

OpenBlocks IoT Family向け 開発者向けガイド



Ver.1.0.14

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。

■ 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をご遠慮ください。
- ・ 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
1-1. BX1 向けパッケージ内容	5
1-2. 各部の名称(BX1 本体)	6
1-3. BX3 向けパッケージ内容	7
1-4. 各部の名称(BX3 本体)	8
1-5. EX1 向けパッケージ内容	9
1-6. 各部の名称(EX1 本体)	10
1-6-1. EX1 RS-232C ピンアサイン	12
1-6-2. EX1 RS-485 ピンアサイン	12
1-7. BX0 向けパッケージ内容	13
1-8. 各部の名称(BX0 本体)	14
1-9. BX3L 向けパッケージ内容	15
1-10. 各部の名称(BX3L 本体)	16
第 2 章 ご利用の前に	17
2-1. 製品概要	17
2-2. SIM について	17
2-3. eMMC ストレージパーティション情報	17
2-4. RAM Disk モード	18
2-5. ストレージ併用モード	19
2-6. その他の運用方法	20
2-7. ウォッチドッグ	21
2-8. RTC	21
2-9. PC と OpenBlocks IoT Family の接続	21
2-10. WEB UI	22
第 3 章 ご利用方法	24
3-1. 省電力化	24
3-2. ステータスインジケータの表示色	24
3-3. モバイル回線向けモデム制御	27
3-4. モバイル回線接続	29
3-4-1. オンデマンド接続	29
3-4-2. モバイル回線モデムのユーザー制御	31
3-5. BT の起動	33
3-6. 専用コマンド(flashcfg)	33
3-7. バックアップ	35

3-8. リストア	36
3-9. 応用	37
3-10. WEB UI 拡張機能	38
3-10-1. スクリプト編集	38
3-10-2. コマンド実行	39
3-10-3. SMS コマンド実行	39
3-10-4. PD Subscriber	40
3-11. WEB UI フィルタテーブル特殊設定	43
3-12. SMS 送信	43
3-13. LTE モジュール for KDDI 運用切り替え	44
3-14. クロス開発環境の構築	44
3-15. Yocto Linux の起動	45
第 4 章 製品仕様	47
4-1. OpenBlocks IoT BX1 本体仕様	47
4-2. OpenBlocks IoT BX3 本体仕様	49
4-3. OpenBlocks IoT EX1 本体仕様	50
4-4. OpenBlocks IoT EX1 オプション	51
4-4-1. 3G モジュール For NTT ドコモ	51
4-4-2. 3G モジュール For ソフトバンク	51
4-4-3. LTE モジュール For KDDI	52
4-4-4. LTE モジュール For Docomo	52
4-4-5. EnOcean モジュール	52
4-4-6. Wi-SUN モジュール	53
4-5. OpenBlocks IoT BX0 本体仕様	53
4-5. OpenBlocks IoT BX3L 本体仕様	55
第 5 章 注意事項	56
5-1. スクリプト処理による遅延処理等について	56
5-2. SSD ベースのシステム開発における注意点	56

第 1 章 はじめに

本書は、OpenBlocks IoT Family(OpenBlocks IoT BX シリーズ及び OpenBlocks IoT EX シリーズ)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。

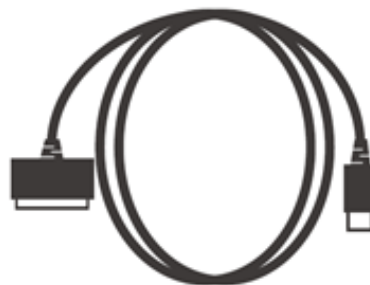
1-1. BX1 向けパッケージ内容

OpenBlocks IoT BX1 の標準品構成は以下となります。

BX1 本体 1 台



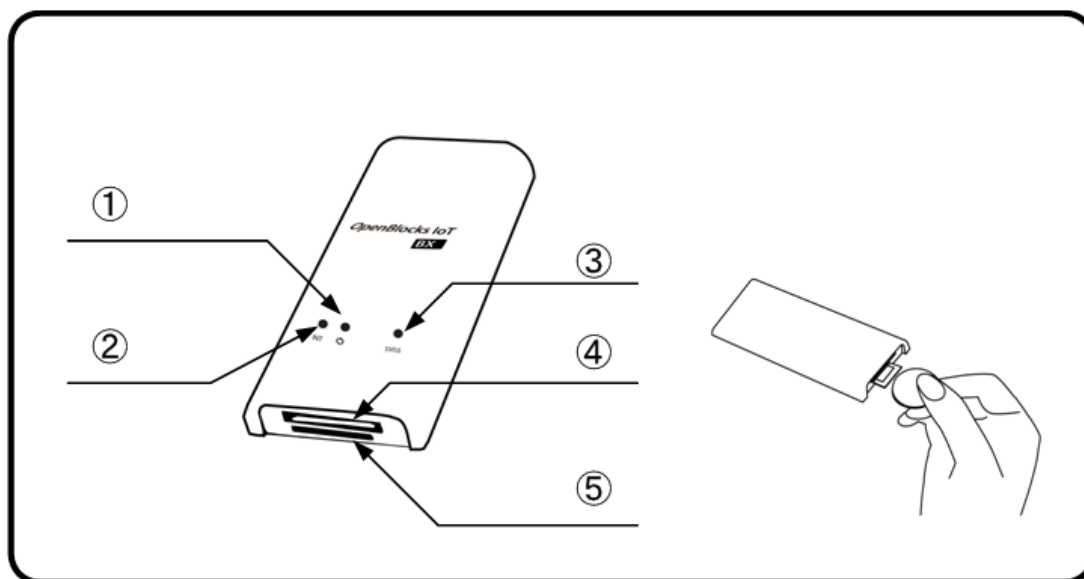
USB 給電コンソールケーブル 1 本



ご使用にあたって 1 部



1-2. 各部の名称(BX1 本体)



No.	名称	備考
①	パワースイッチ	短押しで OS をシャットダウンします。 (INIT スイッチの 4 秒以上長押しと同一) また、8 秒以上の長押しで強制的に電源 OFF します。
②	INIT スイッチ	2~4 秒の長押しで OS の再起動をします。 また、5 秒以上の長押しで OS のシャットダウンします。
③	ステータスインジケータ	7 色の LED で点灯、点滅をします。
④	BX1 コネクタ	様々な IO に対応したコネクタです。 それぞれの IO に合わせたケーブルを接続可能です。
⑤	SIM スロット	3G 回線(NTT ドコモ系列)を契約した SIM を挿入するスロットです。 ※対応する SIM の形状は mini-SIM(2FF)となります。 (一般的に標準 SIM と呼ばれる規格)

※SIM の挿入はコネクタ面を上にし、コイン等を使って奥まで入れてください。また、抜くとき時も同様にコインを使用し、SIM スロットの奥まで差し込むとロックが外れてせり出してきます。

1-3. BX3 向けパッケージ内容

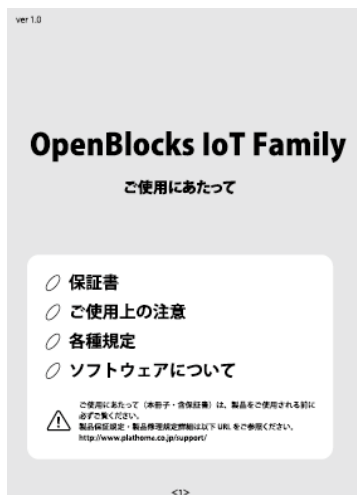
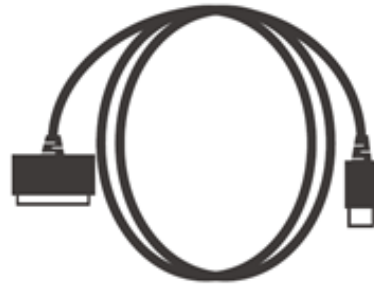
OpenBlocks IoT BX3 の標準品構成は以下となります。

BX3 本体 1 台

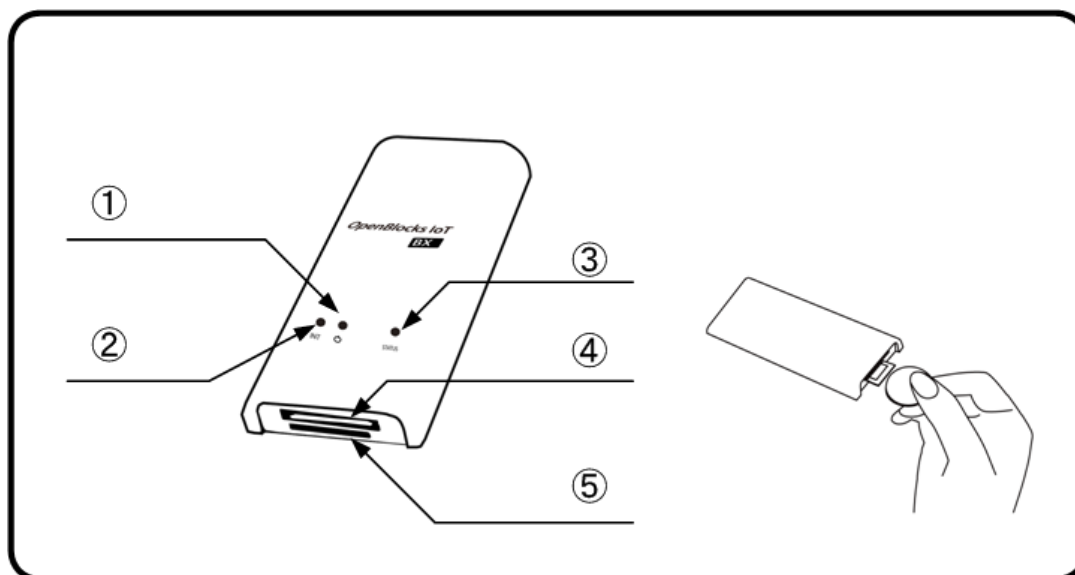


ご使用にあたって 1 部

USB 給電コンソールケーブル 1 本



1-4. 各部の名称(BX3 本体)



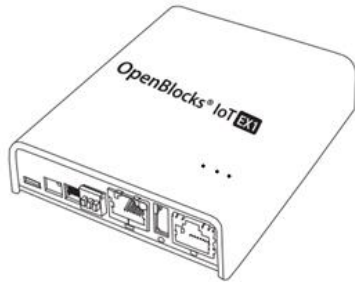
No.	名称	備考
①	パワースイッチ	短押しで OS をシャットダウンします。 (INIT スイッチの 4 秒以上長押しと同一) また、8 秒以上の長押しで強制的に電源 OFF します。
②	INIT スイッチ	2~4 秒の長押しで OS の再起動をします。 また、5 秒以上の長押しで OS のシャットダウンします。
③	ステータスインジケータ	7 色の LED で点灯、点滅をします。
④	BX1 コネクタ	様々な IO に対応したコネクタです。 それぞれの IO に合わせたケーブルを接続可能です。
⑤	SIM スロット	3G 回線(ソフトバンク系列)を契約した SIM を挿入するスロットです。 ※対応する SIM の形状は mini-SIM(2FF)となります。 (一般的に標準 SIM と呼ばれる規格)

※SIM の挿入はコネクタ面を上にし、コイン等を使って奥まで入れてください。また、抜くとき時も同様にコインを使用し、SIM スロットの奥まで差し込むとロックが外れてせり出してきます。

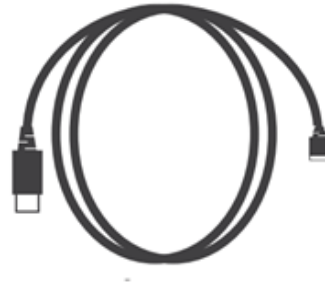
1-5. EX1 向けパッケージ内容

OpenBlocks IoT EX1 の標準品構成は以下となります。

EX1 本体 1 台



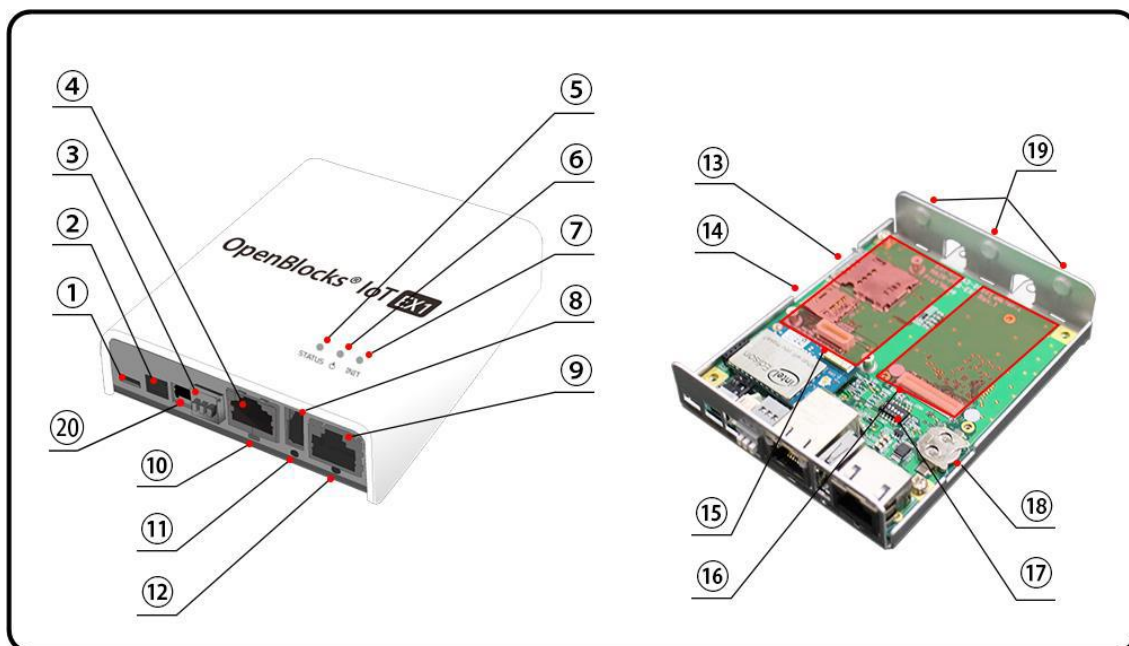
USB Type-A – Micro USB ケーブル 1 本



ご使用にあたって 1 部



1-6. 各部の名称(EX1 本体)



No.	名称	備考
①	USB シリアルコンソールポート	Micro USB。 バスパワーに対応した USB シリアルコンソールポートです。
②	専用 AC アダプタ入力	DC5V
③	RS-485(半二重)コネクタ	
④	イーサネットポート	100Base
⑤	ステータスインジケータ-1	7色の LED で点灯、点滅をします。
⑥	パワースイッチ 1	短押しで OS をシャットダウンします。 (INIT スイッチ 1 及び 2 の 4 秒以上長押しと同一) また、8 秒以上の長押しで強制的に電源 OFF します。
⑦	INIT スイッチ 1	2~4 秒の長押しで OS の再起動をします。 また、5 秒以上の長押しで OS をシャットダウンします。
⑧	USB ホストモードポート	A-Type
⑨	RS-232C ポート	RJ-45。 オプションで D-Sub9 ピンとの接続コネクタを販売しています。

No.	名称	備考
		接続ケーブルは一般のストレートネットワークケーブルが利用できます。
⑩	ステータスインジケータ-2	ステータスインジケータ-1 と同機能
⑪	パワースイッチ 2	パワースイッチ 1 と同機能
⑫	INIT スイッチ 2	INIT スイッチ 1 と同機能
⑬	SIM スロット	SIM を挿入するスロットです。 ※対応する SIM の形状は mini-SIM(2FF)となります。(一般的に標準 SIM と呼ばれる規格)
⑭	SD カードスロット	Micro SDXC 対応。 SD カードはシステム運用に十分な信頼性を確保できない為、ファイル交換やログ保存用等にご利用ください。
⑮	拡張スロット 1	EnOcean や Wi-SUN モジュール等の拡張スロットです。
⑯	拡張スロット 2	モバイル回線用のモバイルアダプタカードの拡張スロットです。 使用するキャリア対応のモバイルアダプタカードを取り付けます。原則的に工場出荷オプションとなります。
⑰	DIP スイッチ	工場出荷オプションで設定されるので通常は変更しないでください。 SW1：常時 ON SW2,3：モデム種類判別 OFF/OFF：3G モジュール(NTT ドコモ/ソフトバンク系列) ON/OFF：LTE モジュール(KDDI 系列) ON/ON：モデム無し SW4：- SW5：OFF=RS-232C 使用(デフォルト)、 ON=RS-485 使用 SW6：OFF=RS485 ターミネータ ON(デフォルト)
⑱	RTC 用電池ホルダー	
⑲	外部アンテナ取付穴	画像では穴埋めされています。
⑳	ワイドレンジ電源入力	OBSEX1 では使用不可。 DC 7～48V 対応

※SIM の挿入は EX1 本体を裏返しにし、SIM スロットの奥まで挿入してください。また、抜くときも同様に EX1 本体を裏返しにし取り出してください。

1-6-1. EX1 RS-232C ピンアサイン

ピン	信号
1	RTS
2	未使用
3	TXD
4	GND
5	GND
6	RXD
7	未使用
8	CTS

1-6-2. EX1 RS-485 ピンアサイン

ピン	信号
1	B (-)
2	A (+)
3	GND

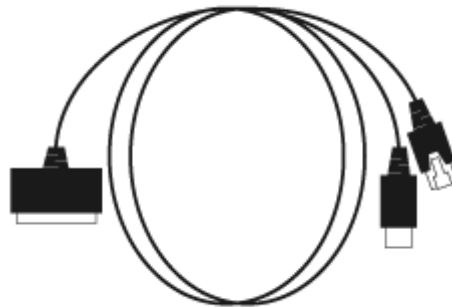
1-7. BX0 向けパッケージ内容

OpenBlocks IoT BX0 の標準品構成は以下となります。

BX0 本体 1 台



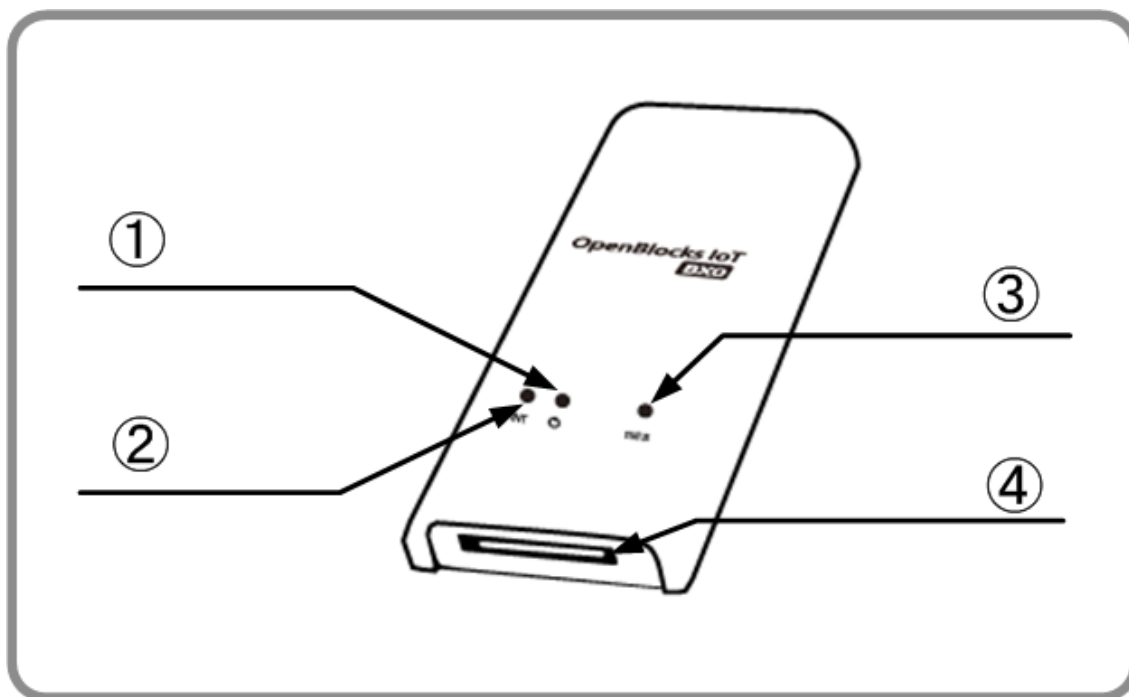
USB 給電二又ケーブル/Ethernet 付き 1 本



ご使用にあたって 1 部



1-8. 各部の名称(BX0 本体)



No.	名称	備考
①	パワースイッチ	短押しで OS をシャットダウンします。 (INIT スイッチの 4 秒以上長押しと同一) また、8 秒以上の長押しで強制的に電源 OFF します。
②	INIT スイッチ	2~4 秒の長押しで OS の再起動をします。 また、5 秒以上の長押しで OS のシャットダウンします。
③	ステータスインジケータ	7 色の LED で点灯、点滅をします。
④	BX1 コネクタ	様々な IO に対応したコネクタです。 それぞれの IO に合わせたケーブルを接続可能です。

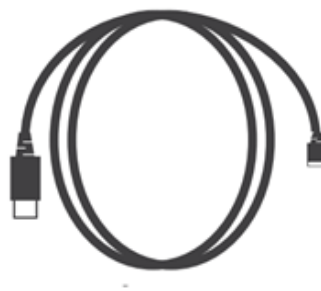
1-9. BX3L 向けパッケージ内容

OpenBlocks IoT BX3L の標準品構成は以下となります。

BX3L 本体 1 台



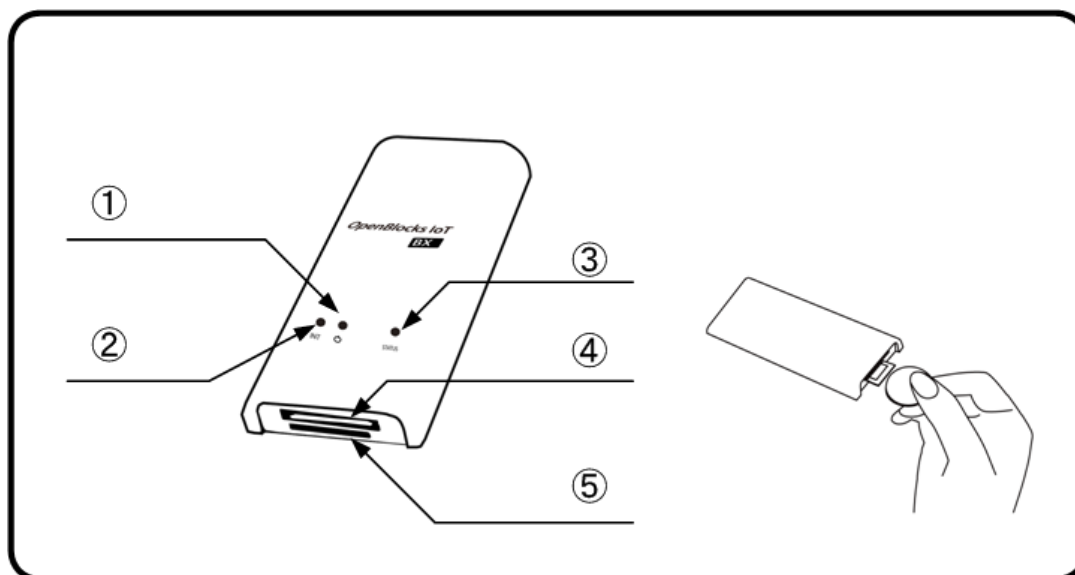
USB Type-A – Micro USB ケーブル 1 本



ご使用にあたって 1 部



1-10. 各部の名称(BX3L 本体)



No.	名称	備考
①	パワースイッチ	短押しで OS をシャットダウンします。 (INIT スイッチの 4 秒以上長押しと同一) また、8 秒以上の長押しで強制的に電源 OFF します。
②	INIT スイッチ	2~4 秒の長押しで OS の再起動をします。 また、5 秒以上の長押しで OS のシャットダウンします。
③	ステータスインジケータ	7 色の LED で点灯、点滅をします。
④	BX1 コネクタ	様々な IO に対応したコネクタです。 それぞれの IO に合わせたケーブルを接続可能です。
⑤	SIM スロット	LTE 回線(ソフトバンク系列)を契約した SIM を挿入するスロットです。 ※対応する SIM の形状は mini-SIM(2FF)となります。 (一般的に標準 SIM と呼ばれる規格)

※SIM の挿入はコネクタ面を上にし、コイン等を使って奥まで入れてください。また、抜くとき時も同様にコインを使用し、SIM スロットの奥まで差し込むとロックが外れてせり出してきます。

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には、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。

また、本装置は M2M ゲートウェイとしての利用を想定している為、実運用する場合には RAM Disk モードでの利用を推奨しております。RAM Disk モードの場合、電源切断などが起こっても内部ストレージデータの破損を行わない為です。

但し、本装置のデフォルト出荷設定は WEB UI 等の利用を前提としている為、ストレージ併用モードとしております。

2-2. SIM について

OpenBlocks IoT Family にて、搭載可能な SIM 形状は mini-SIM(2FF)です。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。

2-3. 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。

番号	名前	format	サイズ	OBS 用途	デバイス名
0	rpmb	-	1MB	未使用	mmcblk0rpmb
1	u-boot0	-	2MB	通常起動用 u-boot	
2	u-boot-env0	-	1MB	通常起動用 u-boot 環境変数	
3	u-boot1	-	2MB	非常時起動用 u-boot	
4	u-boot-env1	-	1MB	非常時起動用 u-boot 環境変数	
5	factory	ext2	1MB	シリアル番号、MAC アドレス等	mmcblk0p5
6	panic	-	24MB	カーネルパニックメッセージ保存用	mmcblk0p6

番号	名前	format	サイズ	OBS 用途	デバイス名
7	boot	fat16	32MB	未使用	mmcblk0p7
8	rootfs	ext4	512MB	ファームウェア格納	mmcblk0p8
9	update	fat16	768MB	未使用	mmcblk0p9
10	home	ext4	2768MB	ストレージ併用モード	mmcblk0p10

2-4. RAM Disk モード

基本的な FW を 4GB の eMMC システム領域から RAM Disk 上に展開し、Linux のユーザーランドまでの全てをその RAM Disk 上で動作する為、不意な電源断が起こっても物理ストレージ上のファイルを破損することが無く、この場合でも電源の再投入のみでサービスの再スタートが行えます。

データの蓄積が不要な用途では、この利用用途により堅牢なシステムを実現することが可能です。

RAM Disk モード及び後述のストレージ併用モードでは aufs(Another Unionfs)を使用します。

RAM Disk モードは /.rw を tmpfs としてマウントし、/配下のディレクトリ毎に aufs を利用して /.rw/配下のディレクトリを被せます。(例 : /.rw/usr を /usr 上に被せる)

/.rw 以下に集約された全ての変更差分をは、専用コマンドの flashcfg を用いて eMMC 内に保存します。

また、起動時に eMMC から RAM ディスクに書きもどし、保存時の環境を復元します。

※mount コマンドの実行結果

```

root@obsiot:~# mount
/dev/root on / type ext2 (rw,relatime,errors=continue)
devtmpfs on /dev type devtmpfs (rw,relatime,size=450880k,nr_inodes=112720,mode=755)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=98436k,mode=755)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /run/shm type tmpfs (rw,nosuid,nodev,noexec,relatime,size=196860k)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620)
tmpfs on /.rw type tmpfs (rw,relatime,size=393216k)
aufs on /etc type aufs (rw,relatime,si=6cc6682d)
aufs on /bin type aufs (rw,relatime,si=60eea82d)

```

```
aufs on /home type aufs (rw,relatime,si=6cc6182d)
aufs on /lib type aufs (rw,relatime,si=60eed82d)
aufs on /sbin type aufs (rw,relatime,si=6cc6082d)
aufs on /usr type aufs (rw,relatime,si=60eec82d)
aufs on /var type aufs (rw,relatime,si=6cc6282d)
aufs on /root type aufs (rw,relatime,si=459ad82d)
aufs on /opt type aufs (rw,relatime,si=459d582d)
aufs on /srv type aufs (rw,relatime,si=459ab82d)
aufs on /media type aufs (rw,relatime,si=459d782d)
/dev/mmcblk0p5 on /factory type ext4 (ro,nosuid,nodev,noatime,discard)
```

※ラベル解除例

```
# e2label /dev/mmcblk0p10 ""
```

2-5. ストレージ併用モード

RAM ディスクモードにおける`/rw`のRAM ディスク領域を、eMMC等のストレージに変更した動作モードです。これによりRAM Disk上のシステムが起動した後に更新されたファイルはeMMCストレージ上に書き残されるので、RAM Diskモードのようにファイルを書き換える度にeMMCストレージへファイルを書き戻す必要がありません。反面、不意な電源切断によりeMMCストレージのユーザーパーティション上の更新ファイルが破壊される危険性が残ります。

ストレージ併用モードは`ext2/ext3/ext4`でフォーマットを行ったラベルが「DEBIAN」ストレージを`/rw`としてマウントし、配下のディレクトリ毎にaufsを利用して`/rw/配下のディレクトリ`を被せます。

実ストレージに対してアクセスを行う為、書き込み等を行った時点でストレージへの反映が行われます。そのため、`flashcfg`コマンドによる保存は不要です。

※「DEBIAN」ラベルの消去やラベル変更後の再起動等によりRAM ディスクモードとなります。

※mount コマンドの実行結果

```
root@obsiot:~# mount
/dev/root on / type ext2 (rw,relatime,errors=continue)
devtmpfs on /dev type devtmpfs (rw,relatime,size=450880k,nr_inodes=112720,mode=755)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=98436k,mode=755)
```

```
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /run/shm type tmpfs (rw,nosuid,nodev,noexec,relatime,size=196860k)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620)
/dev/mmcblk0p10 on /.rw type ext4 (rw,relatime,discard,data=ordered)
aufs on /etc type aufs (rw,relatime,si=1321ee97)
aufs on /bin type aufs (rw,relatime,si=1321fe97)
aufs on /home type aufs (rw,relatime,si=364a5e97)
aufs on /lib type aufs (rw,relatime,si=364a1e97)
aufs on /sbin type aufs (rw,relatime,si=3a7c4e97)
aufs on /usr type aufs (rw,relatime,si=364a6e97)
aufs on /var type aufs (rw,relatime,si=364a3e97)
aufs on /root type aufs (rw,relatime,si=36026e97)
aufs on /opt type aufs (rw,relatime,si=36022e97)
aufs on /srv type aufs (rw,relatime,si=36027e97)
aufs on /media type aufs (rw,relatime,si=3635de97)
/dev/mmcblk0p5 on /factory type ext4 (ro,nosuid,nodev,noatime,discard)
```

※ラベル設定例

```
# e2label /dev/mmcblk0p10 DEBIAN
```

2-6. その他の運用方法

RAM Disk モードでアプリの基本部分を運用し、ログだけを eMMC のストレージに保管したいケースの場合などがあります。この場合、home パーティションを `mkfs` コマンド等にて初期化し任意ディレクトリにマウントすることで利用可能です。この場合、そのディレクトリに書き込まれた内容は電源断された場合でも残ります。

但し、不意な電源切断された場合、後の再起動時にその部分だけファイルのチェック (`fsck`) が入り起動に時間がかかる場合があります。

※初期化例

```
# mkfs -t ext4 -L "" /dev/mmcblk0p10
```

2-7. ウォッチドッグ

OpenBlocks IoT Family には HW ウォッチドッグが組み込まれています。本体起動時に wd-keepalive プロセスが起動し、30 秒毎にタイムアウト値である 90 秒のタイマをクリアしています。wd-keepalive プロセスからタイマクリア信号が送信されない状態に 90 秒間おちいった場合、リセット処理が行われます。

また、wd-keepalive プロセスを kill コマンド等にて終了させた場合には、即時にリセット処理が実施されます。

尚、クリア間隔及びタイムアウト値の変更は行えません。

2-8. RTC

OpenBlocks IoT Family の BX1 及び BX3 には RTC のバックアップ電池が用意されていませんが、スーパーキャパシタにて最大約 10 分程度 RTC の保持が可能です。(EX1 には RTC の電池が搭載されています。)

BX1 及び BX3 に有効な SIM が挿入されている場合には、OS 起動時にモバイル回線の基地局からシステム時間を取得し設定しております。基地局からの時刻取得については、ある程度の誤差が発生する為、NTP の利用を推奨します。

2-9. 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。



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

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 : x x x x x x

※パスワードは使用環境によって異なります。WEB UI 使用時は WEB UI で設定したパスワードを使用してください。また、RAM Disk モード時のデフォルトは”root”となります。

2-10. 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。

尚、WEB UI の機能と根本的に干渉する機能を実装する場合には、RAM Disk モードにて開発することを推奨します。

※WEB UI での設定項目等については、以下を参照してください。

OpenBlocks IoT Family 向け WEB UI セットアップガイド

http://openblocks.plathome.co.jp/common/pdf/obsiot_webui_setup_guide.pdf

WEB UI はストレージ側に存在しています。そのため、以下の手順で WEB UI 等が入っているストレージのフォーマットが行えます。

```
# e2label /dev/mmcblk0p10 ""
```

```
# reboot
```

再起動後にログインしてください。

●RAM Disk モードとして運用する場合

```
# mkfs -t ext4 -L "" /dev/mmcblk0p10
```

●ストレージ併用モードとして運用する場合

```
# mkfs -t ext4 -L DEBIAN /dev/mmcblk0p10
```

WEB UI 削除後に復旧したい場合は、弊社 WEB サイトのファクトリーリセットページを参照してください。

<http://openblocks.plathome.co.jp/support/guide/bx1/reset.html>

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で構いません。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. 省電力化

本装置は USB コンソールケーブル経由で Linux のコマンドラインの操作が可能です。実際の運用に入った際には、コンソールでの操作が不要になるため、USB シリアルチップの電源を自動制御にすることで、より省電力での待機が可能になります。反面、自動制御にした場合、コンソールでのキー入力時に省電力からの復帰時の最初のキーを取りこぼす場合があります。

※USB シリアルチップの電源制御を自動にする

```
# echo auto > /sys/devices/pci0000:00/0000:00:04.3/power/control
```

※USB シリアルチップの電源制御を無効にする

```
# echo on > /sys/devices/pci0000:00/0000:00:04.3/power/control
```

3-2. 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。尚、RAM Disk モード時はモバイル回線に関しては LED と連動いたしません。また、BX0 に限り、Ethernet を使用している場合には Tx/Rx の状況により LED が白点滅します。

状態	色	点灯状態	備考
OS 起動中	黄	点灯 ↓ 消灯 ↓ 点滅	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

状態	色	点灯状態	備考
モバイル回線電波：強	白	点滅	電波強度-87dBm 以上。または、LTE モジュール For KDDI の場合アンテナ 2 本以上、LTE モジュール For Docomo の場合アンテナ 3 本時。
モバイル回線電波：中	水色	点滅	電波強度-88~-108dBm または、LTE モジュール For KDDI の場合アンテナ 1 本、LTE モジュール For Docomo の場合アンテナ 2 本時。
モバイル回線電波：弱	青	点滅	電波強度-109~-112dBm。または、LTE モジュール For KDDI の場合アンテナ 0 本、LTE モジュール For Docomo の場合アンテナ 1 本時。 ※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	電波強度-113dBm 以下。または、LTE モジュール for KDDI 及び、LTE モジュール for Docomo の場合、圏外時。
INIT スイッチによるリブート時	黄	点灯	OS リブート
INIT スイッチによるシャットダウン電源 OFF	赤	点灯	LED が消灯するまで長押しが必要

WEB UI 使用時の LED 点灯制御スクリプト

```
/var/webui/bin/set_signal_value.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	点灯時間(msec)	1以上
2行目	消灯時間(msec)	1以上
3行目	色番号	下表参照

色番号	色
1	赤
2	緑
3	黄
4	青
5	紫
6	水色
7	白

※1 秒毎に黄色点滅

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

3-3. 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。

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	
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 EX1 Docomo LTE モデム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 SMONI 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、各 IoT Family 製品によって 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/ttyACM0	使用可能
/dev/ttyACM1	使用不可
/dev/ttyACM2	使用不可
/dev/ttyACM3	使用可能
/dev/ttyACM4	使用可能
/dev/ttyACM5	使用不可
/dev/ttyACM6	使用不可

●BX3

デバイスファイル	atcmd 使用可否
/dev/ttyACM0	使用可能
/dev/ttyACM1	使用可能
/dev/ttyACM2	使用可能
/dev/ttyACM3	使用不可
/dev/ttyACM4	使用不可
/dev/ttyACM5	使用不可
/dev/ttyACM6	使用不可

●EX1 (3G モジュール)

デバイスファイル	atcmd 使用可否
/dev/ttyACM0	使用可能
/dev/ttyACM1	使用可能
/dev/ttyACM2	使用可能
/dev/ttyACM3	使用不可
/dev/ttyACM4	使用不可
/dev/ttyACM5	使用不可

デバイスファイル	atcmd 使用可否
/dev/ttyACM6	使用不可

●EX1 (LTE モジュール for KDDI)

デバイスファイル	atcmd 使用可否
/dev/ttyACM0	使用可能

●EX1 (LTE モジュール for Docomo)

デバイスファイル	atcmd 使用可否
/dev/ttyACM0	使用可能
/dev/ttyACM1	使用可能

3-4. モバイル回線接続

WEB UI を使用している場合、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 ver.1.0.7 以降を使用している場合、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。

※WEB UI を使用していない RAM Disk 方式についてはこの限りではありません。

3-4-1. 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) (2) 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

APN

ユーザ名

パスワード パスワードを表示する

認証方式

自動接続 自動接続する 自動接続しない

通信確認用ホスト (2)

定期再接続設定 定期再接続をする 定期再接続をしない

SMSコントロール (2) 無効 有効

必要な情報は以下となります。

- ・APN(LTE モジュール(KDDI 4G LTE)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 4G LTE)時は項目がありません)

※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、モバイル回線の接続及び切断のコマンドは以下となります。

●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 "# Disonnect (`date`)"
echo "#-----#"

```

```
/var/webui/scripts/mobile_control.sh coff 1  
  
sleep 2  
  
exit 0
```

3-4-2. モバイル回線モデムのユーザー制御

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) (2) 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 (2) 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	
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/ttyACM[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/ttyACM[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、atmcd 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。

3-5. BT の起動

OpenBlocks IoT Family にて、BTを使用する場合には以下の手順で起動してください。尚、WEB UI を使用している場合には、WEB UI から設定が可能です。

※下記の赤字部が実施コマンドとなります。

```
# bluetooth_rfkill_event &
[1] 2041
1415343811.654239: idx 2 type 2 op 0 soft 1 hard 0
# rfkill unblock bluetooth
1415343829.262611: idx 2 type 2 op 2 soft 0 hard 0
execute brcm_patchram_plus --use_baudrate_for_download --no2bytes --enable_fork --enable_lpm
--enable_hci --baudrate 3000000 --patchram /etc/firmware/bcm43341.hcd --bd_addr xx:xx:xx:xx:xx:xx
/dev/ttyMFD0
Done setting line discipline
1415343829.720741: idx 3 type 2 op 0 soft 0 hard 0
```

実際の運用方法については、BT デバイスに依存します。そのため、各 BT の情報サイトをご参照ください。

3-6. 専用コマンド(flashcfg)

本装置の動作に関わる設定を行う為に専用のコマンド「flashcfg」を搭載しています。RAM Disk モードで利用する場合は、追加したソフトウェアや変更した設定の保存を行う為に必ず利用することになります。ストレージ併用モードの場合には、基本的な利用の範囲では使用せずとも運用可能です。尚、オプションを指定せず実行した場合、実行可能なコマンドの簡単なヘルプを表示します。

■変更内容を eMMC へ保存 - 全て

全てを保存する場合には、オプション「-S(大文字の S)」を指定します。/rw 以下全てを

eMMC に保存します。また、内部的に次の「-s(小文字の s)」も呼び出します。特に理由がない限りは、「-S(大文字の S)」を指定して実行してください。

```
# flashcfg -S

Overwrites the current data.

Are you ok? [y|N] y

Archiving userland files... done (Approximately 0 MBytes)

done (Approximately 0 MBytes)
```

■変更内容を eMMC へ保存 - /etc のみ

設定変更(/etc 以下のみ)を保存する場合は、オプション「-s(小文字の s)」を指定します。/.rw/etc のみを eMMC に保存します。対象が/etc 以下のみの為、短時間で保存が完了しますが、他の領域との不整合が発生する可能性がありますので、特に理由がない限りは使用しないことをお勧めします。

```
# flashcfg -s

Overwrites the current data.

Are you ok? [y|N] y

Archiving userland files... done (Approximately 0 MBytes)
```

■eMMC へ保存した変更内容を削除

保存内容を削除し、設定を初期化したい場合には、オプション「-e」を指定します。eMMC 内ユーザーエリアのデータを削除し、起動時にデータ復元が行えないようにします。全ての領域を削除する場合には、オプション「-E」を指定します。

```
# flashcfg -e

Erase userarea).

Are you ok? [y|N] y
```

■前回保存した際のサイズを表示

RAM Disk モードでの、ソフトウェアの追加可否を判断する為に、前回保存を行った際の

使用サイズの情報を表示します。

```
# flashcfg -l

Show FlashROM last saved size

Use userland(-S) : 0 KBytes (MAX: unknown MBytes)
Use /etc config(-s) : 0 KBytes (MAX: unknown MBytes)
```

3-7. バックアップ

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)を接続し、以下のコマンドを実行することで環境のバックアップが取得できます。

```
# flashcfg -B
```

接続したストレージの中に、/.rw 以下をアーカイブした `userland.tgz` を作成します。

尚、/etc/exclude.list 内に記載されているパスのデータはバックアップ対象に含まれません。そのため、アプリケーションの起動等により対象パス内のディレクトリが必要な場合には、/ext/exclude.list を編集しバックアップ対象に含ませてください。また、/ext/exclude.list 自体はバックアップ対象に含まれません。

※WEB UI を用いている場合には、以下のディレクトリが必要となります。

```
/var/log/lighttpd
```

```
/var/log/supervisor
```

そのため、/ext/exclude.list を以下のように修正してください。尚、kernel-image-obsbx1 のバージョンが 3.10.17-15 以降は修正されている為、/ext/exclude.list は不要です。

●変更後/ext/exclude.list

```
etc/exclude.list
etc/init.d/openblocks-setup
etc/init.d/pshd
etc/init.d/runled
etc/init.d/telinitq
etc/init.d/umountfs
etc/mtab
etc/udev/rules.d/70-persistent-net.rules
```

```
usr/lib/jre
usr/sbin/flashcfg
usr/sbin/flashcfg-debian
var/cache/apt
var/lib/apt/lists
var/lib/dpkg/updates
var/log/*log*
var/log/*tmp*
var/log/*mail*
var/log/*dmesg*
var/log/*ppp*
var/log/*message*
var/log/*debug*
var/log/*/*log*
var/log/*/*tmp*
var/log/*/*mail*
var/log/*/*dmesg*
var/log/*/*ppp*
var/log/*/*message*
var/log/*/*debug*
var/run
```

3-8. リストア

前述の方法でバックアップを取得したストレージを接続し、電源 ON すると起動時に自動的に展開します。

尚、弊社の出荷状態のデフォルトでは WEB UI 等のデータが含まれております。この状態でリストアを実施した場合、データの不一致が発生するためリストア前に対象領域を初期化する必要があります。そのため、後述の **FORCE** ファイルの使用を推奨します。

またモデル間でインターフェース等の差分がありますので、モデル間をまたいでリストアを行う場合には注意してください。

3-9. 応用

バックアップファイル(`userland.tgz`)の他、所定のファイル名でファイルを用意することで、スクリプト実行などが可能です。以下、実行(チェック)順に列挙します。いずれもファイルが存在する場合のみ有効になります。

- ・ `FORCE` ファイル(中身は空)

`RAM Disk` モードではユーザーエリアの展開は行いません。また、ストレージ併用モードでは、ストレージの初期化を行います。

※`kernel-image-obsbx1` のバージョンが 3.10.17-14 以前では `ext3` にてフォーマットされます。IoT Family 本製品でのストレージ領域は `ext4` を用いておりますので、フォーマットする際には手動にて実施してください。

- ・ `userland.tgz`(`/.rw` 以下を `tar+gzip` でアーカイブしたもの)

`aufs` のマウント前に、`/.rw` への展開を行います。(前述のリストア動作)

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

`aufs` のマウント前に、実行を行います。

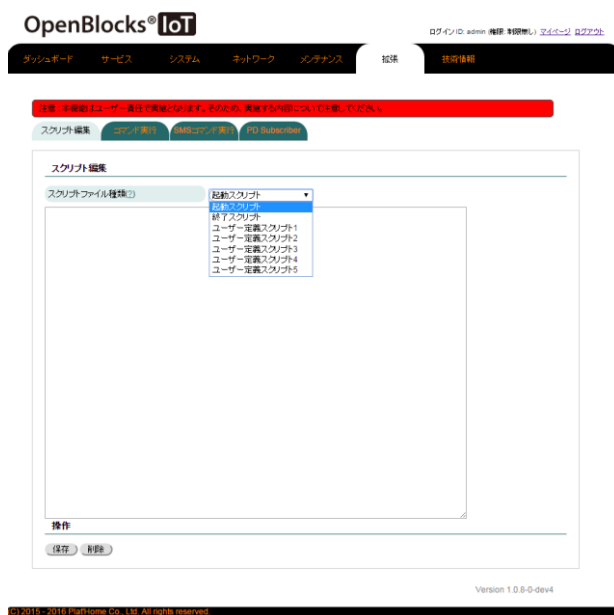
- ・ `post-init.sh`(`sh` スクリプト、改行コードは `LF` のみ)

`aufs` のマウント完了後に、実行を行います。

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. スクリプト編集



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

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。	

3-10-2. コマンド実行



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。

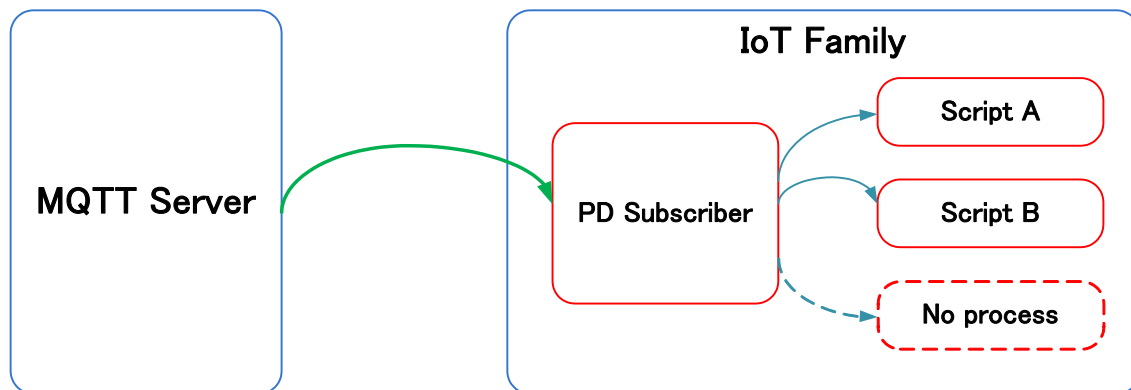
3-10-3. SMS コマンド実行

本項については、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。

3-10-4. PD Subscriber

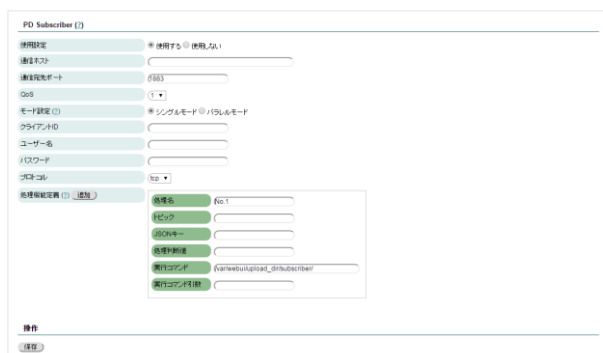
本製品は MQTT プロトコルにおけるサブスクライバーとなります。サーバサイドからの情報を元にスクリプト等の処理を実行することが出来ます。

サーバサイドから特定のトピックに対して送信された JSON メッセージにおいて、条件に一致した場合、一致対象のスクリプト等を実行します。



PD Subscriber タブにアクセスした場合、左図のように表示されます。

PD Subscriber を使用する場合は、”使用する”を選択してください。



通信先の MQTT サーバ情報を設定します。設定情報は以下となります。

- 通信ホスト
- 通信宛先ポート
- QoS
- クライアント ID
- ユーザー名
- パスワード
- プロトコル
- トラストア(ssl 時のみ)
- キーストア(ssl 時のみ)
- プライベートキー(ssl 時のみ)

条件に一致した際に実行するスクリプトを設定します。尚、追加ボタンにて最大 8 個まで追加することができます。

処理名：

実行する処理の名前を定義します。自動で”No.X”として設定されます。

トピック：

処理を行う対象の判断としてトピックを設定します。尚、”#”がワイルドカードとなります。

JSON キー：

サーバサイドから受信したメッセージ内で条件判断を行うキーを設定します。

処理判断値：

サーバサイドから受信したメッセージ内で判断を行うキーの値を設定します。

※サーバサイドからのメッセージ及び本設定では文字列として扱う為、完全一致での判断となります。

実行コマンド：

設定した JSON キー及び処理判断値に一致した場合、実行するコマンドを定義します。

※ファイルの存在チェックをしている為、事前にスクリプトファイルをシステムのファイル管理から作成してください。

実行コマンド引数：

コマンドを実行する際に用いる引数を設定します。



設定をし、保存ボタンを押すことで実行されます。



PD Subscriber が実行された場合、ログが生成されます。

先頭部に存在するログ選択からログが閲覧可能です。

① PD Subscriber にて実行されるスクリプト等には、MQTT サーバから送られてきた JSON データを環境変数にて設定し実行されます。

※MQTT サーバ送信メッセージサンプル

```
{  
  "TEST1": 1  
}
```

※環境変数追加分

TEST1=1

① 本機能は試験的な実装となっております。ご使用する場合には、十二分に検証の上、ご使用ください。

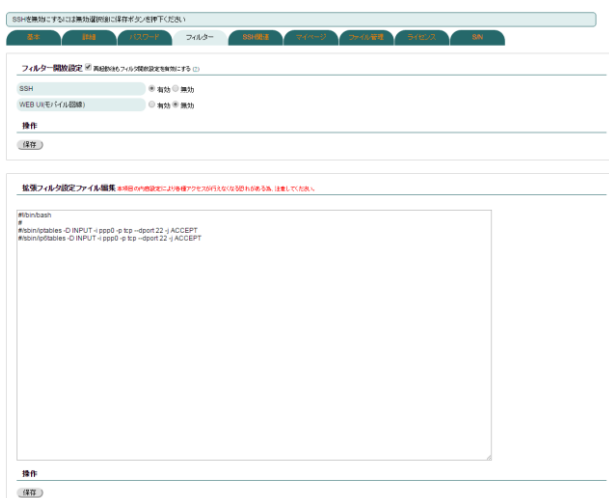
※弊社での検証では、サーバとの接続が一時的に切断されることを確認しております。

3-11. 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。

3-12. 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 for 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。

●EX1 LTE モジュール for Docomo 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"
```

●EX1 LTE モジュール for Docomo 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"
```

3-13. LTE モジュール for KDDI 運用切り替え

LTE モジュール for 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
```

3-14. 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

3-15. Yocto Linux の起動

OpenBlocks IoT Family ではエマージェンシーブート用に Yocto Linux がインストールされています。Yocto Linux は以下の方法で起動可能です。

本体起動時に以下のメッセージが表示されます。

```
*** Ready to receive application ***
```

この時点で U-Boot のプロンプトが表示されるまで、スペースキーを 2 回以上連打します。

```
*****  
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 (Aug 20 2014 - 16:08:32)  
  
Watchdog enabled
```

```
DRAM: 980.6 MiB
MMC: tangier_sdhci: 0
In: serial
Out: serial
Err: serial
Hit any key to stop autoboot: 0
boot >
```

U-Boot のプロンプトが表示されたら”run bootYocto”を入力します。

```
boot > run bootYocto
```

Yocto Linux が起動したらログイン可能となります。また、root ログインでのパスワードは不要です。

```
edison login: root
```

第 4 章 製品仕様

4-1. OpenBlocks IoT BX1 本体仕様

型番	OBSBX1	
CPU	モデル	インテル® Atom™ プロセッサ
	動作速度	500MHz (デュアルコア)
メインメモリ	オンボード	1GB(LPDDR3)
FLASH ROM	4GB(eMMC)	
追加ストレージ	-	
無線インターフェース	3G(W-CDMA)*1	
	BT 4.0	
	IEEE802.11a/b/g/n	
SIM インターフェース	通信用 miniSIM(25mm x 15mm x 0.76mm) カードスロット	
有線インターフェース	BX1 コネクタ 標準添付 : USB2.0(バスパワー・type-A)	
寸法	41.6mm(W) x 96mm (D) x 11.3mm(H) (突起部 含まず)	
重量	45g	
電源	DC 4.75~48V ※USB ケーブルによる給電時 DC4.75~5.25V	
消費電力	アイドル時	3G 通信アクティブ時 : 0.8W / 3G 通信非アクテ ィブ時 : 0.1W*2
	高負荷時	4.6W
MTBF	445,951h	
EMC 規格	VCCI Class-B 適合	
省エネ法に基づくエネルギー消費効率 [単位 : W/GTOPS]*3	区分 : H	
	0.17	
環境保護	RoHS 指令 PFOS 規制 省エネルギー法	
認証	JATE/TELEC	
RTC バックアップ時間	最大約 10 分	
出荷時 OS	Debian GNU/Linux	

※1：NTT ドコモ FOMA 網に対応

※2：Power Save = Auto 設定時

※3：エネルギー消費効率とは省エネ法で定める測定方法により測定した消費電力を省エネ法で定める複合理論性能で除したものです

4-2. OpenBlocks IoT BX3 本体仕様

型番		OBSBX3
CPU	モデル	インテル® Atom™ プロセッサ
	動作速度	500MHz (デュアルコア)
メインメモリ	オンボード	1GB(LPDDR3)
FLASH ROM		4GB(eMMC)
追加ストレージ		-
無線インターフェース		3G(W-CDMA) ^{※1}
		BT 4.0
		IEEE802.11a/b/g/n
SIM インターフェース		通信用 miniSIM(25mm x 15mm x 0.76mm) カードスロット
有線インターフェース		BX1 コネクタ 標準添付 : USB2.0(バスパワー・type-A)
寸法		41.6mm(W) x 96mm (D) x 11.3mm(H) (突起部 含まず)
重量		45g
電源		DC 4.75~48V ※USB ケーブルによる給電時 DC4.75~5.25V
消費電力	アイドル時	1.3W
	高負荷時	4.6W
MTBF		224,858h
EMC 規格		VCCI Class-B 適合
省エネ法に基づくエネルギー消費効率 [単位 : W/GTOPS] ^{※2}		区分 : H
		0.17
環境保護		RoHS 指令 PFOS 規制 省エネルギー法
認証		JATE/TELEC
RTC バックアップ時間		最大約 10 分
出荷時 OS		Debian GNU/Linux

※1 :Softbank 3G 網に対応

※2 : エネルギー消費効率とは省エネ法で定める測定方法により測定した消費電力を省エネ法で定める複合理論性能で除したものです

4-3. OpenBlocks IoT EX1 本体仕様

型番		OBSEX1	OBSEX1G
CPU	モデル	インテル® Atom™ プロセッサ	
	動作速度	500MHz (デュアルコア)	
メインメモリ	オンボード	1GB(LPDDR3)	
FLASH ROM		4GB(eMMC)	
追加ストレージ		microSD カードスロット x 1	
無線インターフェース		BT 4.0	
		IEEE802.11a/b/g/n	
SIM インターフェース		通信用 miniSIM(25mm x 15mm x 0.76mm) カードスロット	
有線インターフェース	USB(HOST)	2.0 x 1 (type-A)	
	USB(Console)	Micro USB(type-B) x 1 ※バスパワー給電可	
	Ethernet	10Base-T/100BASE-TX x 1	
	RS-232C/RS-485	RS-232C x 1 : RJ45 コネクタ / RS-485 x 1 : ターミナルブロック (排他)	RS-232C x 1 : RJ45 コネクタ / RS-485 x 1 : ターミナルブロック
寸法		91.9mm(W) x 114.8mm (D) x 25mm(H) (突起部含まず)	
重量		135g	
電源		DC-Jack : DC4.75 ~ 5.25V	DC-Jack : DC4.75 ~ 5.25V
		USB ケーブル給電時 : DC4.75~5.25V	USB ケーブル給電時 : DC4.75~5.25V XA コネクタ給電時 : DC7~48V
消費電力	アイドル時	1.8W	
	高負荷時	3.3W	
MTBF		485,265h	468,231h
EMC 規格		-	VCCI Class-B 適合
省エネ法に基づくエネルギー消費効率 [単位 : W/GTOPS] ※1		区分 : H	
		0.23	0.23
環境保護		RoHS 指令 PFOS 規制 省エネルギー法	

認証	TELEC	
RTC バックアップ時間	最大 250 日	約 10 年
出荷時 OS	Debian GNU/Linux	

※1：エネルギー消費効率とは省エネ法で定める測定方法により測定した消費電力を省エネ法で定める複合理論性能で除したものです

4-4. OpenBlocks IoT EX1 オプション

4-4-1. 3G モジュール For NTT ドコモ

対応周波数	GSM/GPRS/EDGE: Quad band 850/900/1800/1900MHz W-CDMA(UMTS/HSPA+): Five band 800/850/900/1900/2100MHz	
データ通信速度	下り：7.2Mbps 上り：5.7Mbps ※理論値	
制御方法	AT コマンド	
認証	JATE/TELEC	
電源電圧	DC 3.3～4.4V	
消費電力	アイドル時	0.18W ※平均電力
	高負荷時	2.6W ※平均電力

4-4-2. 3G モジュール For ソフトバンク

対応周波数	GSM/GPRS/EDGE: Quad band 850/900/1800/1900MHz W-CDMA(UMTS/HSPA+): Five band 800/850/900/1900/2100MHz	
データ通信速度	下り：7.2Mbps 上り：5.7Mbps ※理論値	
制御方法	AT コマンド	
認証	JATE/TELEC	
電源電圧	DC 3.3～4.4V	
消費電力	アイドル時	0.18W ※平均電力
	高負荷時	2.6W ※平均電力

4-4-3. LTE モジュール For KDDI

対応周波数	受信：860.0MHz～875.0MHz 送信：815.0MHz～830.0MHz	
データ通信速度	下り：75Mbps 上り：25Mbps ※理論値	
制御方法	AT コマンド・特殊コマンド	
認証	JATE/TELEC	
電源電圧	DC 3.3～4.2V	
消費電力	アイドル時	0.16W ※平均電力
	高負荷時	2.4W ※平均電力

4-4-4. LTE モジュール For Docomo

対応周波数	LTE：2GHz / 800MHz	
データ通信速度	下り：112.5Mbps 上り：37.5Mbps ※理論値	
制御方法	AT コマンド、特殊コマンド	
認証	JATE/TELEC	
電源電圧	DC 3.3～4.2V ※推奨 DC 3.7V	
消費電力	アイドル時	0.0185W
	高負荷時	3.7W

4-4-5. EnOcean モジュール

周波数	928.35MHz	
データ通信速度	125kbps	
HOST I/F	UART	
認証	TELEC	
電源電圧	DC 2.6～5.0V	
消費電力	アイドル時	0.13W
	高負荷時	0.11W

4-4-6. Wi-SUN モジュール

周波数	922.5～927.9MHz(28ch)	
データ通信速度	100kbps	
HOST I/F	UART	
認証	TELEC	
電源電圧	DC 2.7～3.6V	
消費電力	アイドル時	0.1W
	高負荷時	0.15W

4-5. OpenBlocks IoT BX0 本体仕様

型番	OBSBX0	
CPU	モデル	インテル® Atom™ プロセッサ
	動作速度	500MHz (デュアルコア)
メインメモリ	オンボード	1GB(LPDDR3)
FLASH ROM	4GB(eMMC)	
追加ストレージ	-	
無線インターフェース	BT 4.0	
	IEEE802.11a/b/g/n	
有線インターフェース	BX1 コネクタ 標準添付：USB2.0(バスパワー・type-A、 Ether 二股)	
寸法	41.6mm(W) x 96mm (D) x 11.3mm(H) (突起部 含まず)	
重量	45g	
電源	・ DC 4.75～5.25V (USB) ・ PoE 受電(DC48V) TypeA(1-2,3-6)	
消費電力	アイドル時	1.9W
	高負荷時	2.4W
EMC 規格	VCCI Class-B 適合	
省エネ法に基づくエネルギー消費効率 [単位：W/GTOPS]*1	区分：H	
	0.24	
環境保護	RoHS 指令 PFOS 規制 省エネルギー法	
認証	TELEC	

MTBF	479,898h
RTC バックアップ時間	約 16 年
出荷時 OS	Debian GNU/Linux

※1: エネルギー消費効率とは省エネ法で定める測定方法により測定した消費電力を省エネ法で定める複合理論性能で除したものです

4-5. OpenBlocks IoT BX3L 本体仕様

型番	OBSBX3L	
CPU	モデル	インテル® Atom™ プロセッサ
	動作速度	500MHz (デュアルコア)
メインメモリ	オンボード	1GB(LPDDR3)
FLASH ROM	4GB(eMMC)	
追加ストレージ	-	
無線インターフェース	LTE(W-CDMA 含む) ^{※1}	
	BT 4.0	
	IEEE802.11a/b/g/n	
SIM インターフェース	通信用 miniSIM(25mm x 15mm x 0.76mm) カードスロット	
有線インターフェース	BX1 コネクタ 標準添付 : USB2.0(バスパワー・type-A)	
寸法	41.6mm(W) x 96mm (D) x 11.3mm(H) (突起部 含まず)	
重量	45g	
電源	DC 4.75~48V ※USB ケーブルによる給電時 DC4.75~5.25V	
消費電力	アイドル時	1.8W
	高負荷時	4.6W
MTBF	510,256h	
EMC 規格	VCCI Class-B 適合	
省エネ法に基づくエネルギー消費効率 [単位 : W/GTOPS] ^{※2}	区分 : H	
	0.23	
環境保護	RoHS 指令 PFOS 規制 省エネルギー法	
認証	JATE/TELEC	
RTC バックアップ時間	最大約 10 分	
出荷時 OS	Debian GNU/Linux	

※1 : Sotbank LTE 網に対応

※2 : エネルギー消費効率とは省エネ法で定める測定方法により測定した消費電力を省エネ法で定める複合理論性能で除したものです

第 5 章 注意事項

5-1. スクリプト処理による遅延処理等について

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。

5-2. 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。

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込ま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% ほぼ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様になります。

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

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 書くと寿命がつかけてしまいます。(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。

●RAM ディスクの利用で 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 RAM disk 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 RAM disk の領域にリンクを張って対処します。こういった、オープンソースなど想定外のストレージアクセスを見つけるためには、iotop コマンドがお奨めです。

●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 **RAM disk** 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。

OpenBlocks IoT Family 向け 開発者向けガイド
(2017/07/20 第9版)

ぷらっとホーム株式会社

〒102-0073 東京都千代田区九段北 4-1-3 日本ビルディング九段別館 3F