://tezi.toradex.com/Colibri-iMX7_LXDE-Image-Tezi_2.8b2.97/
- Toradex Windows Embedded Compact 7 1.1, 2018-05-14, http://tezi.toradex.com/Colibri-iMX7_WinCE7Image-Tezi_1.1-20180514/
- Toradex Windows Embedded Compact 2013 1.1, 2018-05-14, http://tezi.toradex.com/Colibri-iMX7_WinCE8Image-Tezi_1.1-20180514/
- Toradex Windows Embedded Compact 7 Fast boot demo 1.1b4, 2018-03-22, http://tezi.toradex.com/Colibri-iMX7_WinCE7Image-FastBoot-Tezi_1.1b4...
- Toradex Easy Installer 1.5, 20180507, http://tezi.toradex.com/Colibri-iMX7_ToradexEasyInstaller_1.5-20180507/

At the bottom, there are three panels:

- Network Information:**

Ethernet	10.22.1.67/24
USB RNDIS	192.168.11.1/24
- Module Version:**

Product	Colibri iMX7 Dual 512MB
Version	V1.1C
Serial	02911038
- Flash Space:**

Required	Unknown
Capacity	512 MB

Toradex Easy Installer 1.5 (gf4e21de) - Built: May 7 2018

TEZIの起動

TEZIを起動するには3つの方法があります。

1. TEZIがインストールされているモジュールを起動する

購入直後はTEZIがインストールされているため開発ボードに電源を入れるだけでTEZIが起動します。

ただしTEZIでOSイメージを書き込むと内部のTEZIは消えてしまいますのでそれ以降はTEZIは起動せず書き込んだものが起動します。主にモジュール購入直後の評価開始のタイミングや量産時の初期書き込みに利用します。

2. リカバリーモードで起動

リカバリーモードは通常起動とは別に内部ROMを使用せずに直接外部からRAMへプログラムを展開して起動するモードです。主にブートローダーが潰れるなどモジュールが復帰できなくなった時に利用します。

3. 外部メディアなどからの起動

通常はモジュール内蔵のeMMCに格納されているLinuxを起動しますがSDカードやUSBメモリー、Ethernetなど別の領域に格納されたTEZIを起動させることも可能です。

開発時に何度も書き直す場合にはこの方法が一番使いやすいです。

起動イメージ

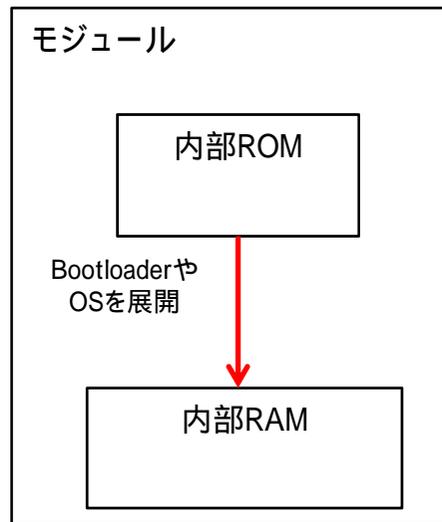
3つの起動方法の起動イメージです。

リカバリーモードは内部ROMを使わずにパソコンから直接RAMへBootloaderやOSを展開します。

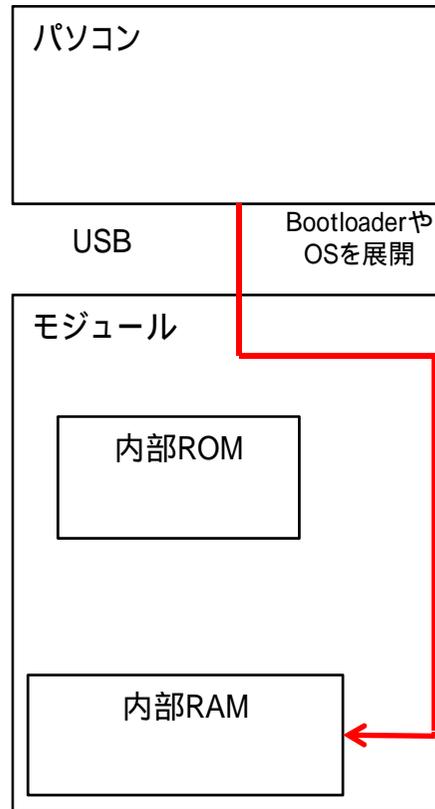
内部ROMの状態にかかわらず常に同じ状態で起動できます。

外部メディアから起動する場合は最初に内部ROMのBootloaderが起動して外部メディアのOS起動するためそれぞれ互換性がない場合うまく動作しません。

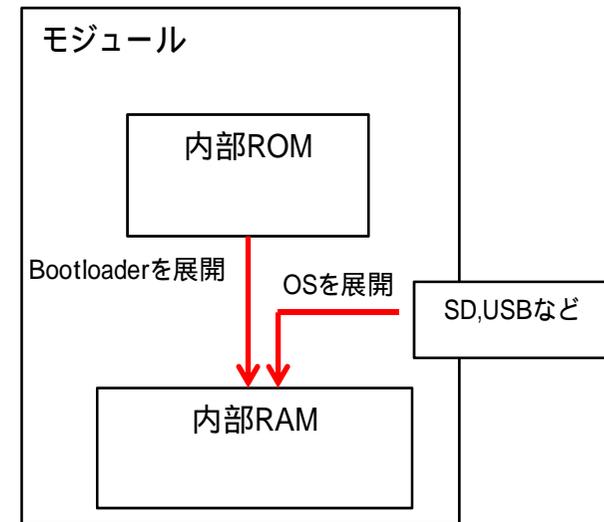
通常の起動



リカバリーモードで起動



外部メディアからの起動



TEZIとOSイメージの入手

OSを書き込むにはTEZIとOSイメージ(TEZIフォーマットタイプ)が必要です。

・TEZIの入手

TEZIは下記からダウンロードできます。対象のモジュールのファイルをダウンロードしてください。

https://developer-archives.toradex.com/knowledge-base/load-toradex-easy-installer#Download_Toradex_Easy_Installer

・OSイメージの入手

下記にデモ用のOSイメージがあります。書き込みたいOSイメージをSDカードなどに格納してください。

<https://developer.toradex.com/knowledge-base/toradex-easy-installer-os-and-demo-images>

OSイメージをToradexのサーバーから書き込む場合はダウンロード不要です。書き込み時に自動でダウンロードされます。自作カスタムOSを書き込む場合は作成したOSイメージを使用してください。

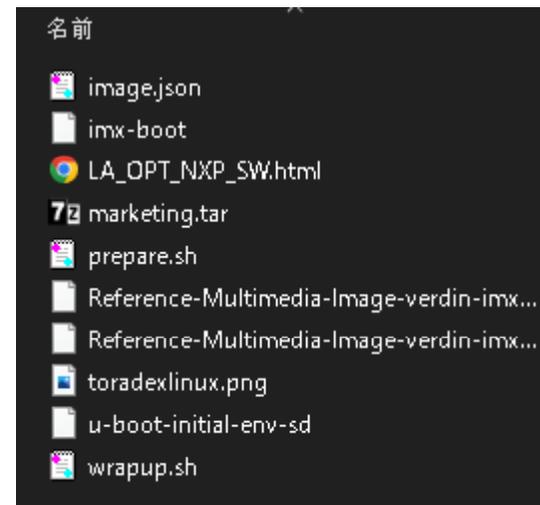
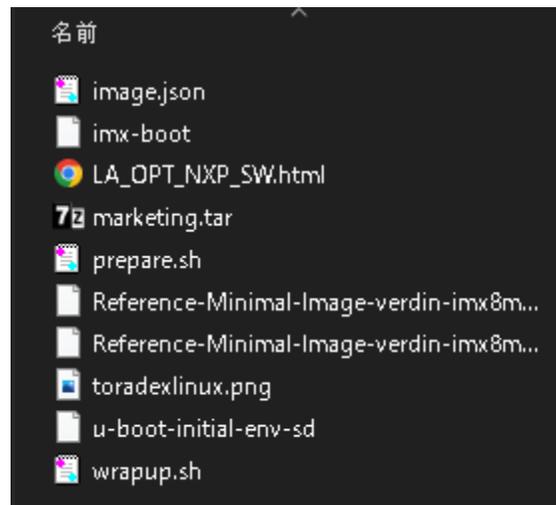
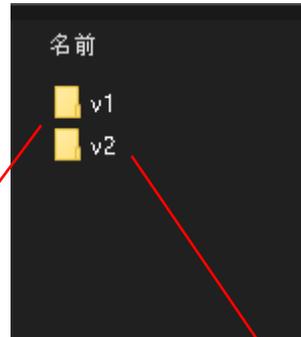
リカバリーモードでTEZIを起動する方法

OSイメージ解凍後をFAT32(もしくはEXT4)フォーマットのSDカードやUSBメモリに格納してください。

ルートディレクトリに任意の名前でディレクトリを作成してその中に格納してください。ディレクトリを分けて複数のOSイメージを格納することもできます。例えばSDカードがFドライブであれば F:¥v1、F:¥v2配下など。

ToradexのサーバーからOSイメージを書き込む場合はSDカードは準備不要です。

それぞれのディレクトリにOS
イメージを格納



ハードウェアの準備

以下を行います。

X1にモジュールを挿します。ヒートシンクも装着します。

X37にHDMIケーブルを指してディスプレイと繋がます。TightVNC Viewerなどで代用できます。

X53にUSBマウスを挿入します。

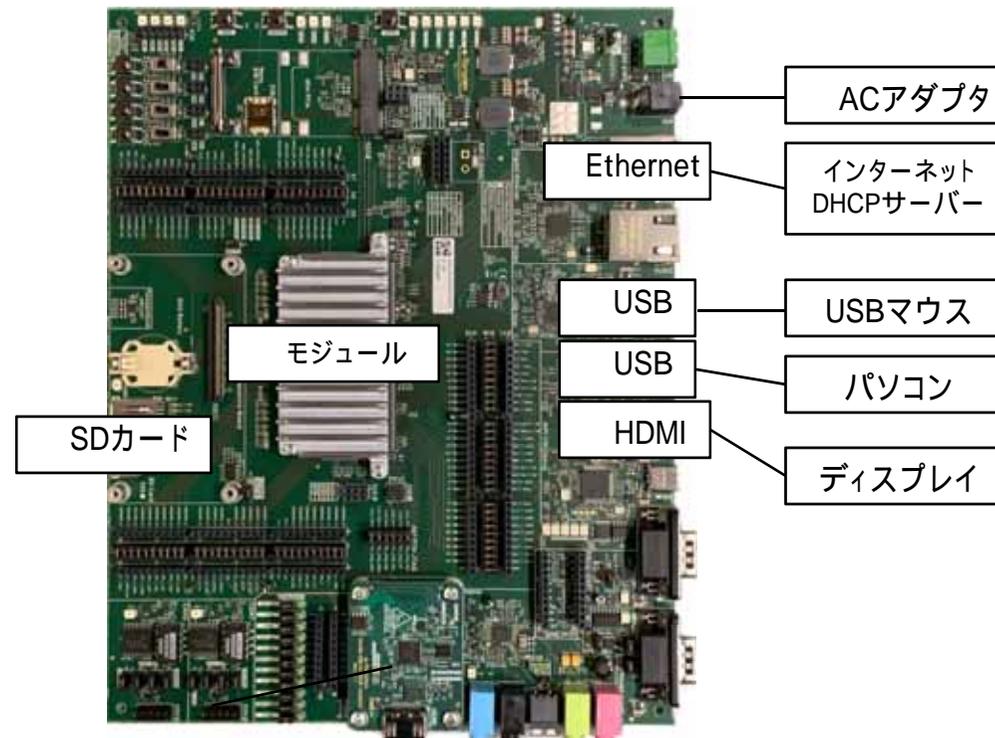
X34にUSBケーブルを差し込みパソコンと接続します。

SDカードでアップデートする場合はX55にSDカードを挿入します。

アップデートをインターネット越しで行う場合はX25にEthernetケーブルを挿入します。

インターネットに接続できる環境が必要です。モジュールはDHCPで接続を行いますのでDHCPサーバーが必要です。

X58にACアダプタを接続します。



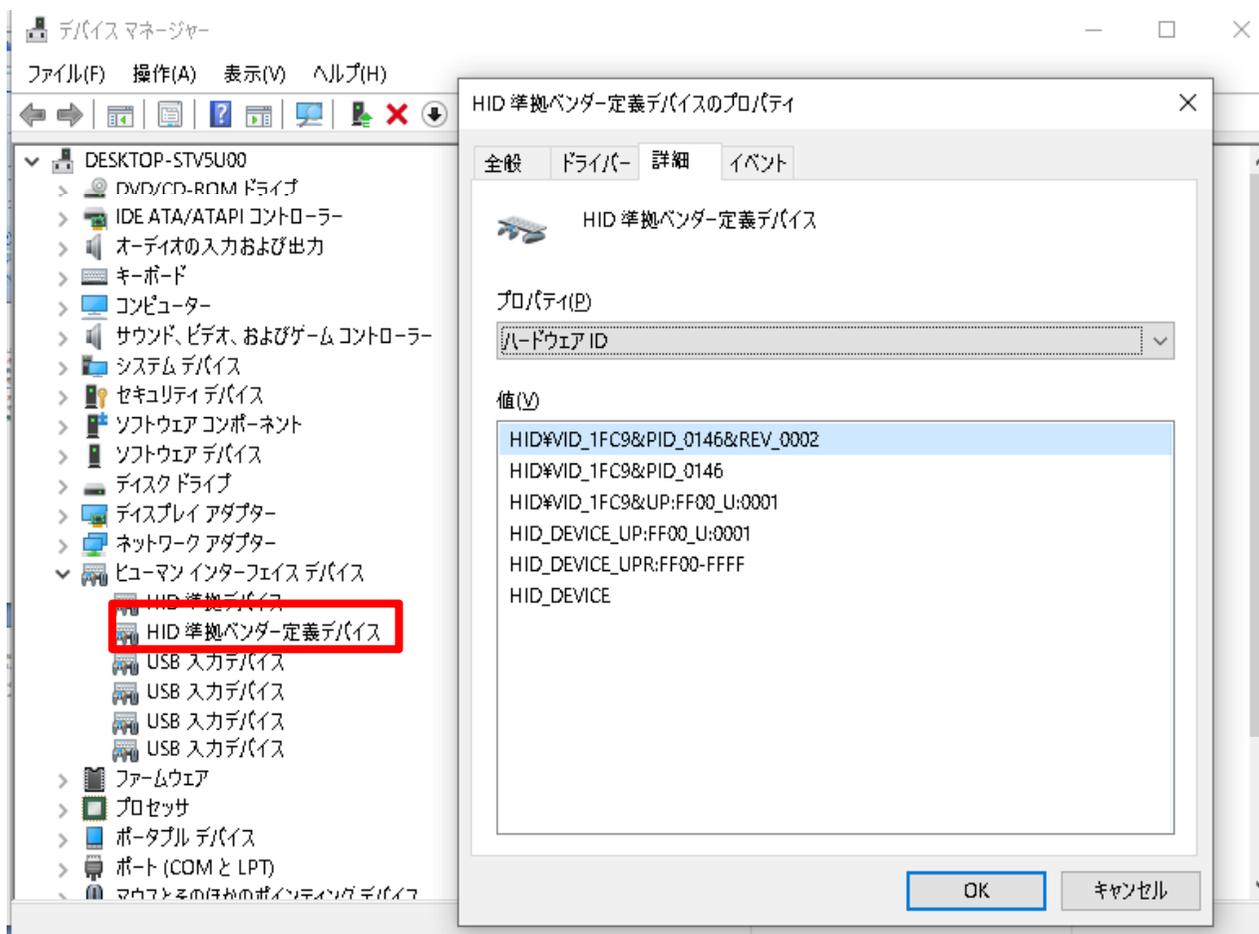
リカバリーモードで起動

Recovery Modeボタン(SW3)を押しながら電源ボタン(SW2)を押してください。
電源ボタンを離した後少しおいてRecovery Modeボタンを離してください。リカバリーモードで起動します。

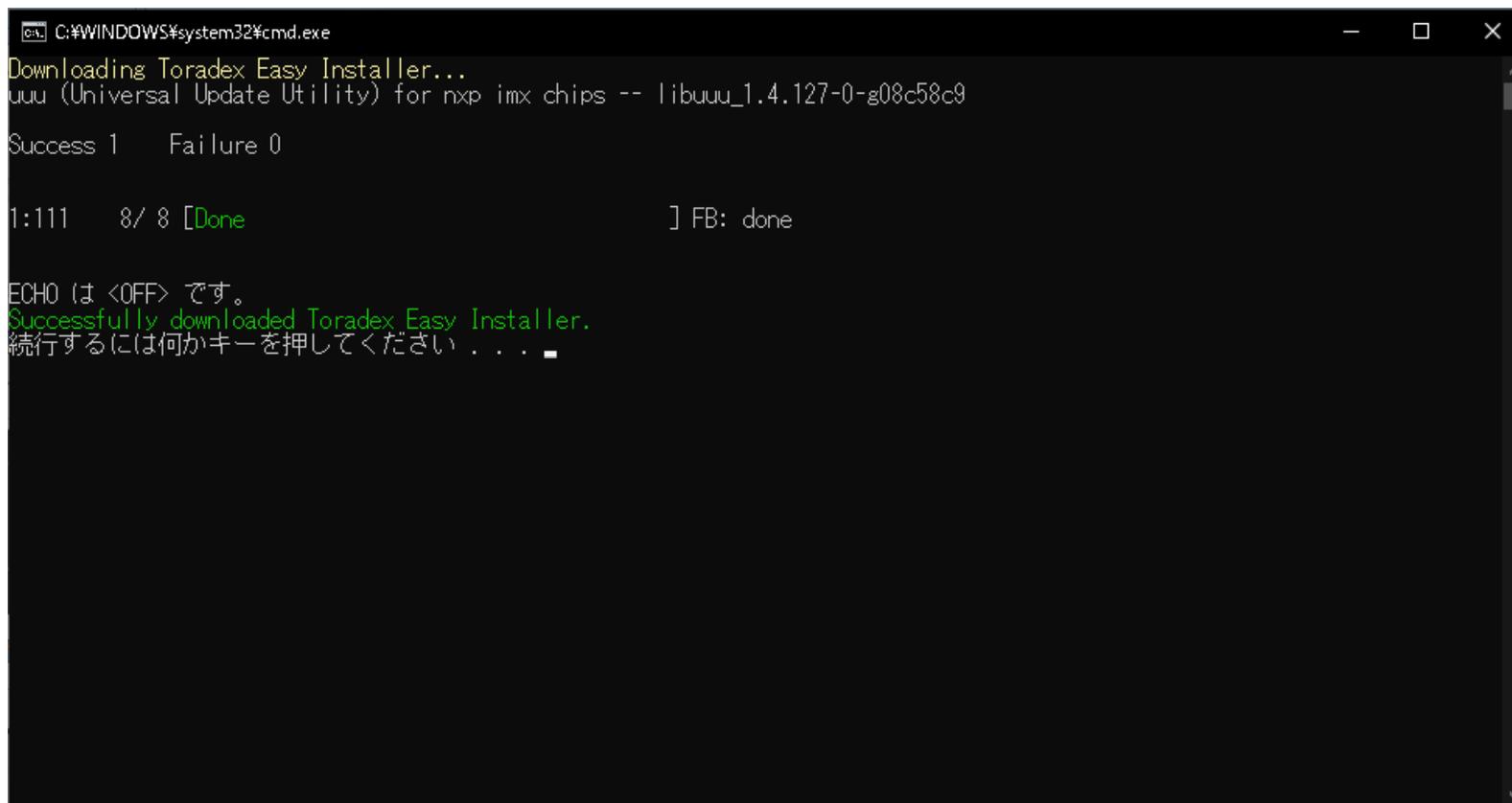
Verdin-iMX8M Plus以外のモジュールのリカバリーモードへの遷移方法は下記をご参照ください。
https://developer.toradex.com/knowledge-base/imx-recovery-mode#Enter_Recovery_Mode



リカバリーモードで起動するとブートローダーが動作しないためteratermなどに起動ログが出力されません。
リカバリモードに入ったかどうかを確認する場合はWindows PCのデバイスマネージャを見ます。
下記のようなベンダーID,プロダクトIDのHID準拠デバイスが追加されています。



リカバリーモードで起動したらTEZIを起動します。ダウンロードしたTEZIの中にあるrecovery-windows.batを実行します。下記のようなログが出るとTEZIの起動成功です。

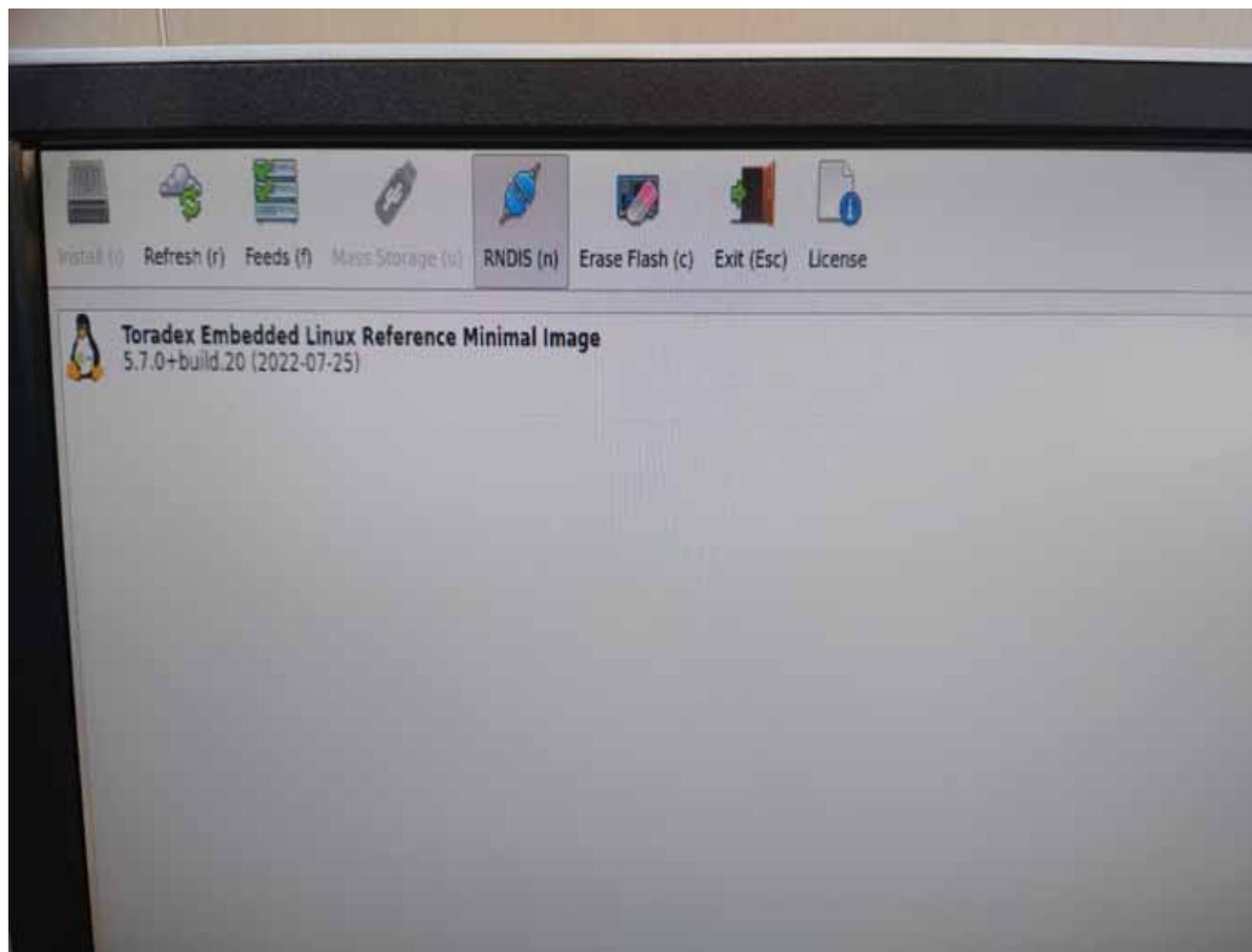


```
C:\WINDOWS\system32\cmd.exe
Downloading Toradex Easy Installer...
uuu (Universal Update Utility) for nxp imx chips -- libuuu_1.4.127-0-g08c58c9
Success 1 Failure 0

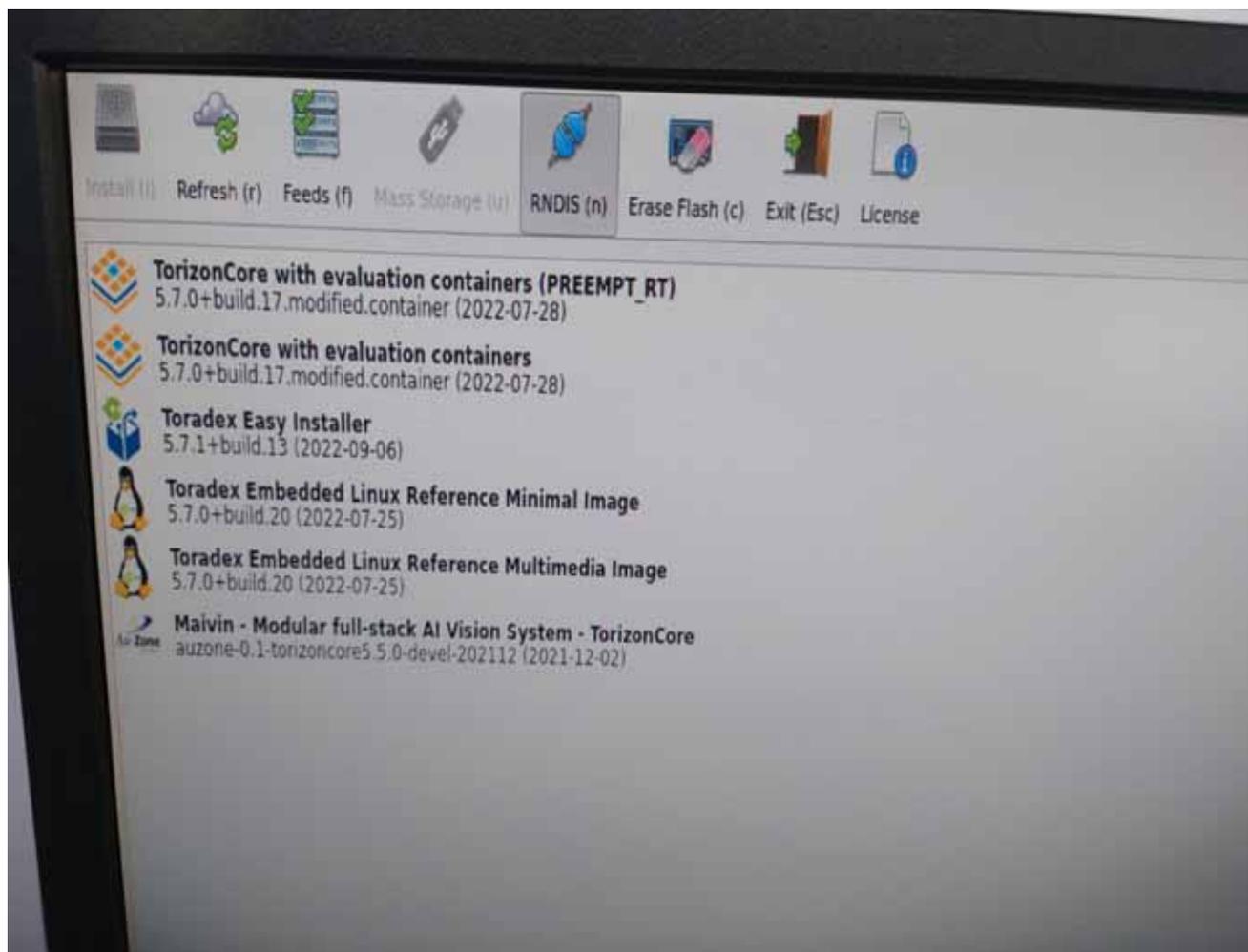
1:111 8/8 [Done] ] FB: done

ECHO は <OFF> です。
Successfully downloaded Toradex Easy Installer.
続行するには何かキーを押してください . . . .
```

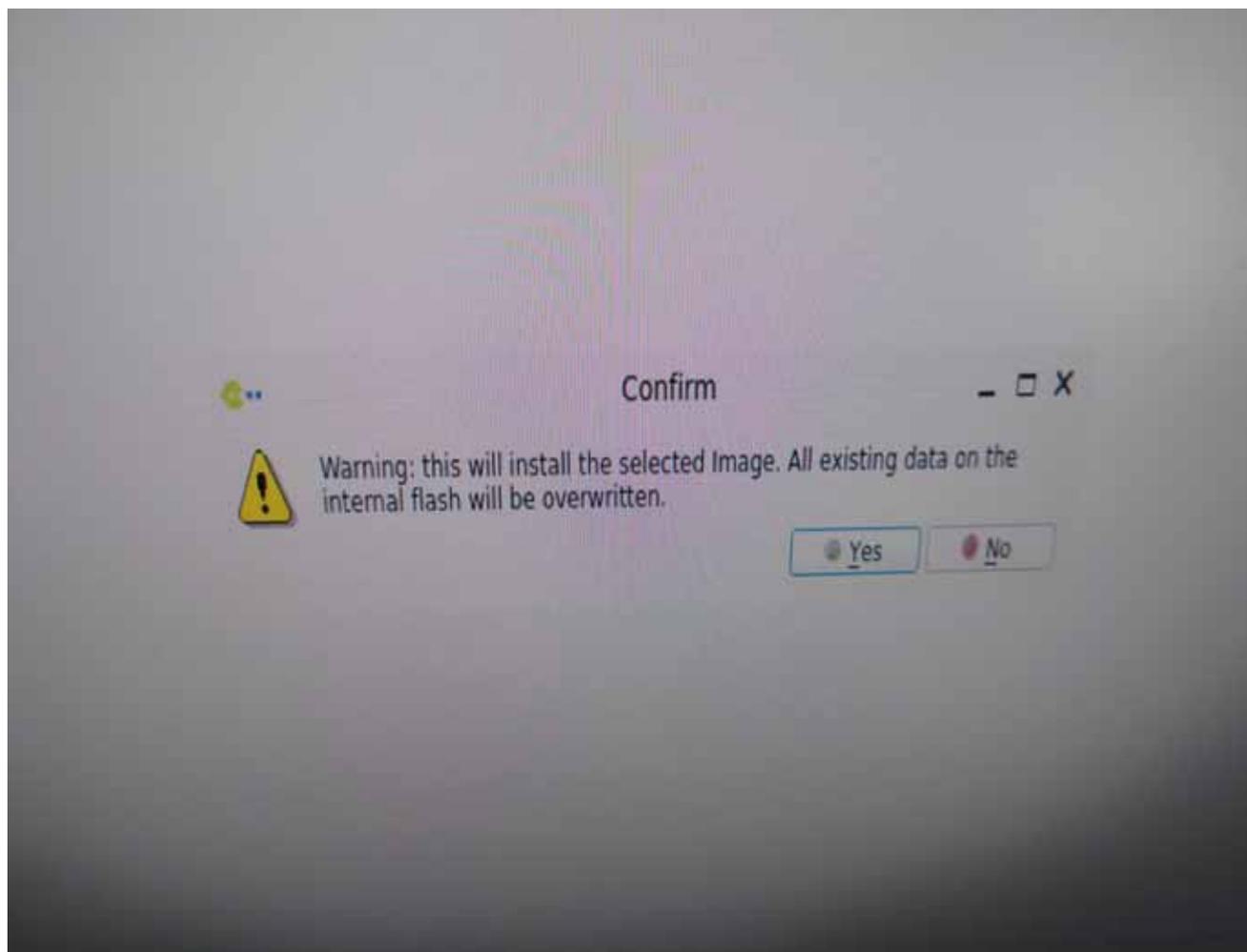
ディスプレイに下記のような画面が表示されます。SDカードに入れたOSイメージがリストされています。複数入れている場合は複数表示されます。



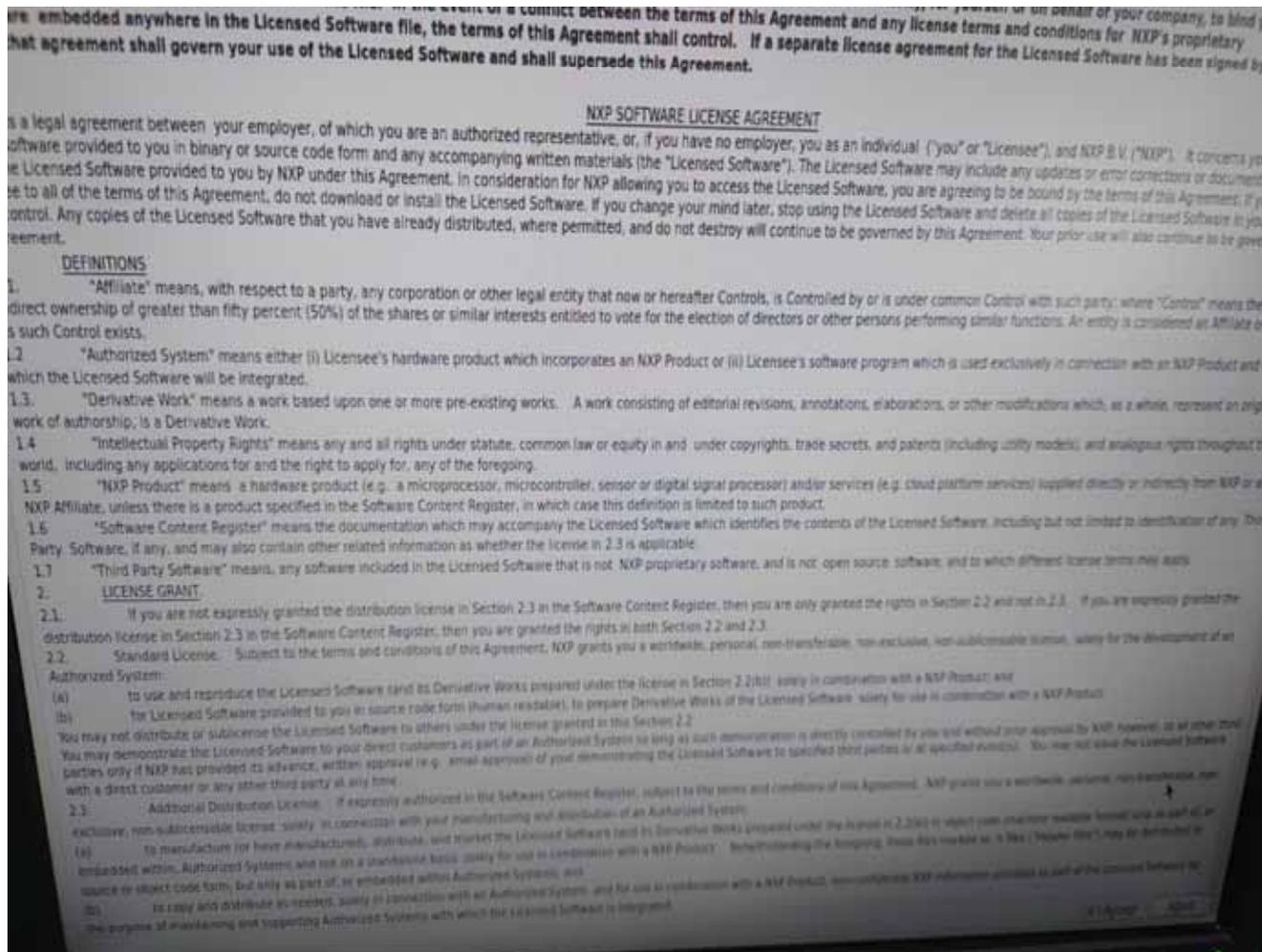
Ethernetでアップデートする場合は下記のようにいくつかの項目がリストされます。
この項目はサーバーへ接続する時期により内容が変わります。



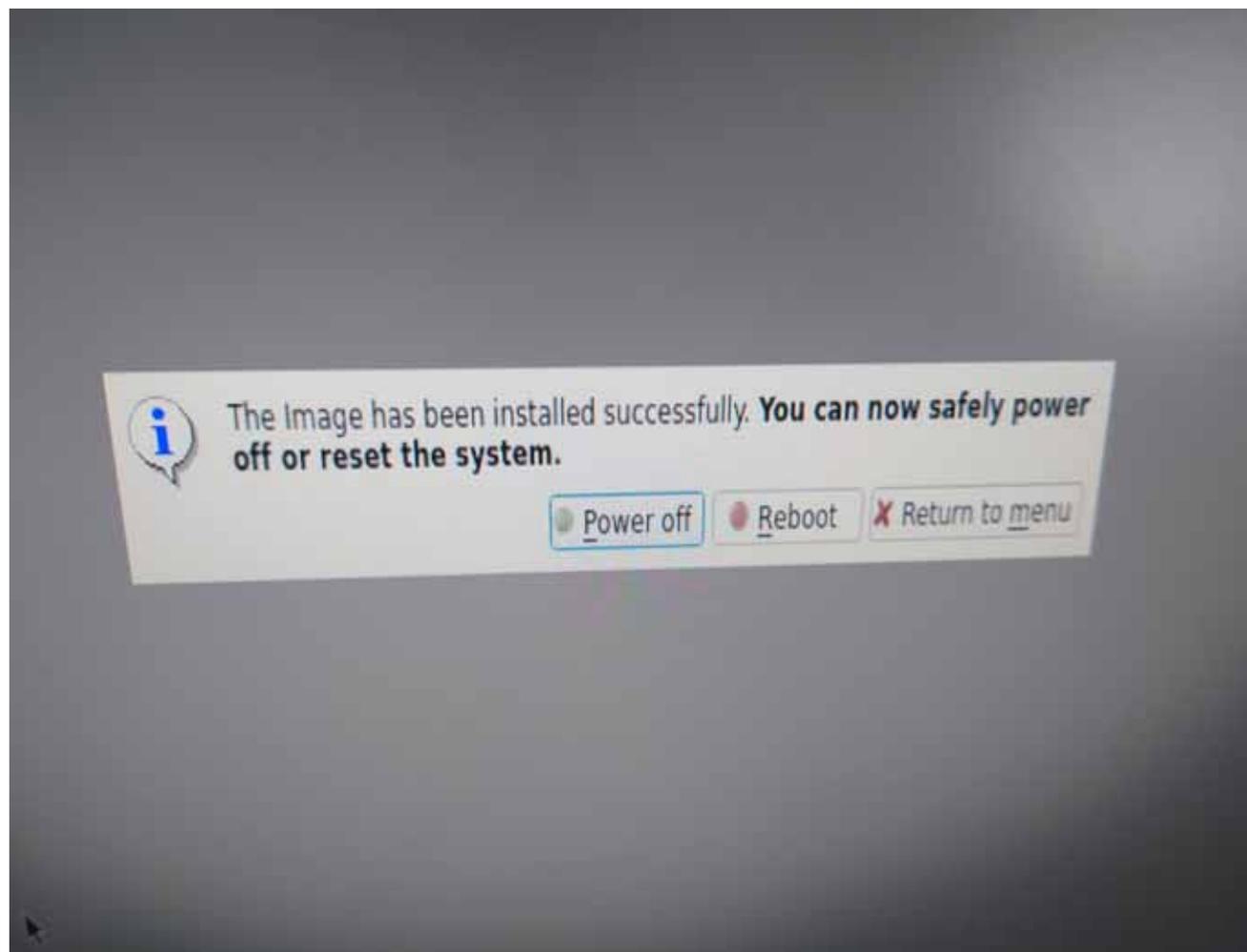
リストされている項目をダブルクリックすると下記のような確認がでできます。Yesをクリックするとインストールが始まります。



ライセンス規約が表示されます。同意できる場合は「I Accept」をクリックしてください。
ライセンス規約に同意できない場合はソフトウェアを使用することはできません。



インストールが終了すると下記のような選択が表示されます。Rebootをクリックして書き込んだOSが起動するかどうかを確認します。

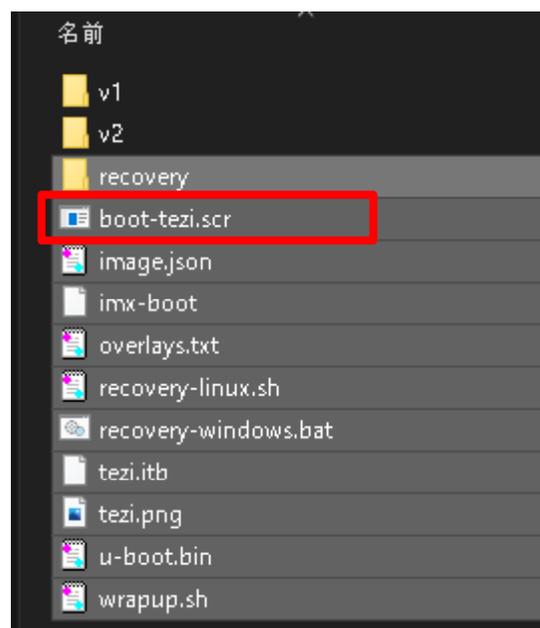


外部メディアからTEZIを起動する方法

TEZIを解凍してFAT32(もしくはEXT4)フォーマットのSDカードやUSBメモリのルートディレクトリに格納してください。OSイメージも同様に解凍してリカバリーモード時同様に任意のディレクトリに格納してください。ToradexのサーバーからOSイメージを書き込む場合はOSイメージの準備は不要です。

TEZIとOSイメージは各々を別のメディアに格納することもできます。

ルートディレクトリにあるboot-tezi.scrというファイルをboot.scrにリネームします。boot.scrはU-Boot起動時に起動スクリプトが監視するファイルです。



BSP5系では起動処理が内部eMMCや外部メディアなど起動する場所によらずすべて同じ処理となっています。
U-Bootの起動スクリプトはSDカード、モジュール上のeMMC、USBメモリ、Ethernetという優先順位で各々のルートディレクトリのboot.scrを探しboot.scrを実行することで起動します。

この優先順位はU-Bootの設定のboot_targetsという設定で決まっています。

下記コマンドで確認できます。

```
[U-Boot]$ printenv boot_targets
```

```
boot_targets=mmc1 mmc2 usb0 dhcp
```

この順序は設定を変更することにより変えることができます。(左から順番に実行)

通常eMMCにはLinuxを書き込みます。SDカードにはTEZIを入れておきます。
優先順位を変更することで何を起動するかを切り替えることができます。

U-Bootのfatlsコマンドで中身を確認できます。

```
[U-Boot]$ fatls mmc 1:1
```

mmc1のパーティション1を参照します。SDカードの中身が参照できます。

```
[U-Boot]$ fatls mmc 2:1
```

mmc2のパーティション1を参照します。内部eMMCの中身が参照できます。

mmc1には本マニュアル通りのTEZIとインストールするOSイメージが格納されています。

boot.scrも含まれているためmmc1が優先されてmmc1から起動が行われます。

USBを使う場合はUSBの優先するようにboot_targetsを修正してください。修正方法は後述。

```
COM16:115200bps - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) 漢字コード(K) ヘルプ(H)

Verdin iMX8MP # fatls mmc 1:1
      recovery/
3084  boot.scr
1474  image.json
1234432 imx-boot
      86  overlays.txt
      397 recovery-linux.sh
      273 recovery-windows.bat
46971668 tezi.itb
      2997 tezi.png
      783185 u-boot.bin
      18  wrapup.sh
Verdin-iMX8MP_Reference-Minimal-Image-Tezi_5.7.0+

10 file(s), 2 dir(s)

Verdin iMX8MP # █
```

```
COM12:115200bps - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) 漢字コード(K)

Verdin iMX8MP # fatls mmc 2:1
      87885 imx8mp-verdin-nonwifi-dahlia.dtb
10734485 Image.gz
      88299 imx8mp-verdin-wifi-dahlia.dtb
      89831 imx8mp-verdin-wifi-dev.dtb
      89421 imx8mp-verdin-nonwifi-dev.dtb
      5754 boot.scr
      overlays/
      86  overlays.txt

7 file(s), 1 dir(s)

Verdin iMX8MP # █
```

ハードウェアの準備

以下を行います。

X1にモジュールを挿します。ヒートシンクも装着します。

X37にHDMIケーブルを指してディスプレイと繋がります。 TightVNC Viewerなどで代用できます。

X53にUSBマウスを挿入します。

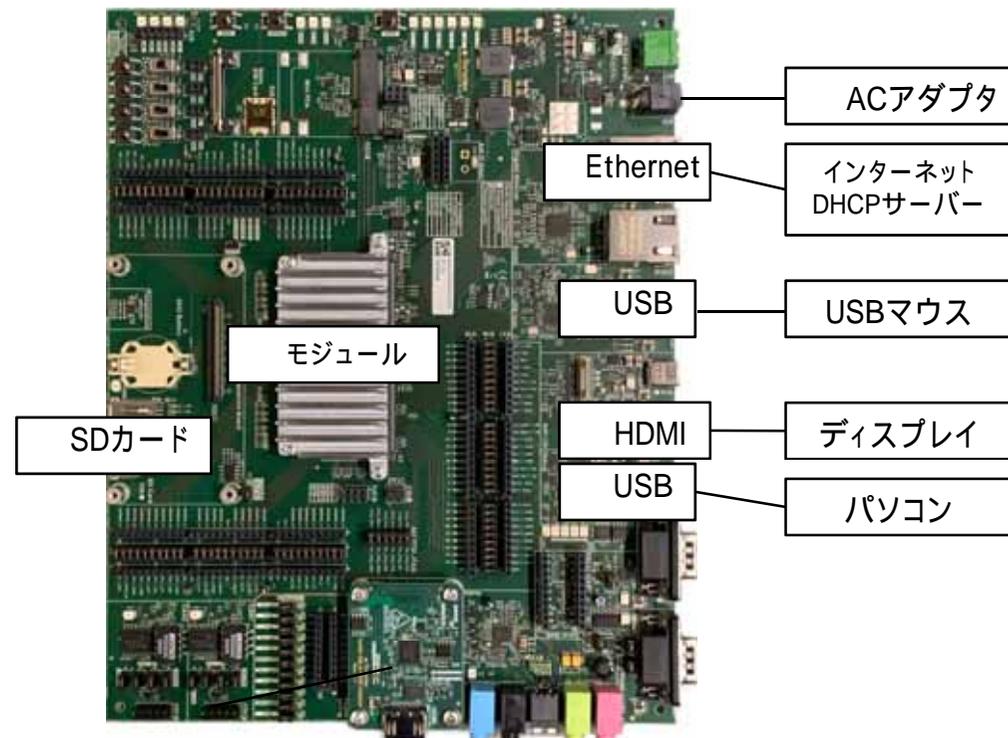
X66にUSBケーブルを差し込みパソコンと接続します。

SDカードでアップデートする場合はX55にSDカードを挿入します。

アップデートをインターネット越しで行う場合はX25にEthernetケーブルを挿入します。

インターネットに接続できる環境が必要です。モジュールはDHCPで接続を行いますのでDHCPサーバーが必要です。

X58にACアダプタを接続します。



シリアル通信ソフトの準備

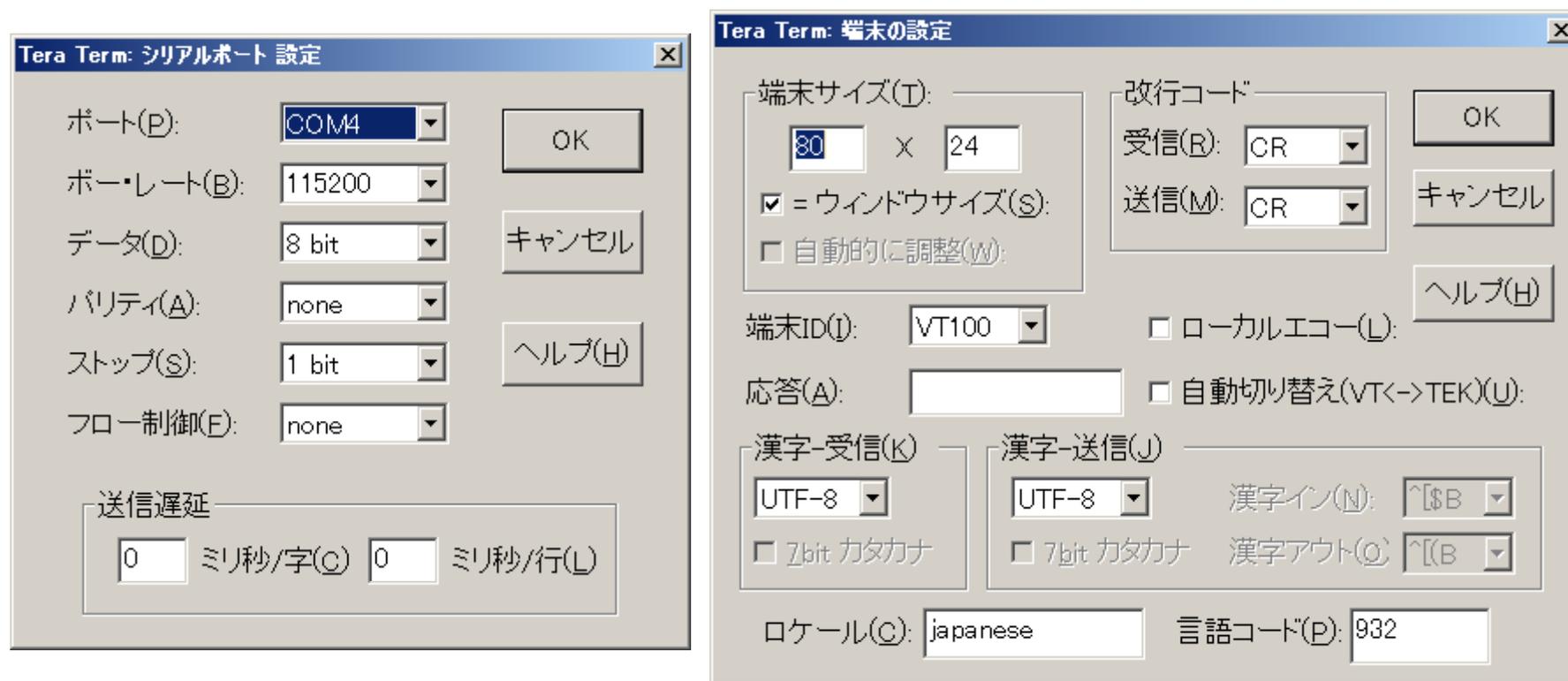
モジュールとシリアル通信を行います。本マニュアルではTeraTermを使います。

TeraTermを下記、設定でCOMポートを開きます。

Verdin評価ボードと接続すると2つのCOMポート番号が追加されます。

そのうち大きな番号のCOMポート番号に接続してください。

例えば10,11と追加された場合は11を指定します。



起動

SDカードから起動する場合は電源ボタン(SW2)を押し電源ONすると自動的にSDカードから起動します。

SDカード以外から起動する場合は電源ON後、下記のように「Hit any key to stop autoboot:」とTeratermに出力されます。

この時に何かしらのキーを入力するとブートローダー(U-Boot)で止まりますが入力しないとU-BootがLinuxを起動させます。(書き込まれている場合) ブートローダーで止めるためにスペースキーなどを入力します。

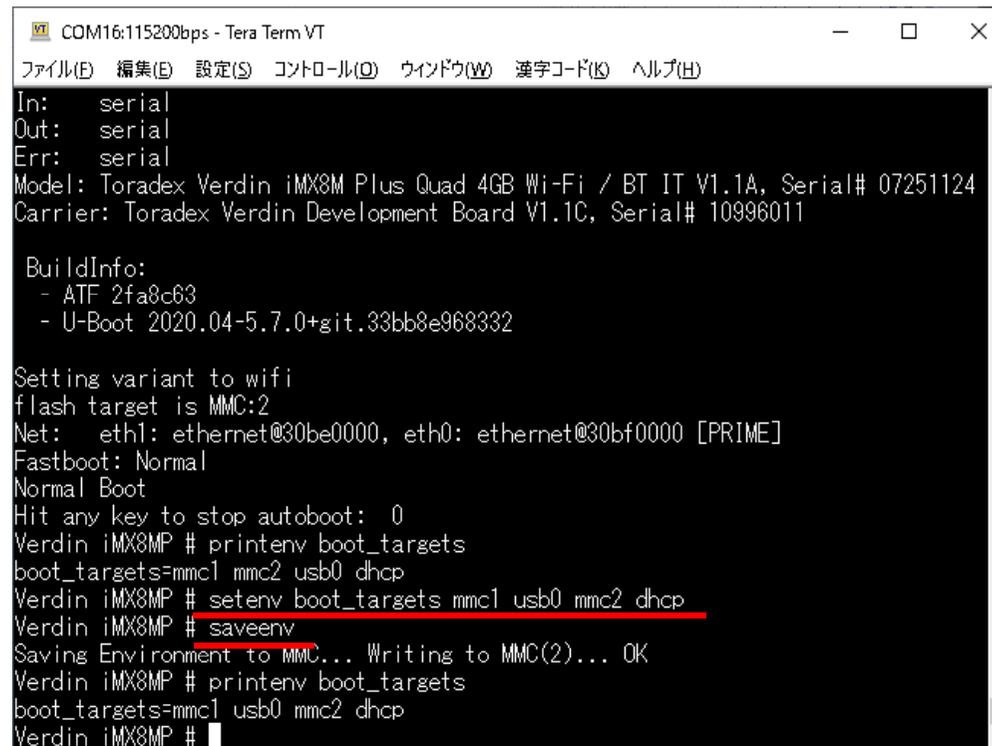
例えばUSBを使う場合はUSBの優先するようにboot_targetsを修正してからresetコマンドで再起動します。

mmc2より左にusb0があれば優先されます。

```
[U-Boot]$ setenv boot_targets mmc1 usb0 mmc2 dhcp
```

```
[U-Boot]$ saveenv
```

```
[U-Boot]$ reset
```



```
COM16:115200bps - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(C) ウィンドウ(W) 漢字コード(K) ヘルプ(H)
In: serial
Out: serial
Err: serial
Model: Toradex Verdin iMX8M Plus Quad 4GB Wi-Fi / BT IT V1.1A, Serial# 07251124
Carrier: Toradex Verdin Development Board V1.1C, Serial# 10996011

BuildInfo:
- ATF 2fa8c63
- U-Boot 2020.04-5.7.0+git.33bb8e968332

Setting variant to wifi
flash target is MMC:2
Net: eth1: ethernet@30be0000, eth0: ethernet@30bf0000 [PRIME]
Fastboot: Normal
Normal Boot
Hit any key to stop autoboot: 0
Verdin iMX8MP # printenv boot_targets
boot_targets=mmc1 mmc2 usb0 dhcp
Verdin iMX8MP # setenv boot_targets mmc1 usb0 mmc2 dhcp
Verdin iMX8MP # saveenv
Saving Environment to MMC... Writing to MMC(2)... OK
Verdin iMX8MP # printenv boot_targets
boot_targets=mmc1 usb0 mmc2 dhcp
Verdin iMX8MP #
```

設定にかかわらずブートメディアを指定して起動することもできます。

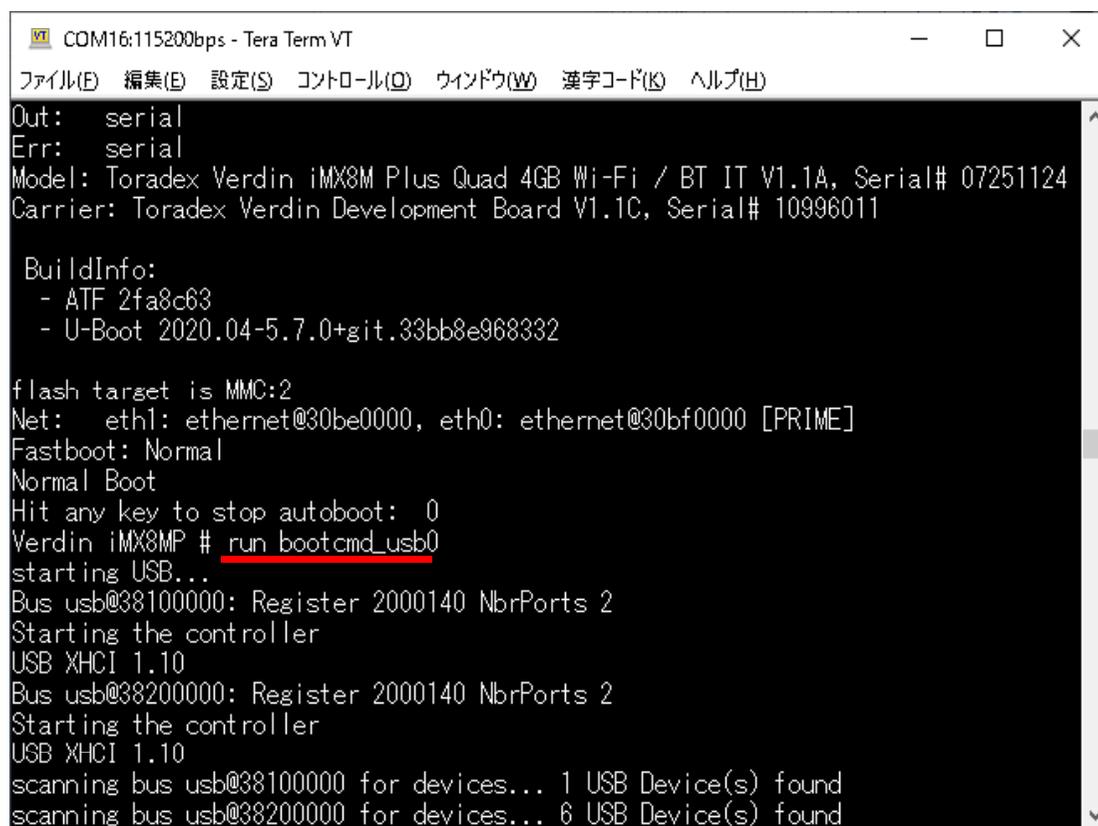
bootcmd_mmc1やbootcmd_usb0など起動メディアを指定する起動コマンドが定義されています。

runコマンドでこの定義を実行することで指定した場所から起動することができます。

```
[U-Boot]$ run bootcmd_usb0
```

どのような定義が存在するかはprintenvコマンドで確認できます。

TEZI起動後はリカバリーモード時と同様の操作でOSイメージの書き込みを行ってください。



```
COM16:115200bps - Tera Term VT
ファイル(E) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) 漢字コード(K) ヘルプ(H)
Out: serial
Err: serial
Model: Toradex Verdin iMX8M Plus Quad 4GB Wi-Fi / BT IT V1.1A, Serial# 07251124
Carrier: Toradex Verdin Development Board V1.1C, Serial# 10996011

BuildInfo:
- ATF 2fa8c63
- U-Boot 2020.04-5.7.0+git.33bb8e968332

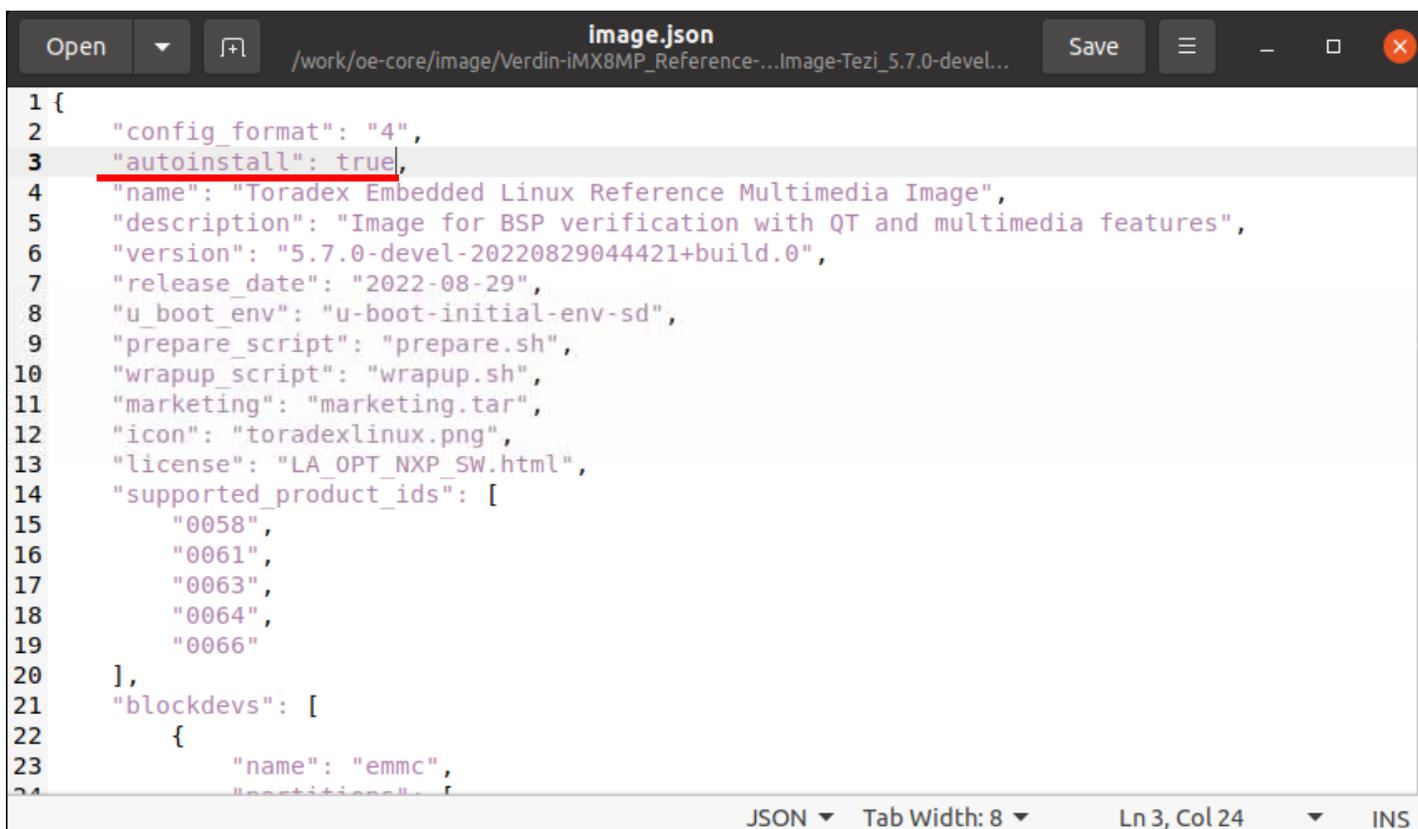
flash target is MMC:2
Net: eth1: ethernet@30be0000, eth0: ethernet@30bf0000 [PRIME]
Fastboot: Normal
Normal Boot
Hit any key to stop autoboot: 0
Verdin iMX8MP # run bootcmd_usb0
starting USB...
Bus usb@38100000: Register 2000140 NbrPorts 2
Starting the controller
USB XHCI 1.10
Bus usb@38200000: Register 2000140 NbrPorts 2
Starting the controller
USB XHCI 1.10
scanning bus usb@38100000 for devices... 1 USB Device(s) found
scanning bus usb@38200000 for devices... 6 USB Device(s) found
```

自動インストール機能

OSイメージの中にimage.jsonというファイルがあります。このファイルはTEZIの動作を定義するものです。

この中にautoinstallという設定があります。この設定をtrueに変更すると自動でインストールするようになります。TEZIが起動した後リムーバブルメディアをスキャンしてautoinstallがtrueがあればそのままインストールされます。この設定を使用すると量産工程などでGUIなどを操作することなく電源ONからインストールまでが自動で行われます。複数OSイメージがある場合はどれがインストールされるかは保証されていないためひとつだけtrueに設定してください。

TEZIで扱うテキストファイルはgeditで編集することをお勧めします。改行コードは¥nで処理されます。¥rが含まれると誤動作します。



```
1 {
2   "config_format": "4",
3   "autoinstall": true,
4   "name": "Toradex Embedded Linux Reference Multimedia Image",
5   "description": "Image for BSP verification with QT and multimedia features",
6   "version": "5.7.0-devel-20220829044421+build.0",
7   "release_date": "2022-08-29",
8   "u_boot_env": "u-boot-initial-env-sd",
9   "prepare_script": "prepare.sh",
10  "wrapup_script": "wrapup.sh",
11  "marketing": "marketing.tar",
12  "icon": "toradexlinux.png",
13  "license": "LA_OPT_NXP_SW.html",
14  "supported_product_ids": [
15    "0058",
16    "0061",
17    "0063",
18    "0064",
19    "0066"
20  ],
21  "blockdevs": [
22    {
23      "name": "emmc",
24      "partitions": [
```

TEZIの自動アップデート機能

TEZIの設定ファイルimage.jsonにはフォーマットのバージョンがありconfig_formatで指定されています。

TEZIは新しい機能が追加されていき、場合によっては新しい設定項目が追加されフォーマットのバージョンが上がっていきます。

古いバージョンのTEZIを使った場合、新しいフォーマットのimage.jsonを読み込むことができません。

例えば在庫として長く置かれていたモジュールにはインストールされているTEZIのバージョンが古く、製品製造時に書き込むOSイメージのconfig_formatで指定されているバージョンに対応していない場合書き込むことができません。このような問題を解決するためにTEZIの自動アップデート機能があります。OSイメージを書き込む前に一旦TEZIのバージョンアップをすることで問題に対処します。

TEZIのimage.json(**OSイメージにもimage.jsonがあるため注意してください。**)にはisinstallerという設定がありtrueになっています。isinstallerがtrueかつautoinstallがtrueの場合、OSイメージよりも優先してTEZIの書き込みが行われます。実行中のTEZIのバージョン(versionの設定)と比較してリムーバルメディアにあるTEZIのバージョンが新しければ最初にTEZIをインストールして再起動が行われます。再起動後は新しいTEZIが立ち上がりバージョンが同じになるため再度TEZIがインストールされることはありません。次にOSイメージのインストールが行われます。



```
1 {
2   "config_format": "4",
3   "autoinstall": true,
4   "name": "toradex Easy Installer",
5   "description": "Toradex Easy Installer for verdin-imx8mp machine",
6   "version": "5.7.1+build.13",
7   "release_date": "2022-09-06",
8   "wrapup_script": "wrapup.sh",
9   "icon": "tezi.png",
10  "isinstaller": true,
11  "supported_product_ids": [
12    "0058",
13    "0061",
14    "0063",
15    "0064",
16    "0066"
17  ],
18  "blockdevs": [
19    {
20      "name": "ennc",
21      "partitions": [
22        {
23          "partition_size_nominal": 128,
24          "want_maximised": false,
25          "content": {
26            "label": "BOOT",
27            "filesystem_type": "FAT",
28            "mkfs_options": "",
29            "filelist": [
30              "tezi.itb",
31              "tezi.itb"
32            ]
33          }
34        }
35      ]
36    }
37  ]
38 }
```

スクリプト実行機能

image.jsonに定義されているprepare_script、wrapup_script、error_scriptにシェルフファイルを設定することでそれぞれインストール前、インストール後、エラー発生時に自動でスクリプトを実行することが可能です。

例えばwrapup.shにrebootコマンドを記述することでOS書き込み後自動でrebootします。ただしrebootしてもそのままほっておくと再度SDからの起動が優先されてTEZIが起動してしまいます。自動的に書き込んだOSがeMMCから起動するようにするためにはSDカードにあるboot.scrをリネームすることで防止できます。

例えば下記のような記述をするとOSイメージインストール後boot.scrをリネームして再起動します。TEZIはリムーバルメディアを読み込み専用でマウントします。そのままではboot.scrをリネームできないため一度、書き込み可能でマウントし直して最後に読み込み専用に戻しています。

```
#!/bin/sh
```

```
MOUNT_POINT=/run/media/src
```

```
mount -o remount,rw $MOUNT_POINT
```

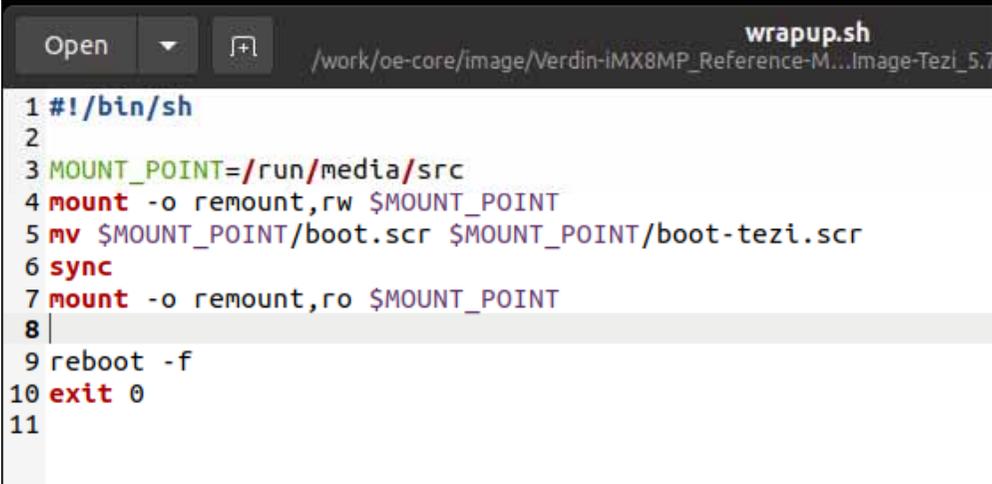
```
mv $MOUNT_POINT/boot.scr $MOUNT_POINT/boot-tezi.scr
```

```
sync
```

```
mount -o remount,ro $MOUNT_POINT
```

```
reboot -f
```

```
exit 0
```



```
wrapup.sh
/work/oe-core/image/Verdin-IMX8MP_Reference-M...Image-Tezi_5.7
1 #!/bin/sh
2
3 MOUNT_POINT=/run/media/src
4 mount -o remount,rw $MOUNT_POINT
5 mv $MOUNT_POINT/boot.scr $MOUNT_POINT/boot-tezi.scr
6 sync
7 mount -o remount,ro $MOUNT_POINT
8
9 reboot -f
10 exit 0
11
```

書き込みのトレース機能

prepare_script、wrapup_script、error_scriptで実行されるシェルスクリプトにはパラメータが付与されて起動します。\$1,\$2,\$3,\$4にそれぞれプロダクトID、モジュールリビジョン、シリアルナンバー、イメージのディレクトリが入っています。

例えば下記のような記述をするとOSイメージインストール後SDカードにログを出力して再起動します

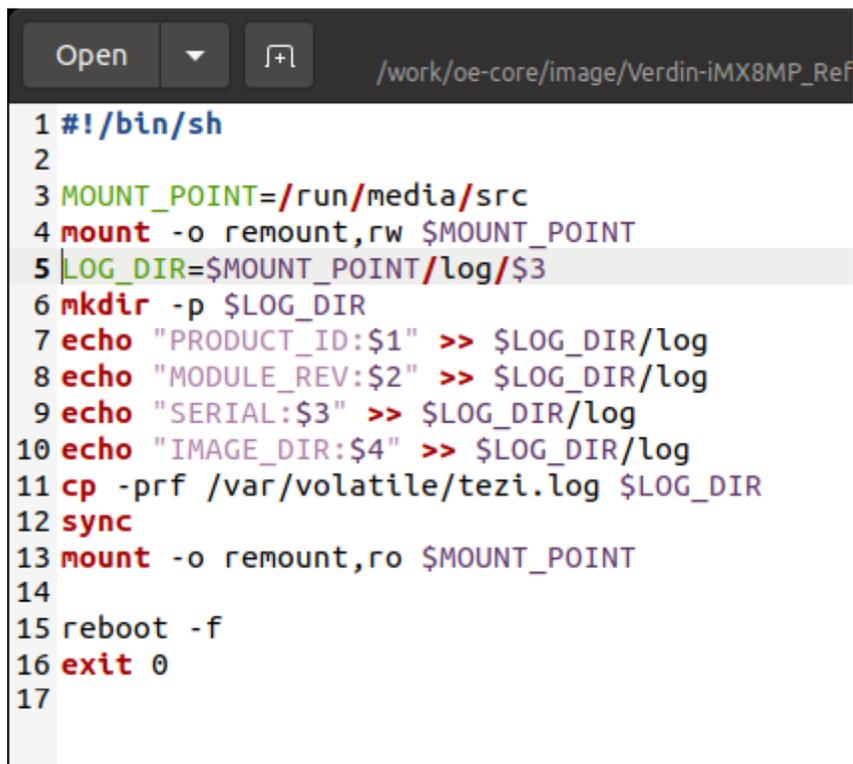
TEZIが内部でどのようなことをしているかは/var/volatile/tezi.logにログが出力されています。

下記シェルではtezi.logも保存しています。

```
#!/bin/sh
```

```
MOUNT_POINT=/run/media/src
mount -o remount,rw $MOUNT_POINT
LOG_DIR=$MOUNT_POINT/log/$3
mkdir -p $LOG_DIR
echo "PRODUCT_ID:$1" >> $LOG_DIR/log
echo "MODULE_REV:$2" >> $LOG_DIR/log
echo "SERIAL:$3" >> $LOG_DIR/log
echo "IMAGE_DIR:$4" >> $LOG_DIR/log
cp -prf /var/volatile/tezi.log $LOG_DIR
sync
mount -o remount,ro $MOUNT_POINT

reboot -f
exit 0
```

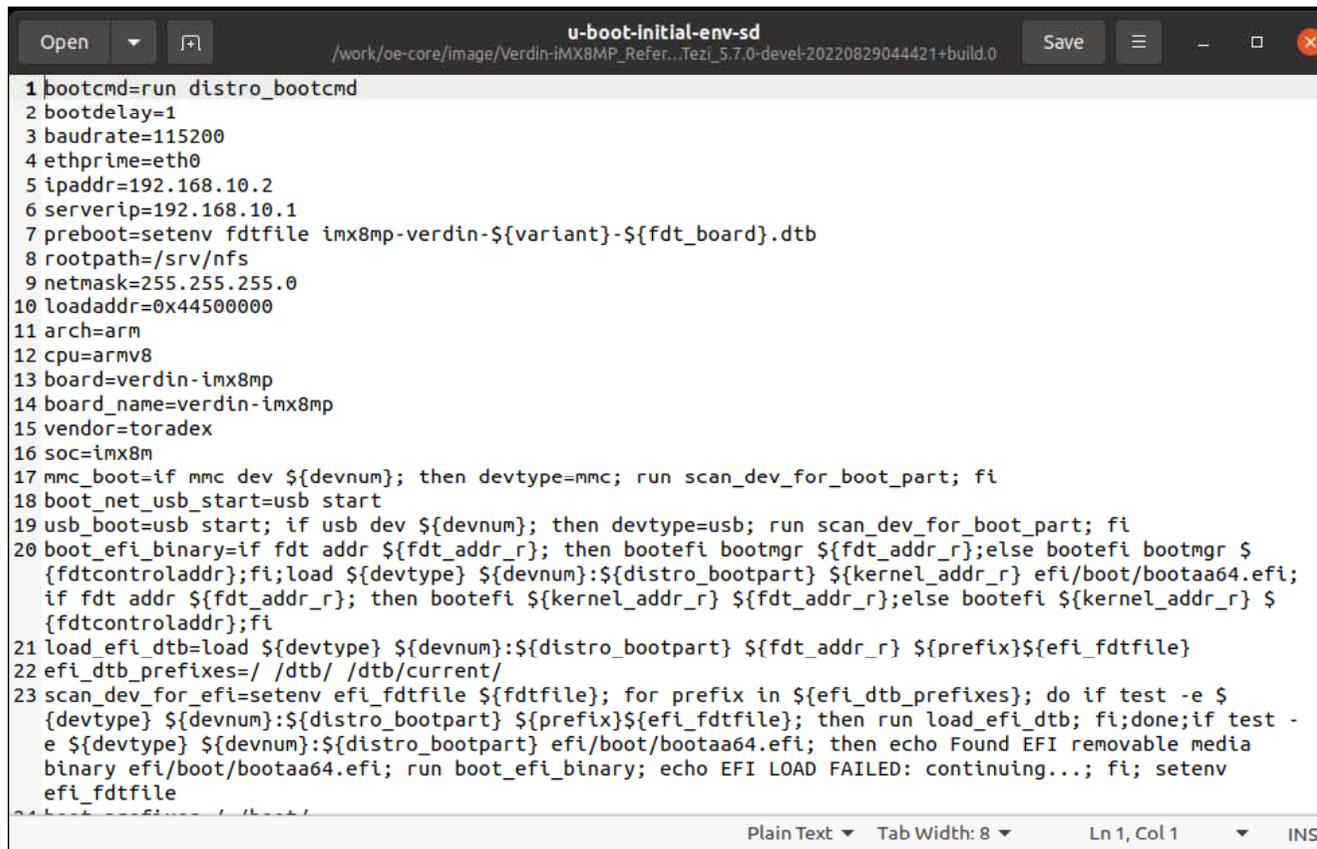


```
Open [v] [+] /work/oe-core/image/Verdin-iMX8MP_Ref
1 #!/bin/sh
2
3 MOUNT_POINT=/run/media/src
4 mount -o remount,rw $MOUNT_POINT
5 LOG_DIR=$MOUNT_POINT/log/$3
6 mkdir -p $LOG_DIR
7 echo "PRODUCT_ID:$1" >> $LOG_DIR/log
8 echo "MODULE_REV:$2" >> $LOG_DIR/log
9 echo "SERIAL:$3" >> $LOG_DIR/log
10 echo "IMAGE_DIR:$4" >> $LOG_DIR/log
11 cp -prf /var/volatile/tezi.log $LOG_DIR
12 sync
13 mount -o remount,ro $MOUNT_POINT
14
15 reboot -f
16 exit 0
17
```

U-Boot環境変数設定ファイル

image.jsonのu_boot_envに設定されたファイル(デフォルトはu-boot-initial-env-sd)にU-Bootの環境変数の設定をします。U-Bootのヘッダーを参照して出力されるためU-Bootのソースコードを修正しても修正した箇所がこのファイルにも適用されています。U-Bootのソースコードに設定された値はenv defaultで初期化した時に設定される値です。個別で設定を追加したい場合はこのファイルに追加するだけで設定されますが基本的にはU-Bootのソースコードを修正することで対応する方をお勧めします。

フラッシュ上に有効なU-Boot環境変数がすでにある場合、新しい変数は追加され、既存の変数は上書きされ、空の変数は削除され、他の変数は削除されます。

A screenshot of a text editor window titled "u-boot-initial-env-sd". The window shows a list of environment variables and boot commands for U-Boot. The variables include bootcmd, bootdelay, baudrate, ethprime, ipaddr, serverip, preboot, rootpath, netmask, loadaddr, arch, cpu, board, board_name, vendor, soc, mmc_boot, boot_net_usb_start, usb_boot, boot_efi_binary, load_efi_dtb, efi_dtb_prefixes, and scan_dev_for_efi. The code is written in a shell script style with line numbers from 1 to 23. The editor interface includes a menu bar with "Open", "Save", and window control buttons. The status bar at the bottom shows "Plain Text", "Tab Width: 8", "Ln 1, Col 1", and "INS".

```
1 bootcmd=run distro_bootcmd
2 bootdelay=1
3 baudrate=115200
4 ethprime=eth0
5 ipaddr=192.168.10.2
6 serverip=192.168.10.1
7 preboot=setenv fdtfile imx8mp-verdin-${variant}-${fdt_board}.dtb
8 rootpath=/srv/nfs
9 netmask=255.255.255.0
10 loadaddr=0x44500000
11 arch=arm
12 cpu=armv8
13 board=verdin-imx8mp
14 board_name=verdin-imx8mp
15 vendor=toradex
16 soc=imx8m
17 mmc_boot=if mmc dev ${devnum}; then devtype=mmc; run scan_dev_for_boot_part; fi
18 boot_net_usb_start=usb start
19 usb_boot=usb start; if usb dev ${devnum}; then devtype=usb; run scan_dev_for_boot_part; fi
20 boot_efi_binary=if fdt addr ${fdt_addr_r}; then bootefi bootmgr ${fdt_addr_r};else bootefi bootmgr $
  {fdtcontroladdr};fi;load ${devtype} ${devnum}:${distro_bootpart} ${kernel_addr_r} efi/boot/bootaa64.efi;
  if fdt addr ${fdt_addr_r}; then bootefi ${kernel_addr_r} ${fdt_addr_r};else bootefi ${kernel_addr_r} $
  {fdtcontroladdr};fi
21 load_efi_dtb=load ${devtype} ${devnum}:${distro_bootpart} ${fdt_addr_r} ${prefix}${efi_fdtfile}
22 efi_dtb_prefixes=/dtb/ /dtb/current/
23 scan_dev_for_efi=setenv efi_fdtfile ${fdtfile}; for prefix in ${efi_dtb_prefixes}; do if test -e $
  {devtype} ${devnum}:${distro_bootpart} ${prefix}${efi_fdtfile}; then run load_efi_dtb; fi;done;if test -
  e ${devtype} ${devnum}:${distro_bootpart} efi/boot/bootaa64.efi; then echo Found EFI removable media
  binary efi/boot/bootaa64.efi; run boot_efi_binary; echo EFI LOAD FAILED: continuing...; fi; setenv
  efi_fdtfile
```

書き込み時間の短縮

書き込み時間はOSイメージの大きさに依存します。できるだけOSイメージから不要な機能を削除することでOSイメージのサイズを小さくすることで書き込み時間を短くすることができます。

OSイメージを出力するとrootfsやbootfsはLZMA2アルゴリズム(xzファイル)で出力されます。SDカードやUSBメモリなど高速に読み込みできる場所にある場合、読み込速度に対して解凍の方が時間がかかるため遅くなります。(インターネット越しでダウンロードする場合は解凍時間に比べてダウンロード速度のほうが遅いため圧縮は効果的です。)

tar.xzを解凍してtarにした状態でインストールすると大幅に早くなります。この場合image.jsonのファイル名をtar.xzからtarに修正してください。xzの解凍は下記のコマンドで行います。

```
sudo xz -d ./Reference-Minimal-Image-verdin-imx8mp.tar.xz
```

uncompressed_sizeは書き込み中のインジケータの分母に使われます。正確な値ではなく目安として使われます。

気になる場合は微調整してください。

```
20 ],
21 "blockdevs": [
22   {
23     "name": "emmc",
24     "partitions": [
25       {
26         "partition_size_nominal": 48,
27         "want_maximised": false,
28         "content": {
29           "label": "BOOT",
30           "filesystem_type": "FAT",
31           "mkfs_options": "",
32           "filename": "Reference-Multimedia-Image-verdin-imx8mp.bootfs.tar",
33           "uncompressed_size": 10.91015625
34         }
35       },
36       {
37         "partition_size_nominal": 512,
38         "want_maximised": true,
39         "content": {
40           "label": "RFS",
41           "filesystem_type": "ext4",
42           "mkfs_options": "-E nodiscard",
43           "filename": "Reference-Multimedia-Image-verdin-imx8mp.tar",
44           "uncompressed_size": 932.22265625
45         }
46       }
47     ]
48   },
49 ]
```

インストール中の背景の変更

OSイメージのファイル一式の中にmarketing.tarがあります。このファイルはインストール中の背景画像が入っています。この中のファイルを変更することでインストール中の背景画像を変更することができます。

下記コマンドで解凍します。

```
tar xf marketing.tar
```

slides_vgaというディレクトリに下記が入っています。これらの画像が順番に表示されていきます。

Embedded-Linux-Splash-Screen_V3_01.png

Embedded-Linux-Splash-Screen_V3_02.png

Embedded-Linux-Splash-Screen_V3_03.png

Embedded-Linux-Splash-Screen_V3_04.png

Embedded-Linux-Splash-Screen_V3_05.png

使用したい画像をpng形式で格納してください。

再度アーカイブ化してオリジナルに上書きしてください。

```
tar cf ./marketing.tar ./slides_vga/
```

インストール中の背景が変わっていることが確認できます。