

# OpenBlocks IoT Family向け 開発者向けガイド



Ver.3.4.2-2

ぷらっとホーム株式会社

## ■ 商標について

- ・ Linux は、Linus Torvalds 氏の米国およびその他の国における商標あるいは登録商標です。
- ・ NTT ドコモは日本電信電話株式会社の登録商標です。
- ・ SoftBank およびソフトバンクの名称、ロゴは、日本国およびその他の国におけるソフトバンクグループ株式会社の登録商標または商標です。
- ・ au(KDDI)は KDDI 株式会社の登録商標または商標です。
- ・ 文中の社名、商品名等は各社の商標または登録商標である場合があります。
- ・ その他記載されている製品名などの固有名詞は、各社の商標または登録商標です。
- ・ Docker and Docker logo are trademarks or registered trademarks of Docker, Inc. in the United States and/or other countries. Docker, Inc. and other parties may also have trademark rights in other terms used herein.

## ■ 使用にあたって

- ・ 本書の内容の一部または全部を、無断で転載することとはご遠慮ください。
- ・ 本書の内容は予告なしに変更することがあります。
- ・ 本書の内容については正確を期するように努めていますが、記載の誤りなどにご指摘がございましたら弊社サポート窓口へご連絡ください。  
また、弊社公開の **WEB** サイトにより本書の最新版をダウンロードすることが可能です。
- ・ 本装置の使用にあたっては、生命に関わる危険性のある分野での利用を前提とされていないことを予めご了承ください。
- ・ その他、本装置の運用結果における損害や逸失利益の請求につきましては、上記にかかわらずいかなる責任も負いかねますので予めご了承ください。

## 目次

第 1 章 はじめに .....	5
第 2 章 ご利用の前に .....	6
2-1. 製品概要 .....	6
2-2. SSD ベースのシステム開発における注意点 .....	6
2-3. SIM について .....	9
2-4. eMMC ストレージパーティション情報 .....	9
2-5. ストレージモード .....	11
2-6. PC と OpenBlocks IoT Family の接続 .....	12
2-7. WEB UI .....	13
第 3 章 ご利用方法 .....	14
3-1. ステータスインジケータの表示色 .....	14
3-2. モバイル回線向けモデム制御 .....	17
3-3. モバイル回線接続 .....	22
3-3-1. オンデマンド接続 .....	22
3-3-2. モバイル回線モデムのユーザー制御 .....	23
3-4. バックアップ .....	26
3-5. リストア .....	28
3-6. 応用 .....	29
3-7. 推奨使用デバイスファイル .....	29
3-8. リカバリー起動 .....	31
3-9. クロス開発環境の構築 .....	33
3-10. WEB UI 拡張機能 .....	33
3-10-1. WEB UI への機能拡張インストール .....	33
3-10-2. スクリプト編集 .....	33
3-10-3. コマンド実行 .....	34
3-10-4. WEB UI フィルタテーブル特殊設定 .....	35
3-10-5. SMS 送信 .....	35
3-10-6. LTE モジュール(KDDI)運用切り替え .....	36
3-10-7. WEB UI 自動外部ストレージマウント機能 .....	36
3-10-8. ブートモード切替 .....	37
3-10-9. GPS 情報について .....	38
3-10-10. DHCP における DGW、DNS について .....	38
3-11. Docker について .....	39
第 4 章 注意事項及び補足 .....	40

4-1. スクリプト処理による遅延処理等について .....	40
4-2. 使用ポート一覧.....	40

# 第 1 章 はじめに

本書は、OpenBlocks IoT Family の開発者向けガイドです。

利用者向けについては、OpenBlocks IoT Family 向け WEB UI セットアップガイドを参照してください。

## 第 2 章 ご利用の前に

### 2-1. 製品概要

OpenBlocks IoT Family は OS として Debian GNU/Linux を採用した汎用サーバー製品です。ハードウェア特性を活かす為のカスタマイズを行っていますが、本カスタマイズ以外は、Debian やその他 Linux の一般的な操作方法でご利用いただけます。

### 2-2. SSD ベースのシステム開発における注意点

近年、フラッシュメモリーのコストダウンによってスマートフォンを始めノート PC や様々な機器では、従来のハードディスクに代わって SSD が採用されています。また、本製品に使用されている eMMC は SSD の一種です。

SSD はランダムアクセス性能が高速であり、かつ、ソリッドステートである利点からメカニカルな障害耐性や環境性能が非常に良くなりますが、その反面、ハードディスクに比べてデータの書き換え制限が大幅に少なくなります。

SSD には大きく分けると SLC と MLC にわかれ、書き換え性能が数万回の SLC タイプが数ギガの容量帯では主流でしたが、マルチビットセルで書き換え回数数千回の低コストの MLC が大容量化を実現し、スマートフォンや PC に多く使われるようになり、現在では SLC の SSD は徐々に姿を消し始めています。

当社マイクロサーバー製品のオプションでも SLC 製品は一部だけ小容量帯で残すのみです。

ゆえに MLC の SSD 搭載でのマイクロサーバー運用が大変多くなります。

MLC では概ね 3000 回の書き換え性能があり、3000 回を超えた辺りからビットエラーが発生し始めますが ECC エラー回復します。

しかし、その ECC 回復可能な条件を超えるとリードエラーが発生します。

ゆえにこの状況に追いつかないようなシステム作りが必要になります。

#### ●セルあたりの書き換え数とフラッシュメモリーのブロックサイズ

セルあたり 3000 回と言っても、SSD に対する書き込みが 1 バイトでも 1 回は 1 回となります。

最近のフラッシュメモリーは少ないアドレス線で大容量をサポートするため 512KB くらいの大きなブロックで読み書きされます。

つまり、書き込みにおいて1バイト書いても512KB書いても書き換えは1回なのです。  
ゆえにSSDへの書き込みは、小さなデータサイズで細かくよりも、なるべくバッファに多くデータを貯めこんで大きなデータサイズで一気を書く方がSSDに対する書き込み回数を最小限にすることが出来ます。

#### ●ウェアレベリング機能

SSDは書き換え制限が少ないため、同じブロックアドレスを何回も書き換える動作に対して、同一の実ブロックアドレスへの書き換えを集中させないように平均化しています。  
これはブロックアドレスを仮想化して実現しています。

OSはSSDに対して使っているブロックと使われていないブロックを通知しておりSSDは次に書き込みがあった場合、一番書き換え数の少ないブロックをその書き込みのために準備しています。

これによって結果的に書き換え数が平均化されます。

#### ●スタティックウェアレベリング

従来のウェアレベリングの場合、使われている領域でかつほとんど書き換えのないデータが例えば50%あった場合、OSのインストール時から1回だけ書かれたブロックが50%を占め、その部分がほぼ新品のまま、再利用されている残りの50%の領域がどんどん書き換えられてしまい、全体で平均化した場合よりも2倍早くSSDの寿命が来てしまいます。  
この対策に考えられたのがスタティックウェアレベリングで、ほとんど書き換えられていないブロックのデータを再利用の多いブロックに移動し、ほぼ新品のままだったブロックを再利用領域にあてがいます。

これにより、50%ほど書き換えのないブロックがあっても、ほぼSSDの寿命をまっとうできます。

#### ●SSDの全体の書き換え回数を想定する

例えば4GBのSSDで512KBブロックタイプであった場合、セルあたり3000回の書き換えできるとして512KB以下のデータを書いた場合、SSDの総書き換え回数を想定すると以下のようになります。

$$4294967296B \div 524288 = 8192 (4GB \div 512KB)$$

物理ブロック数は8192個あることになります。

それぞれが3000回書き換えられるとしたら

$$8192 \times 3000 = 24576000 \text{ 回}$$

つまり 1 回ずつ 1 バイトのデータを書くとしたの 18.4MB 書くと寿命がつきてしまいます。(実際にはこういった書き込みは SSD 側で効率化してます) さらに 512KB サイズの書き込みを想定すると、ブロックのセグメントがキッチリ 512KB の境界をまたがないでならば 1 回の書き込みですが、OS からのファイルアクセスでは、ブロックの途中から書かれるケースもあるので、512KB 以下のデータ書きとはいえ、半分の確率で 2 回書き込まれます。つまり

24576000 回 × 75 % = 18432000 回 (これは 512KB だけの書き込みを想定)

さらに OS からのアクセスではもう一つ書き換え回数が追加されます。それはファイルのクローズ処理によるファイルコントロールブロックの更新で最低 1 回書き換えが発生します。もちろん SSD ではキャッシュを使ったりしてさらに書き換え回数を低減しておりますが、基本的にはこういった処理系になります。

#### ●なるべく大きな SSD を使用する

例えば前記の 4GB の SSD では 8192 個のブロックが存在しますが、8GB ならその 2 倍の 16384 個のブロックが存在するので書き換え回数が単純に 2 倍になります。

SSD が同じブロックサイズならサイズに応じて単純に比例して書き換え回数がアップするわけです。

ゆえに可能な限りサイズの大きい SSD を利用すれば書き換え回数によるトラブルへの耐性が上がっていくわけです。

#### ●tmpfs の利用で SSD 書き込みを軽減する

Linux システムの場合、何の考慮もしないでシステム開発すると、ストレージは無限に使えるデバイスとしての前提で利用されてしまいます。

データ保存の必要もなくともストレージ領域をワーキングのためのバッファとして普通に使われてしまいます。

こういった事で SSD の寿命を縮める事の無いように、ワーキング処理に必要なストレージはなるべく tmpfs に配置する様、考慮してシステムを設計して行きます。

また、オープンソースのソフトウェアでも、自分用に確保したストレージ領域をテンポラリーとして利用するケースも多く、この場合は、そのファイルを tmpfs の領域にリンクを張って対処します。



## ●ログ

Linux システムでは何事においてもストレージにログを残す事が一般的ですが非常に細かくログを残すプロセスがある場合が、一度 **tmpfs** に書かせてから定期的にまとめてログを **SSD** に移すなどの工夫が必要です。

こういった対策では、突然の電源断には対応出来ませんが、この点はトレードオフしてあきらめるか、例えば **UPS** 対策されたシスログサーバーにログを送信するかそういった考慮を検討する必要があります。

## 2-3. SIM について

OpenBlocks IoT Family にて、搭載可能な SIM 形状は一部の型番の物を除き **mini-SIM(2FF)**です。**mini-SIM** 対応モデルにて **micro-SIM** 及び **nano-SIM** を使用する場合には、脱落防止フィルム有及び接着テープ有で **SIM** を固定できるアダプタを使用してください。尚、**SIM** アダプタを使用した場合での **SIM** スロットの破損は有償修理対象となります為、ご注意ください。

## 2-4. eMMC ストレージパーティション情報

本装置に使用されている eMMC のパーティション情報は以下となります。

### ●OpenBlocks IoT VX1 の場合

番号	format	サイズ	OBS 用途	デバイス名
1	fat16	1.5Gbyte	Boot	mmcblk0p1
2	ext4	6.5Gbyte	Primary	mmcblk0p2

### ●OpenBlocks IoT VX2 の場合

番号	format	サイズ	OBS 用途	デバイス名
1	fat16	1.5Gbyte	Boot	mmcblk0p1
2	ext4	30.5Gbyte	Primary	mmcblk0p2

●OpenBlocks IoT BX 及び EX シリーズの場合

番号	format	サイズ	OBS 用途	デバイス名
1	-	1MB	通常起動用 u-boot	mmcblk0p1
2	-	2MB	通常起動用 u-boot 環境変数	mmcblk0p2
3	-	1MB	非常時起動用 u-boot	mmcblk0p3
4	-	2MB	非常時起動用 u-boot 環境変数	mmcblk0p4
5	ext2	1MB	シリアル番号、MAC アドレス等	mmcblk0p5
6	-	1MB	カーネルパニックメッセージ保存用	mmcblk0p6
7	fat16	210MB	通常起動用 FW 用	mmcblk0p7
8	ext4	105MB	バックアップ領域	mmcblk0p8
9	fat16	210MB	リカバリー起動用 FW 用	mmcblk0p9
10	ext4	3351MB	Primary	mmcblk0p10

## 2-5. ストレージモード

基本的なユーザーランドデータを eMMC から参照し動作を行います。不意な電源断が起こった場合、物理ストレージ上のファイルが破損する恐れがありますが、**Docker** 等のアプリケーション側では **unionfs** にてストレージ上のデータ参照を行いますので問題なく稼働が行えます。(Docker は OpenBlocks IoT BX 及び EX シリーズでは動作いたしません)

唐突な電源断によるファイル破損は、主に書き込み中のファイルが対象となります。そのため、基本システムへの影響は発生しにくいよう通常書き込むファイルはログファイル程度に抑えるのを推奨いたします。

### ※mount コマンドの実行結果

```
root@obsiot:/var/webui/docroot# mount
/dev/mmcblk0p2 on / type ext4 (rw,relatime,data=ordered)
devtmpfs on /dev type devtmpfs (rw,relatime,size=956312k,nr_inodes=239078,mode=755)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,mode=755)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup
(rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/lib/systemd/systemd-cgroups-agent,name=systemd)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/debug type cgroup (rw,nosuid,nodev,noexec,relatime,debug)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs
(rw,relatime,fd=37,pgrp=1,timeout=0,minproto=5,maxproto=5,direct)
mqueue on /dev/mqueue type mqueue (rw,relatime)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
tmpfs on /tmp type tmpfs (rw,relatime)
```

## 2-6. PC と OpenBlocks IoT Family の接続

OpenBlocks IoT Family と PC を添付品の USB ケーブルを用いて接続します。

また、OpenBlocks IoT VX シリーズについては AC アダプタ経由での電源供給またはワイドレンジ電源入力経由での電源供給をしている必要があります。



Windows PC の場合、USB ポートに接続されると自動的に USB シリアルドライバがインストールされます。(Windows PC がインターネット環境につながっている場合です。)ドライバのインストールが完了したら、TeraTerm や PuTTY などのターミナルソフトでシリアルポート接続が可能となります。

尚、OpenBlocks IoT Family のシリアルポートのデフォルト通信パラメータは以下の通りです。

通信速度 : 115200bps

データ長 : 8bit

パリティ : 無し

ストップ : 1bit

通信が確立後、起動が完了するとログインプロンプトが表示されます。デフォルトの root 権限でログイン操作を開始して下さい。

login : root

password : 0BSIoT

※パスワードはデフォルトの場合となります。WEB UI にてパスワードを変更している場合には、設定したパスワードを用いてください。

## 2-7. WEB UI

本システムには基本的なシステム設定が行えるように WEB UI を搭載しています。

ネットワーク設定やモバイル回線の制御等は、WEB UI から実施してください。

また、WEB UI から各種機能の追加等が行えますので、基本的な使用方法等については WEB UI セットアップガイドを参照してください。追加した機能については各種機能のマニュアルを参照してください。

## 第 3 章 ご利用方法

本装置は Debian GNU/Linux を採用した汎用サーバー製品です。そのため、通常の Debian と同様にご利用いただいて構いません。また、本項では本装置特有の情報を記載しております。

### 3-1. ステータスインジケータの表示色

OpenBlocks IoT Family の正面にある LED は RGB の組み合わせで 7 色に点灯し、それぞれの表示色で点滅などをスクリプト制御しています。

WEB UI が使用されている場合には、デフォルトの LED 点灯仕様は以下の通りになります。

状態	色	点灯状態	備考
本体及び OS 起動中	黄	点灯	本体起動及び OS 起動が終わるとモバイル回線の電波受信チェックへ移行します。 ※SIM が挿入されていない場合は緑点滅。
SIM スロット未使用時	緑	点滅	SIM が無い状態での正常稼働または電波受信待機状態への移行待ち状態。
モバイル回線電波：強	白	点滅	電波状態詳細参照。
モバイル回線電波：中	水色	点滅	電波状態詳細参照。
モバイル回線電波：弱	青	点滅	電波状態詳細参照。 ※この電波強度での通信はリトライが多発する可能性があります。そのため、モバイル回線を使用する場合にはなるべく電波強度が中以上の状態にて使用してください。
モバイル回線電波：圏外	紫	点滅	電波状態詳細参照。
FUNC ボタンによる機能有効時	黄	点滅	モバイル回線や SIM スロット未使用時におけるステータスインジケータと交互点滅となります。
OS 終了中	黄	点灯	
AirManage 初回アクセス失敗時	赤	点灯	AirManage リモート管理サーバへの初回アクセスが失敗した際に表示となります。WEB UI 未使用時の場合は 5 分後に OS が終了します。

状態	色	点灯状態	備考
強制 SIM モード時におけるモデムデバイスファイルまたは SIM カード認識失敗時	赤	点灯	モデム搭載モデルにおいて強制 SIM モードを有効にし、起動時にモデムのデバイスファイルが存在しないまたは SIM カードが認識できない場合の表示となります。5 分後に OS が再起動します。
AirManage 及び SIM 認識失敗時の OS 終了中	赤	点灯	

※電波状態詳細

HW / モデム種別	電波：強	電波：中	電波：弱	電波：圏外
BX1	-87dBm 以上	-88～-108dBm	-109～-112dBm	-113dBm 以下
BX3	-87dBm 以上	-88～-108dBm	-109～-112dBm	-113dBm 以下
BX5 ※3G 時	アンテナ 3 本時	アンテナ 2 本時	アンテナ 1 本時	圏外時
BX5 ※LTE 時	-95dbm 以上	-95.1～-105dBm	-105.1～-120dBm	-120.1dBm 以下
3G モジュール (NTT ドコモ)	-87dBm 以上	-88～-108dBm	-109～-112dBm	-113dBm 以下
3G モジュール (ソフトバンク)	-87dBm 以上	-88～-108dBm	-109～-112dBm	-113dBm 以下
LTE/3G モジュール (ソフトバンク)	AT+CSQ 値： 14 以上	AT+CSQ 値： 13～3	AT+CSQ 値： 2～1	AT+CSQ 値： 0 または 99
LTE モジュール (KDDI)	アンテナ 2 本以上	アンテナ 1 本時	アンテナ 0 本時	圏外時
LTE モジュール (NTT ドコモ)	アンテナ 3 本時	アンテナ 2 本時	アンテナ 1 本時	圏外時
LTE モジュール (NTT ドコモ/KDDI) ※3G 時	アンテナ 3 本時	アンテナ 2 本時	アンテナ 1 本時	圏外時
LTE モジュール (NTT ドコモ/KDDI) ※LTE 時	-95dbm 以上	-95.1～-105dBm	-105.1～-120dBm	-120.1dBm 以下
BWA モジュール	-95dbm 以上	-95.1～-105dBm	-105.1～-120dBm	-120.1dBm 以下

### ※電波状態判定

モデム種別	回線	電波状態判定
BX1 (3G)	3G	RSSI による判定
BX3 (3G))	3G	RSCP によるマッピング値判定
BX5 (LTE/3G)	3G	ECIO 及び RSCP による判定
	LTE	RSRP による判定
3G モジュール(NTT ドコモ)	3G	RSCP によるマッピング値判定
3G モジュール(ソフトバンク)	3G	RSCP によるマッピング値判定
LTE/3G モジュール(ソフトバンク)	LTE/3G	AT+CSQ 値による判定
LTE モジュール(KDDI)	LTE	モデムモジュールによる判定
LTE モジュール(NTT ドコモ)	LTE	モデムモジュールによる判定
LTE モジュール (NTT ドコモ/KDDI)	3G	ECIO 及び RSCP による判定
	LTE	RSRP による判定
BWA モジュール	LTE	RSRP による判定

### WEB UI 使用時の LED 点灯制御スクリプト

/var/webui/bin/set\_signal\_value.sh

/var/webui/scripts/led\_updater.sh

### ●LED の制御

LED の表示色・点灯状態を変更する場合は、/tmp/.runled ファイルの内容を編集します。  
尚、WEB UI 使用時における SIM 挿入時は電波強度と連動する為、定期的に本ファイルが更新されます。そのため、意図的に変更する場合は、WEB UI を未使用とするか LED 点灯制御スクリプトを終了させてください。

行	設定内容	備考
1 行目	点灯 1 時間(msec)	1 以上
2 行目	点灯 2 時間(msec)	1 以上
3 行目	点灯 1 色番号	下表参照
4 行目	点灯 2 色番号	下表参照(省略した場合、0)



色番号	色
0	非点灯
1	赤
2	緑
3	黄
4	青
5	紫
6	水色
7	白
範囲外	非点灯

※1 秒毎に黄色点滅

```
# echo -e "1000¥n1000¥n3" > /tmp/.runled
```

※1 秒毎に黄色/緑色交互点滅

```
# echo -e "1000¥n1000¥n3¥n2" > /tmp/.runled
```

## 3-2. モバイル回線向けモデム制御

本装置搭載のモデムモジュールの電源 ON/OFF や電波状態をチェックするツールを搭載しています。

コマンド名 : atcmd

起動方法 1 : atcmd [コマンド]

起動方法 2 : atcmd [コマンド 1] [コマンド 2] [コマンド 3]

起動方法 3 : atcmd -d [デバイスファイル] [コマンド]

起動方法 2 のように、コマンドを列挙して順次実行することが可能です。

また、起動方法 3 のように使用するデバイスファイルを指定実行することも可能です。

※FW3.3.1 以降ではデフォルトで”/dev/ttyMODEM0”のデバイスファイルを参照します。  
(FW3.3.0 以前のデフォルトは”/dev/ttyACM0”を参照します。) HW 構成によっては  
ttyACM0 にモデム以外のデバイスが割り当てられることがある為、起動方法 3 のように明  
示的にデバイスファイルの指定またはアップデートの適用を強く推奨いたします。

コマンド	機能	備考
PON	モデム電源 ON	
POFF	モデム電源 OFF	
PRST	モデム reboot	ソフトウェアリセット (一部、ハードウェアリセット)
HRST	モデム reboot	ハードウェアリセット。
SMONI	電波強度取得	BX1 限定
CSQ	電波強度取得	LTE モジュール(NTT ドコモ/KDDI)及び BWA モジュールでは、正しい電波強度取得は行えません。
SIND	基地局からの時刻取得	BX1 限定
CCLK	基地局からの時刻取得	BX3 限定
CCID	SIM の番号取得	
CTZU 1	タイムゾーン自動取得	BX1 及び EX/VX シリーズ LTE モジュール(NTT ドコモ)以外のモデルのみ対応 ※スペースが入ります。
ATI	モデム型番取得	
CGSN	モデムシリアル番号取得	

コマンドは上記表の内容で指定します。

※電源 ON、SIM 番号取得、電源 OFF

```
# atcmd PON CCID POFF
XXXXXXXXXXXXXXXX
```

※電源 ON、電波強度取得、電源 OFF

```
# atcmd PON CSQ POFF
-86
```

WEB UI では常時電波状況取得によりデバイスファイルを占有しています。そのため、本コマンドは WEB UI にて使用していないデバイスファイルを指定し実行してください。尚、WEB UI を使用していない環境またはユーザー制御では本制約はありません。

尚、使用モジュールによって **atcmd** にて使用できるデバイスファイルが異なります。

●BX1

デバイスファイル	atcmd 使用可否
/dev/ttyMODEM0	使用可能
/dev/ttyMODEM1	使用不可
/dev/ttyMODEM2	使用不可
/dev/ttyMODEM3	使用可能
/dev/ttyMODEM4	使用可能
/dev/ttyMODEM5	使用不可
/dev/ttyMODEM6	使用不可

●BX3

デバイスファイル	atcmd 使用可否
/dev/ttyMODEM0	使用可能
/dev/ttyMODEM1	使用可能
/dev/ttyMODEM2	使用可能
/dev/ttyMODEM3	使用不可
/dev/ttyMODEM4	使用不可
/dev/ttyMODEM5	使用不可
/dev/ttyMODEM6	使用不可

●BX5

デバイスファイル	atcmd 使用可否
/dev/ttyMODEM0	使用可能
/dev/ttyMODEM1	使用不可
/dev/ttyMODEM2	使用不可
/dev/ttyMODEM3	使用可能
/dev/ttyMODEM4	使用不可

●EX/VX シリーズ LTE/3G モジュール(ソフトバンク)

デバイスファイル	atcmd 使用可否
/dev/ttyMODEM0	使用可能

●EX/VX シリーズ LTE モジュール(KDDI)

デバイスファイル	atcmd 使用可否
/dev/ttyMODEM0	使用可能

●EX/VX シリーズ LTE モジュール(NTT ドコモ)

デバイスファイル	atcmd 使用可否
/dev/ttyMODEM0	使用可能
/dev/ttyMODEM1	使用可能

●EX/VX シリーズ LTE モジュール(NTT ドコモ/KDDI)

デバイスファイル	atcmd 使用可否
/dev/ttyMODEM0	使用可能
/dev/ttyMODEM1	使用不可
/dev/ttyMODEM2	使用不可
/dev/ttyMODEM3	使用可能
/dev/ttyMODEM4	使用不可

●EX/VX シリーズ BWA モジュール

デバイスファイル	atcmd 使用可否
/dev/ttyMODEM0	使用可能
/dev/ttyMODEM1	使用不可
/dev/ttyMODEM2	使用不可
/dev/ttyMODEM3	使用可能
/dev/ttyMODEM4	使用不可

●EX1/3G モジュール(NTT ドコモ)

デバイスファイル	atcmd 使用可否
/dev/ttyMODEM0	使用可能
/dev/ttyMODEM1	使用可能
/dev/ttyMODEM2	使用可能
/dev/ttyMODEM3	使用不可
/dev/ttyMODEM4	使用不可
/dev/ttyMODEM5	使用不可
/dev/ttyMODEM6	使用不可

●EX1/3G モジュール(ソフトバンク)

デバイスファイル	atcmd 使用可否
/dev/ttyMODEM0	使用可能
/dev/ttyMODEM1	使用可能
/dev/ttyMODEM2	使用可能
/dev/ttyMODEM3	使用不可
/dev/ttyMODEM4	使用不可
/dev/ttyMODEM5	使用不可
/dev/ttyMODEM6	使用不可

### 3-3. モバイル回線接続

モバイル回線モデムの制御は WEB UI にて行っております。

WEB UI 制御における必要時のみモバイル回線接続する方式(オンデマンド接続)及び WEB UI からモデムの制御を完全に排他したユーザー制御方式をサポートしています。

#### 3-3-1. オンデマンド接続

WEB UI のネットワーク→基本タブにアクセスし、モバイル回線の接続に必要な情報を設定します。

サービスネットワーク (モバイル回線) (?) モデム制御項目を表示する

使用設定	<input checked="" type="radio"/> 使用する <input type="radio"/> 使用しない
APN	<input type="text" value="xxxx"/>
ユーザ名	<input type="text"/>
パスワード	<input type="password"/> <input type="checkbox"/> パスワードを表示する
認証方式	CHAP ▼
自動接続	<input checked="" type="radio"/> 自動接続する <input type="radio"/> 自動接続しない
通信確認用ホスト (?)	<input type="text" value="8.8.8.8"/>
定期再接続設定	<input type="radio"/> 定期再接続をする <input checked="" type="radio"/> 定期再接続をしない
SMSコントロール (?)	<input checked="" type="radio"/> 無効 <input type="radio"/> 有効

必要な情報は以下となります。

- ・ APN(LTE モジュール(KDDI)時は項目が有りません)
- ・ ユーザ名
- ・ パスワード
- ・ 認証方式
- ・ 自動接続：“自動接続しない”を選択してください。
- ・ 通信確認用ホスト
- ・ 定期再接続設定  
(・ モバイル回線再接続時間)
- ・ SMS コントロール(LTE モジュール(KDDI)時は項目が有りません)  
※SMS コントロールは使用する場合にのみ設定してください。

保存ボタンを押し、再起動することでモバイル回線モデムへの反映は完了します。

また、モバイル回線のオンデマンド接続及び切断のコマンドは以下となります。尚、下記のコマンドラインにおける数字の”1”はオンデマンド接続を前提とするものです。”0”を指定した場合には、接続を保持する常時接続となります。

#### ●モバイル回線の接続

```
# /var/webui/scripts/mobile_control.sh con 1
```

### ●モバイル回線の切断

```
# /var/webui/scripts/mobile_control.sh coff 1
```

上記のコマンドを組み合わせたシェルスクリプトのサンプルは以下です。  
内容は以下となっております。

- 1, モバイル回線の接続。
- 2, DNS サーバに対して **ping** コマンドを実行。
- 3, モバイル回線の切断。

```
#!/bin/bash

echo "#-----#"
echo "# Connect (`date`)"
echo "#-----#"

/var/webui/scripts/mobile_control.sh con 1

sleep 2

echo ""
echo "#-----#"
echo "# Command Exec (`date`)"
echo "#-----#"

ping -c 3 8.8.8.8

echo ""
echo "#-----#"
echo "# Disconnect (`date`)"
echo "#-----#"

/var/webui/scripts/mobile_control.sh coff 1

sleep 2

exit 0
```

## 3-3-2. モバイル回線モデムのユーザー制御

本項は OpenBlocks IoT Family では未対応の 3G モジュール向けの説明となっております。  
通常ラインナップのモジュールではございませんのでご注意ください。

WEB UI のネットワーク→基本タブにアクセスし、モバイル回線の接続に必要な情報を設定します。尚、LTE モジュールは本機能に対応しておりません。

本項目を実施する場合には、「モデム制御項目を表示する」にチェックをしてください。

サービスネットワーク (モバイル回線) (?) ☒ モデム制御項目を表示する

使用設定 ☒ 使用する ☐ 使用しない

モデム制御 (?) ☒ ユーザー制御 ☐ WEB UI制御

APN

ユーザ名

パスワード  ☐ パスワードを表示する

認証方式

設定に必要な情報は以下となります。

- ・モデム制御：**”ユーザー制御”を選択してください。**
- ・APN
- ・ユーザ名
- ・パスワード
- ・認証方式

WEB UI にて設定した情報は以下のコマンドでモデムへ反映可能です。

●モデム設定反映コマンド

```
# /var/webui/scripts/setapn.sh
```

モデム設定反映コマンドで使用しているコンフィグファイルは以下となります。

●モデム設定反映用コンフィグファイル

```
/var/webui/config/ppp0_device.sh  
/var/webui/upload_dir/modem.sh
```

設定情報をさらに変更した場合には、以下のファイルを作成し設定することにより情報を上書きすることが可能です。

●情報上書き用ファイル

```
/var/webui/upload_dir/user_modem.sh
```

●変数内容

変数名	内容	備考
modem_ppp0_apn	APN	
modem_ppp0_user	ユーザー名	
modem_ppp0_pass	パスワード	



変数名	内容	備考
modem_ppp0_authtype	認証方式	PAP または CHAP
modem_ppp0_provier	プロバイダ名	pon/poff コマンドでの指定
DEVICE_CONNECT	モバイル回線接続用 デバイスファイル	/dev/ttyMODEM[0-9]* と な ります。
DEVICE_SETTING	モバイル回線設定用 デバイスファイル	/dev/ttyMODEM[0-9]* と な ります。

#### ●設定サンプル

```
modem_ppp0_apn="iixxxx.jp"
modem_ppp0_user="test@iixxxx"
modem_ppp0_pass="xxxx"
modem_ppp0_authtype="PAP"
modem_ppp0_provier="usermobile"
```

以下の各コマンドにより、モバイル回線の接続及び切断が可能です。


#### ●モバイル回線接続コマンド


```
# pon <modem_ppp0_provier>
```

#### ●モバイル回線切断コマンド

```
# poff <modem_ppp0_provier>
```

※<modem\_ppp0\_provier>は変数で設定した物となります。user\_modem.sh にて変更していない場合は”mobile”となります。

- 

●電波強度等の取得については、atmed を使用してください。尚、回線接続しているデバイスファイルに対しては取得できません。
- 

●モバイル回線の接続前にデフォルトゲートウェイが設定されている場合、モバイル回線接続時には反映されません。そのため、モバイル回線接続までにデフォルトゲートウェイを解除してください。  
また、モバイル回線切断後はデフォルトゲートウェイを解除している状態となります。そのため、インターネット環境へのアクセスを別途する場合等では、デフォルトゲートウェイを設定してください。

## 3-4. バックアップ

作りこみを行った本システムデータのバックアップは以下の方法にて行えます。

### ◆OpenBlocks IoT VX シリーズの場合

#### 1. 本体再起動

```
# sync  
# reboot
```

#### 2. エマージェンシーブートモードによる起動

OpenBlocks IoT VX シリーズの場合、起動時の GRUB 画面にて『Emergency boot』を選択して下さい。

```
GNU GRUB version 0.97 (252K lower / 523264K upper memory)

+-----+
| o OBS IoT VX - Normal boot  
| x OBS IoT VX - WebUI init boot  
| x OBS IoT VX - Emergency boot  
| x OBS IoT VX - Factory Image  
+-----+

Board: Aptio CRB  
Platform: Intel Bay Trail-I  
Hardware Secure Boot: Inactive  
UEFI Secure Boot: Inactive  
System Mode: Setup  
UEFI Secure Boot Mode: Custom  
GRUB Verified Boot: Unsupported  
Boot Device: UEFI OS  
Initial Root Device: (hd0,0)

Use the ^ and v keys to select which entry is highlighted.  
Press enter to boot the selected OS, 'e' to edit the  
commands before booting, 'a' to modify the kernel arguments
```

### 3. 不要データの削除及びバックアップ

```
obsiot login: root
Password:root

# mount /dev/mmcblk0p2 /mnt
    ※/mnt 以下の HW 固有のファイル等は削除してください。以下は BT 情報を削除して
    います。
# rm -rf /mnt/var/lib/bluetooth/*

# cd /mnt

# tar --exclude=lost+found --exclude=<tgz ファイル名> -cpzf <tgz ファイル名> .
```

/mnt/<tgz ファイル名>がバックアップしたファイルとなります。

#### 注意)

/var/lib/bluetooth/ディレクトリ配下に HW 固有の情報が含まれています。

また、他にも作りこみの際の中間ファイル等も削除した方が良いです。

### ◆OpenBlocks IoT BX/EX シリーズの場合

#### 1. リカバリーブート起動

OpenBlocks IoT BX/EX シリーズでは起動時に uboot 起動中にキー入力を行う事で、uboot プロンプトが表示されます。uboot プロンプトにて、リカバリーFW 起動用コマンドを実施し RAM ディスクモードにて起動することが行えます。

```
*****
PSH KERNEL VERSION: b0182727
WR: 20104000
*****

SCU IPC: 0x800000d0 0xffffce92c

PSH miaHOB version: TNG.B0.VVBD.0000000c

microkernel built 23:15:13 Apr 24 2014

***** PSH loader *****
PCM page cache size = 192 KB
Cache Constraint = 0 Pages
Arming IPC driver ..
Adding page store pool ..
PagestoreAddr(IMR Start Address) = 0x04899000
pageStoreSize(IMR Size)           = 0x00080000

*** Ready to receive application ***

U-Boot 2014.04 (Oct 14 2014 - 15:19:04)
```

```
Watchdog enabled
DRAM: 980.6 MiB
MMC: tangier_sdhci: 0
In: serial
Out: serial
Err: serial
Hit any key to stop autoboot: 0 ※キー入力
boot> run bootRecovery
```

## 2. 不要データの削除及びバックアップ

```
obsiot login: root
Password: root

# mount /dev/mmcb1k0p10 /mnt
  ※/mnt 以下の HW 固有のファイル等は削除してください。以下は BT 情報を削除しています。
# rm -rf /mnt/var/lib/bluetooth/*

# cd /mnt

# tar --exclude=lost+found --exclude=<tgz ファイル名> -cpzf <tgz ファイル名> .
```

/mnt/<tgz ファイル>がバックアップしたファイルとなります。

### 注意)

/var/lib/bluetooth/ディレクトリ配下に HW 固有の情報が含まれています。  
また、他にも作りこみの際の中間ファイル等も削除した方が良いです。

## 3-5. リストア

エマージェンシーブート(リカバリブートモード)にてボリュームラベルに「DEB\_CONFIG」を設定したストレージ(ext2/ext3/vfat の USB メモリ等)に後述の応用の init.sh を用い、Primary パーティションにバックアップした<tgz ファイル>の展開を行いリストアします。

### 注意)

- ・バックアップしたファイルの展開前に Primary となるパーティションストレージをフォーマットしてください。(OpenBlocks IoT VX シリーズ: /dev/mmcb1k0p2、OpenBlocks IoT BX/EX シリーズ: /dev/mmcb1k0p10)
- ・バックアップしたファイルと kernel-image は一致している必要があります。そのため、リストア対象の筐体の kernel-image は別途確認してください。

## 3-6. 応用

エマージェンシーブート(リカバリブートモード)にてバックアップファイルの他、所定のファイル名でファイルをボリュームラベルに「DEB\_CONFIG」内に用意することで、スクリプト実行などが可能です。いずれもファイルが存在する場合のみ有効になります。

- ・ `init.sh`(`sh` スクリプト、改行コードは LF のみ)

KERNEL 起動中において `overlayfs`(または `aufs`)のマウント前に、実行を行います。

- ・ `post-init.sh`(`sh` スクリプト、改行コードは LF のみ)

KERNEL 起動中において `overlayfs`(または `aufs`)のマウント完了後に、実行を行います。

尚、エマージェンシーブート(リカバリブートモード)では、**RAMDISK** モードで起動します。そのため、上記のマウント作業については特に考慮する必要はありません。

※本製品では `systemd` を採用しております。そのため、KERNEL 起動中の段階では `systemd` が起動していないため一部コマンド(`poweroff/reboot` コマンド等)が使用できません。これらのコマンド等を使用する場合には、処理したいコマンド等をバックグラウンドで起動すべき `daemon` 等が起動完了後に実行してください。

## 3-7. 推奨使用デバイスファイル

Linux では各デバイスファイル等は認識順に名前がアサインされます。

そのため、デバイス自体の電源 ON/OFF 状況によってデバイスファイルが異なる恐れがあります。

モデル毎に自動でデバイスファイルのリンクを張りますので、デバイスファイルへのアクセスを行う場合には以下の推奨デバイスファイルを使用してください。

### ●OpenBlocks IoT VX シリーズの場合

推奨デバイスファイル	対象デバイス
<code>/dev/ttyRS485</code>	RS-485 用デバイスファイル
<code>/dev/ttyEX1</code>	拡張スロット 1 用デバイスファイル (LoRaWAN 等)
<code>/dev/ttyEX2</code>	拡張スロット 2 用デバイスファイル (EnOcean 等)

推奨デバイスファイル	対象デバイス
/dev/ttyS4	RS-232C 用デバイスファイル (VX1 のみとなります。)

●OpenBlocks IoT EX1 の場合

推奨デバイスファイル	対象デバイス
/dev/ttyEX1	拡張スロット 1 用デバイスファイル (LoRaWAN 等)
/dev/ttyEX2	拡張スロット 2 用デバイスファイル (EnOcean 等)
/dev/ttyMFD1	RS-232C または RS-485 用デバイスファイル

●OpenBlocks IoT EX1G の場合

推奨デバイスファイル	対象デバイス
/dev/ttyRS485	RS-485 用デバイスファイル
/dev/ttyEX1	拡張スロット 1 用デバイスファイル (LoRaWAN 等)
/dev/ttyEX2	拡張スロット 2 用デバイスファイル (EnOcean 等)
/dev/ttyMFD1	RS-232C 用デバイスファイル

### 3-8. リカバリー起動

各モデルにて通常起動で用いている FW データやストレージデータが破損した場合、リカバリー用の FW にて起動することが出来ます。

OpenBlocks IoT VX シリーズでは GRUB メニューの「Emergency boot」を選択することで、RAMdisk モードにて起動することが行えます。

本起動によるコンソールによるログインアカウント及びパスワードは”root”/”root”となります。

```
GNU GRUB version 0.97 (252K lower / 523264K upper memory)

+-----+
| o OBS IoT VX - Normal boot          |
| x OBS IoT VX - WebUI init boot      |
| x OBS IoT VX - Emergency boot      |
| x OBS IoT VX - Factory Image       |
+-----+

Board: Aptio CRB
Platform: Intel Bay Trail-I
Hardware Secure Boot: Inactive
UEFI Secure Boot: Inactive
System Mode: Setup
UEFI Secure Boot Mode: Custom
GRUB Verified Boot: Unsupported
Boot Device: UEFI OS
Initial Root Device: (hd0,0)

Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, 'a' to modify the kernel arguments
```

OpenBlocks IoT BX/EX シリーズでは起動時に uboot 起動中にキー入力を行う事で、uboot プロンプトが表示されます。

uboot プロンプトにて、リカバリーFW 起動用コマンドを実施し RAM ディスクモードにて起動することが行えます。

リカバリーFW 起動後に、kernel-image データを再インストールやストレージデータの復旧等を実施してください。

```
*****
PSH KERNEL VERSION: b0182727
WR: 20104000
*****

SCU IPC: 0x800000d0 0xffce92c

PSH miaHOB version: TNG.B0.VVBD.0000000c

microkernel built 23:15:13 Apr 24 2014
```

```
***** PSH loader *****
PCM page cache size = 192 KB
Cache Constraint = 0 Pages
Arming IPC driver ..
Adding page store pool ..
PagestoreAddr(IMR Start Address) = 0x04899000
pageStoreSize(IMR Size)          = 0x00080000

*** Ready to receive application ***

U-Boot 2014.04 (Oct 14 2014 - 15:19:04)

        Watchdog enabled
DRAM:  980.6 MiB
MMC:   tangier_sdhci: 0
In:    serial
Out:   serial
Err:   serial
Hit any key to stop autoboot:  0          ※キー入力
boot > run bootRecovery
```

尚、WEB UI を用いている場合通常起動時においてコマンドラインにより以下のコマンドを実行することで次回以降の起動をリカバリーモード起動することが行えます。また、同様にリカバリーモード時から次回以降の起動を通常起動とするコマンドについても以下となります。

※通常起動からリカバリーモード起動への切り替えコマンド

```
# /var/webui/bin/bxex_uboot_runrecovery.sh
```

※リカバリーモード起動から通常起動への切り替えコマンド

```
# /usr/sbin/fw_setenv bootcmd 'setenv firm_part 0:7;run bootDebian'
```

### ***Attension)***

通常起動からリカバリーモードへの切り替えコマンドを実施した場合、次回以降の起動は常にリカバリーモードとなります。そのため、実配置している OpenBlocks では実行しないでください。



## 3-9. クロス開発環境の構築

OpenBlocks IoT Family のファームウェアの作成方法を、以下のページに用意しています。  
開発環境の構築を行う方はご確認ください。

[https://github.com/plathome/debian\\_based\\_firmware](https://github.com/plathome/debian_based_firmware)

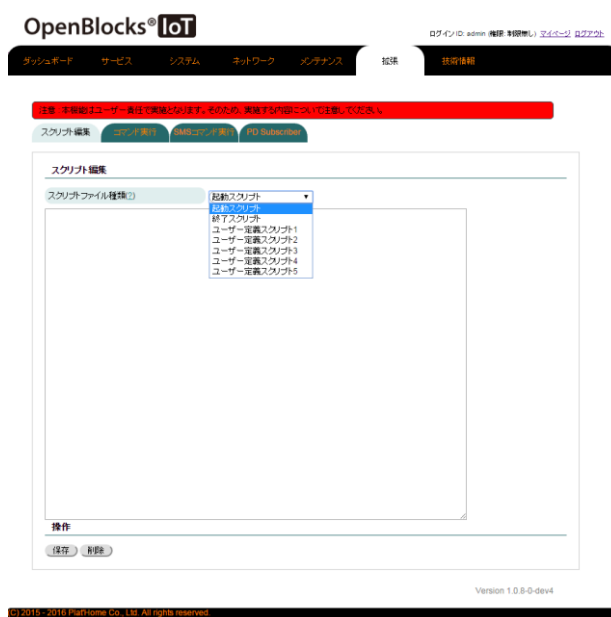
## 3-10. WEB UI 拡張機能

WEB UI 上にて、拡張機能として以下を用意しております。

### 3-10-1. WEB UI への機能拡張インストール

出荷直後状態の本筐体では、ネットワーク設定等を設定するソフトウェアのみ組み込まれています。IoT Gateway として使用する場合や Node-RED 等をインストールする等の機能を拡張することが行えます。インストール方法については、OpenBlocks IoT Family 向け WEB UI セットアップガイドに記載しておりますので、そちらをご参照ください。

### 3-10-2. スクリプト編集



「拡張」→「スクリプト編集」タブにて、以下のスクリプトを WEB UI 上から作成・編集することが出来ます。

- ・起動スクリプト
- ・終了スクリプト
- ・ユーザー定義スクリプト 1~5
- ・ユーザー定義スクリプト(button)
- ・動体検知イベント

本機能はユーザー様の実装内容依存となりますので、ご注意ください。

#	スクリプト種類	実行タイミング	備考
1	起動スクリプト	本製品の起動時における WEB UI の起動処理完了後に実行されます。	
2	終了スクリプト	本製品の終了時における WEB UI の終了処理開始直後に実行されます。	
3	ユーザー定義スクリプト	通常処理中には実行されません。 SMS コントロール機能にて命令適用時に実施されます。	
4	ユーザー定義スクリプト (button)	FUNC 機能割当機能にて設定した場合に、FUNC スイッチを押すことで実施されます。	
5	動体検知イベント	カメラ機能における動体検知時の動画保存時に、イベントトリガーが有効となっている場合に実施されます。	

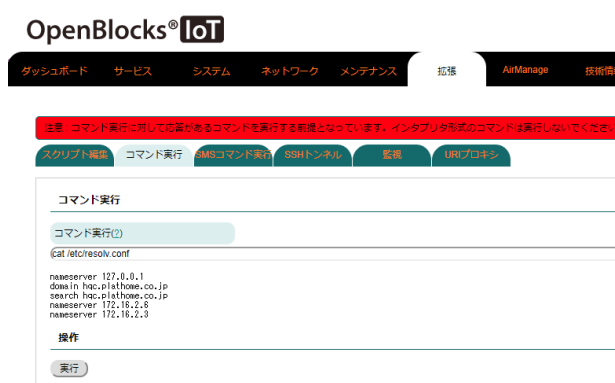
### 3-10-3. コマンド実行



「拡張」→「コマンド実行」タブから 1 行程度の  
コマンド実行することが可能です。

コマンドを実行すると応答結果が表示されま  
す。

本機能にてフォアグラウンドで永続稼働する  
ようなコマンドを実行した場合、WEB 応答し  
なくなりますので注意してください。

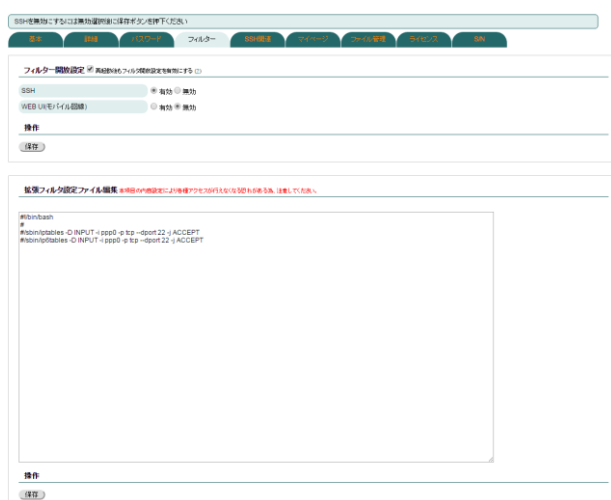


## 3-10-4. WEB UI フィルタテーブル特殊設定

“/var/webui/local/bin/iptables-ext.sh”というファイルが存在している場合、WEB UI のシステム→フィルタータブにて、“拡張フィルター設定編集”項目が表示されます。

本項目では、iptables 及び ip6tables コマンドによりフィルター設定をカスタマイズすることを前提としています。

iptables-ext.sh の実行タイミングは起動時やフィルター設定変更時となります。



本項目の内容はシェルスクリプトとなります。  
適宜 iptables コマンドにて編集を実施してください。

## 3-10-5. SMS 送信

WEB UI を用いておりモデムモジュール(LTE モジュール(KDDIを除く)が搭載され、SMS 送信可能な SIM が挿入されている場合、コマンドライン上から SMS を送信することが出来ます。

以下のコマンドにより、SMS データの雛形を作成します。

※雛形データを作成後、自動で送信されます。

### ●LTE モジュール(NTT ドコモ)のモデルの場合

```
# /var/webui/bin/create_sms_um04.php <宛先電話番号> <本文>
```

※実行例

```
# /var/webui/bin/create_sms_um04.php 09012345678 "TEST MESSAGE"
```

●LTE モジュール(NTT ドコモ)以外のモデル

```
# /var/webui/bin/create_sms.py <宛先電話番号> <本文>
```

※実行例

```
# /var/webui/bin/create_sms.py 09012345678 "TEST MESSAGE"
```

### 3-10-6. LTE モジュール(KDDI)運用切り替え

LTE モジュール(KDDI)はモジュール自体が SIM になる機能を持っています。以下のコマンドにより、モジュール自体が SIM の機能となる内部 SIM モード及び挿入されている SIM を参照する外部 SIM モードの切り替えが可能です。

●内部 SIM モードへの切替コマンド

```
# /var/webui/scripts/kym_set_mode.sh in
```

●外部 SIM モードへの切替コマンド

```
# /var/webui/scripts/kym_set_mode.sh out
```

### 3-10-7. WEB UI 自動外部ストレージマウント機能

WEB UI において特定のボリュームラベルの付いたデバイスが見つかった場合、自動でマウントされます。

WEB UI の機能等で保存先管理等を行う場合にご使用ください。

ボリュームラベル	マウント先	補足
WEBUI_STORAGE	/var/tmp/storage	ファイルシステムは NTFS をご使用ください。

## 3-10-8. ブートモード切替

### 【OpenBlocks IoT VX シリーズのみ】

「メンテナンス」→「ブートモード切替」タブから次回起動時のブートモードを設定することができます。



通常出荷状態では選択可能なブートモードは1つのみとなっています。

※/dev/mmcblk0p1/EFI/boot/bootx64.conf ファイル部の一部を書き換えることによって、この部分に表示することができますが、選択可能なブートモードでは工場出荷状態へ戻す等の実運用では危険な内容が含まれています。そのため、内容を判断の上表示追加等を行ってください。

### 3-10-9. GPS 情報について

WEB UI を用いており LTE モジュール(NTT ドコモ/KDDI)及び BWA モジュール搭載品において、GPS の使用設定を有効にし SIM が挿入されている場合、最終取得時点の位置情報を JSON 形式にて以下のファイルに書き込みを行っております。  
位置情報を活用したい場合にはこちらのファイルをご使用ください。

#### ●GPS 情報 JSON ファイル

/tmp/.gps\_posi.json

また、GPS の位置情報取得の際に上記ファイル内容を Node-RED の Unix ドメインソケットに対して書き込みを行います。

#### ●GPS 情報書き込み Unix ドメインソケット(Abstract ソケット)

¥0/tmp/node-red-gpsin.sock

※Node-RED にて GPS 情報を JSON へ追加する手法については、『Nod-RED スターターガイド』をご確認ください。

### 3-10-10. DHCP における DGW、DNS について

OpenBlocks IoT Family の WEB UI 機能にて、使用するネットワークインターフェースの IP アドレス取得設定にて DHCP を設定している場合、DGW(デフォルトゲートウェイ)及び DNS(DNS サーバー)情報は DHCP で設定された情報が反映されます。

そのため、通常では DHCP サーバー側にてデフォルトゲートウェイ及び DNS サーバー情報が設定されている為問題ありません。しかしセキュリティの関係上、デフォルトゲートウェイまたは DNS サーバー情報を配布しない DHCP サーバーのネットワークを運用している場合があります。このような場合、以下のファイルを対象フォーマットで用意することで DHCP 環境において強制的にデフォルトゲートウェイ及び DNS サーバーを設定することができます。

尚、上記における強制的に付与するデフォルトゲートウェイは DHCP における IP 付与時に設定されます。また、DNS サーバー設定に関してはローカルネットワークのインターフェースの Up 時(DHCP)に適用されます。

●対象ファイル

/var/webui/upload\_dir/force\_network.sh

●ファイルフォーマット

```
force_network_defaultroute=<デフォルトゲートウェイ IP アドレス>  
force_network_nameservers="<DNS サーバーIP アドレス 1> <DNS サーバーIP アドレス 2>"
```

●ファイルサンプル

```
force_network_defaultroute=192.168.130.1  
force_network_nameservers="127.0.0.1 192.168.130.1"
```

## 3-11. Docker について

OpenBlocks IoT VX シリーズでは Docker の動作が可能です。

WEB UI の拡張機能追加により Docker をインストールすることができます。

また、以下のコマンドでも同様にインストール可能です。

●Docker インストール方法(インターネット環境につながっている必要があります)

```
# apt-get update  
# apt-get -y install docker-ce
```

現状では、Docker コンテナを WEB UI から制御する機能はありません。コンテナ操作を行う場合には、コマンドラインにて実施してください。また、本番環境に Docker を用いる場合には、コンテナ起動・停止等のコマンドに関しては Docker 公式ページを参照してください。

## 第 4 章 注意事項及び補足

### 4-1. スクリプト処理による遅延処理等について

エマージェンシーブート時における `init.sh` 及び `post-init.sh` 等のスクリプトにおいてストレージへの書き込みを行った場合、書き込み完了前に次のコマンド処理へと実施される場合があります。そのため、明示的に `sleep` 及び `sync` コマンド等を実施してください。

### 4-2. 使用ポート一覧

WEB UI 込みでの OpenBlocks IoT Family では以下のポートを使用及び使用する可能性があります。

サービス種類	ポート番号	補足
FTP	21	FTP インストール時
SSH	22	ポート番号変更可能。
DNS	53	
DHCP	67	
NetBIOS	137	Samba インストール時(UDP)
NetBIOS	138	Samba インストール時(UDP)
NetBIOS	139	Samba インストール時
Samba	445	Samba インストール時
Modbus	502	IoT データ制御インストール時
WEB UI(HTTP アクセス)	880	
Node-RED	1880	Node-RED インストール時。 (ポート番号変更可能。)
ECHONET	3610	IoT データ制御/HVSMC 使用時
Shell in a box(WEB SSH)	4200	
WEB UI(HTTPS アクセス)	4430	
SSH	50022	LTE/3G モジュール(ソフトバンク) / WAN 側のみ
WEB UI(HTTP アクセス)	50880	LTE/3G モジュール(ソフトバンク) / WAN 側のみ
WEB UI(HTTPS アクセス)	54430	LTE/3G モジュール(ソフトバンク) / WAN 側のみ



サービス種類	ポート番号	補足
Node-RED	51880	LTE/3G モジュール(ソフトバンク) / WAN 側のみ
WEB UI 独自サービス	63003	

