

OpenBlocks IoT BX1

先行開発キット ユーザーズガイド



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。
- ・ 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。

ご使用上の注意

本装置を安全にお使いいただくために

本書は、本装置を安全にお使い頂くために必要な注意事項を記載しております。

注意書きに従い正しくお使い頂けない場合、けが・火災・装置の破損などの原因になる事があるので、必ず注意事項を守ってください。

警告および注意表示のマーク

△警告	誤った取り扱いをしたときに、死亡や重傷に結び付く可能性があるもの。
△注意	誤った取り扱いをしたときに、障害または家屋家財などの損害に結び付くもの。
!	守らなければいけない指示。
⊘	やってはいけない事。

やってはいけないこと

△警告	⊘	本装置の内部にクリップなどの異物を入れないでください。火災や感電の原因になります。
	⊘	石油類やマニキュアの除光液など引火性のある液体を本装置の近くで使わないでください。火災の原因になります。
	⊘	ケーブルを抜く時はケーブル部分を引っ張って抜かないでください。ケーブルが痛むと発煙・火災の原因になります。
	⊘	本装置を改造しないでください。誤った改造を行った場合、火災の原因になる事があります。
△警告	⊘	本装置の掃除にシンナーやベンジンなどを使用しないでください。乾いた布で汚れを拭き取る程度にしてください。
	!	本装置の電源を入れたまま、フタを空けないでください。本装置が壊れたり感電の原因になる事があります。
	!	AC ケーブルは特に、引っ張らない、折り曲げない、熱器具のそばで使わない様に扱ってください。雑に取り扱くと銅線の露出などで感電や火災の原因になります。

設置・移設の際の注意

本装置の設置や移設の際に守らないと、けが・火災の原因になります。

△警告	⊘	不安定な場所へは設置しないでください。ぐらついた台の上や傾いた不安定な場所に設置すると本装置が落ちたりするので危険です。
	⊘	本装置を湿気やほこりの多い場所に設置しないでください。火災や感電の原因になる事があります。
	⊘	電源ケーブルやネットワークケーブルの上に重いものを置かないでください。ケーブルの破損が火災の原因になる事があります。
	⊘	ゆるいAC コンセントにつながらない。AC コンセントの接触不良が火災の原因になる場合があります。
	⊘	本装置専用の AC アダプタ・AC ケーブル以外を使わないでください。
△注意	⊘	本装置の上に物を置いたり、布などをかけないでください。本装置が放熱できなくなり火災の原因になる場合があります。
	!	長期間、本装置を使わない場合は、ケーブル類を外して、湿気のない場所で保管してください。
	!	本装置は AC100V50/60Hz の AC コンセントにつないでお使いください。

運用中における注意

△警告	⊘	煙がでたり、変なにおい・音がしたら使うのを中止してください。
	!	故障した時、AC ケーブルやネットワークケーブルをつないだままにしないでください。
	!	製品寿命を超えて本装置を使用することはお勧めしません。特にACアダプタは電解コンデンサを使用しているため寿命を超えた継続利用は発煙/故障の原因になる場合があります。
△注意	⊘	電池を適切でない種類のものとの交換した場合、爆発の危険があります。電池は、絶対に取り外し又は交換しないでください。
	!	製品の利用完了後に電池を廃棄する場合は、電池の充電、分解、炎の中へ投げ込む様なことはしないでください。また、電池は、各自治体の指示に従って処分して下さい。

目次

1. はじめに	5
1.1. パッケージの内容	5
1.2. 各部の名称 (OpenBlocks IoT BX1 本体)	6
1.3. 各部の名称 (開発ボード)	7
2. 利用の準備	8
2.1. OpenBlocks IoT BX1 本体の実装	8
2.2. 電源の接続	8
2.3. USB コンソールケーブルの接続	9
2.4. WiFi クライアントの設定	9
2.5. フラッシュストレージ eMMC(4GB)のパーティション内容	10
2.6. OpenBlocks が採用している RAM Disk モードとストレージ併用モード	11
2.7. ウォッチドッグとシャットダウン、電源切断について	12
2.8. RTC	12
2.8. 省電力化	12
2.9. LED の表示色	13
2.10. Bluetooth の起動	14
2.11. Yocto Linux の起動方法	14
3. 開発ボード	15
3.1. システム構成ブロック図	15
3.2. 開発ボード内部ブロック図	15
3.3. 開発ボードインターフェース仕様	16
3.4. 開発ボードインターフェースコネクタピン配置	17
3.5. 開発ボードジャンパピン配置	19
3.6. 開発ボードオンボードデバイス	19

1. はじめに

1.1. パッケージの内容

OpenBlocks IoT BX1 本体 1台

(3G 非対応版です。)



開発ボード 1台



USB コンソールケーブル 1本



USB Type-A micro USB ケーブル 1本



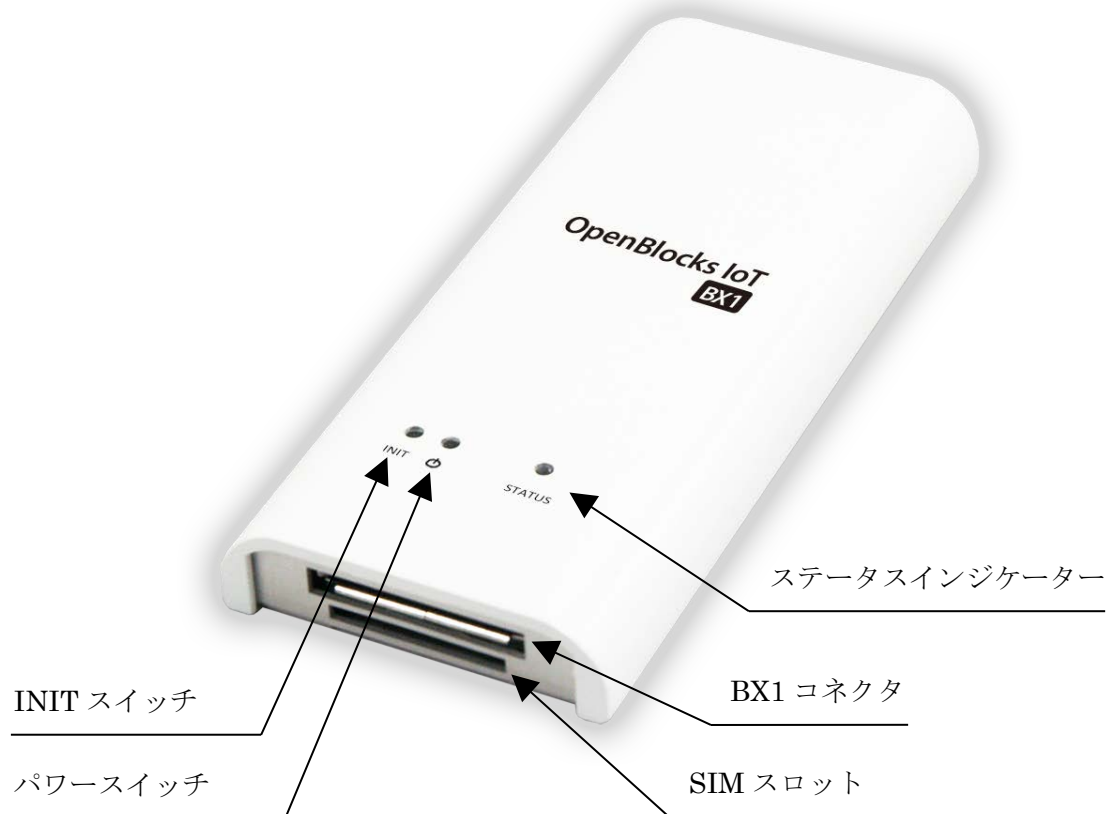
スペーサーとネジ 各4コ



ご使用にあたって 1部



1.2. 各部の名称 (OpenBlocks IoT BX1 本体)



ステータスインジケータ
7色のLEDで点灯します。

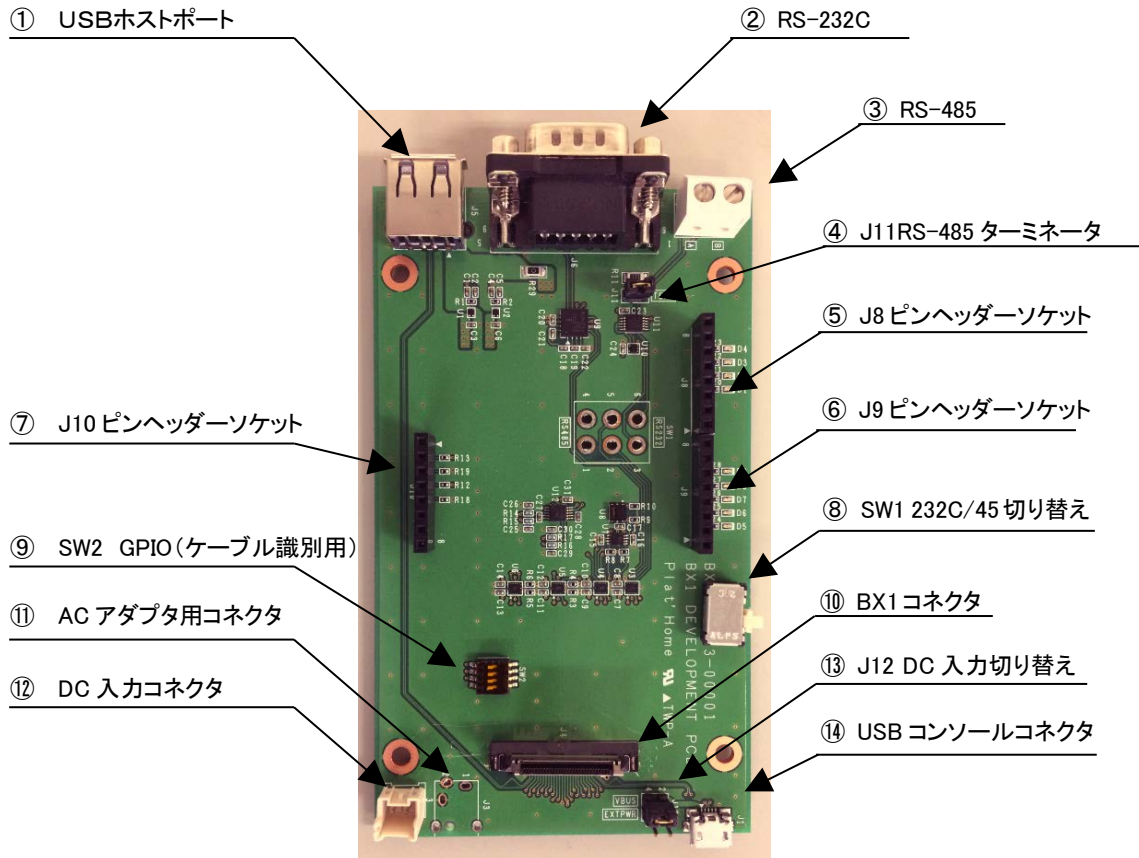
BX1 コネクタ
様々な I/O ケーブルを接続するコネクタです。

SIM スロット (本モデルでは未使用)
3G 回線の契約した SIM を挿入するスロットです。(NTT Docomo 系)

INIT スイッチ
初期化などの機能を割り当てるスイッチです。(GPIO でステータスを読みます。)

パワースイッチ (RAM Disk モードでの運用でない時はこのボタンで必ずシャットダウン)
長押しすると本装置をシャットダウンしての電源を切ります。
※電源を ON する時は BX1 コネクタを挿しなおしてください。

1.3. 各部の名称（開発ボード）



- ① USB ホストポート
USB メモリーなどのデバイスを接続するコネクタです。(Type-A)
- ② RS-232C DTE ポート (Edison の UART1 に接続)
RS-232C 用ポートで⑤のスイッチで RS-485 と切り替えて使います。
- ③ RS-485 半二重ポート (Edison の UART1 に接続)
RS-485 用ポートで⑤のスイッチで RS-232C と切り替えて使います。
- ④ RS-485 用ターミネータインーブルジャンパ
RS-485 のデバイスとして終端位置にある時ジャンパします。
- ⑤ J8 ピンヘッダーソケット
各種 I/O 接続用コネクタです。ピン配置は別項を参照ください。
- ⑥ J9 ピンヘッダーソケット
各種 I/O 接続用コネクタです。ピン配置は別項を参照ください。
- ⑦ J10 ピンヘッダーソケット
各種 I/O 接続用コネクタです。ピン配置は別項を参照ください。
- ⑧ RS-232C/RS-485 切り替えスイッチ
RS-232C か RS-485 のどちらを使うか切り替えるためのスイッチです。
- ⑨ SW2 ケーブル識別用 DIP スイッチ
ケーブル識別用に BX1 コネクタに3ビット GPIO を割り振っています。
- ⑩ BX1 コネクタ
このコネクタに OpenBlocks IoT BX1 を実装します。
- ⑪ AC アダプタコネクタ
オプションの AC アダプタを接続するコネクタです。⑬の J12 の設定が必要です。
- ⑫ DC 入力コネクタ
DC5~48V の電源入力が可能です。(48V には対応できません。)
- ⑬ J12 DC 入力切り替えジャンパ
電源入力が⑪か⑫を使う場合 EXTPWR ヘジャンパ、⑭の USB コンソールのバスパワーを使う場合は VBUS ヘジャンパします。
- ⑭ USB コンソールコネクタ (バスパワー供給可能、マイクロ USB コネクタ)
USB 接続されていますが OpenBlocks IoT BX1 内で USB シリアル変換され Edison の UART2 に接続されます。PC の USB ポートに接続すると自動的に USB シリアルドライバがインストールされ BX1 の CUI が利用出来ます。

2. 利用の準備

2.1. OpenBlocks IoT BX1 本体の実装

下記写真の通り、OpenBlocks IoT BX1 を開発ボードの BX1 コネクタに接続します。
また開発ボードが直置きにならないように添付のスペーサー4つをネジ止めします。



2.2. 電源の接続

OpenBlocks IoT BX1 の開発ボードへの電源ケーブルは三通りあります。

- (1) USB コンソールコネクタ⑭からバスパワーでの電源供給 (J12 ジャンパ⑬を VBUS に設定)
OpenBlocks IoT BX1 が内蔵する 3G モジュールを使ってネットワーク通信する場合や、USB ホストポートコネクタ①に何らかの USB デバイスを接続した時、消費電流が 500mA を超えるケースもあるのでバスパワー1000mA で電流供給できる装置へ接続してください。
- (2) オプションの AC アダプタ⑩による電源供給 (J12 ジャンパ⑬を EXTPWR に設定)
消費電力の大きい USB デバイスを接続する場合は、オプションの AC アダプタを使用し、ここへ接続してください。※AC アダプタは OpenBlocks A6/A7 用のものが利用できます。
- (3) DC 入力コネクタからの電源供給
OpenBlocks IoT BX1 はワイドレンジの直流入力ができます。
DC+5~48V とオートバイや自動車のバッテリーでの利用も可能です。
接続用ケーブルは圧着コネクタを自作する必要があります。
推奨コネクタ

メーカー : 日本圧着端子製造

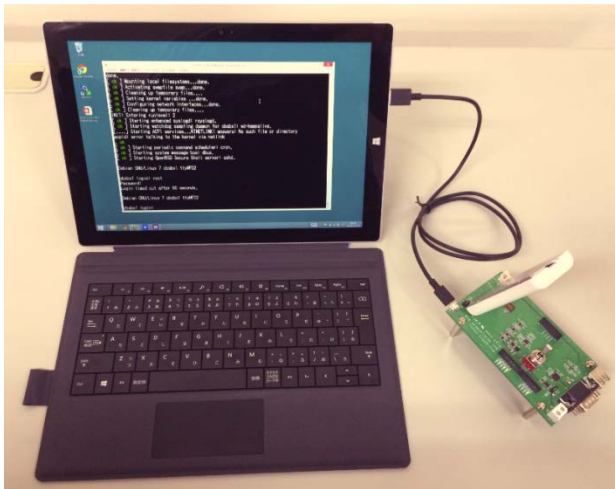
ハウジング : XAP-02V-1

コンタクト : SXA-001T-0.6

2.3. USB コンソールケーブルの接続

下の写真の様に、⑭USB コンソールコネクタと PC を接続します。

(この例では USB コンソールケーブルからのバスパワーで電源供給しています。)



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。ドライバのインストールが済んだら PuTTY などのターミナルソフトでシリアルポート接続出来ます。OpenBlocks IoT BX1 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

パリティ: 無し

データ長: 8bit

ストップ: 1bit

通信が確立したらログインプロンプトが表示されるのでデフォルト root 権限でログインし操作を開始してください。(以下は出荷時のデフォルト)

login: root

Password: root

2.4. WiFi クライアントの設定

本装置を WiFi ネットワークに参加させることで SSH などでの利用可能になるので便利です。

WiFi クライアント利用する時は先に wpa.conf ファイルを記述します。

```
# cat /etc/wpa_supplicant/wpa.conf
ctrl_interface=/var/run/wpa_supplicant
ap_scan=1
network={
    ssid="SSID"
    key_mgmt=WPA-PSK
    proto=WPA WPA2
    pairwise=CCMP TKIP
    group=CCMP TKIP
    psk="PASSWORD"
}
```

SSID と PASSWORD の部分はアクセス
ポイント毎の設定です。

WiFi 起動

```
# /sbin/wpa_supplicant -s -B -P /var/run/wpa_supplicant.wlan0.pid -i wlan0 -D nl80211,wext -c  
/etc/wpa_supplicant/wpa.conf
```

また、OS 起動毎に WiFi を自動起動する時は /etc/network/interfaces に API の接続パラメータを追記し

ます。

```
# vi /etc/network/interfaces
# Used by ifup(8) and ifdown(8). See the interfaces(5) manpage or
# /usr/share/doc/ifupdown/examples for more information.
auto lo
iface lo inet loopback
auto wlan0
iface wlan0 inet dhcp
        wpa-conf /etc/wpa_supplicant/wpa.conf
#auto wlan0
#iface wlan0 inet static
#       address 192.168.254.254
#       network 192.168.254.0
#       netmask 255.255.255.0
#       broadcast 192.168.254.255
```

2.5. フラッシュストレージ eMMC(4GB)のパーティション内容

本装置に搭載している CPU モジュール Edison のフラッシュストレージ eMMC(4GB)は GPT で領域確保しているため fdisk ではパーティションを見ることが出来ません。

パーティション内容

```
Model: MMC H4G1d (sd/mmc)
Disk /dev/mmcblk0: 3909MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Number  Start   End     Size    File system  Name          Flags
  1      1049kB  3146kB  2097kB                u-boot0
  2      3146kB  4194kB  1049kB                u-boot-env0
  3      4194kB  6291kB  2097kB                u-boot1
  4      6291kB  7340kB  1049kB                u-boot-env1
  5      7340kB  8389kB  1049kB  ext4            factory
  6      8389kB  33.6MB  25.2MB                panic
  7      33.6MB  67.1MB  33.6MB  fat16           boot
  8      67.1MB  604MB   537MB   ext4            rootfs
  9      604MB   1409MB  805MB   ext4            update
 10     1409MB  3909MB  2500MB  ext4            home
```

パーティション変更禁止

なお、本システムではこのパーティション内容の変更を禁止しております。

Edison のフラッシュストレージには、ベース OS として Yocto Linux が搭載されており、本システムでは エマージェンシーboot 用に残してあります。

BX1 のファームウェアは、/dev/mmcblk0p8 に保存してあります。

```
root@obsbx1:~# mount /dev/mmcblk0p8 /mnt
root@obsbx1:~# ls /mnt
bin          factory      nodeappslot  sketch
boot        first-install.log  opt          sys
bzImage     home         proc         tmp
bzImage.bak lib          ramdisk-wheezy.obsbx1.img.gz  userland.tgz
dev         lost+found   ramdisk-wheezy.obsbx1.img.gz.bak  usr
etc         media       run          var
etc.tgz     mnt         sbin
```

bzImage が Linux カーネルで ramdisk-wheezy.obsbx1.img.gz が基本ファームウェアの Ram Disk イメージです。etc.tgz と userland.tgz が本装置を Ram Disk モードで使った時の保存対象ファイルの圧縮イメージです。

etc.tgz と userland.tgz は flashcfg コマンドで Ram Disk のイメージを書き戻したファイルで、その詳細については当社の WEB 情報サイト OBDN での解説を参照ください。

このパーティションに基本ファームウェアが無い場合は Yocto Linux が起動します。

2.6. OpenBlocks が採用している RAM Disk モードとストレージ併用モード

本装置は M2M などのゲートウェイとしての利用を想定しているため、実運用する場合には RAM Disk モードでの利用をお奨めしております。RAM Disk モードの場合、不意な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起こさないためです。

ただし、本装置のデフォルトの出荷設定ではストレージ併用モードとしております。

RAM Disk モードとは、基本的なファームウェアを 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がその RAM Disk 上で動作するので、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壊することがなく、この場合でも電源の再投入だけでサービスを再スタートできます。このモードで利用するアプリケーションやサーバ固有の設定情報も、eMMC システム領域に専用コマンドで書き戻せば決まったアプリケーションを RAM Disk 起動することが出来ます。

ストレージ併用モードとは、基本ファームウェアを展開した RAM Disk に、AUFS を使い EMMC ストレージのユーザーパーティション home の領域を被せるような作りになっています。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 home 上の更新ファイルが破壊される危険性は残ります。ただ、こういったファイル破壊はフラッシュ ROM 上の基本ファームウェア部分には全く影響しないので、トラブルの際には最悪 AUFS で上乗せした SSD を一度アンマウントし再初期化してからマウントすれば、初期状態でシステムを起動できるので非常に短い時間で障害からの回復が可能になります。

このように AUFS で RAM Disk に eMMC ストレージを上乗せするかしないかは、ストレージのボリュームラベルに"DEBIAN"と設定した場合、自動的に適用されます。ゆえに単純に RAM Disk モードで運用したい場合は、OpenBlocks に接続されているストレージデバイスに"DEBIAN"というラベルを付けなければ良いわけです。

設定例

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

エンベデットでの利用シーンが多い OpenBlocks シリーズ全般でこの仕組を取り入れています。

その他の運用方法では、RAM Disk モードでアプリの基本部分を運用し、ログだけは SSD などのストレージに保管したいケースの場合は、出荷時点で home というパーティションを初期化し、そのまま単純に SSD の空いている領域を mkfs し任意のディレクトリにマウントすれば利用できます。もちろん、そのディレクトリに書き込まれた内容は電源が切られても残ります。

ただし、不意な電源切断された後の再起動時、稀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。

2.7. ウォッチドッグとシャットダウン、電源切断について

OpenBlocks IoT BX1 には HW ウォッチドッグが組み込まれており、OS の万が一のハングアップ時には強制リセットによる再起動が起きます。

また、本開発キットの場合の制限事項として、以下の3つの仕様があります。

1. シャットダウンをコマンド入力して `halt` 状態になるとウォッチドッグによって強制リセットされるので OS が再起動します。
2. 電源ボタンを短く押した場合も、1と同様に `halt` 後に再起動します。
3. 電源を切るには電源ボタンを長押しします。長押しは LED が消灯するまで続けます。また電源を再投入するには USB コンソールケーブルの抜き差しが必要です。



※この3つの仕様は量産モデルで変更される予定です。

2.8. RTC

本装置の RTC にはバックアップ電池が用意されていませんが、電源やバッテリーの切り替え程度の時間、満充電で約 10 分程度をスーパーキャパシタでバックアップしています。

したがって基本的には `ntp` などを利用してシステムクロックとの時間同期を利用するようお願いします。

手動で時刻を設定する時の手順 (`hwclock` コマンドを使用する時は `-directisa` オプションを付けます。)

```
# date 110716092014          ← 現在の時刻と日付を入力してください。
Fri Nov  7 16:09:00 JST 2014
# hwclock --systohc
hwclock: select() to /dev/rtc to wait for clock tick timed out: Success
# hwclock --systohc --directisa
```

2.8. 省電力化

本装置は USB コンソールケーブル経由で Linux のコマンドラインを操作しますが、実際の運用に入った時はほとんどコンソールが不要になるので、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。USB シリアルチップの電源を常時オン状態での Linux 無負荷状態では消費電流が 60mA 程度ですが、自動制御にすると 35mA 程度に抑えられるのでバッテリーなどでの運用時間が有利になります。反面、自動制御にした場合、コンソールでのキー入力時に電源 OFF 状態から復帰時に最初のキーを取りこぼす場合があります。

USB シリアルチップの制御

電源制御を自動にする

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

電源を常時オンにする

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

2.9. LED の表示色

OpenBlocks IoT BX1 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可能です。

量産時の OpenBlocks IoT BX1 標準的な LED 点灯色は以下を予定していますが、開発キットでは 3G モジュールが搭載されていないので、下表の太字の通りに機能します。

状態	色	点灯状態	備考
OS 起動中	黄	点灯	OS 起動が終わると 3G 回線電波受信チェックへ移行します。※3G が起動できない時は緑点灯。
3G 回線電波受信チェック	白(電波強) 水色(電波強) 青(電波弱) 紫(圏外)	高速点滅	BX1 本体を手で持って電波状況の良いところ探すためのモードです。 OS 起動後数分間 5 秒おきに 3G の電波状況をチェックし LED の点灯色を電波強度に従って更新します。
3G 未使用での運用	緑	点灯	SIM がない状態での正常稼働状態
3G 電波(強 Lv4,5)待機中	白	点灯	電波強度-87dB 以上
3G 電波(中 Lv3)待機中	水色	点灯	電波強度-88~-106dB
3G 電波(弱 Lv1,2)待機中	青	点灯	電波強度-105~-129dB
INIT ボタンを押下 リブート	黄	点灯	OS リブート。
電源ボタンを押下 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

LED 制御

LED の表示色・点灯状態を変更する場合は、/dev/.runled ファイルの内容を編集します。

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
0	黒
1	赤
2	緑
3	黄
4	青
5	紫
6	水色
7	白

例) 1 秒ごとに黄色を点滅

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

2.10. Bluetooth の起動

OpenBlocks IoT BX1 は以下の手順で起動してください。

```
# 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
# hciconfig
hci0: Type: BR/EDR Bus: UART
      BD Address: xx:xx:xx:xx:xx:xx ACL MTU: 1021:8 SCO MTU: 64:1
      UP RUNNING PSCAN
      RX bytes:954 acl:0 sco:0 events:42 errors:0
      TX bytes:987 acl:0 sco:0 commands:42 errors:0
```

以上で起動状態です。

実際の運用方法についてはデバイスによるので、各 Bluetooth の情報サイトを参照ください。

2.11. Yocto Linux の起動方法

OpenBlocks IoT BX1 には Edison の Yocto Linux がエマージェンシーブート用に残されています。

デフォルトでは OpenBlocks シリーズ用の Debian Linux が起動しますが、u-boot 上で Yocto Linux ブートに切り替えることができます。なお、リブートすると Debian Linux に戻ります。

```
*****
PSH KERNEL VERSION: b0182727
      WR: 20104000
*****                               ← スペースを押しっぱなしで電源をいれてください。

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

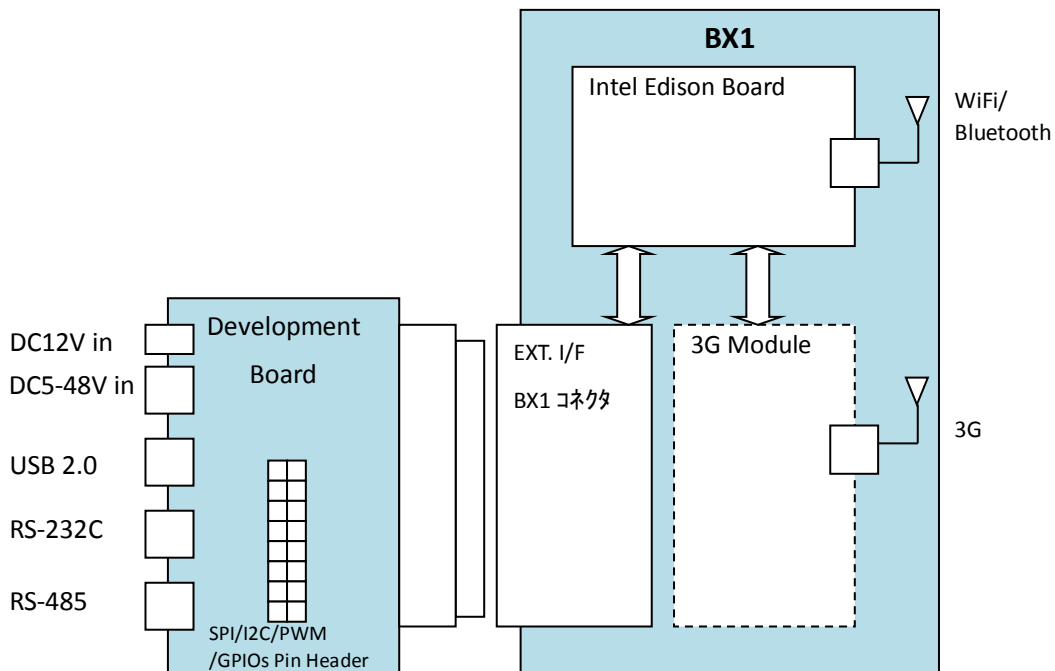
*** Ready to receive application ***

U-Boot 2014.04 (Aug 20 2014 - 16:08:32)

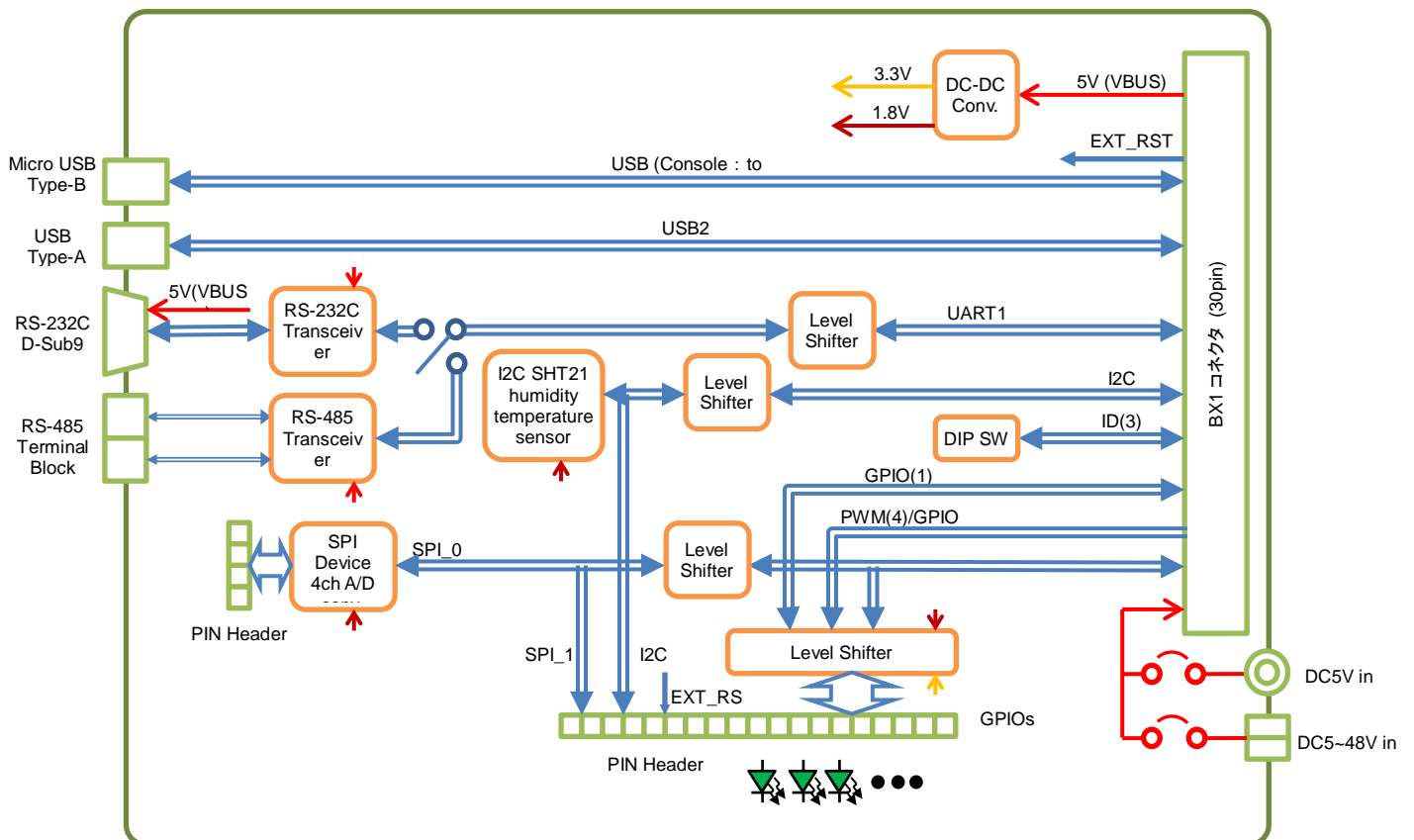
      Watchdog enabled
DRAM: 980.6 MiB
MMC: tangier_sdhci: 0
In: serial
Out: serial
Err: serial
Hit any key to stop autoboot: 0
boot > run bootYocto                               ← 止まったら run bootYocto と入力
```

3. 開発ボード

3.1. システム構成ブロック図



3.2. 開発ボード内部ブロック図



3.3. 開発ボードインターフェース仕様

モデル		BX1
外部 I/F	UART to USB	microUSB Type-B
	USB 2.0×1	Type-A
	RS-232C×1 *1	D-Sub9,(Tx,Rx,CTS,RTS) ※DSR,DTR はボード内でループバック
	RS-485×1 *1	2極 端子台
	I ² C×1 *2, *3	3.3V I/O
	SPI×1 *2, *3	1 デバイス 3.3V I/O
	PWM×4 *2, *3	3.3V I/O
	GPIO×1 *2	3.3V I/O
	A/D 入力(SPI_0)	差動 2ch または シングルエンド 4ch、0-3.3V input
オンボード ・デバイス	温湿度センサー(I2C)	Alps 社製 HSHCAL001B
	A/D コンバーター(SPI)	TI 社製 ADS1018
Board 本体	Board 寸法 (mm)	70 (W) × 120 (D)
	重量 (g)	50
	付属品	スペーサー×4, ネジ×4
温度・湿度	周囲温度 動作時	0~40℃
	周囲湿度 動作時	20~80%Rh ※結露なきこと
Power	入力	DC5V(microUSB Type-B) or DC-Jack(EIAJ Type-2) or DC5~48V(AUX Power)

*1 排他使用(ボード上のスライドスイッチにて切替)

*2 ピンヘッダーソケット

*3 GPIO として使用可

3.4. 開発ボードインターフェースコネクタピン配置

J1 Console & BUS Power IN (micro USB Type-B)

No.	信号
1	VBUS
2	D-
3	D+
4	ID
5	GND

J2 AUX Power IN (XA 2.5mm 2P)

No.	信号
1	VIN (5-48V)
2	GND

J3 DC Jack Power IN (EIAJ Type2)

No.	信号
1	VIN (5V)
2	GND
3	GND

J4 BX1 Connector 30P

No.	信号	No.	信号
1	DC_IN	16	GND
2	DC_IN	17	SPI_FS1 / GP111
3	UART2USB_D-	18	EXT_RST# (Open collector)
4	UART2USB_D+	19	PWM0 / GP12
5	I2C_SCL / GP27	20	PWM1 / GP13
6	I2C_SDA / GP28	21	PWM2 / GP182
7	UART_Tx / GP131	22	PWM3 / GP183
8	UART_Rx / GP130	23	SD_CD# / GP77
9	UART_RTS / GP129	24	ID0 / GP44
10	UART_CTS / GP128	25	ID1 / GP45
11	SPI_CLK / GP119	26	ID2 / GP46
12	SPI_TXD / GP115	27	USB2_D+
13	SPI_RXD / GP114	28	USB2_D-
14	SPI_FS0 / GP110	29	GND
15	V_BUS	30	GND

J5 USB2.0 (Type-A)

No.	信号
1	VBUS
2	D-
3	D+
4	GND

J6 RS-232C (D-Sub9) ※J7 RS-485 と排他。(ボード上のトグルスイッチで切替え)

No.	信号	No.	信号
1	N.C.	6	DSR
2	RxD	7	RTS
3	TxD	8	CTS
4	DTR	9	VIN (5V)
5	GND	-	-

J7 RS-485 (端子台 2P) ※J8 RS-485 と排他。(ボード上のトグルスイッチで切替え)

No.	信号
1	B
2	A

J8 GPIO (ピンヘッダーソケット(角ピン)2.54mmピッチ ストレート(1列))

No.	信号	No.	信号
1	+3.3V	5	SPI_2_TXD
2	I2C_6_SCL	6	SPI_2_RXD
3	I2C_6_SDA	7	GP111/SPI_2_FS1
4	SPI_2_CLK	8	GND

※各 GPIO ポートには LED が接続されており、Hight で点灯。

J9 GPIO (ピンヘッダーソケット(角ピン)2.54mmピッチ ストレート(1列))

No.	信号	No.	信号
1	+3.3V	5	GP183/PWM3
2	GP12/PWM0	6	GP77
3	GP13/PWM1	7	EXT_RST# (Open collector)
4	GP182/PWM2	8	GND

※各 GPIO ポートには LED が接続されており、Low で点灯。

J10 A/D IN (ピンヘッダーソケット(角ピン)2.54mmピッチ ストレート(1列))

No.	信号	No.	信号
1	+3.3V	5	AIN_B-
2	AIN_A+	6	GND
3	AIN_A-	7	N.C.
4	AIN_B+	8	N.C.

3.5. 開発ボードジャンパピン配置

J11 RS-485 Termination (ピンヘッダー ピン(角ピン)2.54mm ピッチ ストレート(1 列))

No.	信号
1	B
2	A

1-2pin Short : Term On

J12 Power Select (ピンヘッダー ピン(角ピン)2.54mm ピッチ ストレート(2 列))

No.	信号
1	VIN (5-48V)
2	VBUS
3	VOUT
4	VOUT

1-3pin Short : DC Jack or AUX Power

2-4pin Short : MicroUSB BUS Power

3.6. 開発ボードオンボードデバイス

これらはサンプルデバイスとして本開発ボードに搭載しています。

I2C デバイス: Alps 社製 HSHCAL001B 温湿度センサー

SPI デバイス: TI 社製 ADS1018 4ch-A/D コンバーター

利用方法については順次当社開発者向け情報サイト OBDN Magazine にて公開いたします。

OBDN Magazine

<http://obdnmagazine.blogspot.jp/>

OpenBlocks IoT BX1 先行開発キットユーザーズガイド

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