

# CONEXIO BlackBear 取扱説明書



Ver. 2.0.5

2022年1月11日

コネクシオ株式会社

## 変更履歴

Version	Date	Updates
1.0.0	Oct. 10, 2019	初版発行
1.0.1	Oct. 23, 2019	文言の微修正及び誤記修正を実施
1.0.2	Oct. 29, 2019	文言の微修正及び追記を実施
1.0.3	Oct. 30, 2019	仕様変更事項の修正を実施
1.0.4	Nov. 11, 2019	WiFi AP 設定に関する制限事項の修正を実施
1.0.5	Nov. 18, 2019	時刻設定に関する記述の修正及び追記を実施
1.0.6	Nov. 28, 2019	WiFi 機能の仕様変更に対応して記述の修正及び追記を実施
1.0.7	Dec. 9, 2019	WiFi 機能の仕様変更に対応して記述の再修正を実施
1.0.8	Dec. 19, 2019	MCU Firmware 更新に関する注意事項を追記
1.0.9	Dec. 20, 2019	文言の微修正及び追記を実施。
1.1.0	Jan. 16, 2020	起動時の自動的な WiFi 機能有効化に関する記述を追記
1.1.1	Jan. 22, 2020	ケーブル接続の記載を修正
1.1.2	Apr. 1, 2020	Pre-installed ソフトウェアの表を修正 文言の微修正を実施
1.1.3	May 22, 2020	Resume 実行方法に関する記述を追記 LTE Module 電源の OFF/ON コマンドに関する記述を追記 OSS に関する注意事項の追記を実施
1.1.4	May 26, 2020	Pre-installed ソフトウェアの表を修正
1.1.5	May 28, 2020	Pre-installed ソフトウェアの表を修正 Resume 実行方法に関する記述の一部追記を実施 文言の微修正及び追記を実施
1.1.6	Jun. 4, 2020	USB I/F に関する記述の修正および追記を実施
1.1.7	Jun. 12, 2020	GNSS Antenna の図を修正
1.1.8	Jun. 29, 2020	車載用 Power Cable の図及びケーブル接続の文言を修正
2.0.0	Sep. 1, 2020	これまでの問い合わせ事項を考慮して大幅な追加・修正を実施
2.0.1	Oct. 1, 2020	仕様一覧表”Certification”および”CAN I/F”記載内容の修正を実施 CAN の仕様に関する記述の追記を実施 Resume 機能に関する記述の修正を実施 GNSS 機能に関する記述の修正を実施
2.0.2	Nov. 4, 2020	付属品アンテナのコネクタに関する記述の修正を実施
2.0.3	Jan. 22, 2021	exFAT ファイルシステムの mount に関する注意事項の追記を実施
2.0.4	Mar. 26, 2021	PAP 認証の場合の LTE 接続設定に関する記述の追記を実施 OS の差分更新に関する記述の追記を実施
2.0.5	Jan. 11, 2022	文言の微修正を実施 Pre-installed ソフトウェアの表を修正 付属品のケーブル色に関する記述の追記を実施 CAN I/F で使用可能な機能等に関する記述の追記を実施 DI/DO の使い方に関する記述の追記を実施 差分更新後の version 情報表記に関する記述の追記を実施 セキュリティ基準適合認定に関する記述の追記を実施 本社移転に伴い問合せ先を変更

# [ CONEXIOBlackBear 取扱説明書 目次 ]

<b>1.製品概要</b> .....	<b>1</b>
1.1.はじめに.....	1
1.2.製品外観.....	3
<b>2.製品仕様</b> .....	<b>4</b>
2.1.筐体寸法図 .....	4
2.2.仕様一覧表 .....	5
2.3.ブロック図.....	8
2.4.インターフェースレイアウト図 .....	9
2.5.付属品 .....	10
2.5.1.屋内用セット・車載用セット共通 .....	10
2.5.2.屋内用セットのみ .....	13
2.5.3.車載用セットのみ .....	16
2.6.OS および 主なpre-installedソフトウェア.....	18
<b>3.本体使用方法（準備/起動/終了）</b> .....	<b>19</b>
3.1.microSIMおよびmicroSD挿入.....	19
3.2.アンテナ装着 .....	20
3.3.ケーブル接続 .....	20
3.4.起動（電源ON/ログイン/時刻設定/RTC）.....	22
3.4.1.電源ON .....	22
3.4.2.ログイン .....	23
3.4.3.時刻設定/RTC .....	23
3.5.Suspend – Resume .....	25
3.5.1.Suspend.....	25
3.5.2.Resume .....	26
3.6.終了（電源OFF/再起動）.....	28
3.6.1.電源OFF .....	28
3.6.2.再起動 (Reboot).....	29

<b>4.インターフェース使用方法</b> .....	<b>30</b>
<b>4.1.Ethernet</b> .....	<b>30</b>
4.1.1.DHCP接続 .....	30
4.1.2.固定IP接続 .....	30
4.1.3.DHCP/DNSサーバー機能.....	31
4.1.4.RJ45 connector.....	31
<b>4.2.LTE</b> .....	<b>32</b>
<b>4.3.WiFi</b> .....	<b>34</b>
4.3.1.アクセスポイントへの接続.....	34
4.3.2.2.4GHz アクセスポイント設定 .....	35
4.3.3.5GHz アクセスポイント設定 .....	36
4.3.4.CONEXIOBlackBear 起動時の自動的なWiFi機能有効化 .....	36
<b>4.4.Bluetooth</b> .....	<b>37</b>
<b>4.5.Serial</b> .....	<b>38</b>
<b>4.6.CAN</b> .....	<b>40</b>
<b>4.7.DI/DO</b> .....	<b>40</b>
4.7.1.DI/DO制御用ポート.....	40
4.7.2.DIの使い方 .....	42
4.7.3.DOの使い方 .....	43
<b>4.8.GNSS</b> .....	<b>44</b>
<b>4.9.USB</b> .....	<b>45</b>
<b>4.10.SD (Storage)</b> .....	<b>45</b>
<b>4.11.LED</b> .....	<b>46</b>
4.11.1.USER LED .....	46
4.11.2.WiFi/Bluetooth LED.....	47
4.11.3.LTE電波強度LED .....	47
<b>4.12.USER Switch</b> .....	<b>48</b>
<b>4.13.CPU温度</b> .....	<b>48</b>
<b>4.14.SMS</b> .....	<b>49</b>
<b>5.OS更新 (初期化)</b> .....	<b>50</b>
<b>5.1.OSのVersion確認方法</b> .....	<b>50</b>
<b>5.2.microSD Cardを使用したOS更新</b> .....	<b>51</b>
5.2.1.“SD_burn_EMMC”のダウンロード .....	51
5.2.2.Install Media (microSD Card) 作成手順 .....	51
5.2.3.CONEXIOBlackBear 本体でのOS更新手順 .....	51

<b>6.OS差分更新</b> .....	<b>53</b>
<b>6.1.概要</b> .....	<b>53</b>
<b>6.2.更新手順</b> .....	<b>53</b>
6.2.1.差分更新用OSパッケージの取得 .....	53
6.2.2.OS差分更新の実行 .....	53
6.2.3.OS差分更新後のversion情報 .....	55
<b>7.開発用ツール</b> .....	<b>56</b>
7.1.Debug Cable .....	56
7.2.Debug Port.....	57
<b>8.注意事項</b> .....	<b>58</b>
8.1.安全に使用するための取扱注意事項 .....	58
8.2.電波に関する注意事項 .....	59
8.2.1.Bluetoothの周波数帯 .....	60
8.2.2.WiFi(無線LAN)の周波数帯(2.4GHz帯) .....	60
8.2.3.WiFi(無線LAN)の周波数帯(5GHz帯) .....	60
8.3.電波法 および 電気通信事業法対応 .....	61
8.4.オープンソースソフトウェアに関する注意事項 .....	62
8.5.海外での使用について.....	62
8.6.保証事項.....	63
<b>9.お問い合わせ先</b> .....	<b>64</b>

## [図表目次]

図1.1-1 IoT Gateway Concept.....	1
図1.1-2 CONEXIOBlackBear 活用ソリューション例.....	2
図1.2-1 CONEXIOBlackBear 外観 - 上面およびコネクタ端子側.....	3
図1.2-2 CONEXIOBlackBear 外観 - アンテナ端子側 .....	3
図2.1-1 筐体寸法図 .....	4
図2.3-1 ブロック図 .....	8
図2.4-1 インターフェースレイアウト図 .....	9
図2.5-1 Serial Cable .....	10
図2.5-2 DI/DO Cable .....	11
図2.5-3 CAN Cable .....	12
図2.5-4 Holeless EX-Door.....	12
図2.5-5 LTE Antenna .....	13
図2.5-6 WiFi/Bluetooth Antenna .....	13
図2.5-7 GNSS Antenna .....	13
図2.5-8 AC Adapter .....	14
図2.5-9 AC Adapter Conversion Cable.....	15
図2.5-10 4 in 1 Antenna/Cable .....	16
図2.5-11 Power Cable.....	17
図3.1-1 microSIM および microSDの挿入 .....	19
図3.2-1 アンテナ装着.....	20
図3.3-1 ケーブル接続.....	21
図3.3-2 車載用電源ケーブル接続.....	21
図3.4-1 起動時のLED表示 .....	22
図3.5-1 Suspend時のLED表示 .....	25
図3.5-2 Resume時のLED表示 .....	26
図3.6-1 終了時のLED表示.....	28
図3.6-2 Reboot (go-sys-reboot実行) 時のLED表示 .....	29
図4.1-1 RJ45 connector LEDs .....	31
図4.5-1 Serialインターフェース切り替えスイッチ.....	38
図5.2-1 OS更新時のLED表示.....	52
図6.1-1 Debug Cable A .....	56
図6.1-2 Debug Cable B .....	56
図6.2-1 本体へのDebug Cable接続 .....	57
図7.3-1 技術基準適合証明/認定情報 - 底面ラベル .....	61

図7.6-1 TAMPER SEAL .....	63
<hr/>	
表2.2-1 仕様一覧表 .....	5
表2.2-2 モデル別型番 .....	7
表2.6-1 OS および 主なPre-installedソフトウェア .....	18
表3.4-1 電源投入 - 起動 実行方法 .....	22
表3.5-1 Suspend 実行方法 .....	25
表3.5-2 Resume 実行方法 .....	26
表3.6-1 終了 - 電源OFF 実行方法 .....	28
表3.6-2 Reboot 実行方法 .....	29
表4.1-1 Ethernet デバイス名 .....	30
表4.1-2 リンク速度LED .....	32
表4.1-3 リンク状態LED .....	32
表4.2-1 LTE Module デバイス名 .....	32
表4.2-2 LTE 有効化/無効化コマンド .....	33
表4.3-1 WiFi デバイス名 .....	34
表4.3-2 WiFi 有効化/無効化コマンド .....	34
表4.4-1 Bluetooth デバイス名 .....	37
表4.4-2 Bluetooth 有効化/無効化コマンド .....	37
表4.5-1 Serial通信 デバイスファイル .....	38
表4.5-2 Serialインターフェース切り替え設定 .....	39
表4.5-3 終端抵抗設定 .....	39
表4.6-1 CAN デバイス名 .....	40
表4.7-1 DI/DO制御用ポート .....	40
表4.7-2 各ポートの機能設定 .....	41
表4.7-3 PWR_CTLポートの設定値 .....	41
表4.8-1 GNSS Module デバイスファイル .....	44
表4.8-2 NMEAメッセージ .....	44
表4.9-1 USB制御用ポートと設定値 .....	45
表4.9-2 USB Memory デバイスファイル .....	45
表4.10-1 microSD Card デバイスファイル .....	45
表4.11-1 USER LED .....	46
表4.11-2 USER LED 制御方法 .....	46
表4.11-3 WiFi/Bluetooth LED .....	47
表4.11-4 WiFi/Bluetooth LEDの点灯条件 .....	47

表4.11-5 LTE電波強度LEDの点灯条件.....	47
表4.12-1 USER Switch インputデバイスファイル .....	48
表7.2-1 ターミナルエミュレータのシリアル通信設定 .....	57
表8.3-1 技術基準適合証明/認定情報 .....	61
表8.3-2 セキュリティ基準適合認定情報.....	61

# 1. 製品概要

## 1.1. はじめに

本書はエッジコンピューティング・ゲートウェイ CONEXIOBlackBear に関する取扱説明書です。本書では CONEXIOBlackBear の仕様や、本体および搭載されたI/Fの使用方法について記載しています。CONEXIOBlackBear へ搭載するソフトウェアの開発方法に関しては別紙「簡易プログラミングマニュアル」を参照してください。

※「簡易プログラミングマニュアル」は CONEXIOBlackBear ユーザー向けサポートサイトで入手できます。

CONEXIOBlackBear は「図1.1-1 IoT Gateway Concept」に記載したように、多様なソリューションへ対応するために多様な環境に設置された機器やセンサー等のデバイスを、インターネットを通してIoTクラウドのようなサーバーへと接続して、デバイスデータの有効活用や新たな価値を創造することを可能にするエッジコンピューティング・ゲートウェイです。



図1.1-1 IoT Gateway Concept

CONEXIOBlackBear には以下のような特長があります。

- **エッジコンピューティング能力強化** : QuadコアCPUおよびGPUを搭載して処理能力を強化したエッジコンピューティング・ゲートウェイです。これによりクラウドを介することなく現場でのエッジAI処理実行が可能になるなど、高性能かつ即時性を求められる用途にも活用することができます。
- **グローバル対応** : 日本国内の”技適”や”PSE”だけではなく、世界各国・地域の電波関連法規や製品安全に関する認証を取得しており、日本国内で開発したシステムを再開発することなくそのまま海外展開することができます。(「8.5. 海外での使用について」も参照してください)
- **車載対応** : 車載も想定した堅牢性、動作温度、動作電圧を確保し、CANインターフェース2系統およびGNSSを標準搭載するなど、特殊車両や建設機械へも後付けでIoT化を図ることができます。これにより車両の稼働分析や遠隔保守などを実現することができます。
- **多種多様なI/Fを搭載** : Ethernet、LTE、WiFi、Bluetooth、RS485/RS422/RS232C(選択)、CAN(2系統)、DI/DO、GNSSを標準搭載し、さらにWi-SUN、EnOceanなどのモジュールをオプションで追加可能(予定)であり、多種多様な機器やセンサーと接続して使用することができます。

このように CONEXIOBlackBear はグローバルに、多様な状況へ対応することが可能です。  
 CONEXIOBlackBear の活用ソリューション例を「図1.1-2 CONEXIOBlackBear 活用ソリューション例」に記載  
 します。

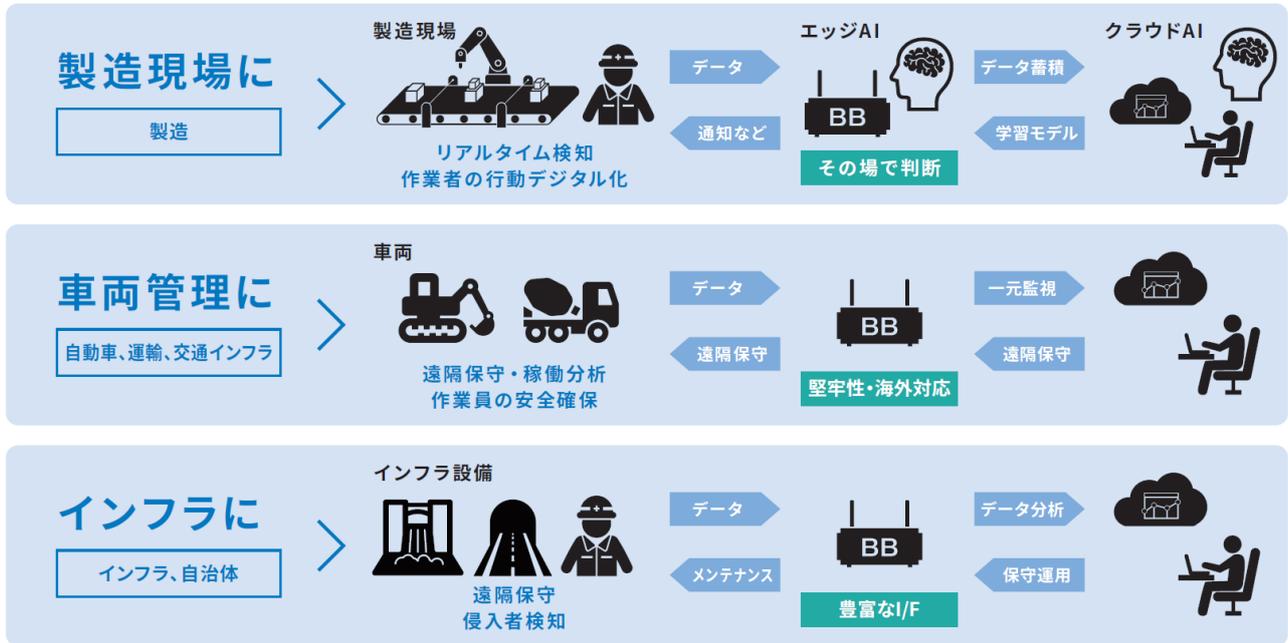


図1.1-2 CONEXIOBlackBear 活用ソリューション例

## 1.2. 製品外観

CONEXIOBlackBear の外観を「図1.2-1 CONEXIOBlackBear 外観 - 上面およびコネクタ端子側」および「図1.2-2 CONEXIOBlackBear 外観 - アンテナ端子側」に記載します。



図1.2-1 CONEXIOBlackBear 外観 - 上面およびコネクタ端子側



図1.2-2 CONEXIOBlackBear 外観 - アンテナ端子側

## 2. 製品仕様

### 2.1. 筐体寸法図

CONEXIO BlackBear 本体の筐体寸法を「図2.1-1 筐体寸法図」に記載します。

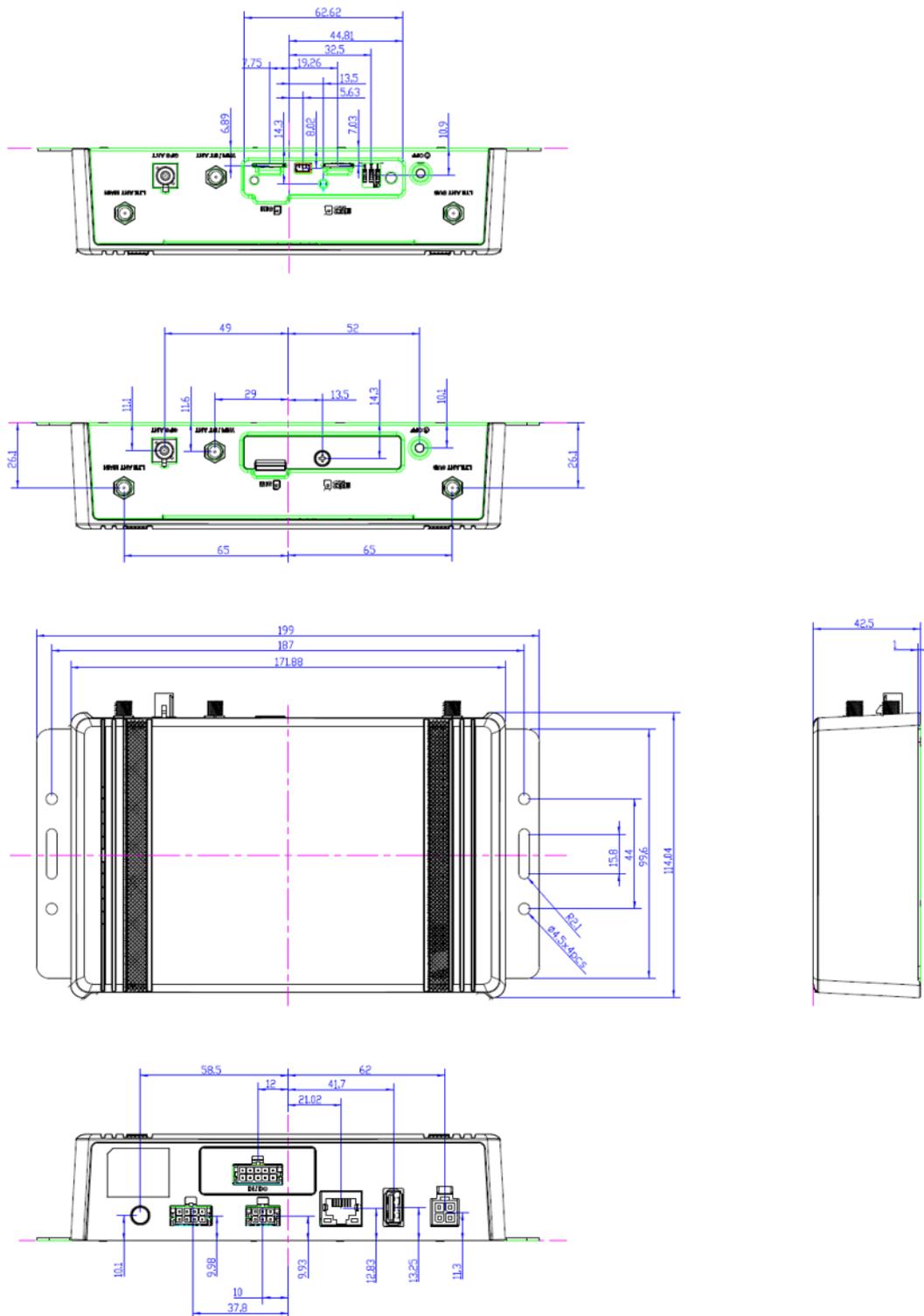


図2.1-1 筐体寸法図

## 2.2. 仕様一覧表

CONEXIO BlackBear の主な仕様を「表2.2-1 仕様一覧表」に記載します。

表2.2-1 仕様一覧表

ITEM	DESCRIPTION	DETAILS
CPU	FREESCALE/MCIMX6Q5EYM10AD	ARM Cortex-A9 1.0GHz Quad Core Built-in GPU : Vivante GC2000
Memory	DDR	DDR3, 2GB
Main Storage	Storage	eMMC (MLC), 32GB
OS	Linux Yocto	Linux Kernel Version : 4.9.88 NXP BPS Version : 4.9.88-2.0.0 Yocto Version : rocko (2.4.2)
Top Cover	LED x 6	Power : 1 color LED (Blue) LTE : 3 color LEDs (green) WiFi/BT : 2 color LEDs (Red/Green) combined into one User : 2 color LEDs (Red/Green) combined into one
Front I/O Functions	Power connector	4 pin x 1
	RJ45 connector	10BASE-Te/100BASE-TX/1000BASE-T Terminal : RJ45 x 1 (with 2 color LEDs)
	USB connector	USB2.0 Type A x 1
	Can Bus Vehicle signal connector	CAN x 2 Terminal : CAN_H,CAN_L,GND Total 6pins CAN Version 2.0B active
	Serial connector	RS232C/RS485/RS422 (Isolated) x 1 (Select) Terminal : Signal line x 4+GND x 4 : 8pins (TX,RX,DR,ER,GND/TX,RX,GND)
	DI/DO connector	Extension board DI x 4, DO x 4, (COMMON [GND] x 2)
	USER Switch	x 1
Back I/O Functions	LTE Antenna	SMA connector x 2
	WiFi/Bluetooth Antenna	SMA connector x 1 (Reverse Type)
	GNSS Antenna	Fakra connector x 1
	Power OFF button	x 1
	microSD Card	x 1 (microSDXC UHS-I SDR104)
	microSIM Card	x 1
	Serial Port Setting Switch	3 ports
	Debug Port	x 1
LTE	Interface	Mini-PCIe
	Bands/Frequency	FDD-LTE : B1(2100MHz) / B3(1800MHz) / B8(900MHz) / B26(800MHz,B18/19) (depends on JP)
	Up/Downlink rate	Cat.4 Down 150Mbps / Up 50Mbps (Ideal value of LTE Chipset)
	LTE module	SIM7600JC-H PCIE support docomo,SoftBank and KDDI
	Antenna	With SMA connector for external antenna x 2
GNSS	Module	u-blox NEO-M8N
	Frequency	GPS L1C/A, SBAS L1C/A, QZSS L1C/A, GLONASS L1OF, BeiDou B1, GALILEO E1B/C (Up to Module spec. The capability value depends on integration with the system.)
	Antenna	With Fakra connector for external antenna x 1
WiFi / Bluetooth	Module	WM-BAC-BM25 (Broadcom)
	WiFi Dual band	IEEE 802.11 ac/a/b/g/n
	Bluetooth	Bluetooth 4.2 (Bluetooth Low Energy)
	Bluetooth Profile	Depends on BlueZ
	Antenna	With SMA (Reverse Type) connector for external WiFi + Bluetooth Antenna x 1

Power supply	Power supply voltage	Applied Voltage : DC12V or 24V Car ACC/IG / Car BATT /GND : Total 3 Operation Voltage : 9~32V Absolute maximum rated voltage : 0~40V
	USB voltage	5V/0.5A (Power supply to USB device)
Power Consumption	DC12V Current Consumption	MAX : about 1.5A Suspend Mode : about 100mA Leak Current : 2mA
RTC battery	RTC backup battery	BCR1220 3V Lithium Coin Cell
Environment	Operation temperature	-20°C~70°C
	Storage Temperature	-30°C~80°C
	Humidity	5% to 95% non-condensing
	Environment	Compliant with ISO 16750-3, Class C. (RJ45 and USB are only for testing to confirm there is no problem in their ability value.)
	Vibration	2.83G (ISO 16750-3)
	Mechanical shock	50G (ISO 16750-3)
	IP code	IP40
Electrostatic Discharge	Survive $\pm 10$ kV contact discharge and $\pm 10$ kV air discharge ESD testing (Criterion C)	
Certification	Compliant with Telec, VCCI, CE, FCC, IC and NCC (Up to LTE module chip certification, CONEXIO BlackBear can apply for TELEC, VCCI, NCC with SIM7600JC-H PCIE, but other Certification is related to the acquisition of regional LTE module.)	
Mechanical	Dimensions	171.9mm x 114.0mm x 42.5 mm (w/o Flange)
	Weight	600g $\pm$ 20g (Estimated)
Accessories (Common)	Serial Cable	Cable with connector
	DI/DO Cable	Cable with connector
	CAN Cable	Cable with connector
	Holeless EX-Door	
Accessories (Indoor set)	LTE Antenna	SMA connector for external
	GNSS Antenna	Fakra connector for external
	WiFi/Bluetooth Antenna	SMA (Reverse Type) connector for external
	AC Adapter	AC Adapter & Conversion cable
Accessories (Commercial set)	4 in 1 Antenna/Cable	LTE x 2, GNSS, WiFi Antenna & Cable
	Power Cable	4 pins power cable

CONEXIOBlackBear には屋内用のIndoorセットと車載用のCommercialセットが有ります。この2種類のセットの違いは付属品だけです。(「表2.2-1 仕様一覧表 Accessories」の項を参照してください)

また対象となる国・地域別に”Jモデル”、”Aモデル”、”Eモデル”が有ります。LTE Moduleの対応周波数、WiFiの対応channel以外のCONEXIOBlackBear の構成は各国・地域対応モデル間で同一で、開発したシステムは各モデル間共通で使用することができます。

それぞれのモデル別型番を「表2.2-2 モデル別型番」に記載します。

表2.2-2 モデル別型番

モデル	対象国・地域	屋内用/車載用	型番
Jモデル	日本国内、台湾向け	屋内用	MS-5776-JC-H Indoor
		車載用	MS-5776-JC-H Commercial
Aモデル	米国、カナダ向け	屋内用	MS-5776-A-H Indoor
		車載用	MS-5776-A-H Commercial
Eモデル	欧州、東南アジア諸国向け(※)	屋内用	MS-5776-E-H Indoor
		車載用	MS-5776-E-H Commercial

(※) 対応国・地域は、下記URLの「対応国・予定国について」を参照してください。

[https://conexio-iot.jp/serviceproduct/blackbear\\_faq.html](https://conexio-iot.jp/serviceproduct/blackbear_faq.html)

本書では日本国内および台湾向けの”Jモデル”を念頭に置いた記載をしておりますが、上記のように各モデル間で動作するシステムは共通です。

各モデルは対象となるそれぞれの国・地域の認証を取得してありますので、必ず使用する国・地域での認証を取得したモデルを使用してください。(「8.5. 海外での使用について」も参照してください)

## 2.3. ブロック図

CONEXIOBlackBear のブロック図を「図2.3-1 ブロック図」に記載します。

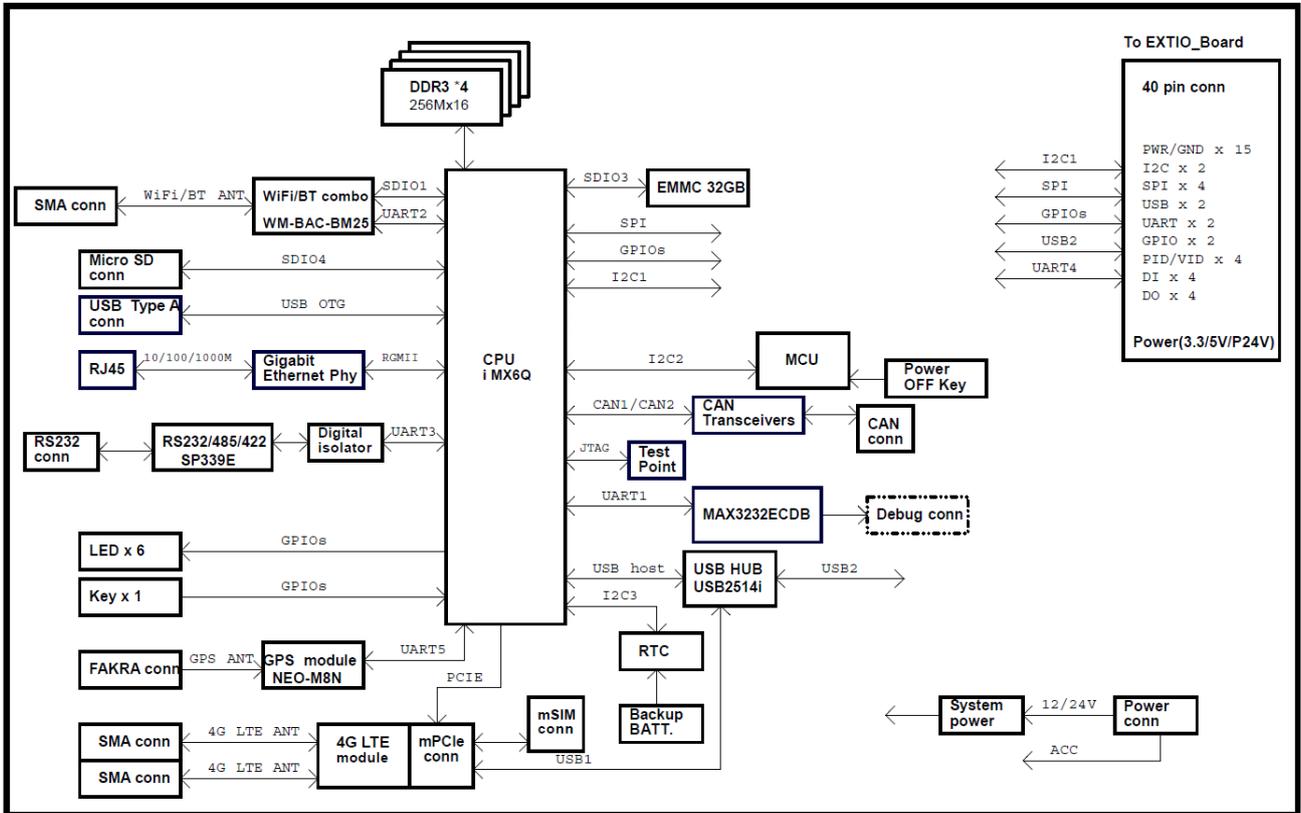


図2.3-1 ブロック図

## 2.4. インターフェースレイアウト図

CONEXIO BlackBear のインターフェースレイアウトを「図2.4-1 インターフェースレイアウト図」に記載します。

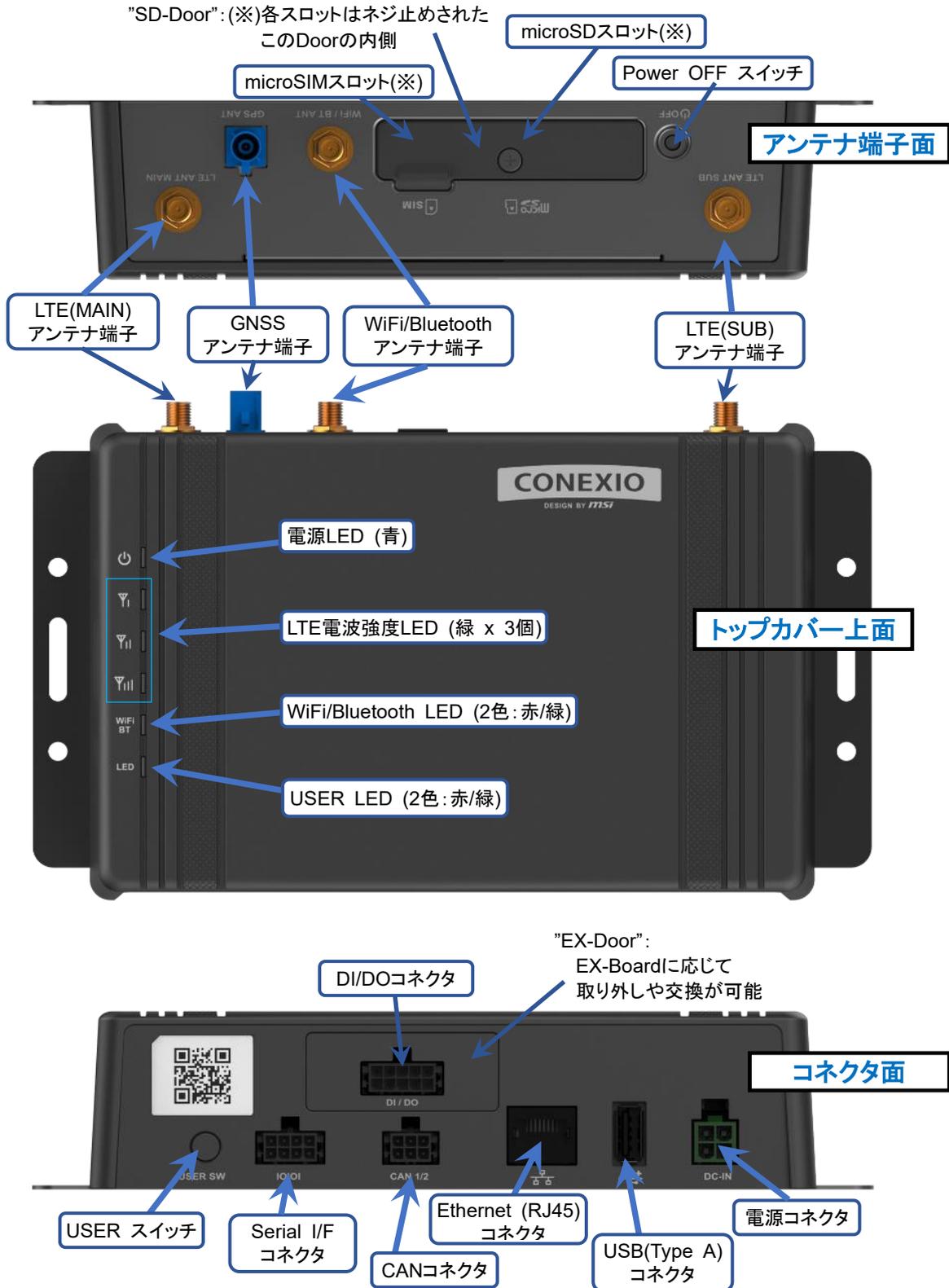


図2.4-1 インターフェースレイアウト図

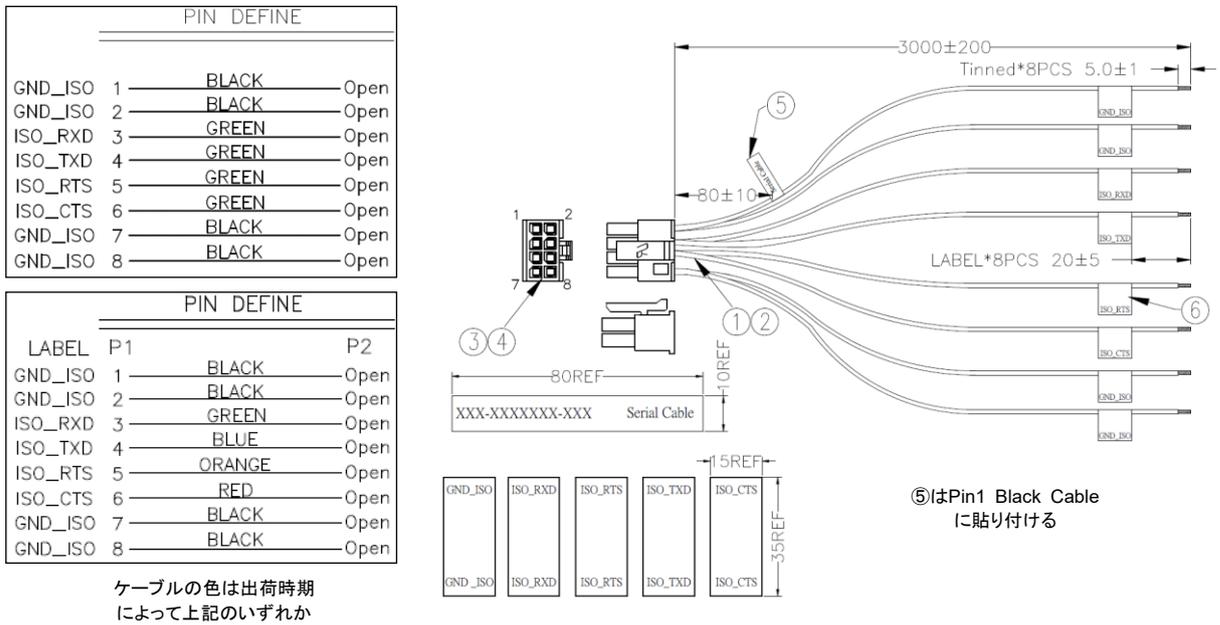
## 2.5. 付属品

CONEXIOBlackBear には”屋内用セット (Indoor set)”と”車載用セット (Commercial set)”の2種類のセットがあり、一部の付属品が異なります。CONEXIOBlackBear の付属品を、2種類のセットに共通して入っている付属品およびそれぞれのセットのみに入っている付属品に分けて以下に記載します。

なお”Serial Cable”と”DI/DO Cable”のケーブルの色は、出荷時期によって2種類のものが有ります。これらのケーブルについては、2種類のケーブルの色を併記してありますので、ご注意ください。

### 2.5.1. 屋内用セット・車載用セット共通

“屋内用セット”および”車載用セット”共通に添付される付属品は”Serial Cable”、”DI/DO Cable”、”CAN Cable”、”Holeles EX-Door”の4品です。図2.5-1 ~ 図2.5-4 にこれらの付属品の詳細を記載します。



ピン番号	RS-232C		RS-485 Half Duplex		RS-485/RS-422 Full Duplex	
	信号名	I/O	信号名	I/O	信号名	I/O
1	GND ISO	GND	GND ISO	GND	GND ISO	GND
2	GND ISO	GND	GND ISO	GND	GND ISO	GND
3	ISO_RXD	IN	DATA+	IN/OUT	TX+	OUT
4	ISO_TXD	OUT			RX+	IN
5	ISO_RTS	IN	DATA-	IN/OUT	TX-	OUT
6	ISO_CTS	OUT			RX-	IN
7	GND ISO	GND	GND ISO	GND	GND ISO	GND
8	GND ISO	GND	GND ISO	GND	GND ISO	GND

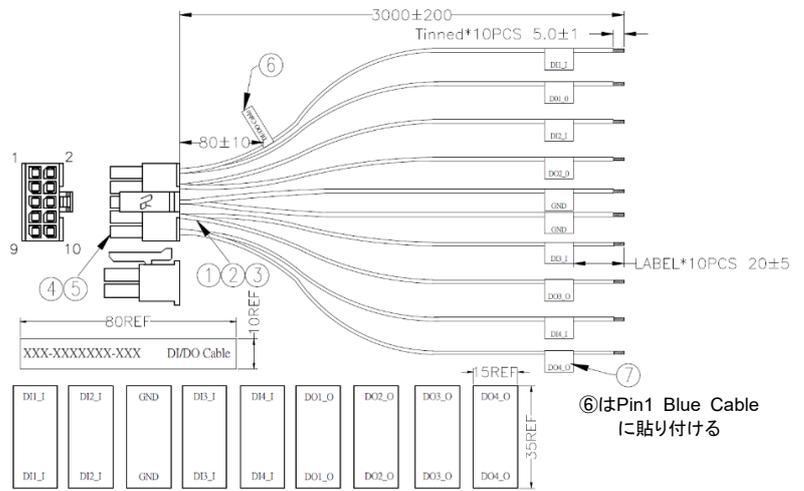
※空欄とGND\_ISOで使用しないピンは未接続としてください。

①	UL1007 20AWG(21/0.178)TIN PLATED COPPER WIRE BLACK INSULATION(-40°C~+85°C) Ø1.8mm L=3010mm (RoHS)	4pcs
②	UL1007 20AWG(21/0.178)TIN PLATED COPPER WIRE GREEN INSULATION(-40°C~+85°C) Ø1.8mm L=3010mm (RoHS)	4pcs
③	HOUSING:(WSTP/N:PD08-I30002) 2*4P/M PH:3.0mm PA66 BLACK (RoHS)	1pcs
④	TERMINAL: (WSTP/N:I30002PS-2) PH:3.0mm PHOSPHOR BRONZE TIN-PLATED (RoHS)	8pcs
⑤	テドロン防水ラベル 80*10mm Silver地Black文字 フォントTimes New Roman (RoHS)	1pcs
⑥	テドロン防水ラベル 35*15mm Silver地Black文字 フォントTimes New Roman (RoHS)	8pcs

図2.5-1 Serial Cable

PIN DEFINE		
DI1_I	1	BLUE Open
DO1_O	2	ORANGE Open
DI2_I	3	BLUE Open
DO2_O	4	ORANGE Open
GND	5	BLACK Open
GND	6	BLACK Open
DI3_I	7	BLUE Open
DO3_O	8	ORANGE Open
DI4_I	9	BLUE Open
DO4_O	10	ORANGE Open

PIN DEFINE		
LABEL	P1	P2
DI1_I	1	BLUE Open
DO1_O	2	ORANGE Open
DI2_I	3	RED Open
DO2_O	4	GREEN Open
GND	5	BLACK Open
GND	6	BLACK Open
DI3_I	7	WHITE Open
DO3_O	8	YELLOW Open
DI4_I	9	GRAY Open
DO4_O	10	BROWN Open

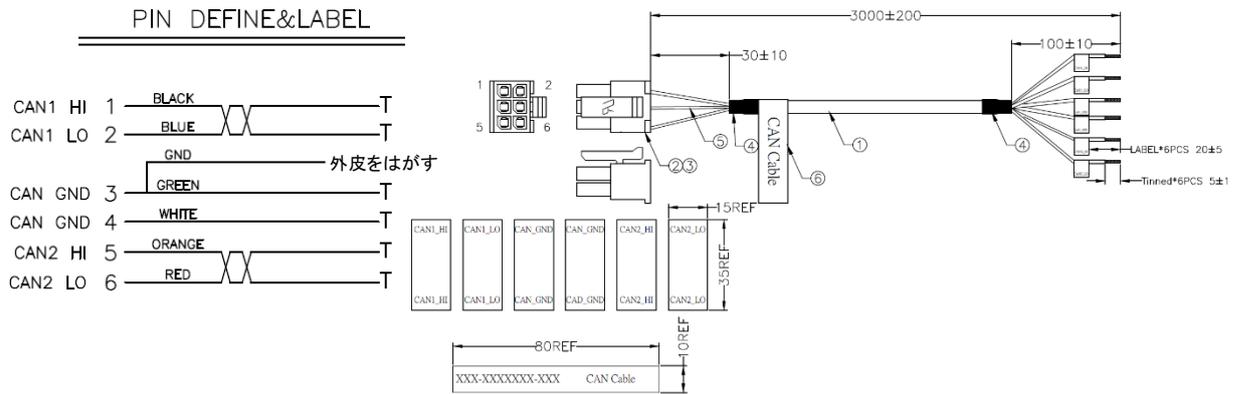


ケーブルの色は出荷時期  
によって上記のいずれか

ピン番号	信号名	I/O
1	DI1_I	IN
2	DO1_O	OUT
3	DI2_I	IN
4	DO2_O	OUT
5	GND	GND
6	GND	GND
7	DI3_I	IN
8	DO3_O	OUT
9	DI4_I	IN
10	DO4_O	OUT

①	UL1007 20AWG(21/0.178)TIN PLATED COPPER WIRE BLACK INSULATION(-40°C~+85°C) Ø1.8mm L=3005mm(RoHS)	2pcs
②	UL1007 20AWG(21/0.178)TIN PLATED COPPER WIRE BLUE INSULATION(-40°C~+85°C) Ø1.8mm L=3005mm(RoHS)	4pcs
③	UL1007 20AWG(21/0.178)TIN PLATED COPPER WIRE ORANGE INSULATION(-40°C~+85°C) Ø1.8mm L=3005mm(RoHS)	4pcs
④	HOUSING:(WSTP/N:PD10-I30002) 2*5P/M PH:3.0mm PA66 BLACK (RoHS)	1pcs
⑤	TERMINAL: (WSTP/N:I30002PS-2) PH:3.0mm PHOSPHOR BRONZE TIN-PLATED (RoHS)	10pcs
⑥	テドロン防水ラベル 80*10mm Silver地Black文字 フォントTimes New Roman (RoHS)	1pcs
⑦	テドロン防水ラベル 35*15mm Silver地Black文字 フォントTimes New Roman (RoHS)	10pcs

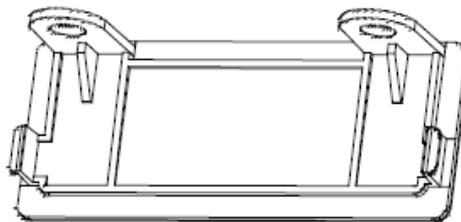
図2.5-2 DI/DO Cable



ピン番号	信号名	I/O
1	CAN1_HI	IN/OUT
2	CAN1_LO	IN/OUT
3	CAN_GND	GND
4	CAN_GND	GND
5	CAN2_HI	IN/OUT
6	CAN2_LO	IN/OUT

①	UL2725(#24*1P+AL)*4P+E+AL BLACK PVC OD:6.3mm LENGTH:3010MM	1pcs
②	HOUSING:(WST P/N:PD06-I30002) 2*3P/M PH:3.0mm PA66 BLACK (RoHS)	1pcs
③	TERMINAL: (WST P/N:I30002PS-2) PH:3.0mm PHOSPHOR BRONZE TIN-PLATED (RoHS)	6pcs
④	H.S.T:Ø7.0*20*0.15mm BLACK (RoHS)	2pcs
⑤	H.S.T:Ø1.0*45*0.15mm BLACK (RoHS)	1pcs
⑥	テドロン防水ラベル 80*10mm Silver地Black文字 フォントTimes New Roman (RoHS)	6pcs
⑦	テドロン防水ラベル 35*15mm Silver地Black文字 フォントTimes New Roman (RoHS)	6pcs

図2.5-3 CAN Cable

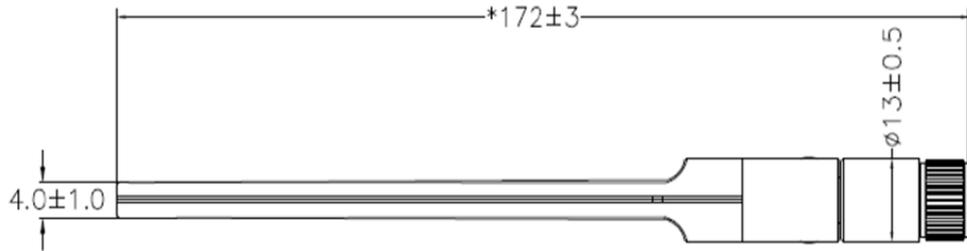


※標準では本体に”DI/DOコネクタ”用の穴が開いた”EX-Door”が取り付けられています。  
 “Holeless EX-Door”はコネクタ用の穴が開いていない”EX-Door”で、EX-Boardを取り外した時などに、標準で取り付けられているEX-Door”と交換して本体に取り付けることができます。  
 ※取り付けに際しては「8.6. 保証事項」にご留意いただき、必要に応じて「9. お問い合わせ先」に相談してください。

図2.5-4 Holeless EX-Door

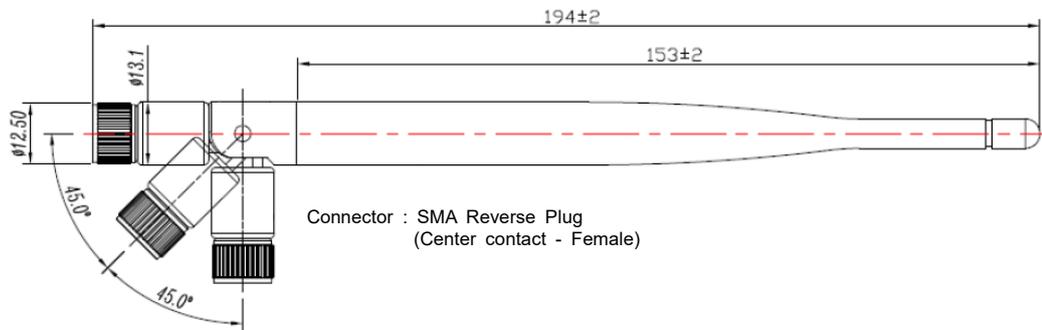
### 2.5.2. 屋内用セットのみ

“屋内用セット”のみに添付される付属品は”LTE Antenna” 2本、” WiFi/Bluetooth Antenna”、”GNSS Antenna”、”AC Adapter”、”AC adapter Conversion Cable”の5品です。図2.5-5 ~ 図2.5-9 にこれらの付属品の詳細を記載します。



Connector : SMA Normal Plug  
(Center contact - Male)

図2.5-5 LTE Antenna



Connector : SMA Reverse Plug  
(Center contact - Female)

図2.5-6 WiFi/Bluetooth Antenna

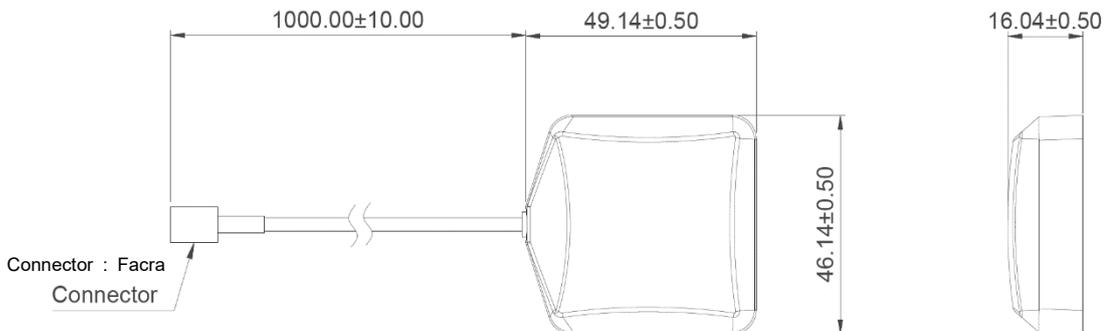
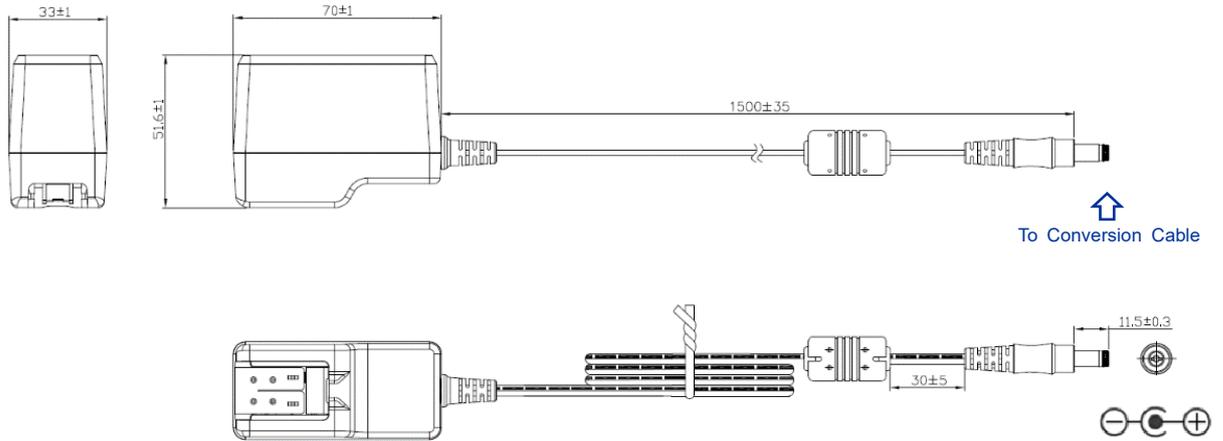
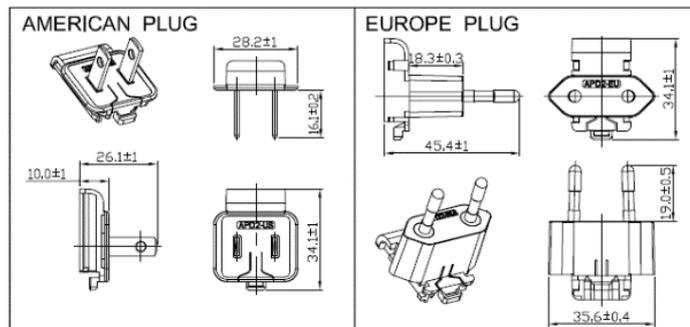


図2.5-7 GNSS Antenna



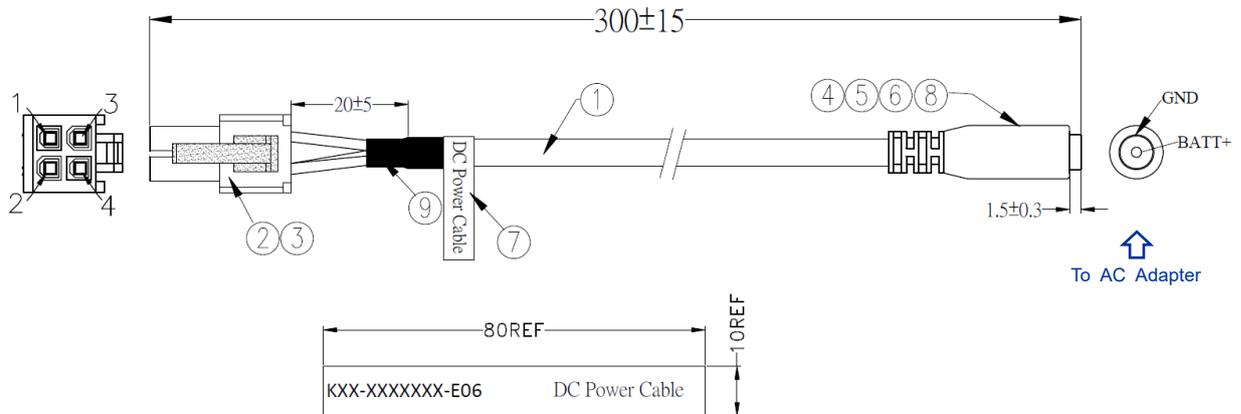
↑  
To Conversion Cable



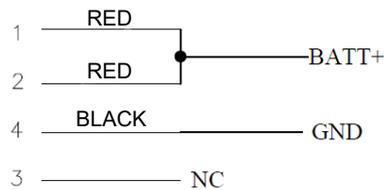
GENERAL Notes	
INPUT	AC100-240V, 50-60Hz
OUTPUT	DC12V / 2A
COLOR	BLACK
AC INPUT PLUG	NEMA 1-15P, 2PINS AMERICA PLUG EN50075, 2PINS EUROPE PLUG
DC POWER CORD	UL2468 #24AWG x 2C
DC POWER PLUG	φ5.5 x φ2.1 x L11.5mm TUNIGH FORK TYPE
CABLE TIE	CABLE TIE (WITH-STEEL)
CORE	RH-φ10.5 x φ5.5 X 20mm OR EQUIVALENT
TOTAL WEIGHT	ABOUT 135g

※PLUG部分を交換して各国・地域に対応するタイプのAC Adapterです。

図2.5-8 AC Adapter



### PIN ASSIGNMENT



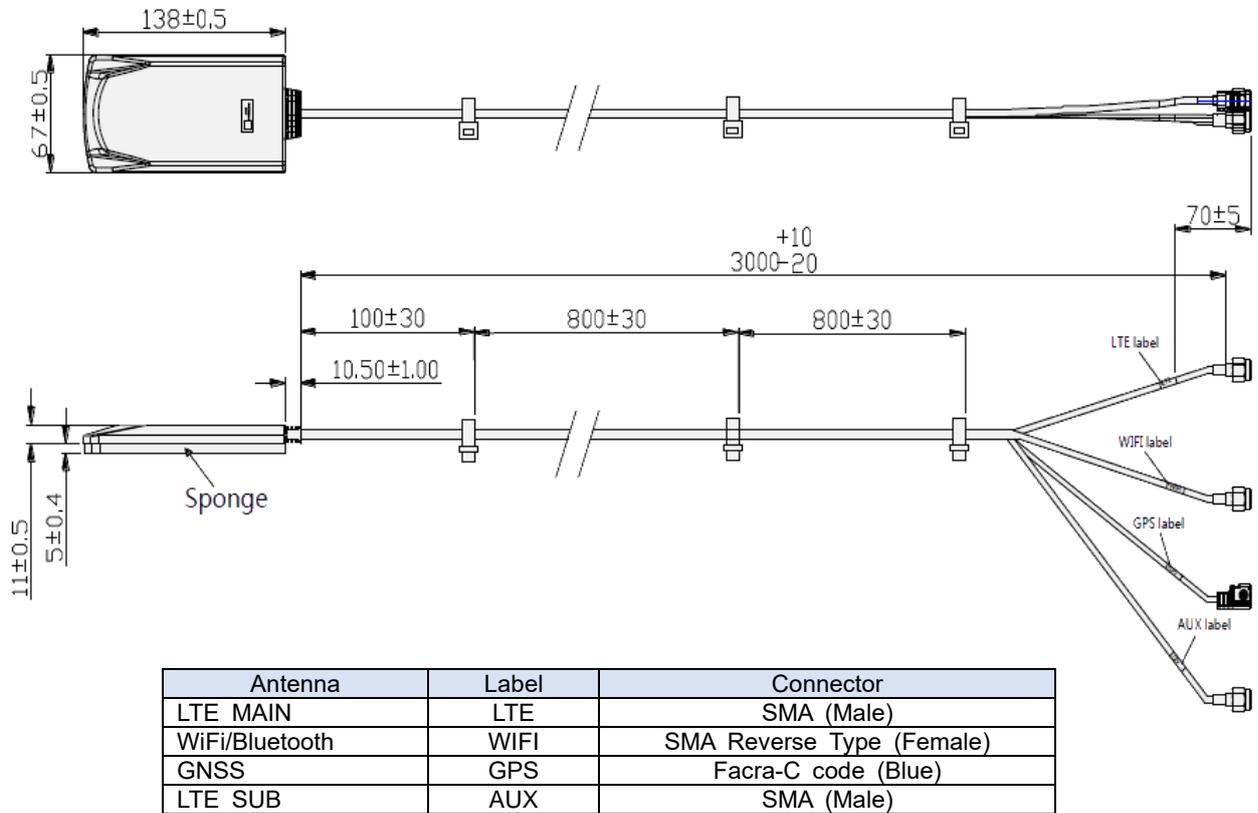
①	UL2464 18AWG*3C(赤*2 / 黒*1) PVC 黒色(-40°C~105°C)(RoHS) L=270±5MM	1pcs
②	HSG:PH4.2mm 2*2P/M NYLON66 透明(WST P/N:P4-I42002)(RoHS) または同等品	1pcs
③	TER:PH:4.2mm適用18#~24#線材,真鍮錫メッキメス端子, (WST P/N:I42002BS-2)(RoHS) または同等品	4pcs
④	DC JACK:ID:5.5*2.1mm OD:7.0mm L=17.5mm ABS BLACK INSULATOR BRASS NICKEL PLATING (RoHS)	1pcs
⑤	INNER MOLD:LOW DENSITY PE CLEAR (RoHS)	AR
⑥	OVER MOLD:PVC 45P BLACK(-40°C~105°C) (RoHS)	AR
⑦	ドローン防水ラベル 80*10mm Silver地Black文字 フォントTimes New Roman (RoHS)	1pcs
⑧	黒熱収縮スリーブ:Φ1.5*0.13(T)mm*6mm EVA 赤燐非含有 (RoHS)	1pcs
⑨	黒熱収縮スリーブ:Φ5.0*0.13(T)mm*20mm EVA 赤燐非含有 (RoHS)	1pcs

※この変換ケーブルを使用してAC Adapterを本体の電源コネクタに接続します。

図2.5-9 AC Adapter Conversion Cable

### 2.5.3. 車載用セットのみ

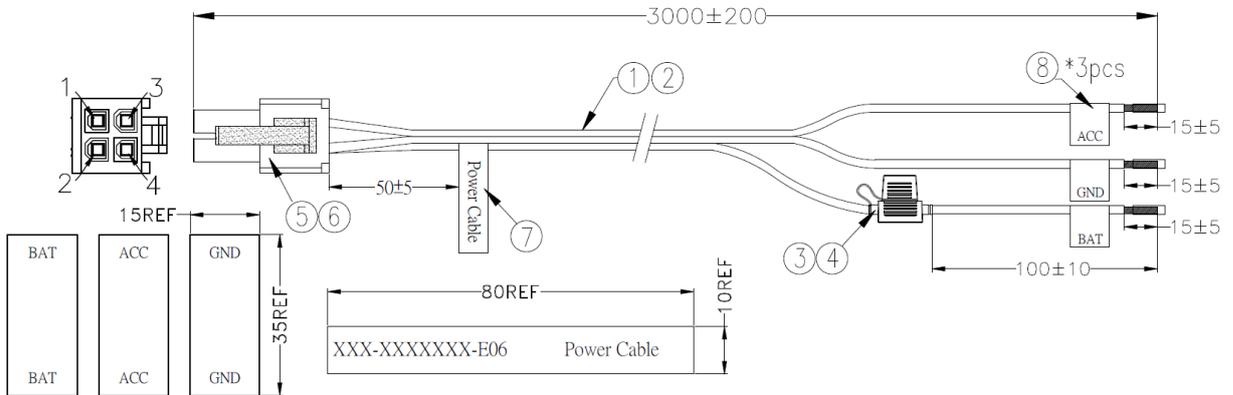
“車載用セット”のみに添付される付属品は“4 in 1 Antenna/Cable”、“Power Cable”の2品です。図2.5-10および 図2.5-11 にこれらの付属品の詳細を記載します。



アンテナ部底面には粘着面の有る5mm厚の発泡ゴムが貼り付けられています。特に金属面に取り付ける場合は、アンテナ特性を確保するため、この発泡ゴムをはがさないでください。粘着面にマグネットシートを貼り付けて、スチール面へ取り付けることができます。



図2.5-10 4 in 1 Antenna/Cable



## PIN ASSIGNMENT

ACC	1	RED	Half Peeling
BAT	2	YELLOW	Fuse Half Peeling
GND	4	BLACK	Half Peeling
	3	NC	

①	電子線:UL1007 18AWG(34/0.178), 錫メッキ銅 OD:2.1mm, PVC絶縁(-40°C~105°C) BLACK (RoHS) L=2990±5mm	1pcs
②	電子線:UL1007 18AWG(34/0.178), 錫メッキ銅 OD:2.1mm, PVC絶縁(-40°C~105°C) RED (RoHS) L=2990±5mm	1pcs
③	ヒューズホルダー: BLACK+電子線 (AVS0.8 黄色L1=2870mm / L2=100mm) 全て(RoHS)	1pcs
④	ブレードヒューズ: 32V 5A 黄褐色 (P/N:MIN-N5) (RoHS)	1pcs
⑤	HSG:PH4.2mm 2*2P/M NYLON66 透明色 (WST P/N:P4-I42002) (RoHS) または同等品	1pcs
⑥	TER:PH:4.2mm適用18#~24#線材, 真鍮錫メッキ メス端子, (WST P/N:I42002BS-2) (RoHS) または同等品	4pcs
⑦	ドローン防水ラベル 80*10mm Silver地Black文字 フォントTimes New Roman (RoHS)	1pcs
⑧	ドローン防水ラベル 35*15mm Silver地Black文字 フォントTimes New Roman (RoHS)	3pcs

図2.5-11 Power Cable

## 2.6. OS および 主な pre-installed ソフトウェア

CONEXIO BlackBear に搭載されているOSおよび主なpre-installedソフトウェアを「表2.6-1 OS および 主な Pre-installedソフトウェア」に記載します。

主なユーティリティツールとしてはbusyboxを使用しています。オープンソースソフトウェア(OSS)の使用においては「8.4. オープンソースソフトウェアに関する注意事項」に記載した事項に留意してください。

表2.6-1 OS および 主なPre-installedソフトウェア

No.	Category	Name	Version
1	OS	Linux Yocto	Kernel : 4.9.88 Yocto : rocko (2.4.2)
2	Driver	FTDI Serial driver	-
3	Package	deb	-
4	Package Manager	apt	1.2.24
5	Init	systemd	234
6	Utils	sudo	1.8.20p2
7		locale	2.26
8		sed	4.2.2
9		less	487
10		expect	5.45
11		minicom	2.7.1
12		screen	4.6.1
13		vim-tiny	8.0.983
14		dosfstools	4.1
15		util-linux	2.3
16		rsyslog	8.29
17		bash-completion	2.7-r0
18		man	1.6g-r1
19		dialog	1.3-20160828-r0
20	cpufrequtils	008-r5	
21	Device	usbutils	008-r0
22		i2c-tools	3.1.2-r0
23		evtest	1.33+AUTOINC+ab140a2dab-r0
24		bluez	5.49-r0
25	canutils	4.0.6	
26	Network	iptables	1.6.1
27		networkmanager	1.10.6
28		modemmanager	1.6.4
29		net-tools	1.60-26
30		avahi-daemon	0.6.32
31		avahi-autoipd	0.6.32
32		ntpd	4.2.8
33		lighttpd	1.4.45
34		curl	7.54.1
35		wget	1.19.1
36		traceroute	2.1.0
37		iputils-ping	20151218
38		iputils-arping	20151218
39		dnsutils	9.10.5
40	mosquitto-clients	1.4.14	
41	openssh-client	7.5p1	
42	openssl	1.0.2n	
43	Development	bundler	1.6.2
44		lua	5.3.4
45		gcc	7.3
46		g++	7.3.0
47		python	2.7.13-r1
48		python3	3.5.3
49		python-pip	9.0.1-r0
50		perl	5.24.1
51		google-glog	0.3.4
52		sodium	1.0.11
53		zmq3(ZeroMQ version3)	4
54		googlegflags	2.2.0
55		opencv	4.0.1

### 3. 本体使用方法（準備/起動/終了）

本章では、CONEXIO BlackBear 本体の起動前の準備から電源の投入、そしてLinuxシステムを終了させて電源を落とすまでの基本的な操作について記載します。

#### 3.1. microSIM および microSD 挿入

「図3.1-1 microSIM および microSDの挿入」に示した”SD-Door”を固定してあるネジを外し、必要に応じてmicroSIMおよびmicroSDを挿入します。挿入する際には”SD-Door”上部に印刷したマークを参照して、挿入スロットや挿入方向を間違えないように注意してください。特にmicroSIMの抜き差しは、microSIMの破損や誤動作を避けるために必ず電源を落とした状態で行ってください。

確実に挿入されたことを確認して、”SD-Door”をネジ止めして固定します。

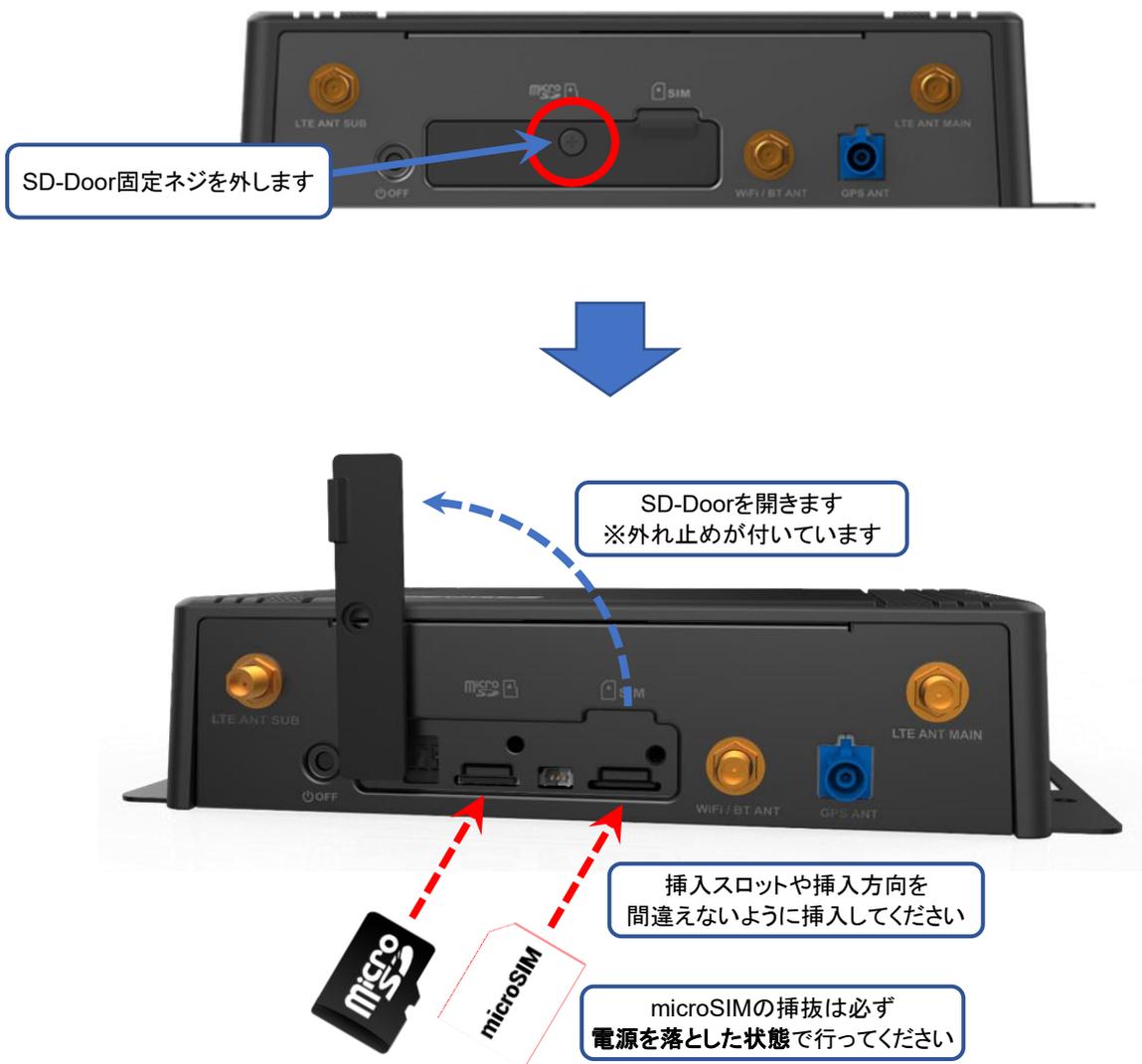


図3.1-1 microSIM および microSDの挿入

### 3.2. アンテナ装着

「図2.4-1 インターフェースレイアウト図」に示した“アンテナ端子面”の各アンテナ端子に、必要に応じてLTE、WiFi/Bluetooth、GNSSアンテナを装着します。

LTEアンテナ端子とWiFi/Bluetoothアンテナ端子は両方ともSMA端子ですが、WiFi/BluetoothアンテナのSMA端子はリバースタイプなので、取り付け位置を間違えないように注意してください。GNSSアンテナ端子(筐体の印刷表記は“GPS ANT”)はFacra端子です。

屋内用セットにおいて、LTEアンテナを2本ではなく1本だけ装着する場合は“LTE ANT MAIN”と印刷されている側のアンテナ端子に装着してください。ただし1本だけしか装着しない場合は特性が確保できない可能性が高いため推奨いたしません。

車載用4in1アンテナの接続は、ケーブルに貼ってあるラベルを参照して接続してください。  
(“LTE SUBアンテナ”のケーブルは“AUX”と記載されているケーブルです)

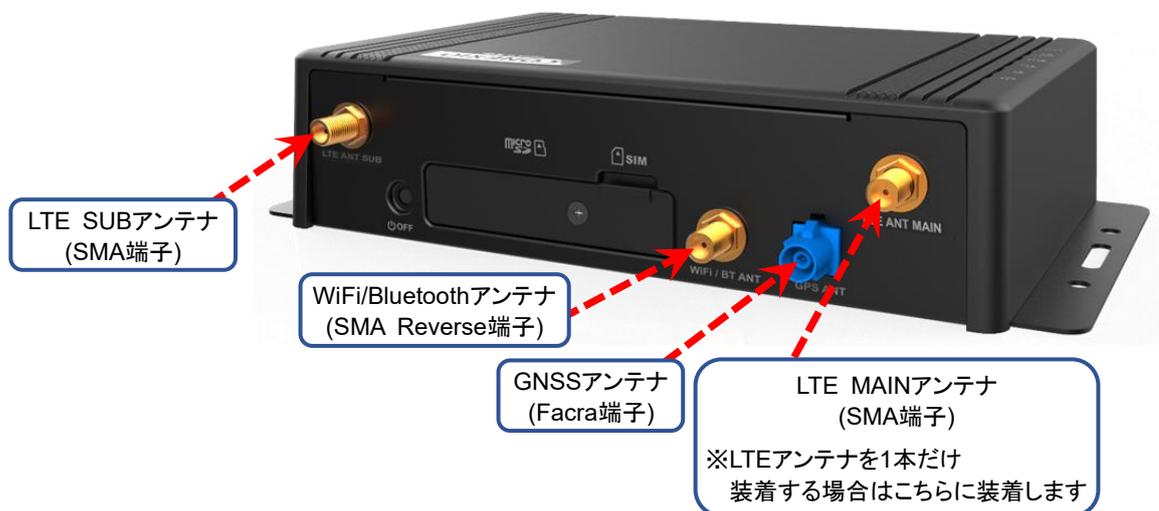


図3.2-1 アンテナ装着

### 3.3. ケーブル接続

「図2.4-1 インターフェースレイアウト図」に示した“コネクタ面”の各コネクタに、必要に応じてEthernetケーブル、Serialケーブル、DI/DOケーブル、CANケーブル、USBケーブルを接続します。

Serialケーブル、DI/DOケーブル、CANケーブルの端子には抜け止めが付いています。奥までしっかりと差し込んで抜け止めが本体側コネクタの爪と噛み合っていることを確認してください。

屋内用セットのACアダプタは、変換ケーブルを介して CONEXIOBlackBear に接続します。ACアダプタの変換ケーブルも接続してかまいませんが、「3.4. 起動 (電源ON/ログイン/時刻設定/RTC)」に記載したように、電源を投入すると CONEXIOBlackBear が起動します。本当に起動したい時までは誤って電源を投入しない(ACアダプタを電源に接続して通電しない)ように注意してください。

車載用セットの車載用電源ケーブルは、黄色の線をバッテリーのプラス端子へ、黒色の線をバッテリーのマイナス端子へ、赤色の線をACCまたはIG電源端子へ接続してください。黄色の線の途中には5Aのヒューズが取り付けられています。

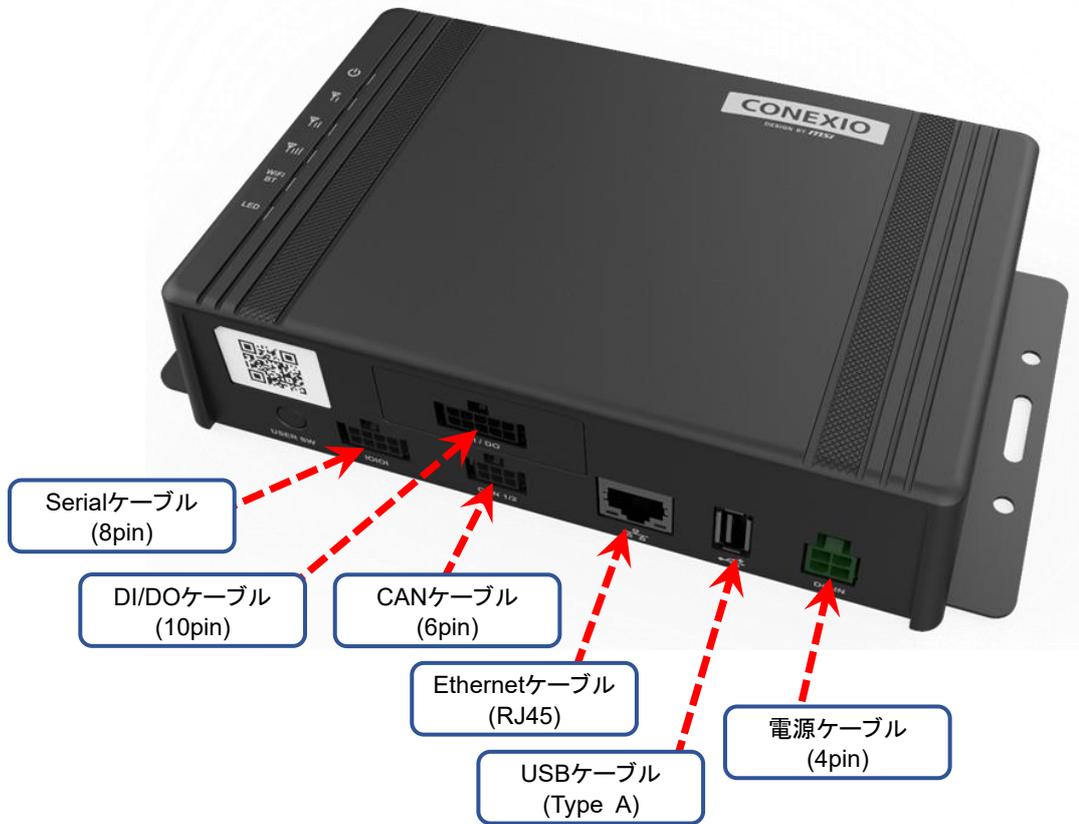


図3.3-1 ケーブル接続

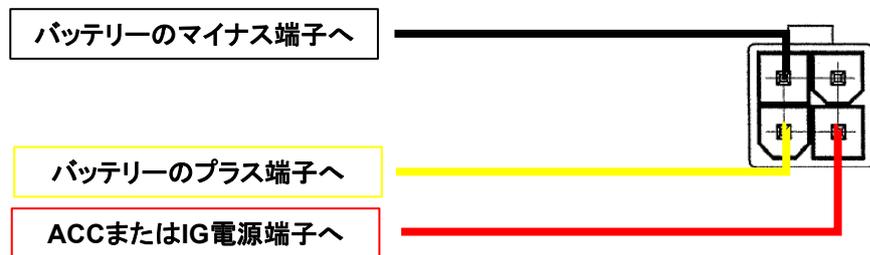


図3.3-2 車載用電源ケーブル接続

### 3.4. 起動（電源 ON/ログイン/時刻設定/RTC）

#### 3.4.1. 電源 ON

表3.4-1 電源投入 - 起動 実行方法

対象セット	電源投入方法
屋内用セット(ACアダプタ変換ケーブル使用時)	ACアダプタを接続して通電する
車載用セット(車載用電源ケーブル使用時)	ACC/IG電源をONにする

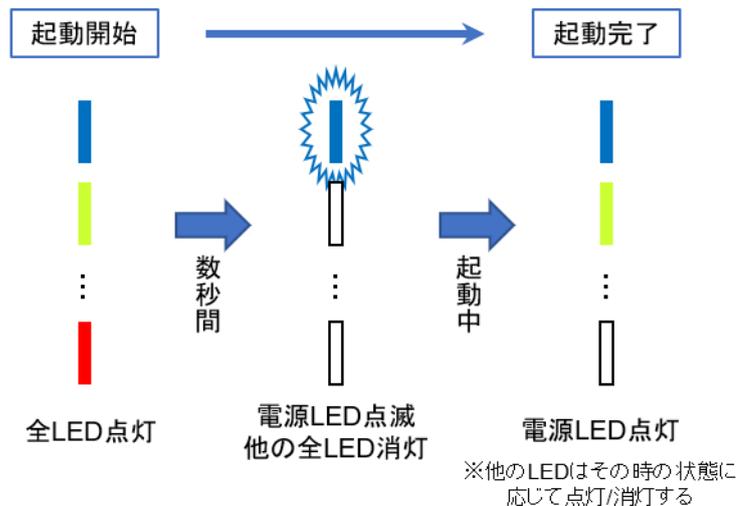


図3.4-1 起動時のLED表示

電源を投入するだけで CONEXIOBlackBear が自動的に起動して、システムおよびインストールされているアプリケーションが立ち上がります。

屋内用セット(ACアダプタ変換ケーブルを使用している場合)の電源投入はACアダプタを接続して通電します。  
 車載用セット(車載用電源ケーブルを使用している場合)の電源投入は、車載用電源ケーブルをバッテリー等の電源に接続した後、ACC/IG電源をONにします。

CONEXIOBlackBear に電源が投入されると、「図2.4-1 インターフェースレイアウト図」に示した”トップカバー上面”のLEDが全て点灯した後、電源LEDが点滅を始めて電源LED以外のLEDは数秒後に全て消灯します。

CONEXIOBlackBear の起動処理中は電源LEDの点滅が継続し、起動処理が完了すると電源LEDは点灯状態に移行します。電源LEDが点灯状態になればCONEXIOBlackBear を使い始めることができます。

すでに設定されている情報やインストール済みのアプリケーションなどを使用してそのまま運用する場合は、上記のように電源を投入するだけで使用可能になります。

### 3.4.2. ログイン

システムを起動した後、ログインする場合に使用するユーザー名の初期値は"root"(superuser権限を持つユーザー)です。工場出荷後に初めてログインする場合には、新しいパスワードを設定するように求められますので、画面の指示に従って適切なパスワードを設定してください。

セキュリティを確保するため、適切なパスワードの管理やユーザー権限設定の実施をしていただきますよう、くれぐれも注意してください。

### 3.4.3. 時刻設定/RTC

CONEXIOBlackBear のRTCは内蔵電池でバックアップされており、設定した時刻は電源をOFFにしても保持されます。内蔵電池はCR1220で、筐体内部の電池ホルダーに装着されており交換することができます。

設定されている時刻やタイムゾーンはtimedatectlコマンドで確認することができます。

例) timedatectlコマンド実行例 ("timedatectl status"でも同じ結果が得られます)

```
# timedatectl
   Local time: Thu 2019-07-11 07:56:56 UTC
   Universal time: Thu 2019-07-11 07:56:56 UTC
   RTC time: Thu 2019-07-11 07:56:56
   Time zone: Universal (UTC, +0000)
   Network time on: yes
   NTP synchronized: yes
   RTC in local TZ: no
```

※行頭の"#"はその行がLinuxの入力コマンドであることを示すために、Linuxのコマンドプロンプトを表しています。(この章以降も本書ではこの表記を使用します)

CONEXIOBlackBear のDefault設定では、Local timeはUTCとなっています。UTCからタイムゾーンを変更する場合にも下記のようにtimedatectlコマンドを使用します。

例) 使用できるタイムゾーンを確認した後、JSTに設定する場合

```
# timedatectl list-timezones          使用できるタイムゾーンを確認
...
# timedatectl set-timezone Asia/Tokyo  タイムゾーンをJSTに設定
# timedatectl
   Local time: Thu 2019-07-11 17:10:11 JST
   Universal time: Thu 2019-07-11 08:10:11 UTC
   RTC time: Thu 2019-07-11 08:10:11
   Time zone: Asia/Tokyo (JST, +0900)
   Network time on: yes
   NTP synchronized: yes
   RTC in local TZ: no
```

CONEXIOBlackBear がネットワークに接続されている場合、"Network time on"が"yes"に設定されていれば、ネットワーク上のNTP Serverに対して時刻同期を実行して時刻が設定されます。時刻同期の実行が完了すると"NTP synchronized"が"yes"になります。

※"Network time on"は、ネットワーク上のNTP Serverを使用して時刻同期を実行するか否かを表します。

CONEXIOBlackBear のdefaultは"yes"です。

※"NTP synchronized"は、NTP Serverを使用して設定された時刻か否かを表します。

※NTP Serverやネットワークの状態によっては、時刻同期に数十秒かかる場合があります。

CONEXIOBlackBear が使用するdefaultのNTP Serverは、“time.google.com”です(time1.google.com～time4.google.com が使用されます)。これを変更したい場合は、“/etc/systemd/timesyncd.conf”に使用するNTP Serverを設定してください。(NTPがメイン、FallbackNTPが予備です。予備として複数のNTP Serverを設定する場合は半角スペースで区切ってください)

NTP Serverの設定を有効にするためには、CONEXIOBlackBear を再起動するか、以下のコマンドで時刻同期デーモンを再起動してください。

```
# systemctl restart systemd-timesyncd
```

NTP Serverを使用せずに手動で時刻を設定することも可能です。ただし“Network time on”および“NTP synchronized”が両方とも“yes”になっている場合は手動での時刻設定はできません。この場合は、以下のように“timedatectl set-ntp”コマンドを実行して“Network time on”を“no”にしてから、“date”コマンドや“timedatectl set-time”コマンドで時刻を設定してください。

例) NTP Serverを使用せずに手動で2019年10月25日 13時39分00秒に設定する場合

```
# timedatectl set-ntp no          (パラメータの“no”は、“0”または“false”でも可能です)
# date -s "2019-10-25 13:39:00"
```

手動で時刻設定ができる状態から、NTP Serverを使用して時刻設定を実行する状態へと設定を変更する場合は、以下の“timedatectl set-ntp”コマンドを実行して、“Network time on”を“yes”にしてください。

```
# timedatectl set-ntp yes        (パラメータの“yes”は、“1”または“true”でも可能です)
```

時刻を設定/補正した後、必要に応じて下記のようにhwclockコマンドを使用してRTCへ時刻を設定してください。

例) システムクロックのローカルタイムをRTCのUTCとして設定する場合

```
# hwclock -w -u
```

※RTCがUTCではなく、JST等のローカルタイムで設定されている場合、予期せぬLinuxカーネルのバグを引き起こす可能性が有ることが広くアナウンスされています。RTCはUTCで設定することを強く推奨します。

逆にhwclockコマンドを使用してRTCのUTCをシステムクロックのローカルタイムとして設定する場合は下記の通りです。

例) RTCのUTCをシステムクロックのローカルタイムとして設定する場合。

```
# hwclock -s
```

### 3.5. Suspend – Resume

#### 3.5.1. Suspend

表3.5-1 Suspend 実行方法

Suspend手段	操作方法
Linuxコマンド	“go-sys-suspend”コマンドを実行する

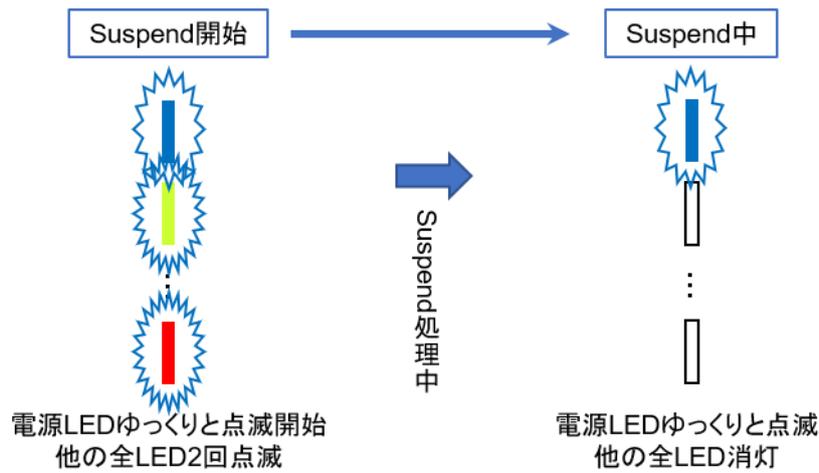


図3.5-1 Suspend時のLED表示

CONEXIOBlackBear は、“go-sys-suspend”コマンド(CONEXIOBlackBear 固有のコマンドです)を実行することにより“Suspend Mode”に入ります。

“Suspend Mode”に入ると、CONEXIOBlackBear での処理状態をメモリ上に保持したまま、MCU(電源管理プロセッサ)以外のプロセッサや周辺I/Oデバイスなどは低消費電力の状態に移行します。

“Suspend Mode”開始時にはトップカバー上面の電源LED以外の全てのLEDが2回点滅し“Suspend Mode”中は電源LEDがゆっくりと点滅を続けます。

### 3.5.2. Resume

表3.5-2 Resume 実行方法

Resume手段	操作方法
USER SW	USER SWを押す
SMS	LTE経由でSMSを受信する
RTC alarm	rtcwakeコマンド等でResumeのタイミングを指定する

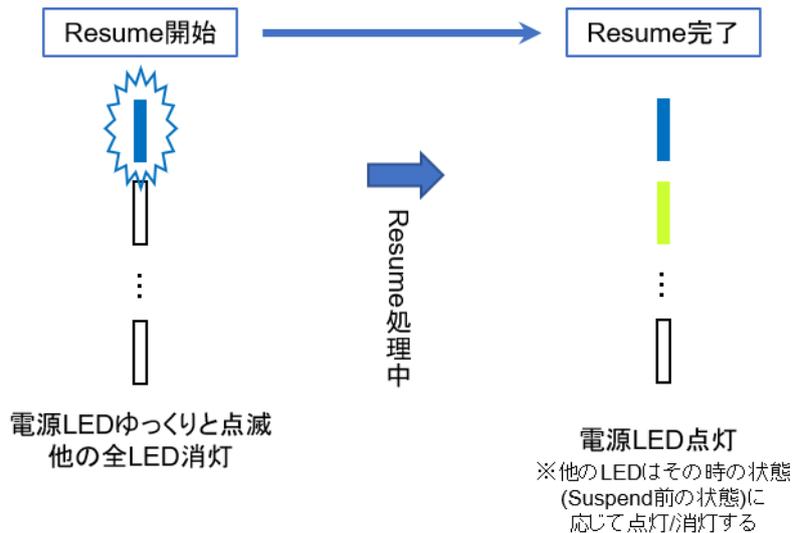


図3.5-2 Resume時のLED表示

“Suspend Mode”からResumeする方法には、以下の3種類があります。

- ・「図2.4-1 インターフェースレイアウト図」に示した“USER スイッチ(USER SW)”を押す。
- ・LTE経由でSMSを受信する。
- ・rtcwakeコマンド等でResumeするタイミングを設定しておく。

SMS受信を行うためには、SMS通信が可能な通信契約が必要ですので注意してください。「4.2. LTE」に記載したLTE接続設定が完了してLTEがすでに接続されている場合、SMSの受信は自動的に実行されます。SMSに関する詳細は「4.14. SMS」を参照してください。

またOS V1.1.5以降では、CONEXIOBlackBear はRTC alarmの割り込みによって“Suspend Mode”からResumeすることができるようになりました。

Resumeするタイミングは以下の例のように“rtcwake”コマンド等によって設定することが可能です。Resumeのタイミングを設定した後に“go-sys-suspend”コマンドを実行して“Suspend Mode”に入ってください。

例) 2020年5月26日15時20分0秒にResumeを実行させたい場合

```
# rtcwake -m no --date "2020-05-26 15:20:00"
# go-sys-suspend
```

例) 1時間後にResumeを実行させたい場合

```
# rtcwake -m no -s 3600
# go-sys-suspend
```

※rtcwakeコマンドを使用する場合"-m"オプションで"no"を指定して、Resumeのタイミングを設定するだけにしてください。"Suspend Mode"へ入るためには必ず"go-sys-suspend"コマンドを使用してください。

※rtcwakeコマンドの"--date"オプションで指定する時刻はシステムクロックの時刻で設定します。

RTCがUTCのまま、システムクロックがUTC以外のローカルタイム(JST等)に設定してある場合でも、rtcwakeコマンドにおいてローカルタイムで設定した時刻でResumeが実行されます。

設定したResumeのタイミングを確認したい場合は、以下のコマンドを実行することにより"alarm\_time"および"alarm\_date"として確認することができます。

例) 設定したRTC alarmを確認したい場合

```
# cat /proc/driver/rtc
rtc_time      : 09:39:29
rtc_date      : 2020-05-25
alarm_time    : 15:20:00
alarm_date    : 2020-05-26
.
.
.
```

“Suspend Mode”からのResumeが完了すると電源LEDの点滅が終了して点灯状態へ移行します。Resumeが完了すると“Suspend Mode”に入った時点からの処理が再開され、電源LED以外のLEDも“Suspend Mode”に入る前の状態に戻ります。

### 3.6. 終了（電源 OFF/再起動）

#### 3.6.1. 電源 OFF

表3.6-1 終了 – 電源OFF 実行方法

電源OFF手段	操作方法
Power OFF SW	10秒間以上長押しする
車載用電源ケーブル	ACC/IG電源をOFFにする
Linuxコマンド	“go-sys-off”コマンドを実行する

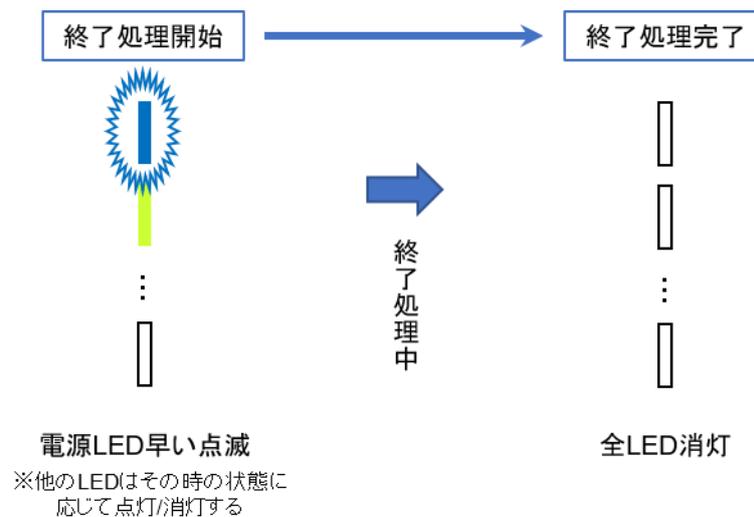


図3.6-1 終了時のLED表示

「図2.4-1 インターフェースレイアウト図」に示した“Power OFFスイッチ”を10秒間以上長押しすることにより、CONEXIOBlackBear が機能を停止して電源OFFが実行されます。車載用セットの車載用電源ケーブルを使用している場合は、ACC/IG電源をOFFにしても電源OFFが実行されます

“Power OFFスイッチ”は誤って電源がOFFになってしまうことを避けるために、“アンテナ端子面”に設けた窪みの中に埋め込まれています。そのため指や爪では押しにくい場合が有ることをご了承願います。必要に応じて細い棒状のもので押すようにしてください。

CONEXIOBlackBear の終了処理実行中は電源LEDが早く点滅し、終了処理が全て完了すると電源LEDが消灯して電源がOFFになります。思わぬ不具合や故障を避けるために、電源ケーブルやACアダプタを抜く場合は電源LEDが消灯してから抜いてください。

ACアダプタを使用している場合で、すぐに電源を再投入する必要がある時の電源遮断は、本体～ACアダプタ間(「図2.5-9 AC Adapter Conversion Cable」のどちらかのコネクタの抜き挿し)で行うことを推奨します。

ACアダプタへのAC100V供給を遮断することによって電源を遮断した場合、ACアダプタ内に蓄積されている電流を CONEXIOBlackBear 本体を通して放電する必要があります。そのためAC100V遮断後約20秒程度は、ACアダプタへのAC100V供給を再開しても電源供給が正常に行われず、CONEXIOBlackBear を起動することができません。ACアダプタへのAC100V供給を遮断した場合のAC100V再投入は、約20秒程度以上待つてから行う必要がありますので注意してください。

Linuxコマンドでシステムを終了させる場合は、shutdownコマンドやpoweroffコマンドではなく、“go-sys-off”コマンド(CONEXIOBlackBear 固有のコマンドです)を使用してください。

shutdownコマンドやpoweroffコマンドでもシステムを終了させることはできますが、MCUと連携して電源やLEDを制御し、終了処理が実行状態であることをLEDで確認しながら終了/電源OFF処理を実行することができる“go-sys-off”の使用を推奨します。

### 3.6.2. 再起動 (Reboot)

表3.6-2 Reboot 実行方法

Reboot手段	操作方法
Linuxコマンド	“go-sys-reboot”コマンドを実行する

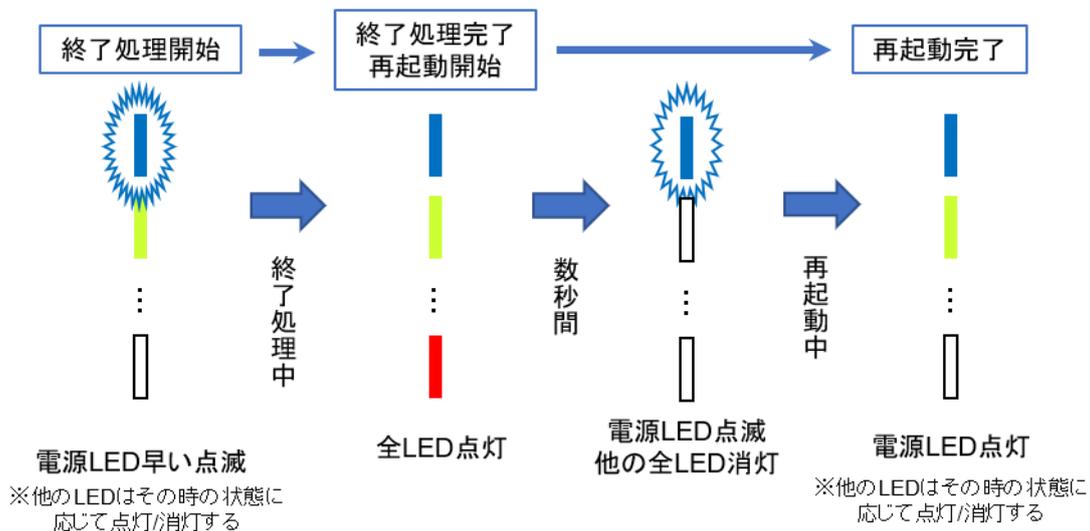


図3.6-2 Reboot (go-sys-reboot実行) 時のLED表示

再起動のためのrebootコマンドも同様で、システムを再起動させることはできますが、“go-sys-reboot”コマンド(CONEXIOBlackBear 固有のコマンドです)の使用を推奨します。

“go-sys-reboot”コマンドを使用するとMCUと連携して電源やLEDを制御し、終了処理中であるか起動処理中であることをLEDで確認しながら再起動処理を実行することができます。LEDの点滅/点灯パターンは、前述の終了処理時および起動処理時のパターンを組み合わせたものとなっています。

## 4. インターフェース使用方法

本章では、「図2.4-1 インターフェースレイアウト図」に記載した CONEXIOBlackBear の各種インターフェースの基本的な使い方について記載します。

### 4.1. Ethernet

#### 4.1.1. DHCP 接続

表4.1-1 Ethernet デバイス名

デバイス名
eth0

CONEXIOBlackBear のDefault設定では、CONEXIOBlackBear のEthernetコネクタにEthernet cableを接続するだけで、DHCPでIPアドレスを取得してEthernetが使用可能になります。Default設定のEthernetコネクション名は"Wired connection 1"です。

Default設定のままの状態では"/etc/NetworkManager/system-connections/"の下にはEthernetのコネクション情報がファイルとして保存されていませんが、Network Managerから変更等の管理を行うことが可能です。Network Managerから変更をかけると"/etc/NetworkManager/system-connections/"の下に変更した内容でEthernetのコネクション情報が保存されます。

Default設定のEthernetコネクション"Wired connection 1"を使用せずに、代わりにDHCPでIPアドレスを取得するEthernetコネクションをNetwork Managerで新たに作成して使用する場合の例を下記に記載します。Network Managerで管理可能なEthernetデバイスはeth0です。

```
# nmcli connection down 'Wired connection 1'
# nmcli connection delete 'Wired connection 1'
# nmcli connection add con-name lan-eth0 type ethernet ifname eth0
# nmcli connection up lan-eth0
```

※"lan-eth0"は、新たに追加作成するEthernetコネクション名の例です。

※Network Managerで新たに追加作成した場合には"/etc/NetworkManager/system-connections/"の下に作成時に設定した内容でEthernetのコネクション情報が保存されます。

#### 4.1.2. 固定 IP 接続

EthernetをDefault設定のDHCP接続ではなく固定IPアドレスで使いたい場合は、設定を変更する必要があります。例えばNetwork Managerを使用して設定を変更する場合の例を下記に記載します。固定IPアドレスを使用する場合は、Defaultで使用しているコネクション名の"Wired connection 1"を使用することは推奨していません。再起動時に不具合が発生する場合は有るため、CONEXIOBlackBear のEthernetコネクションとして固定IPを使用する場合は、別のコネクション名を使用してください。

```
# nmcli connection down 'Wired connection 1'
# nmcli connection delete 'Wired connection 1'
# nmcli connection add con-name lan-eth0 type ethernet ifname eth0 \
  ipv4.method manual ipv4.addresses [IPADDRESS] ipv4.gateway [GATEWAY]
# nmcli connection up lan-eth0
```

- ※"lan-eth0"は、新たに追加作成するEthernetコネクション名の例です。
- ※"[IPADDRESS]"は、使用したい固定IPアドレスです。(例: 192.168.21.117/24)
- ※"[GATEWAY]"は、使用したいDefault GatewayのIPアドレスです。(例: 192.168.21.1)

### 4.1.3. DHCP/DNS サーバー機能

CONEXIOBlackBear には、標準でdnsmasqがプリインストールされています。このdnsmasqによってDHCPサーバーならびにDNSサーバー機能を実行することができます。

EthernetインターフェースでDHCP/DNSサーバー機能を実行するコネクションを作成するための設定例を下記に記載します。

```
# nmcli connection add type ethernet ifname eth0 con-name eth_shared ¥  
ipv4.method shared ipv4.addresses [NETWORK]
```

- ※"eth\_shared"は、DHCPサーバーとなるEthernetコネクション名の例です。
- ※"[NETWORK]"は、使用したいネットワークのIPアドレスです。  
例えば"192.168.9.1/24"を設定した場合、CONEXIOBlackBear のEthernet IPアドレスが192.168.9.1となり、CONEXIOBlackBear にEthernet接続された機器に対して192.168.9.1/24ネットワークのIPアドレスが割り振られます。
- ※上記設定を行うとdnsmasqの機能によって同時に CONEXIOBlackBear がDNSサーバーになります。
- ※またIPv4ルーティングが有効になり、LTE<->Ethernet間の通信を実現するためのIPマスカレードを含むiptablesへの登録が行われます。  
これによりLTE接続が有効な場合は、上記の設定でEthernetインターフェースに接続された機器からLTEによってInternetへ出ることができるようになります。

### 4.1.4. RJ45 connector

Ethernet cableを接続するRJ45 connectorには「図4.1-1 RJ45 connector LEDs」に記載したように、Ethernetの動作状態を示す2つのLEDが取り付けられています。それぞれのLEDが何を示しているかを「表4.1-2 リンク速度LED」および「表4.1-3 リンク状態LED」に記載します。

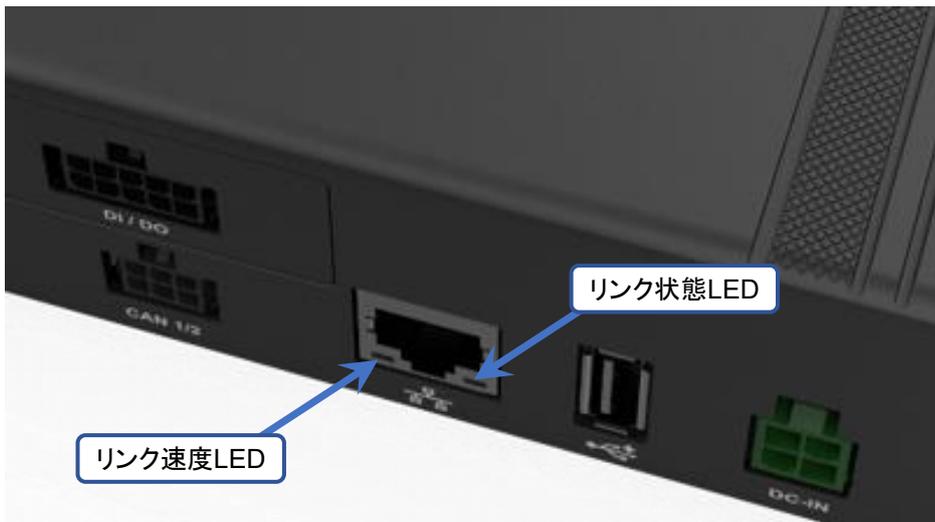


図4.1-1 RJ45 connector LEDs

表4.1-2 リンク速度LED

LEDの状態	リンク速度
消灯	10Mbps / Ethernet未接続
点灯:緑	100Mbps
点灯:オレンジ	1000Mbps

表4.1-3 リンク状態LED

LEDの状態	リンク状態
消灯	リンクが確立されていない
点灯:黄	リンクが確立されている
点滅:黄	データの送受信が発生している

## 4.2. LTE

表4.2-1 LTE Module デバイス名

デバイス名
ttyUSB2

CONEXIOBlackBear でNetwork Managerを使用してLTE接続設定を行う場合の例を下記に記載します。Network Managerで管理可能なLTE ModuleデバイスはttyUSB2です。

```
# nmcli connection add con-name lte-wwan0 type gsm ifname ttyUSB2 \
apn [APN] user [USER] password [PASSWORD]
```

※"lte-wwan0"は、新たに接続設定を行うLTEコネクション名の例です。

※"[APN]"は、接続するAPNです。

※"[USER]"は、APNへ接続する際に使用するユーザー名です。

※"[PASSWORD]"は、APNへ接続する際に使用するパスワードです。

回線の認証タイプとしてPAP認証が指定されている場合は、下記のようにCHAP認証を無効にするオプションを追加して接続設定を実施してください。

```
# nmcli connection add con-name lte-wwan0 type gsm ifname ttyUSB2 \
apn [APN] user [USER] password [PASSWORD] \
ppp.refuse-eap true ppp.refuse-chap true ppp.refuse-mschap true \
ppp.refuse-mschapv2 true ppp.refuse-pap false
```

ifconfigコマンドやip addrコマンドでネットワークインターフェースの状態を確認する場合には、インターフェース"ppp0"にLTE接続の状態が表示されます。

また以下のModemManagerのコマンドを使用して、挿入されたSIMやLTE通信状態を確認することができます。

```
# mmcli -m 0
```

通常はNetwork ManagerやModem Managerが"/dev/ttyUSB2"を使用しています。そのため、これらのserviceを使用中にLTE Moduleに対してATコマンドを実行するなどのアクセスをしたい場合は"/dev/ttyUSB3"を使用します。

なお、SIMを別のSIMに交換して起動した後、SIMの情報とAPN情報を設定したコネクション情報との間に不整合がある場合(例えば交換する前のSIMのAPN情報を設定したコネクション情報が設定されたままの時など)は、“audit: type=1701...”というメッセージが表示されます。この場合は正しいAPN情報を新たなコネクションとして設定し直すことにより、このメッセージは表示されなくなります。

新たにSIMを使用し始める場合や、SIMを別のSIMに交換した場合は、APN情報を新たなコネクション情報として設定した後、両者の情報の整合性を確実にとるために、一旦再起動することを推奨します。

表4.2-2 LTE 有効化/無効化コマンド

コマンド	機能
lte_on	アプリケーションやコマンドラインからLTEを有効化(ttyUSB2を使用可能にする)するLTE Moduleの電源ON
lte_off	アプリケーションやコマンドラインからLTEを無効化(ttyUSB2を使用不能にする)するLTE Moduleの電源OFF

OS Version1.1.5以降では、コマンドによるLTE Module電源のOFF/ONが可能になりました。LTE Moduleの電源をOFFにする場合は“lte\_off”コマンドを、LTE Moduleの電源を再投入する場合は“lte\_on”コマンドを実行してください。(“lte\_off”コマンドの実行完了までには多少時間がかかります。“Power off LTE module”というメッセージが出るまでお待ちください。“lte\_off”および“lte\_on”は CONEXIOBlackBear 固有のコマンドです)

“lte\_off”を実行するとLTE Moduleの電源が完全にOFFになり、ModemManagerも停止しますので、SMSを含めたLTE通信機能は一切使用することはできません。

CONEXIOBlackBear の起動/再起動処理中には、システム設定のためにCPUがLTE Moduleにアクセスしていますので、CONEXIOBlackBear の起動/再起動後は必ず“lte\_on”の状態となります。そのため“lte\_off”の状態でごgo-sys-rebootを実行しても、再起動後は“lte\_on”の状態となります。また“lte\_off”コマンドを実行する場合は、必ず起動/再起動処理が完了してからにしてください。

go-sys-rebootを実行した場合の再起動時にはLTE Module電源のOFF/ONが実行されますが、起動後にLTE Moduleだけをリセットしたい場合には、lte\_off → delay → lte\_onを実行してください。

ちなみに、Network ManagerとModemManagerを停止したうえで、ATコマンドを使用してLTE接続設定を行う場合には、LTE Moduleデバイスとしてwwan0を使用します。ATコマンドを使用した接続方法例を、以下に記載します。

```
# systemctl stop ModemManager.service
# systemctl stop NetworkManager.service
# echo -e 'AT+CSOCKAUTH=1,2,"[PASSWORD]","[USER]"%r%n' > /dev/ttyUSB2
# echo -e 'AT+CGSOCKCONT=1,"IP","[APN]"%r%n' > /dev/ttyUSB2
# echo -e 'AT+CGDCONT=1,"IP","[APN]"%r%n' > /dev/ttyUSB2
# echo -e 'AT+QCRMICALL=1,1"%r%n' > /dev/ttyUSB2
# udhcdc -i wwan0
```

上記のような接続設定を行った場合にifconfigコマンドやip addrコマンドでネットワークインターフェースの状態を確認する時は、“wwan0”にLTE接続の状態が表示されます。

## 4.3. WiFi

### 4.3.1. アクセスポイントへの接続

表4.3-1 WiFi デバイス名

デバイス名
wlan0

CONEXIOBlackBear のdefaultでは、WiFi機能は起動時に自動的に有効になりません。

WiFi機能は、「表4.3-2 WiFi 有効化/無効化コマンド」に記載した、WiFi有効化コマンドである“wifi\_on”を実行することによって有効になります(このコマンドは CONEXIOBlackBear 固有のコマンドです)。このWiFi有効化コマンドは、defaultの状態ではNetwork Managerを使用してWiFiコネクションを設定した後であっても、電源を投入して起動したり再起動したりするたびごとに実行することが必要です。

起動時に自動的にWiFi機能を有効化したい場合は「4.3.4 CONEXIOBlackBear 起動時の自動的なWiFi機能有効化」に記載した方法を実行してください。

表4.3-2 WiFi 有効化/無効化コマンド

コマンド	機能
wifi_on	アプリケーションやコマンドラインからWiFiを有効化(wlan0を使用可能にする) WiFi通信モジュールの電源ON LTE Moduleの型番に合わせてWiFiで使用可能な周波数帯を設定・有効化する
wifi_off	アプリケーションやコマンドラインからWiFiを無効化(wlan0を使用不能にする) WiFi通信モジュールの電源OFF

上記WiFi有効化コマンドを実行すると、Network Managerで管理することができるWiFiデバイスwlan0が使用可能になります。すでにNetwork ManagerでWiFiコネクションが設定済の場合は、上記コマンド実行後すぐに、設定されたWiFiコネクション情報に基づいた処理が実行されます。Network Managerを使用してWiFiコネクションの設定を行う場合(CONEXIOBlackBear をWiFiの子機として使用する場合)の例を下記に記載します。

```
# nmcli connection add con-name wifi-wlan0 type wifi ifname wlan0 ¥
ssid [SSID] 802-11-wireless-security.key-mgmt [SECURITY] ¥
802-11-wireless-security.psk [PASSPHRASE]
```

※“wifi-wlan0”は、新たに接続設定を行うWiFiコネクション名の例です。

※“[SSID]”は、接続したいアクセスポイントのSSIDです。

※“[SECURITY]”は、“wpa-psk”等のアクセスポイントへの接続に適したものを設定してください。

※“[PASSPHRASE]”は、アクセスポイントへ接続するために必要となるパスワードです。

WiFiを無効にするためには、「表4.3-2 WiFi 有効化/無効化コマンド」に記載したWiFi無効化コマンドである“wifi\_off”コマンドを実行します(WiFi有効化コマンド同様にCONEXIOBlackBear 固有のコマンドです)。

WiFi無効化コマンドを実行してもNetwork Managerで設定したWiFiコネクション情報は保存されています。WiFiコネクションがすでに設定されている場合には、WiFi有効化コマンドを実行することにより、WiFiコネクションが有効になり、WiFiコネクション情報に基づいた処理が実行されます。

使用する機能(2.4GHz/5GHz、子機/AP)を変更する場合などWiFiの設定を変更する時は、一旦WiFi無効化コマンドを実行してから設定を変更し、設定完了後にWiFi有効化コマンドを実行してください。

### 4.3.2. 2.4GHz アクセスポイント設定

「4.1.3. DHCP/DNSサーバー機能」に記載したように、CONEXIOBlackBear には標準でdnsmasqがプリインストールされており、WiFi機能でもDHCPサーバーならびにDNSサーバー機能を実行することができます。

CONEXIOBlackBear を2.4GHzのアクセスポイントとして使用する場合の設定例を下記に記載します。

```
# nmcli con add type wifi ifname wlan0 mode ap con-name WIFI_AP ¥
ssid BBtest-g
# nmcli con modify WIFI_AP 802-11-wireless.band bg
# nmcli con modify WIFI_AP 802-11-wireless.channel 1
# nmcli con modify WIFI_AP 802-11-wireless-security.key-mgmt wpa-psk
# nmcli con modify WIFI_AP 802-11-wireless-security.auth-alg open
# nmcli con modify WIFI_AP 802-11-wireless-security.proto rsn
# nmcli con modify WIFI_AP 802-11-wireless-security.group ccmp
# nmcli con modify WIFI_AP 802-11-wireless-security.pairwise ccmp
# nmcli con modify WIFI_AP 802-11-wireless-security.psk blackbear
# nmcli con modify WIFI_AP ipv4.method shared
# nmcli con modify WIFI_AP ipv4.addr 192.168.31.1/24
```

※アクセスポイント の設定内容は、使用する環境に合わせて適宜変更してください。

各パラメータの詳細は"<https://developer.gnome.org/NetworkManager/stable/>"等を参照してください。

※上記の例では以下のような設定を使用しています。

SSID: BBtest-g

PASSPHRASE: blackbear

IP Address: CONEXIOBlackBearが102.168.31.1となり、192.168.31.1/24ネットワークのIPアドレスが割り振られます。

WiFi Security: wpa-psk

※上記設定を行うとdnsmasqの機能によって同時に CONEXIOBlackBear がDNSサーバーになります。

※またIPv4ルーティングが有効になり、LTE<->WiFi間の通信を実現するためのIPマスカレードを含むiptablesへの登録が行われます。

これによりLTE接続が有効な場合は、上記の設定でWiFiインターフェースに接続された機器からLTEによってInternetへ出ることができるようになります。

OS V1.0.1よりも前のOS versionを使用している場合は、アクセスポイントとして設定する前に、下記のようにnamed.serviceを無効化しておく必要が有りますのでご注意ください。OS V1.0.1以降ではnamed.service無効化の設定は不要です。

```
# systemctl stop named.service
# systemctl disable named.service
```

### 4.3.3. 5GHz アクセスポイント設定

CONEXIOBlackBear を5GHzのアクセスポイントとして使用する場合の設定例を下記に記載します。注意事項等は「4.3.2. 2.4GHz アクセスポイント設定」と同様です。

```
# nmcli con add type wifi ifname wlan0 mode ap con-name WIFI_AP_5G ¥
ssid BBtest-a
# nmcli con modify WIFI_AP_5G 802-11-wireless.band a
# nmcli con modify WIFI_AP_5G 802-11-wireless.channel 36
# nmcli con modify WIFI_AP_5G 802-11-wireless-security.key-mgmt wpa-psk
# nmcli con modify WIFI_AP_5G 802-11-wireless-security.auth-alg open
# nmcli con modify WIFI_AP_5G 802-11-wireless-security.proto rsn
# nmcli con modify WIFI_AP_5G 802-11-wireless-security.group ccmp
# nmcli con modify WIFI_AP_5G 802-11-wireless-security.pairwise ccmp
# nmcli con modify WIFI_AP_5G 802-11-wireless-security.psk blackbear
# nmcli con modify WIFI_AP_5G ipv4.method shared
# nmcli con modify WIFI_AP_5G ipv4.addr 192.168.32.1/24
```

なお、「8.2.3. WiFi(無線LAN)の周波数帯(5GHz帯)」記載の使用可能周波数帯にご注意願います。使用できないchannelを指定しても有効な接続を設定することはできません。

### 4.3.4. CONEXIOBlackBear 起動時の自動的なWiFi機能有効化

前述のように CONEXIOBlackBear のdefaultでは、WiFi機能は起動時に自動的に有効になりません。

WiFi機能を CONEXIOBlackBear 起動時に自動的に有効にしたい場合は、下記のコマンドを実行してください。このコマンドを実行後、次の起動時からWiFi機能が自動的に有効になります。

```
# touch /lib/firmware/brcm/auto_wifi_on
```

※本コマンドはOS V1.0.8以降で使用可能です。

「表4.3-2 WiFi 有効化/無効化コマンド」に記載した、WiFi有効化コマンド"wifi\_on"は、LTE Moduleの型番情報を取得した後、その情報に基づいて必要な設定を実行します。そのため CONEXIOBlackBear 起動時に自動的にWiFi機能を有効にしたいという場合、rc.localへの"wifi\_on"の追記では正常に動作しませんので注意してください。

※OS V1.0.3以前の場合は、"wifi\_on"コマンドは、LTE Moduleの型番情報を取得するなどの処理がまだ有効になっていないため、rc.localへの"wifi\_on"の追記でも動作させることができます。

起動時に自動的に有効化されたWiFi機能を一時的に無効化したい場合は、前述の"wifi\_off"コマンドを実行してください。再度WiFi機能を有効化する場合は、前述の"wifi\_on"を実行することによってWiFi機能を有効化することができます。

起動時に自動的にWiFi機能を有効化しない設定に戻したい場合は、下記のコマンドを実行してください。このコマンドを実行後、次の起動時からWiFi機能が自動的に有効には有効にはなりません。

```
# rm /lib/firmware/brcm/auto_wifi_on
```

※本コマンドはOS V1.0.8以降で使用可能です。

これで CONEXIOBlackBear のdefault状態(WiFi機能は起動時に自動的に有効には有効にはなりません)に戻ります。

## 4.4. Bluetooth

表4.4-1 Bluetooth デバイス名

デバイス名
hci0

CONEXIOBlackBear にはBlueZが実装されており、hcitoolやhciconfigなどが使用可能です。これらのツールで使用可能なBluetoothデバイスはhci0です。

Bluetoothを有効にするためには、「表4.4-2 Bluetooth 有効化/無効化コマンド」に記載したBluetooth有効化コマンドである”bt\_on”コマンドを実行します(CONEXIOBlackBear 固有のコマンドです)。このコマンドは電源を投入して起動したり再起動したりするたびごとに実行することが必要です。

このコマンドを実行することにより、hciconfig等で管理することが可能なBluetoothデバイスhci0が有効になります。

表4.4-2 Bluetooth 有効化/無効化コマンド

コマンド	機能
bt_on	アプリケーションやコマンドラインからBluetoothを有効化する (hci0を使用可能かつ”hciconfig up”の状態にする) Bluetoothモジュールの電源ON
bt_off	アプリケーションやコマンドラインからBluetoothを無効化する (hci0を”hciconfig down”の状態にする) BluetoothモジュールをSuspend状態にする

また以下のようなコマンドで CONEXIOBlackBear のBluetoothデバイス情報を確認したり、周辺のBluetoothデバイスをスキャンしたりすることができます。

```
# hciconfig -a
# hcitool scan
```

コマンドラインからBluetooth接続を行う場合には、bluetoothctlを実行して対話的に接続を行うこともできます。ここではbluetoothctlを利用した接続例の概略のみを記載します。(接続するデバイスによって詳細は異なる場合があります)

```
# bluetoothctl
[bluetooth]# power on          コマンドラインでbluetoothctlを実行
                               (プロンプトが変わる) コントローラを起動
[bluetooth]# scan on          周辺をスキャンして接続したいデバイスを探す
[bluetooth]# devices          ペアリングしたいデバイスのBDaddressを確認する
[bluetooth]# pair [BDaddress] ペアリングする
[bluetooth]# trust [BDaddress] 信用登録する
[bluetooth]# connect [BDaddress] 接続する
...
[bluetooth]# help            Helpを見る
...
[bluetooth]# exit            bluetoothctlを終了して対話モードから抜ける
#
```

Bluetoothを無効な状態(コネクションがdownした状態)にするためには、「表4.4-2 Bluetooth 有効化/無効化コマンド」に記載したBluetooth無効化コマンドである”bt\_off”コマンドを実行します(”bt\_on”コマンド同様にCONEXIO BlackBear 固有のコマンドです)。

再度Bluetoothを有効にする場合は、上記のように”bt\_on”コマンドを実行します。

## 4.5. Serial

表4.5-1 Serial通信 デバイスファイル

デバイスファイル
/dev/ttyxc2

CONEXIO BlackBear のSerial通信に使用するデバイスファイルは”/dev/ttyxc2”です。Serialインターフェースは、RS-232C、RS-485 Half Duplex、RS-485/RS-422 Full Duplexに対応しており、これらを「図4.5-1 Serialインターフェース切り替えスイッチ」に記載した”SD-Door”の内側に有る3P DIPスイッチで切り替えて使用します。3P DIPスイッチの切り替えは電源OFFの状態で行ってください。

切り替えスイッチを使用する場合は、SD-Doorを大きく180度開いてください。  
この図ではSD-Doorを90度しか開いていないため切り替えスイッチがSD-Doorの陰に少し隠れています。

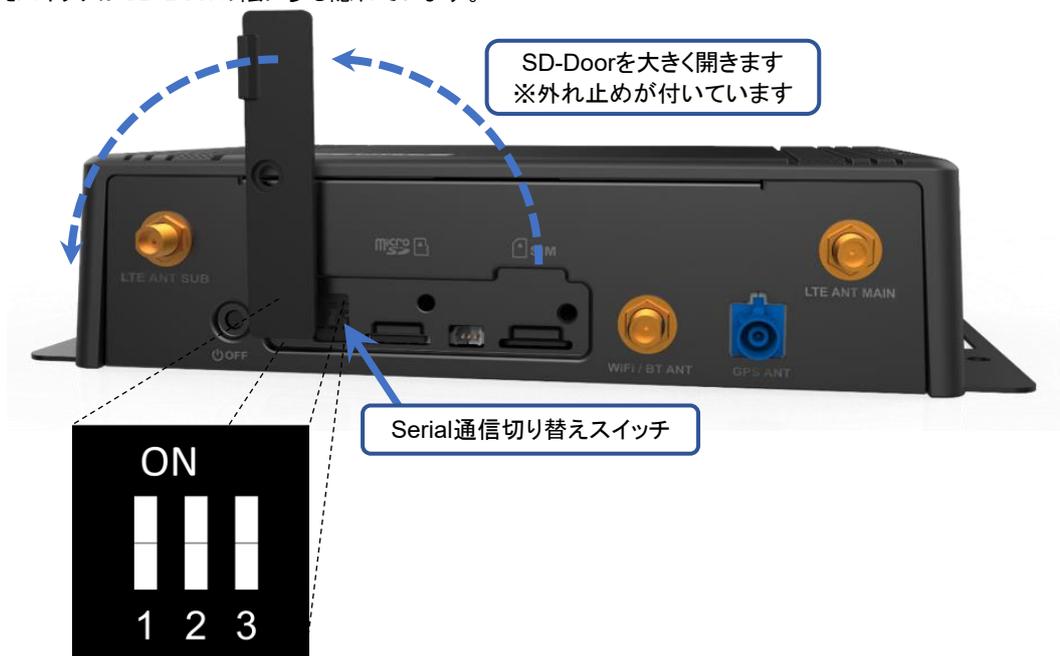


図4.5-1 Serialインターフェース切り替えスイッチ

Serialインターフェース切り替えスイッチの設定を 表4.5-2 および 表4.5-3 に記載します。

図4.5-1 のSW1とSW2をRS-232C、RS-485 Half Duplex、RS-485/RS-422 Full Duplexの切り替えに使用し、SW3はRS-485/RS-422を使用する場合の終端抵抗の設定に使用します。

表4.5-2 Serialインターフェース切り替え設定

SW1	SW2	Serial I/F Mode
OFF	OFF	LOOPBACK
ON	OFF	RS-232C
OFF	ON	RS-485 Half Duplex
ON	ON	RS-485/RS-422 Full Duplex

表4.5-3 終端抵抗設定

SW3	終端抵抗
OFF	OPEN
ON	120Ω

CONEXIOBlackBear には、Serial通信の動作状況を試験するツールであるuarttestが実装されています。下記のコマンド例のようにして、uarttest搭載機器同士の対向またはloopbackでSerial通信の接続を確認できます。

```
# uarttest -s -e -p /dev/ttymx2 -b 115200 -q -c -a 2000
```

ちなみにRS-485をプログラムから使用する場合には、プログラム中に以下の設定が必要になります。

```
/*In userspace, please make sure the configurtion of RS-485 as below:*/

/* enable RS485 mode: */
rs485conf.flags |= SER_RS485_ENABLED;

/* set logical level for RTS pin equal to 1 when sending: */
rs485conf.flags |= SER_RS485_RTS_ON_SEND;

/* set logical level for RTS pin equal to 0 after sending: */
rs485conf.flags &= ~(SER_RS485_RTS_AFTER_SEND);
```

## 4.6. CAN

表4.6-1 CAN デバイス名

デバイス名
can0
can1

CONEXIOBlackBear はCAN Version 2.0B activeに対応しているcan0とcan1という2系統のCANポートを標準搭載しています。これらのCANポートを有効にするためには、有効にするCANポートごとに以下のコマンドを実行します。

例) can0ポートを有効にする場合

```
# ip link set can0 up type can bitrate 250000
# ifconfig can0 up
```

LinuxのSocketCAN機能が使用可能で、生のCANフレーム(標準および拡張フォーマットに対応)を直接送受信できます。またCAN-Utillsが実装されており、“candump”、“canecho”、“cansend”、“cansequence”が使用できます。送信試験は“cansend”コマンド、受信試験は“candump”コマンドで実行できます。なおSAE J1939プロトコルには対応していません。

例) can0ポートへデータ出力をする場合

```
# cansend can0 -i 0x801 0x01 0x22 0x23 0x34 0x45 0x56 0x78 0x89 -e --loop=10
```

例) can1ポートからデータを入力する場合

```
# candump can1
```

CAN-BUSの途中でも末端でも CONEXIOBlackBear を使用可能とするために、CANポートには終端抵抗を設置していませんので注意してください。(終端抵抗の切り替え機能はありません)

## 4.7. DI/DO

### 4.7.1. DI/DO 制御用ポート

CONEXIOBlackBear のDI/DOは「表4.7-1 DI/DO制御用ポート」に記載した制御用gpioポートを使用して制御します。DIポートはデジタル入力用の4つのポート、DOポートはデジタル出力用の4つのポート、PWR\_CTLポートはDIまたはDOポートのpull upを制御するポートです。

表4.7-1 DI/DO制御用ポート

制御用ポート	機能
/sys/class/gpio/gpio9	D11
/sys/class/gpio/gpio205	D12
/sys/class/gpio/gpio101	D13
/sys/class/gpio/gpio171	D14
/sys/class/gpio/gpio167	DO1
/sys/class/gpio/gpio170	DO2
/sys/class/gpio/gpio32	DO3
/sys/class/gpio/gpio33	DO4
/sys/class/gpio/gpio2	PWR_CTL_DI
/sys/class/gpio/gpio5	PWR_CTL_DO

上記のポートは起動時には有効になっていません。使用する場合には上記全ての制御用gpioポートNo.を、下記のように/sys/class/gpio/exportlに対して書き込んで、ポートを有効にする必要が有ります。

例) DI1を制御するgpioポートを有効にする場合  
`# echo 9 > /sys/class/gpio/export`

次にDIポート、DOポート、PWR\_CTLポートとして使用する設定を行います。設定は各制御用gpioポートのdirectionに対して「表4.7-2 各ポートの機能設定」に記載した値を書き込みます。  
 (DIポートは"in"を、DOポートは"out"を、PWR\_CTLポートは"out"を書き込みます)

表4.7-2 各ポートの機能設定

制御用ポート	direction 設定値
/sys/class/gpio/gpio9	in
/sys/class/gpio/gpio205	in
/sys/class/gpio/gpio101	in
/sys/class/gpio/gpio171	in
/sys/class/gpio/gpio167	out
/sys/class/gpio/gpio170	out
/sys/class/gpio/gpio32	out
/sys/class/gpio/gpio33	out
/sys/class/gpio/gpio2	out
/sys/class/gpio/gpio5	out

例) DI2を制御するポートをDIポートとして設定する場合  
`# echo "in" > /sys/class/gpio/gpio205/direction`

さらにPWR\_CTL\_DIポートおよびPWR\_CTL\_DOポートに対して、DIポートおよびDOポートを”無電圧接点”にするか、CONEXIOBlackBear への入力電圧へ”pull up”するかの設定を行います。設定はそれぞれのポートのvalueに対して「表4.7-3 PWR\_CTLポートの設定値」に記載した設定値を書き込みます。

表4.7-3 PWR\_CTLポートの設定値

制御用ポート	value 設定値	
	0	1
/sys/class/gpio/gpio2	DI 無電圧接点	DI pull up
/sys/class/gpio/gpio5	DO 無電圧接点	DO pull up

例) DIポートを CONEXIOBlackBear への入力電圧へpull upする場合  
`# echo 1 > /sys/class/gpio/gpio2/value`

(注): 上記の”無電圧接点”または”pull up”の設定は、DI1～DI4全体への設定になります。ポートごとに個別に設定することはできません。gpio5に対する設定も同様に、DO1～DO4全体への設定になります。

上記のような設定を行った後、各DOポートの制御用gpioポートのvalueに対して”0”を書き込んだ場合は”Hi”レベル(DO pull up 時)または”Hiインピーダンス”(無電圧接点時)の出力が、”1”を書き込んだ場合は”Lo”レベルの出力がDOポートで実行されます。

例) DO3ポートに対して"Lo"レベルの出力を実行する場合

```
# echo 1 > /sys/class/gpio/gpio32/value → DO3ポートの出力:"Lo"レベル
```

また各DIポートの制御用gpioポートのvalueから値を読み出すことによって、DIポートへの入力状態"Hi"レベル(無電圧接点時)または"Hiインピーダンス"(DI pull up 時)の場合は"0"、"Lo"レベルの場合は"1"を取得できます。

例) DI4ポートへの入力状態を取得する場合

```
# cat /sys/class/gpio/gpio171/value
0 → DI4ポートの入力状態:"Hi"レベル/"Hiインピーダンス"
```

なお制御用gpioポートの状態は、以下のコマンドで確認することができます。このコマンドで確認することができる各制御用gpioポートのレベル表記はvalueが"0"の場合は"lo"、"1"の場合は"hi"と表示されます。DI/DOポートの入出力レベルとは逆の表記になりますので注意してください。

```
# cat /sys/kernel/debug/gpio
```

#### 4.7.2. DIの使い方

DIへはオープンコレクタ出力の機器を接続することを想定しています。例えばレーザーセンサー等をつないでそのON/OFFステータスを CONEXIOBlackBear に伝えることができます。DIの回路図を「図4.7-1 DI回路図」に示します。

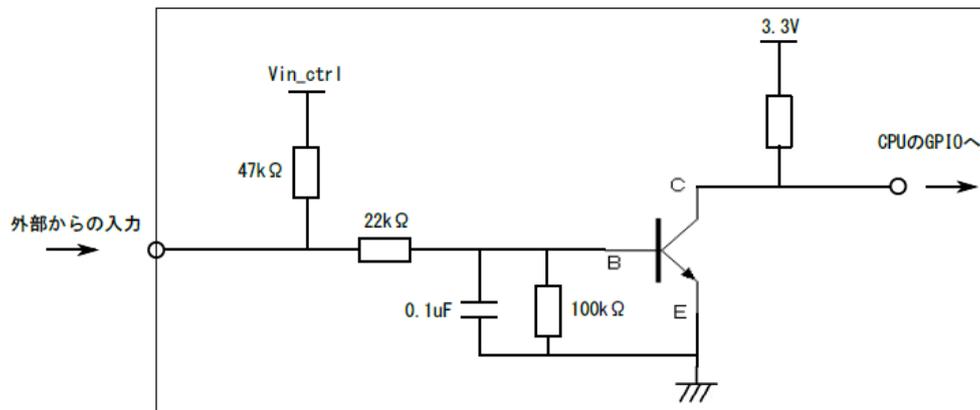


図4.7-1 DI回路図

左側が CONEXIOBlackBear の外部、四角の枠で囲んだ部分が CONEXIOBlackBear 内部のDI回路です。プルアップは CONEXIOBlackBear 内部のFETでON/OFF制御しています。プルアップの電圧 Vin\_ctrl は、CONEXIOBlackBear の電源電圧と同じ電圧値です。

- 無電圧の機器と接続する場合 (内部プルアップ: ON)
  - 接続する機器が以下の条件で問題がないか、ご確認ください。
  - ・ 接続機器がハイインピーダンスの時 : Vin\_ctrl の電圧がかかること
  - ・ 接続機器がローの時 :  $Vin\_ctrl \div 47k\Omega$  の電流を流せること。(24Vの場合で約510uA)
- 有電圧の機器と接続する場合 (内部プルアップ: OFF)
  - DIにかかる電圧は25V以下で、CONEXIOBlackBear の電源電圧と同じ電圧以下を推奨します。

前述のように、DIへの接続機器としてはオープンコレクタ出力の機器を想定しています。同様の回路部品としてはスイッチやリレーがありますが、以下のような注意が必要です。

- スイッチ : 最小電流の規定がある場合、正常動作しない可能性があります。
- リレー : メカ・リレーの最小適用負荷や、ソリッドステート・リレーの出力ON電圧降下の影響で正常動作しない可能性があります。

また「図4.7-1 DI回路図」に示したように、回路保護やノイズ対策のために抵抗やコンデンサを入れているため、パルス等の高速で変化する信号の取得には向いていませんので注意してください。

### 4.7.3. DO の使い方

DOはオープンコレクタ出力によって、例えば CONEXIOBlackBear での処理結果のステータス(ON/OFF状態など)を、接続した機器に伝えることを想定しています。DOの回路図を「図4.7-2 DO回路図」に示します。

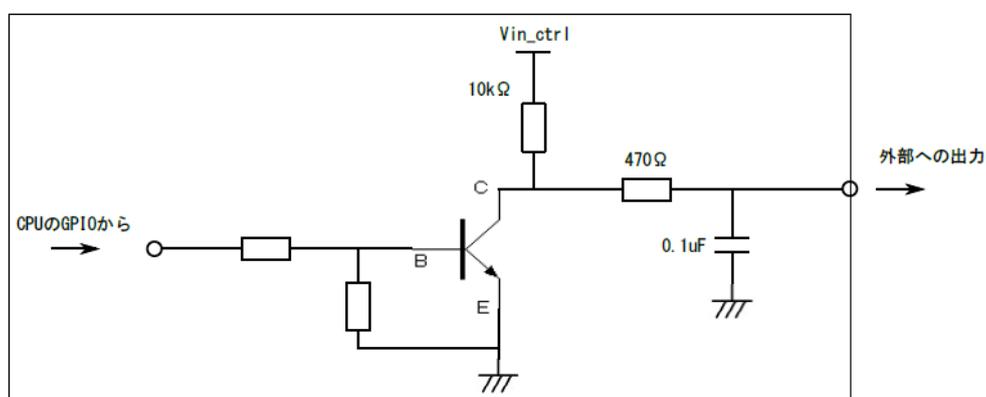


図4.7-2 DO回路図

右側が CONEXIOBlackBear の外部、四角の枠で囲んだ部分が CONEXIOBlackBear 内部のDO回路です。DOもDIと同様にプルアップは CONEXIOBlackBear 内部のFETでON/OFF制御しています。

「図4.7-1 DI回路図」と「図4.7-2 DO回路図」では、プルアップ電圧が同じ名称(Vin\_ctrl)になっていますが、「4.7.1. DI/DO制御用ポート」に記載したように、DIとDOのプルアップ電圧は個別に制御することができます。

- 無電圧の機器と接続する場合 (内部プルアップ: ON)  
接続する機器にプルアップ電圧(Vin\_ctrl)がかかることと、プルアップ抵抗の定数でハイレベルを認識できるかを確認してください。
- 有電圧の機器と接続する場合 (内部プルアップ: OFF)  
外部機器側でプルアップする場合の電圧は40V以下でBBの電源電圧と同じ電圧以下を推奨します。  
ロー出力の時にトランジスタが流せる電流の下限値は約3.7mAです。

また「図4.7-2 DO回路図」に示したように、回路保護のために抵抗を直列に入れていることと、トランジスタのコレクタ電流を制限しているため、DOはリレーのコイルを直接ONにすることができる電流を流せません。そのためリレーを接続機器として使用する場合には、別途リレーのコイルをONにするための外部回路が必要となりますので注意してください。

## 4.8. GNSS

表4.8-1 GNSS Module デバイスファイル

デバイスファイル
/dev/ttymx4

CONEXIOBlackBear はGNSS Moduleとしてu-bloxのNEO-M8Nを標準搭載しています。GNSS Moduleで取得された位置情報は、シリアル通信出力として"/dev/ttymx4"というデバイスファイルに1秒間隔で出力されています。

例) GNSS Moduleからの位置情報を取得する方法その1

```
# ln /dev/ttymx4 /dev/gpsdevice
# stty -F /dev/gpsdevice ispeed 9600 && cat < /dev/gpsdevice
```

GNSS Moduleから位置情報が出力されているかどうかを簡単に確かめるだけならば、以下のような方法でも確認することができます。

例) GNSS Moduleからの位置情報を取得する方法その2

```
# cat /dev/ttymx4
```

CONEXIOBlackBear ではNEO-M8Nをデフォルト設定で使用しています。このデフォルト設定で位置情報取得のために使用しているGNSS衛星は「GPS+GLONASS+SBAS+QZSS」です。

これらの衛星と同時にGalileoを追加して使用することができますが、使用するためには下記のコマンド例のように、NEO-M8Nに対してGalileoを追加使用するための設定を行う必要が有ります。

例) Galileoを追加して「GNSS+GLONASS+Galileo+SBAS+QZSS」を使用するためのコマンド例

```
# echo -e "\xB5\x62\x06\x3E\x3C\x00\x00\x00\x20\x07\x00\x08\x10\x00\x01\x00\x01\x01\x01\x03\x00\x01\x01\x01\x02\x04\x08\x00\x01\x00\x01\x01\x01\x03\x08\x10\x00\x00\x00\x01\x01\x04\x00\x08\x00\x00\x00\x01\x01\x05\x00\x03\x00\x01\x00\x01\x01\x06\x08\x0E\x00\x01\x00\x01\x01\x30\xAD" > /dev/ttymx4
# echo -e "\xB5\x62\x06\x17\x14\x00\x00\x41\x00\x02\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x75\x57" > /dev/ttymx4
```

GNSS Moduleから出力される位置情報は、NMEAフォーマットのメッセージです。GNSS Moduleのデフォルト設定で出力されるメッセージの種類を「表4.8-2 NMEAメッセージ」に記載します。

表4.8-2 NMEAメッセージ

出力メッセージ	主な情報
GNRMC	位置、測位日時、速度、方位
GNVTG	速度、方位
GNGGA	位置、測位時刻、衛星捕捉数、精度
GNGSA	測位使用衛星、精度低下率
GPGSV	衛星情報 (GPS、QZSS)
GLGSV	衛星情報 (GLONASS)

位置情報の取得に使用する衛星として、準天頂衛星みちびき(QZSS)に対応していますが、対応機能は”L1C/A:衛星測位サービス”です。”L1S:サブメータ級測位補強サービス”および”L6:センチメータ級測位補強サービス”には対応していません。

## 4.9. USB

表4.9-1 USB制御用ポートと設定値

制御用ポート	direction 設定値
/sys/class/gpio/gpio174	out

表4.9-2 USB Memory デバイスファイル

デバイスファイル
/dev/sda1

CONEXIOBlackBear のUSBポートは起動時には有効になっていません。USBポートにUSB Memoryやその他のUSB機器を接続して使用するためには、以下のコマンドを実行します。

```
# echo 174 > /sys/class/gpio/export
# echo out > /sys/class/gpio/gpio174/direction
```

これにより、USBポートにUSB Memoryを接続した場合は、「表4.9-2 USB Memory デバイスファイル」に記載したように、「/dev/sda1」というデバイスファイルとして認識されるようになります。このデバイスファイルをいずれかのディレクトリにmountすることによって、USB Memoryに対する読み書きができるようになります。

なお「/dev/sda1」の「1」は一つ目のパーティションであることを示しています。複数のパーティションがmicroSD Card上に存在している場合は、それぞれのパーティションが「/dev/sda2」、「/dev/sda3」のようなデバイスファイルとして認識されます。

その他のUSB機器を接続した場合は、認識されるデバイスファイルが接続したUSB機器によって異なりますので注意してください。

## 4.10. SD (Storage)

表4.10-1 microSD Card デバイスファイル

デバイスファイル
/dev/mmcblk3p1

CONEXIOBlackBear のmicroSDスロットは「UHS-I SDR104」のmicroSDXCに対応しており、exFATへも対応しています。(exFATへの対応はFUSEを利用した実装のため、udevを利用してexFATのファイルシステムをmountすることは推奨していません。Transport endpoint not connected エラーが発生します)

microSDスロットに挿入されたmicroSD Cardは、「/dev/mmcblk3p1」というデバイスファイルとして認識されます。このデバイスファイルをいずれかのディレクトリにmountすることによって、microSD Cardに対する読み書きができるようになります。

なお「/dev/mmcblk3p1」の「p1」は一つ目のパーティションであることを示しています。複数のパーティションがmicroSD Card上に存在している場合は、それぞれのパーティションが「/dev/mmcblk3p2」、「/dev/mmcblk3p3」のようなデバイスファイルとして認識されます。

## 4.11. LED

### 4.11.1. USER LED

CONEXIOBlackBear には「図2.4-1 インターフェースレイアウト図」に示すようにトップカバー上面に2色の“USER LED”が設けられています。

この2色LEDを制御するための“File System Node”を「表4.11-1 USER LED」に記載します。

表4.11-1 USER LED

USER LED	Color	File System Node
user1-led	赤	/sys/class/leds/user1-led
user2-led	緑	/sys/class/leds/user2-led

これらのUSER LEDは 表4.11-1 に記載したそれぞれの“File System Node”下のファイルに値を設定することによって制御できます。USER LED共通の制御方法を「表4.11-2 USER LED 制御方法」に記載します。

表4.11-2 USER LED 制御方法

制御用ファイル	設定値	制御結果
brightness	0	LED OFF
	1	LED ON
trigger	none	LED OFF ※triggerの点灯パターンを停止してLEDをOFFにします。
	timer	Default Value delay_on : 500ms delay_off : 500ms ※delay_on、delay_offの値は“trigger”に“timer”をセットした後でなければ設定できません。 ※delay_on、delay_offの値は“trigger”に何か値を設定するとDefault Valueに戻ります。
	heartbeat	Default Value invert : 0 ※心臓の鼓動のような点滅を実行します。 ※invertに1を設定すると点滅のパターンが逆転します。 ※invertの値は“trigger”に何か値を設定するとDefault Valueに戻ります。
	oneshot	Default Value delay_on : 100ms delay_off : 100ms invert : 0 ※shotに何か任意の値を設定すると上記の設定で1回だけ点灯または消灯します。 ※delay_on、delay_off、invert、shotの値は“trigger”に“oneshot”をセットした後でなければ設定できません。 ※invertに1を設定すると点灯のパターンが逆転します。 ※delay_on、delay_off、invertの値は“trigger”に何か値を設定するとDefault Valueに戻ります。

#### 4.11.2. WiFi/Bluetooth LED

CONEXIOBlackBear には「図2.4-1 インターフェースレイアウト図」に示すようにトップカバー上面に2色の“WiFi/Bluetooth LED”が設けられています。

この2色LEDの色を「表4.11-3 WiFi/Bluetooth LED」に記載します。

表4.11-3 WiFi/Bluetooth LED

USER LED	Color
Bluetooth	赤
WiFi	緑

これらのLEDは、それぞれの通信路の接続・通信状態によって点灯/消灯/点滅します。これらのLEDに共通の通信路の状態と点灯/消灯/点滅との関係を「表4.11-4 WiFi/Bluetooth LEDの点灯条件」に記載します。

表4.11-4 WiFi/Bluetooth LEDの点灯条件

LED	通信路の状態
消灯	未接続 ※通信モジュールに電源が入っていても、相手/ネットワークに対して接続されていない場合は点灯しません。
点灯	接続中 ※相手/ネットワークに対して接続中(connected)で、通信路にパケットが流れていない状態の時に点灯します。
点滅	通信中 ※相手との間で通信を実行中で、通信路にパケットが流れている状態の時に点滅します。

#### 4.11.3. LTE 電波強度 LED

CONEXIOBlackBear には「図2.4-1 インターフェースレイアウト図」に示すようにトップカバー上面に緑色の“LTE電波強度LED”が3個設けられています。

これらのLEDはLTEネットワーク接続中に、LTEモジュールのSIM7600から約5秒間隔で読み出した“rsi”値に応じて点灯/消灯します。“rsi”値とLTE電波強度LEDの点灯/消灯との関係を「表4.11-5 LTE電波強度LEDの点灯条件」に記載します。点灯状態に即時性や正確性はありませので、あくまでも目安として考えてください。

表4.11-5 LTE電波強度LEDの点灯条件

rss1-led 	rss2-led 	rss3-led 	rsi値
			-107 dBm 以下
			-105 dBm ~ -87 dBm
			-85 dBm ~ -67 dBm
			-65 dBm 以上

 : 消灯     : 点灯

## 4.12. USER Switch

表4.12-1 USER Switch インプットデバイスファイル

インプットデバイスファイル
/dev/input/event0

CONEXIOBlackBear はユーザーが使い方を設定できるUSER Switchを1個搭載しています。USER Switchの”押す/離す”は、USER Switchのインプットデバイスファイル”/dev/input/event0”に発生するイベントで検出します。

発生するイベントは下記のように”/usr/include/linux/input.h”に定義されています。

```
struct input_event {
    struct timeval time;
    __u16 type;
    __u16 code;
    __s32 value;
};
```

USER Switchを押して離すと以下のようなイベントが発生します。”XXXXXXXXXX.XXXXXX”はUnixtimeです。

例) User Switchを押した時に発生するイベント

```
time XXXXXXXXXXXX.XXXXXX, type 1 (EV_KEY), code 256 (BTN_0), value 1
time XXXXXXXXXXXX.XXXXXX, ----- SYN_REPORT -----
```

例) User Switchを離した時に発生するイベント

```
time XXXXXXXXXXXX.XXXXXX, type 1 (EV_KEY), code 256 (BTN_0), value 0
time XXXXXXXXXXXX.XXXXXX, ----- SYN_REPORT -----
```

上記のイベントは下記のコマンドで確認ができます。

```
# hexdump /dev/input/event0
```

また CONEXIOBlackBear には”evtest”が実装されているので、このコマンドを実行することによっても確認ができます。(上記の発生イベント例はevtestコマンド実行結果の一部です)

```
# evtest /dev/input/event0
```

## 4.13. CPU 温度

「図2.4-1 インターフェースレイアウト図」に記載した外部インターフェースではありませんが、CONEXIOBlackBear は搭載しているCPUの温度を測定することができますので、参考として記載しておきます。

CPU温度情報を取得するためには以下のコマンドを実行します。コマンドの次の行は実行結果の例です。

```
# cat /sys/class/thermal/thermal_zone0/temp
39315
```

上記実行結果例の”39315”は、”39.315°C”を表しています。

## 4.14. SMS

「3.5.2. Resume」にも記載したように、SMS送受信を行うためには、SMS通信が可能な通信契約が必要ですので注意してください。そして「4.2. LTE」に記載したLTE接続設定が完了してLTEがすでに接続されている場合は、SMSの受信は自動的に実行されます。

以下にModem Managerを使用したSMS操作方法の主なものを記載します。

### [SMSのリスト表示]

```
# mmcli -m 0 --messaging-list-sms
Found 3 SMS messages:
  /org/freedesktop/ModemManager1/SMS/0 (received)
  /org/freedesktop/ModemManager1/SMS/1 (received)
  /org/freedesktop/ModemManager1/SMS/2 (sent)
```

} 応答例

### [SMSの内容表示]

```
# mmcli -s [SMS Number]
```

※"[SMS Number]"は、内容を表示したいSMSの番号です。(上記リスト表示例の0~2)

※"Content"として送信元電話番号およびSMSの内容(本文テキスト)が、"Properties"として関連情報が表示されます。

### [SMSの作成]

```
# mmcli -m 0 --messaging-create-sms="number=[Phone Number],text=' [SMS Message]'"
Successfully created new SMS:
  /org/freedesktop/ModemManager1/SMS/[SMS Number] (unknown)
```

} 応答例

※"[Phone Number]"は、送信先の電話番号です。

※"[SMS Message]"は、送信するSMSの内容(本文テキスト)です。

※SMS作成に成功した時には上記のメッセージが表示されます。

"[SMS Number]"は作成したSMSの番号です。この番号はSMSの送信時に使用します。

### [SMSの送信]

```
# mmcli -s [SMS Number] --send
```

※"[SMS Number]"は送信するSMSの番号です。この番号はSMS作成時に表示されたものを使用します。

### [SMSの削除]

```
# mmcli -m 0 --messaging-delete-sms [SMS Number]
```

※"[SMS Number]"は削除するSMSの番号です。

## 5. OS 更新 (初期化)

CONEXIOBlackBear 本体のOSを更新する方法には、大きく分けて2種類の方法があります。一つは「差分更新」、もう一つは「クリーンインストール」です。

本章では、microSD Cardを使用して CONEXIOBlackBear のOS全体を入れ替える(初期化する)「クリーンインストール」の方法について記載します。

「差分更新」については「6. OS差分更新」を参照してください。また、他のソフトウェアやファームウェアの更新方法については別紙「簡易プログラミングマニュアル」を参照してください。

### 5.1. OS の Version 確認方法

CONEXIOBlackBear に搭載されているOSのversion情報は"/etc/os-release"に保存されています。保存されているversion情報の例を下記に記載します。

```
# cat /etc/os-release
ID="fsl-imx-fb"
NAME="MSI MS5776 Release Distro"
VERSION="4.9.88-2.0.0-1.1.5 (rocko)"
VERSION_ID="4.9.88-2.0.0-1.1.5"
PRETTY_NAME="MSI MS5776 Release Distro 4.9.88-2.0.0-1.1.5 (rocko)"
#
```

上記のversion情報例では"VERSION"行などの末尾の"1.1.5"が、リリースごとに更新されるversion No.です。異なるリリースversionのOSへと変更する場合などは、この箇所を確認してください。

"VERSION"行の詳細内容は下記の通りです。(上記version情報例を用いて記載します)

```
"4.9.88" : Linux Kernel Version
"4.9.88-2.0.0" : NXP BSP(Board Support Package) Version
"1.1.5" : CONEXIOBlackBear Firmware Release Version
"(rocko)" : Yocto Version - rocko (2.4.2)
```

万が一、CONEXIOBlackBearの動作に何か不具合が発生した場合の解析依頼には、必ずOSのversion情報を合わせて伝えてください。

また「5.2. microSD Cardを使用したOS更新」に記載したOS更新を実行すると、同時に電源管理プロセッサであるMCUのファームウェアも(更新内容がある場合には)更新されます。MCU ファームウェアは、この方法でのみ更新可能です。MCU ファームウェアの更新が行われた場合は、その旨ユーザーサポートサイトに掲載します。

MCU ファームウェアのversion情報は"/proc/ms5776\_mcu\_ver/mcu\_version"に保存されています。保存されているversion情報の例を下記に記載します。

```
# cat /proc/ms5776_mcu_ver/mcu_version
ms5776_MCU_AP_Ver=V0.20
```

上記のversion情報例では、"V0.20"の部分がMCUのversion No.です。

OSおよびMCUのversion情報は、OS更新後に再起動を行った後で確認してください。

## 5.2. microSD Card を使用した OS 更新

本章ではmicroSD Cardを使用してOSを更新する手順を記載します。

### 5.2.1. “SD\_burn\_EMMC”のダウンロード

下記URLのサーバーから、更新を行うversionの“SD\_burn\_EMMC-VXXX.zip” ファイルをダウンロードします。(XXXはリリースごとに更新されるversion No.です → 例:OS V1.1.5の場合はSD\_burn\_EMMC-V115.zip)

Server URL: <https://bb.conexio-iot.jp/support> (※CONEXIOBlackBear ユーザーサポートサイトです)

ダウンロードした“SD\_burn\_EMMC”ファイルは解凍して、Windows PCの仮想マシン上やLinux PC上のUbuntuに保存してください。(Install Media作成操作はUbuntu 18.04上での操作を想定しています)

### 5.2.2. Install Media (microSD Card) 作成手順

Install Mediaとして使用するmicroSD CardをPCに挿入します。microSD Cardの容量は4GBのものでは作成できませんので、8GB以上のものを使用してください。

挿入後“lsblk”コマンド等でmicroSD Cardのdevice名を確認しておきます。ここでは例として“/dev/sdX”と記載しておきます。microSD Cardを挿入した時にmountされている場合は、superuser権限でumountしておきます。

Ubuntu上に保存した“SD\_burn\_EMMC”ディレクトリに移動します。

```
# cd SD_burn_EMMC
```

“SD\_burn\_EMMC”ディレクトリ中に有るシェルスクリプト“mkSD.sh”をsuperuser権限で実行します。

```
# sudo ./mkSD.sh /dev/sdX
```

“mkSD.sh”の実行が完了したら、一旦microSD CardをPCから取り出した後、再度PCへ挿入します。

挿入後再度“lsblk”コマンド等でmicroSD Cardのdevice名を確認しておきます。ここでも例として“/dev/sdX”と記載しておきます。この時もmicroSD Cardがmountされている場合は、superuser権限でumountしておきます。

“SD\_burn\_EMMC”ディレクトリ中に有るシェルスクリプト“updateSD.sh”をsuperuser権限で実行します。

```
# sudo ./updateSD.sh /dev/sdX
```

※途中で“Proceed anyway? (y,N)”が表示された場合は、“y”を入力します。

“updateSD.sh”の実行が完了したらInstall Mediaの作成は完了です。microSD CardをPCから取り出してください。

### 5.2.3. CONEXIOBlackBear 本体での OS 更新手順

作成したInstall MediaであるmicroSD Cardを CONEXIOBlackBear のmicroSDスロットへ挿入します。

microSD Card挿入後、USER SWを押しながら CONEXIOBlackBear の電源を投入します。これによりmicroSD Cardからの起動が開始されます。

電源投入後、全てのLEDが点灯した後、電源LED以外の全てのLEDが消灯して、電源LEDが点滅を開始したらUSER SWから指を離します。

microSD Cardからの起動が完了してOS更新のためのeMMC書き換えが始まると全てのLEDが点灯します。そのまましばらくOSの更新が完了するのを待ちます。

OSの更新が完了すると、電源LEDが点灯したままで電源LED以外の全てのLEDが点滅を開始します。この状態になったら電源をOFFにすることができます。**OS V1.0.0以降の本手順による更新には、約11分程度かかりません。**OSの更新が完了する前に誤って電源をOFFにすることが無いよう注意してください。

OSの更新完了時にCONEXIO BlackBear のLinux コンソールには、「microSD Cardを抜いてOSを再起動してください」という旨のメッセージが表示されます。

再起動後には必要に応じて「5.1. OSのVersion確認方法」に記載した方法でOSのversion情報を確認してください。

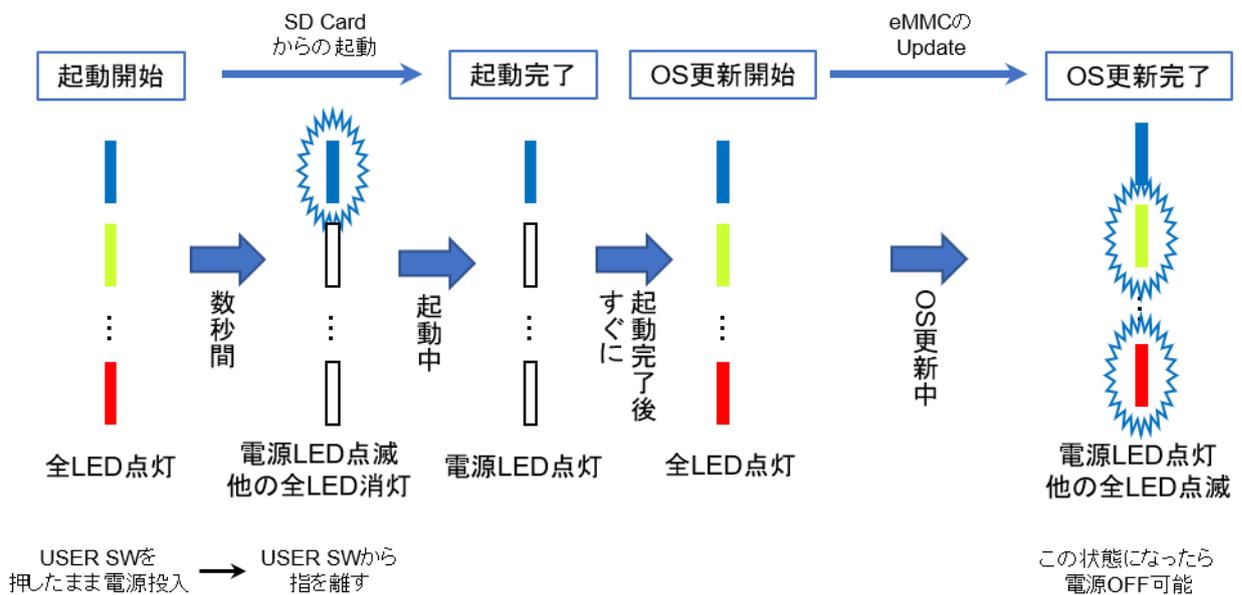


図5.2-1 OS更新時のLED表示

## 6. OS 差分更新

### 6.1. 概要

本章では、「5. OS更新（初期化）」に記載した「OS全体を入れ替える」更新方法とは異なり、「OSの更新された部分のみを変更する」方法である「差分更新」について記載します。

この方法の利点は「クリーンインストール」の場合とは異なり、すでにユーザーアプリケーション等が搭載された後でも、そのアプリケーションが削除されることなくOSの更新を実施できることです。

ただし差分更新用OSのパッケージは、特定のOSバージョンから別の特定のOSバージョンへという特定の組み合わせに対する更新にしか対応しておりませんので、対応バージョンに関する注意が必要です。

以下に、差分更新用OSパッケージを使用して「差分更新」を実施する更新手順を記載します。

### 6.2. 更新手順

#### 6.2.1. 差分更新用 OS パッケージの取得

差分更新用OSパッケージは下記URLの CONEXIOBlackBear サポートサイトに掲載しています。

Server URL: <https://bb.conexio-iot.jp/support>

上記サイトから更新したいバージョンの組み合わせの差分更新用OSパッケージをダウンロードします。例えば OS V1.1.5からOS V1.1.9へと更新する差分更新用OSパッケージの名称は下記の通りです。

OS V1.1.5 → OS V1.1.9 差分更新用OSパッケージ : diffOS-V115V119.zip

差分更新用OSパッケージはサーバーからのダウンロード後、CONEXIOBlackBear の"/tmp"等へ転送してください。転送はネットワーク経由でも、microSDを使用しても結構です。

CONEXIOBlackBearから直接"wget"コマンド等を使用して差分更新用OSパッケージをダウンロードする場合は、同名のファイルを下記URLのディレクトリにも保存してありますので利用してください。

URL: <https://repository.blackbear.conexiot.jp/ms5776/software/diffOS/>

OSには別紙「簡易プログラミングマニュアル」の「eMMC パーティション構成」に記載したように、「A block」と「B block」という2つのblockがあります。差分更新用OSパッケージは、更新したい方のblockへ置いてください。

以下に、OS V1.1.5からOS V1.1.9へと差分更新を実行する場合の例を用いて、差分更新処理の操作手順を記載します。

#### 6.2.2. OS 差分更新の実行

まず更新対象の差分更新用OSパッケージ(以下の例では"diffOS-V115V119.zip")が置いてあるディレクトリ(以下の例では"/tmp")へ移動します。

```
# cd /tmp
```

unzip コマンドを実行して「diffOS-V115V119.zip」を解凍し、インストール用スクリプト「install\_diffOS.sh」およびパッケージ群を取り出します。

```
# unzip diffOS-V115V119.zip
```

取り出されたインストール用スクリプト「install\_diffOS.sh」およびパッケージ群が格納されているディレクトリ diffOS-V115V119へ移動します。

```
# cd diffOS-V115V119
```

インストール用スクリプト「install\_diffOS.sh」を実行して、OS差分更新処理を開始します。

```
# ./install_diffOS.sh
Are you sure you want to update MS5776 OS (1.1.5->1.1.9)? (y/N) :
```

スクリプトを実行すると、上記のようにOSの更新を実施するか否かを問われます。実行する場合は“y”または“Y”を入力してください。OSの更新をやめる場合はそれ以外の文字を入力してください。“y”または“Y”以外の文字を入力した場合は“abort.”が表示されて、更新処理は実行されません。

```
# ./ install_diffOS.sh
Are you sure you want to update MS5776 OS (1.1.5->1.1.9)? (y/N) : N
abort.
```

“y”または“Y”を入力した場合は、OS差分更新処理が開始されます。

```
Are you sure you want to update MS5776 OS (1.1.5->1.1.9)? (y/N) : y
2021-03-16 10:17:20 MS5776 OS update started.
.
.
```

もしOS差分更新を実行しようとしている対象のOSバージョン(CONEXIOBlackBear にすでにインストールされている現状のOSバージョン)が、差分更新用OSパッケージが対象としているOSのバージョン(この例の場合はOS V1.1.5)と異なる場合は、以下のメッセージが出て更新処理は実行されません。インストールされているOSバージョンおよび差分更新用OSパッケージが対象としているOSバージョンを確認してください。

```
Are you sure you want to update MS5776 OS (1.1.5->1.1.9)? (y/N) : y
Exit because it is not the target version.
abort.
```

OS差分更新処理が実行されている間は処理内容のメッセージが出力され、処理が成功した場合は下記の成功した旨のメッセージが出力されて、約10秒間のカウントダウン完了後、自動的に再起動が実行されます。

```
2021-03-16 10:17:50 MS5776 OS update success.
After the countdown, reboot the MS5776 : 10
↓ (countdownが実行されて残り秒数が減っていきます)
After the countdown, reboot the MS5776 : 0
#
↓
(go-sys-rebootが自動的に実行されます)
```

再起動が完了した時点で、OSの差分更新処理は完了です。

再起動が実行されるため、リモート接続でOS差分更新処理を実行していた場合は、接続が一旦切れてしまいますが、CONEXIOBlackBear にOS差分更新処理前に設定されていた接続情報はそのまま残っているため、再起動に必要な時間が経過した後に再度接続することができます。

ただし「4.1.2. 固定IP接続」に記載したように、CONEXIOBlackBear のEthernetコネクションとして固定IPアドレスを設定してEthernet経由でCONEXIOBlackBearにアクセスする場合には、Defaultで使用しているコネクション名の‘Wired connection 1’を使用することは推奨していません。再起動時に不具合が発生する場合がありますため、CONEXIOBlackBear のEthernetコネクションとして固定IPを使用する場合は、別のコネクション名を使用してください。

### 6.2.3. OS 差分更新後の version 情報

OS差分更新を実行した後、OSのversion情報を確認すると、クリーンインストールを実行した場合とは異なり、下記のように“PRETTY\_NAME”には更新前のversion No.が表示されます。その他の項目はクリーンインストールを実行した場合と同様に更新後のversion No.が表示されます。

```
# cat /etc/os-release
ID="fsl-imx-fb"
NAME="MSI MS5776 Release Distro"
VERSION="4.9.88-2.0.0-1.1.9 (rocko)"
VERSION_ID="4.9.88-2.0.0-1.1.9"
PRETTY_NAME="MSI MS5776 Release Distro 4.9.88-2.0.0-1.1.5 (rocko)"
#
```

これにより、クリーンインストールされたOSなのか、差分更新されたOSなのかを識別することができます。万が一、CONEXIOBlackBearの動作に何か不具合が発生した場合の解析依頼には、必ずOSのversion情報を合わせて伝えてください。

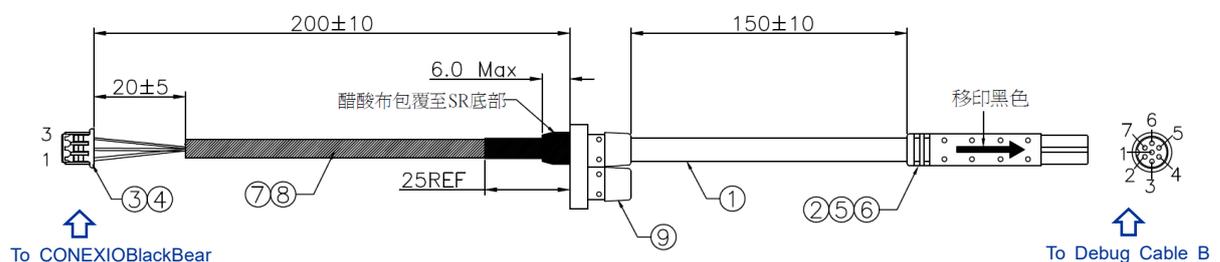
## 7. 開発用ツール

### 7.1. Debug Cable

CONEXIOBlackBear の初期設定を行なったり、CONEXIOBlackBear 上でソフトウェア開発やソフトウェアの動作確認を行なったりする場合のために、“Debug Cable”を提供しています。

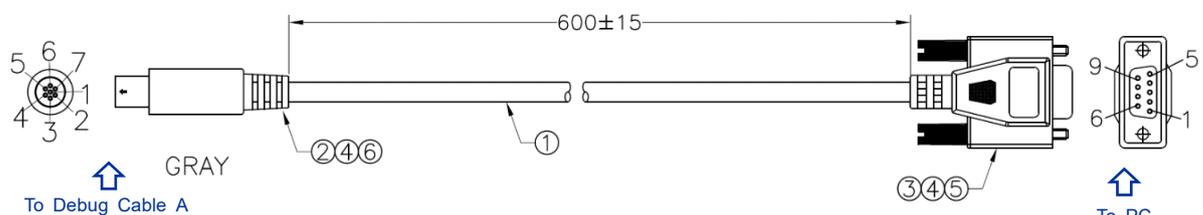
この“Debug Cable”は、標準的なセットには添付されておりませんので、必要な場合は別途ご購入先へ問い合わせをしてください。

“Debug Cable”は 図7.1-1 Debug Cable A および 図7.1-2 Debug Cable B に記載した2本のケーブルで構成されています。2本のケーブルは 図7.1-1 Debug Cable A 右端のコネクタと 図7.1-2 Debug Cable B 左端のコネクタを接続して使用します。“Debug Cable”を用いてPCにCONEXIOBlackBear を接続すると、PCから CONEXIOBlackBear のLinux コンソールを利用できるようになります。



①	UL20276 #28*3COAX+#28*2C+AE OD:5.0mm PVC BLACK(-40°C~105°C) L=400mm(RoHS)	1pcs
②	7PIN FEMALE SOLDER TYPE PBT BLACK 端子銀メッキ (SPD-07F1-01)(RoHS)	1pcs
③	MOLEX 51021 PH:1.25mm PHOSPHOR BRONZE TIN-PLATED または同等品(RoHS)	3pcs
④	MOLEX 51021 PH:1.25mm 1*3P NYLON66 WHITE または同等品(RoHS)	1pcs
⑤	INNER MOLD:PP 透明(RoHS)	0.9g
⑥	OVER MOLD :PVC 30P GRAY(427C)(-40°C~105°C) (RoHS)	3.2g
⑦	黒色アセテート布:25*180mm	1pcs
⑧	黒色アセテート布:25*30mm	1pcs
⑨	SR MOLD:PVC 45P BLACK(-40°C~105°C) (RoHS)	1.7g

図7.1-1 Debug Cable A



①	NOUL 28AWG*3C+DRAIN+AL PVC BLACK OD:3.0mm(-40°C~85°C) L=655mm(RoHS)	1pcs
②	MINI 7P MALE SOLDER TYPE PBT BLACK 端子銀メッキ (RoHS)	1pcs
③	DB 9P/F SOLDER TYPE, CONTACT:BRASS G/F, SHELL NI-PLATED, PLASTIC:PBT BLACK(RoHS)	1pcs
④	INNER MOLD:LOW DENSITY PE(RoHS)	7.2g
⑤	OVER MOLD:PVC 45P BLACK(-40°C~85°C) (RoHS)	8.6g
⑥	OVER MOLD:PVC 45P GRAY(427C)	5.5g

図7.1-2 Debug Cable B

## 7.2. Debug Port

CONEXIOBlackBear 本体への接続は、「図7.2-1 本体へのDebug Cable接続」に記載したようにSD-Doorを開いて、microSDスロットとmicroSIMスロットとの間に有る”Debug Port”に対して、図7.1-1 Debug Cable A 左端の3Pコネクタを接続します。

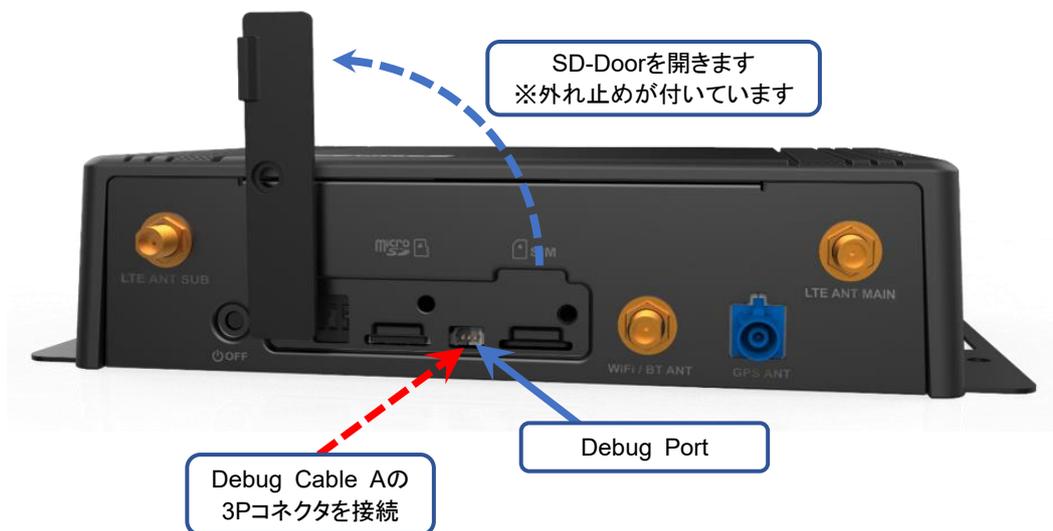


図7.2-1 本体へのDebug Cable接続

PCへの接続は、図7.1-2 Debug Cable B 右端のRS232Cコネクタ(D-sub 9pin メス端子)をPCのシリアルポートに接続するか、市販の”USB – Serial 変換ケーブル”を使用してPCのUSBポートに接続します。

上記のような接続を行うことにより、PC上のTera Termのようなターミナルエミュレータを使用してLinuxのコンソール画面から CONEXIOBlackBear を操作できます。

ターミナルエミュレータのシリアル通信設定を「表7.2-1 ターミナルエミュレータのシリアル通信設定」に記載します。

表7.2-1 ターミナルエミュレータのシリアル通信設定

設定項目	設定値
通信速度	115,200 bps
データ長	8 bits
パリティ・ビット	none
ストップ・ビット	1 bit
フロー制御	none

ネットワーク設定が完了し、CONEXIOBlackBear のIP Addressがわかっている場合は、ネットワーク経由でsshを使用してLoginすることも可能です。

## 8. 注意事項

### 8.1. 安全に使用するための取扱注意事項

マニュアル、関連資料をご確認いただき、使用上の注意を守って正しく使用してください。

以下の区分は、誤った使用をした場合に生じる危害や損傷の程度を説明しています。

 <b>危険</b>	取り扱いを誤った場合、「死亡又は重傷を負う危険が切迫して想定される」内容です。
 <b>警告</b>	取り扱いを誤った場合、「死亡又は重傷を負う危険が想定される」内容です。
 <b>注意</b>	取り扱いを誤った場合、「傷害や物的損害を負う危険が想定される」内容です。

### 危険

- CONEXIOBlackBearを高温になる場所、熱のこもりやすい場所、直射日光の当たる場所などで使用、保管、放置しないでください。火災、故障、感電等の原因となります。
- CONEXIOBlackBearの防塵・防水性能はIP40です。屋内、自動車内等で、ほこりが少なく、雨水等がかからない環境でご使用ください。屋外で使用される場合は防塵・防水性を十分に確保した上でご使用ください。水、ほこり、湿気、油等の多い場所での使用は火災、故障、感電等の原因となります。
- 高い信頼性、安全性(医療機器、航空機器等)を要求される使用を意図しておりません。これらの用途で使用された際に生じた、いかなる損害に対しても弊社は責任を負いかねます。
- 一般電子機器用に製造された材料を使用しています。外来ノイズやサージ等の影響で誤作動や故障が発生する可能性があります。万一の誤作動や故障に備えて、生命・人体・財産が侵害されることのないように装置としての安全設計を行い、十分な措置を講じた上で使用してください。

### 警告

- 落下等の強い衝撃をかけないでください。衝撃により破損した状態で使用した場合、火災、故障、感電等の原因になる場合があります。
- CONEXIOBlackBearには無線機能が搭載されており医療機器等の周辺機器へ影響を及ぼす可能性があります。そのような機器からは十分に離れた場所に取付けて使用してください。
- CONEXIOBlackBearを改造することは、火災、故障、感電等の原因になる場合があります。
- CONEXIOBlackBearにはコイン型二酸化マンガンリチウム電池が組み込まれています。電池の+極と-極をショートさせると、液漏、発熱、破裂、発火の恐れがあります。

## 注意

- CONEXIOBlackBearは取付けた状態での使用を想定しています。取付けていない状態や手に持った使用はやめてください。
- CONEXIOBlackBearを連続して使用した場合、放熱のため本製品の筐体が高温になることがあります。
- CONEXIOBlackBearのOSはLinuxを使用しています。個別に明示されている場合を除き、オープンソースとしてソースコードが提供されています。再配布等の権利については、各ソースコードに記載されたライセンス形態にしたがって、お客様の責任において行使してください。
- プリインストールソフトウェア以外のソフトウェアに関しては、追加される方の責任のもと、使用用途、信頼性等を十分に検証の上、追加してください。追加ソフトウェアに起因する不具合は、いかなる損害に対しても弊社は責任を負いかねます。
- アンテナ、各種ケーブル、microSIMの抜き挿しは電源が入った状態で行わないでください。故障、破損の原因となります。
- 外部インターフェース(アンテナ端子、コネクタ、スロット、スイッチ等)に無理な力を加えないでください。故障、破損の原因となります。
- 溶剤を含む化学薬品や洗剤を使用して清掃をしないでください。

## その他の注意

- CONEXIOBlackBearに貼りつけられた銘板、ラベル、シール等を剥がしたり、傷つけたりしないでください。製品保証の対象外になる場合があります。

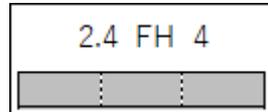
### 8.2. 電波に関する注意事項

CONEXIOBlackBearが使用している周波数帯では、電子レンジ等の産業・科学・医療用機器のほか工場の製造ライン等で使用されている移動体識別用の構内無線局及び特定小電力無線局並びにアマチュア無線局が運用されています。

- CONEXIOBlackBearを使用する前に、近くで移動体識別用の構内無線局及び特定小電力無線局並びにアマチュア無線局が運用されていないことを確認してください。
- 万一、CONEXIOBlackBearから移動体識別用の構内無線局に対して有害な電波干渉の事例が発生した場合には、速やかに使用を中止してください。
- その他、CONEXIOBlackBearから移動体識別用の特定小電力無線局あるいはアマチュア無線局に対する障害となる電波干渉の事例が発生した場合には、速やかに使用を中止してください。

### 8.2.1. Bluetooth の周波数帯

Bluetooth機能が使用する周波数帯は以下の通りです。



2.4 : 2400MHz帯を利用する無線設備であることを示します。

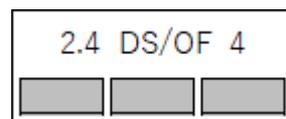
FH : 変調方式がFH-SSであることを示します。

4 : 想定される与干渉距離が40m以下であることを示します。

 : 2400MHz~2483.5MHzの帯域を利用し、かつ移動体識別装置の帯域を回避不可であることを示します。

### 8.2.2. WiFi(無線 LAN)の周波数帯(2.4GHz 帯)

WiFi(無線LAN)機能が使用する周波数帯は以下の通りです。



2.4 : 2400MHz帯を利用する無線設備であることを示します。

DS/OF : 変調方式がDS-SS、変調方式がOFDMであることを示します。

4 : 想定される与干渉距離が40m以下であることを示します。

 : 2400MHz~2483.5MHzの帯域を利用し、かつ移動体識別装置の帯域を回避可能であることを示します。

### 8.2.3. WiFi(無線 LAN)の周波数帯(5GHz 帯)

"SIM7600JC-H"を搭載した国内向けモデルの場合、5GHz帯で使用可能なCHは以下の通りです。

5.2GHz帯(W52:36、40、44、48ch)

5.3GHz帯(W53:52、56、60、64ch)

5.6GHz帯(W56:100、104、108、112、116、120、124、128、132、136、140ch)

※5GHz帯の屋外での使用は電波法の定めにより制限されています。詳しくは、総務省のWEBサイトをご確認ください。

※WiFiの5GHz アクセスポイント設定を使用する場合、CONEXIOBlackBearで使用可能な周波数帯は、5.2GHz帯(W52:36、40、44、48ch)のみに設定されていますので注意してください。

※使用する周波数帯を変更する場合は、必ず日本国の関連法規を遵守してお客様ご自身の責任において変更してください。誤った設定を行った場合、法律により罰せられることが有ります。

### 8.3. 電波法 および 電気通信事業法対応

CONEXIOBlackBear は、LTE Moduleとして国内向けモデルでは”SIM7600JC-H”を搭載しています。またWiFi/Bluetooth Moduleとして”WM-BAC-BM25”を搭載しています。これらの通信Moduleは、日本国内での使用のために必要となる”電波法に基づく技術基準適合証明”および、”電気通信事業法に基づく技術基準適合認定”を受けています。

これらの通信Moduleの認証番号を「表8.3-1 技術基準適合証明/認定情報」に記載します。

表8.3-1 技術基準適合証明/認定情報

技適項目	Module	認証番号
電波法に基づく技術基準適合証明 (工事設計認証)	BM25	020-180109
	SIM7600	003-180005
電気通信事業法に基づく技術基準適合認定 (設計認証)	BM25	D180048020
	SIM7600	DF180082003

また CONEXIOBlackBear は端末として、”電気通信事業法に基づくIoT機器のセキュリティ基準に係る技術基準適合認定”を受けています。

この認証番号を「表8.3-2 セキュリティ基準適合認定情報」に記載します。

表8.3-2 セキュリティ基準適合認定情報

技適項目	機器名	認証番号
電気通信事業法に基づくセキュリティ基準適合認定	CONEXIOBlackBear	D210136001

「表8.3-1 技術基準適合証明/認定情報」および「表8.3-2 セキュリティ基準適合認定情報」に記載した認証番号などは、CONEXIOBlackBear の底面ラベルに「図8.3-1 技術基準適合証明/認定情報 - 底面ラベル」のよう

に記載してありますので、必要に応じて確認してください。

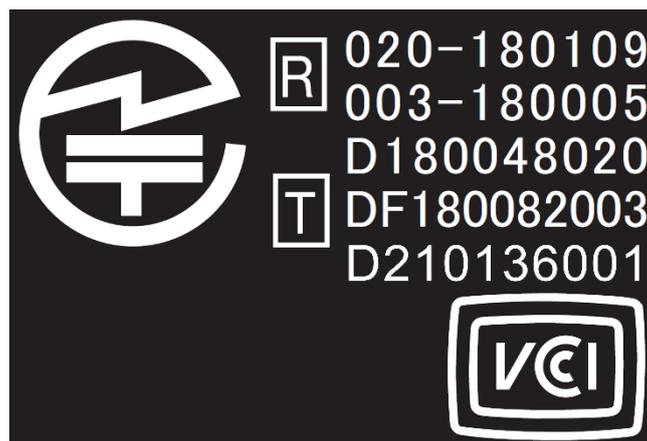


図8.3-1 技術基準適合証明/認定情報 - 底面ラベル

## 8.4. オープンソースソフトウェアに関する注意事項

CONEXIOBlackBear の工場出荷時に標準搭載されているソフトウェアには「表2.6-1 OS および 主なPre-installedソフトウェア」に掲載したものをはじめとするオープンソースソフトウェア(OSS)が含まれています。

これらのOSSには、それぞれの著作権者によってライセンス条件が定められており、CONEXIOBlackBear を購入していただいたお客様にも、定められたライセンス条件を遵守する義務が有ります。

これらのOSSのライセンスに関する詳細については、CONEXIOBlackBear サポートサイトの「Yocto Project - Board Support Package (BSP)」に記載してある注意事項を参照してください。

ソースコードを入手し、それぞれのOSSで定められたライセンス条件に従って、改変、再配布などを実施する場合は、お客様の自己責任で実施してください。

これらのOSSは現状有姿(AS IS)にて提供され、それぞれのOSS自体に対してはいかなる保証も提供されません。またOSSソースコードの内容そのものに関しては、いかなる技術サポートも提供しません。

なお、お客様が CONEXIOBlackBear の工場出荷時に標準搭載されているソフトウェアを改変した場合は、製品保証の対象外となります。改変する場合は全て自己責任にて実施してください。

オリジナルのi.MX BSPの入手に関しては、下記リンクのNXPサイトを参照願います。

NXPサイトURL:

<https://www.nxp.com/design/i-mx-developer-resources/i-mx-software-and-development-tools:IMX-SW>

## 8.5. 海外での使用について

CONEXIOBlackBear には日本国内の“技適”や“PSE”だけではなく、世界各国・地域の電波関連法規や製品安全に関する認証を取得したモデルが有ります。対象国・地域の周波数に対応したLTE Moduleが搭載されており、対象となるそれぞれの国・地域の認証を取得してあります。

本書では“SIM7600JC-H”を搭載した国内向けモデルを念頭に置いた記載をしておりますが、LTE Moduleの対応周波数、WiFiの対応channel以外の CONEXIOBlackBear の構成は、各国・地域対応モデル間で同一です。したがって、日本国内で開発したシステムを再開発することなくそのまま海外展開することが可能ですが、必ず使用する国・地域での認証を取得したモデルを使用してください。

認証取得済の国・地域に関する最新情報については下記URLの情報を確認してください。

URL: [https://conexio-iot.jp/serviceproduct/blackbear\\_faq.html](https://conexio-iot.jp/serviceproduct/blackbear_faq.html)

使用する国・地域に対応した認証取得済モデル以外を使用した場合の製品保証は一切できません。また使用した国・地域において法律により罰せられることがあります。

CONEXIOBlackBear を輸出する場合は、日本国および関連する国・地域の輸出入関連法規を遵守し、必要な輸出入手続きを取ってください。

## 8.6. 保証事項

CONEXIOBlackBear に添付または下記URLに記載した「CONEXIOBlackBear 保証規定」にしたがって保証を行います。

URL: <https://conexio-iot.jp/serviceproduct/blackbear.html>

保証範囲、保証期間がございますので内容を確認して注意してください。

---

### ※注意※

CONEXIOBlackBear底面の4か所のネジ穴のうち1か所に「図8.6-1 TAMPER SEAL」に記載したシールが貼られています。このシールはCONEXIOBlackBearの分解・改造等が不正に行われることを避けるためのものです。このシールを剥がしたり破ったりするなど破損させた場合は、保証期間内でも保証対象外となりますので注意してください。

CONEXIOBlackBearの分解が必要になった場合は、「9. お問い合わせ先」に相談してください。



図8.6-1 TAMPER SEAL

---

## 9. お問い合わせ先

コネクシオ株式会社  
法人スマートサービス部門  
IoTソリューション部 IoT営業課

〒105-6907 東京都港区虎ノ門4-1-1 神谷町トラストタワー7階

**TEL.03-5408-3177**