CONEXIO
人をつなぐ、価値をつなぐ

CONEXIOBlackBear Instruction Manual

Ver. 2.0.7

October 25, 2022

CONEXIO Corporation

Revision History

Version	Date	Updates
1.0.0	Oct.10,2019	Published the first edition.
2.0.0	Sep.1,2020	Made a wide range of additions and corrections in consideration of earlier inquiries.
2.0.1	Oct.1,2020	Corrected the description in “Certification” and “CAN I/F” in Table of Specifications. Added description about CAN specifications. Corrected the description of the Resume function. Corrected the description of the GNSS function.
2.0.2	Nov.4,2020	Corrected the description of the connectors of included antennas.
2.0.3	Jan.22,2021	Added notes on mounting the exFAT file system.
2.0.4	Mar.26,2021	Added description about LTE connection configuration for PAP authentication. Added description about incremental OS update.
2.0.5	Jan.11,2022	Made a small revision of wording. Corrected the table of preinstalled software. Added description about the colors of included cables. Added description about functions available with CAN I/F. Added description about how to use the DI/DO. Added description about version information shown after incremental updates. Added description about security regulations conformity accreditation. Changed the contact information due to the relocation of the headquarters.
2.0.6	Mar.25,2022	Corrected the table of preinstalled software. Corrected the description of power turn-OFF/reboot. Changed the structure of LTE and SMS chapters. Added description about SIM handling. Deleted the description of version information shown after incremental updates.
2.0.7	Oct.25,2022	Corrected reference errors in text. Added description about the cancellation of automatic Ethernet connection. Added notes on the procedure for Install Media(microSD card) creation. Changed the support website URL. Added description about contact information for technical inquiries.

[Contents of the CONEXIOBlackBear Instruction Manual]

[Table of Figures]

----

1. Product Overview

1.1. Introduction

This Instruction Manual is for the Edge Computing Gateway CONEXIOBlackBear. The Instruction Manual shows the specifications of the CONEXIOBlackBear and describes how to use it, as well as the I/Fs installed on it. To learn how to develop software to be implemented in the CONEXIOBlackBear, see the “Simplified Programming Manual”.
*The “Simplified Programming Manual” is available on the CONEXIO IoT support website.

The CONEXIOBlackBear edge computing gateway enables the efficient use of device data and the creation of new value by using the Internet to connect devices such as equipment and sensors installed in different environments to the IoT cloud or other such servers, thereby enabling support for diverse solutions as shown in “Fig. 1.1-1 IoT Gateway Concept”.

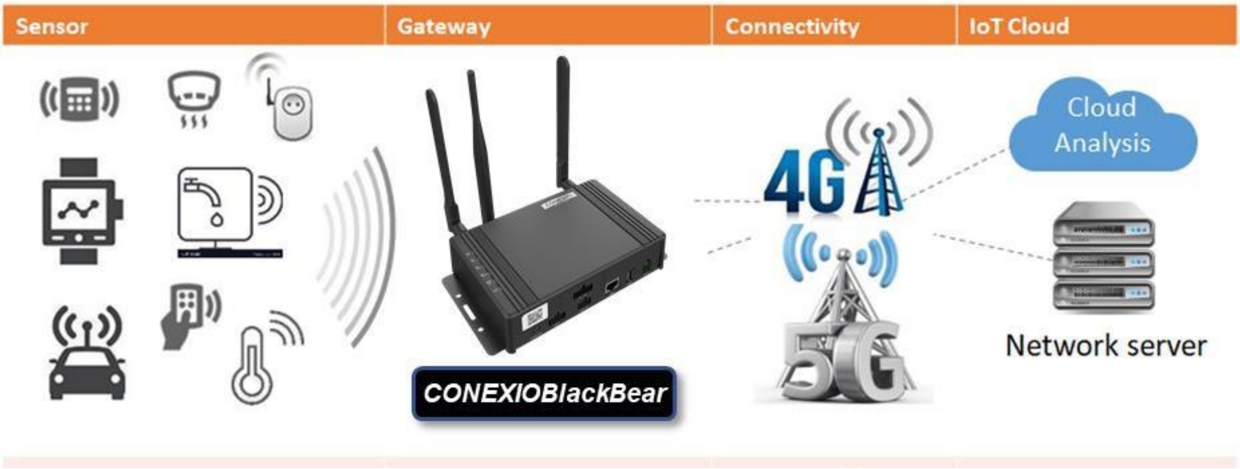


Fig. 1.1-1 IoT Gateway Concept

The CONEXIOBlackBear has the following features.

- **Enhanced edge computing capability**: The CONEXIOBlackBear is an edge computing gateway enhanced in processing capability with a quad-core CPU and GPU. These features make it useful for application in areas where high performance and immediacy are required. For example, it enables onsite edge AI processing without using the cloud.

- **Global support**: Having been accredited as conforming to radio-related regulations and product safety in the relevant countries and regions in addition to Japan’s “Technical Conformity Mark” and “PSE”, the CONEXIOBlackBear can be deployed as is overseas without any redevelopment of the system, which was developed in Japan. (See also “8.5. Overseas Use”.)

- **Automotive support**: Featuring the robustness, operating temperature, and operating voltage required for automotive use, the CONEXIOBlackBear comes standard with two CAN interface channels and a GNSS so that it can be retrofitted to special vehicles or construction machines, allowing them to be operated using IoT technology. This enables the operation analysis and remote maintenance of vehicles.

- **Diverse I/F features**: The CONEXIOBlackBear, which comes standard with Ethernet, LTE, WiFi, Bluetooth, RS485/RS422/RS232C (selectable), CAN (2 channels), DI/DO, and GNSS, and also allows you to add optional Wi-SUN, EnOcean, etc. (TBD) modules, can be connected to a variety of devices and sensors.

As such, the CONEXIOBlackBear can be applied to a wide range of situations around the world. Examples of solutions using the CONEXIOBlackBear are shown in “Fig. 1.1-2 Examples of Solutions Using the CONEXIOBlackBear”.

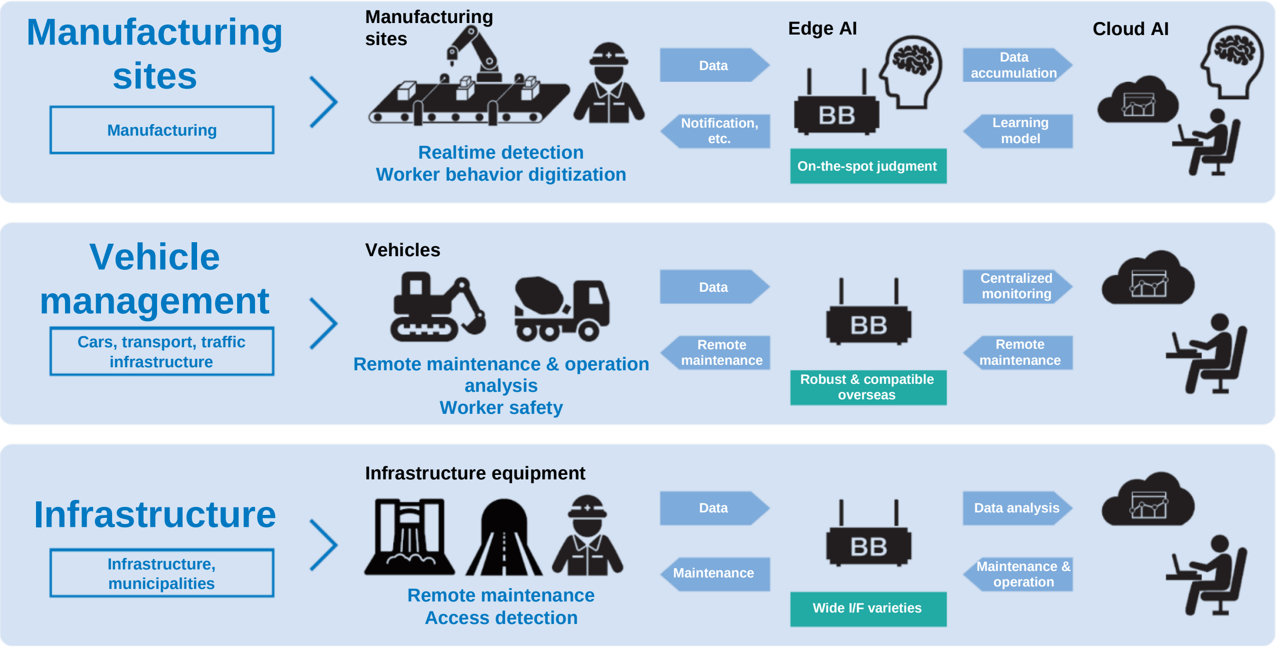


Fig. 1.1-2 Examples of Solutions Using the CONEXIOBlackBear

1.2. External Product Appearance

The external appearance of the CONEXIOBlackBear are shown in “Fig. 1.2-1 External Appearance of the CONEXIOBlackBear - Top and Connector Terminal Side” and “Fig. 1.2-2 External Appearance of the CONEXIOBlackBear - Antenna Terminal Side”.

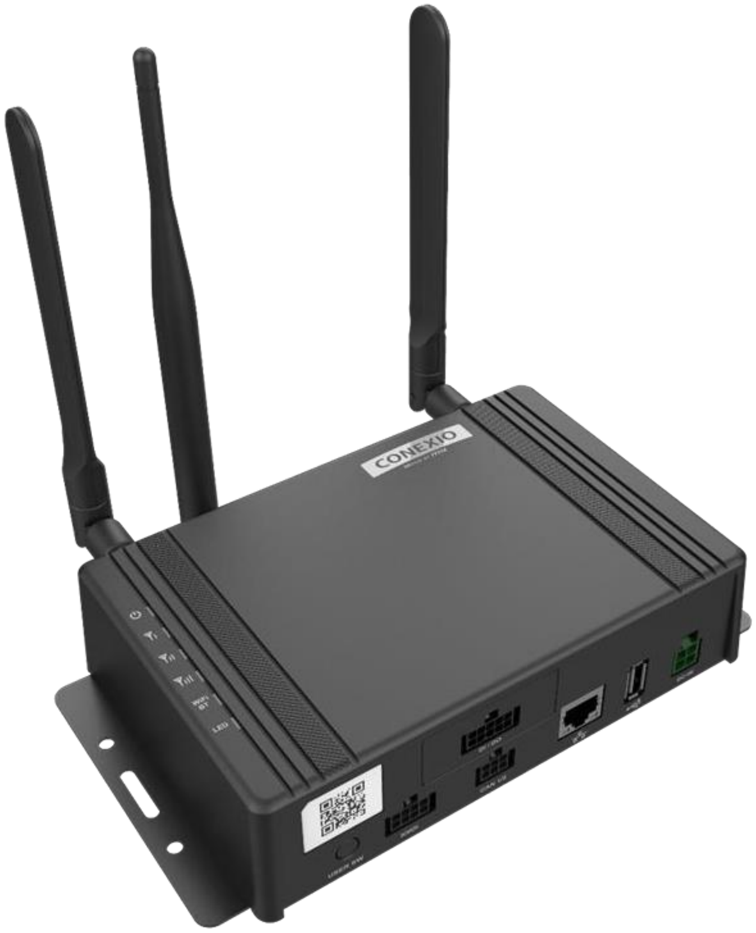


Fig. 1.2-1 External Appearance of the CONEXIOBlackBear - Top and Connector Terminal Side

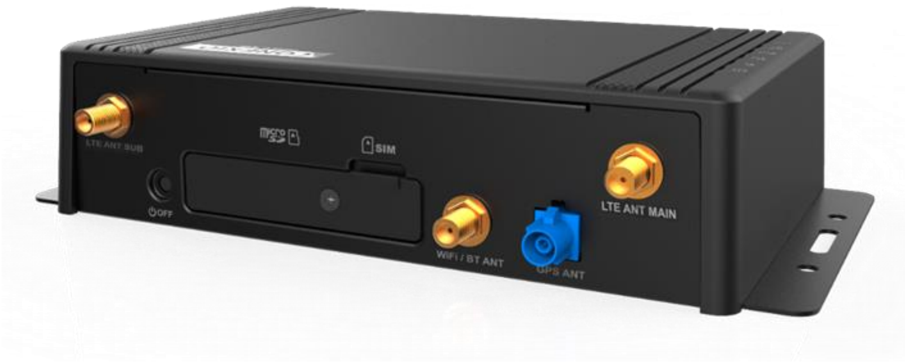


Fig. 1.2-2 External Appearance of the CONEXIOBlackBear - Antenna Terminal Side

2. Product Specifications

2.1. Dimensional Drawings of the Housing

The dimensions of the housing of the CONEXIOBlackBear are shown in “Fig. 2.1-1 Dimensional Drawings of the Housing”.

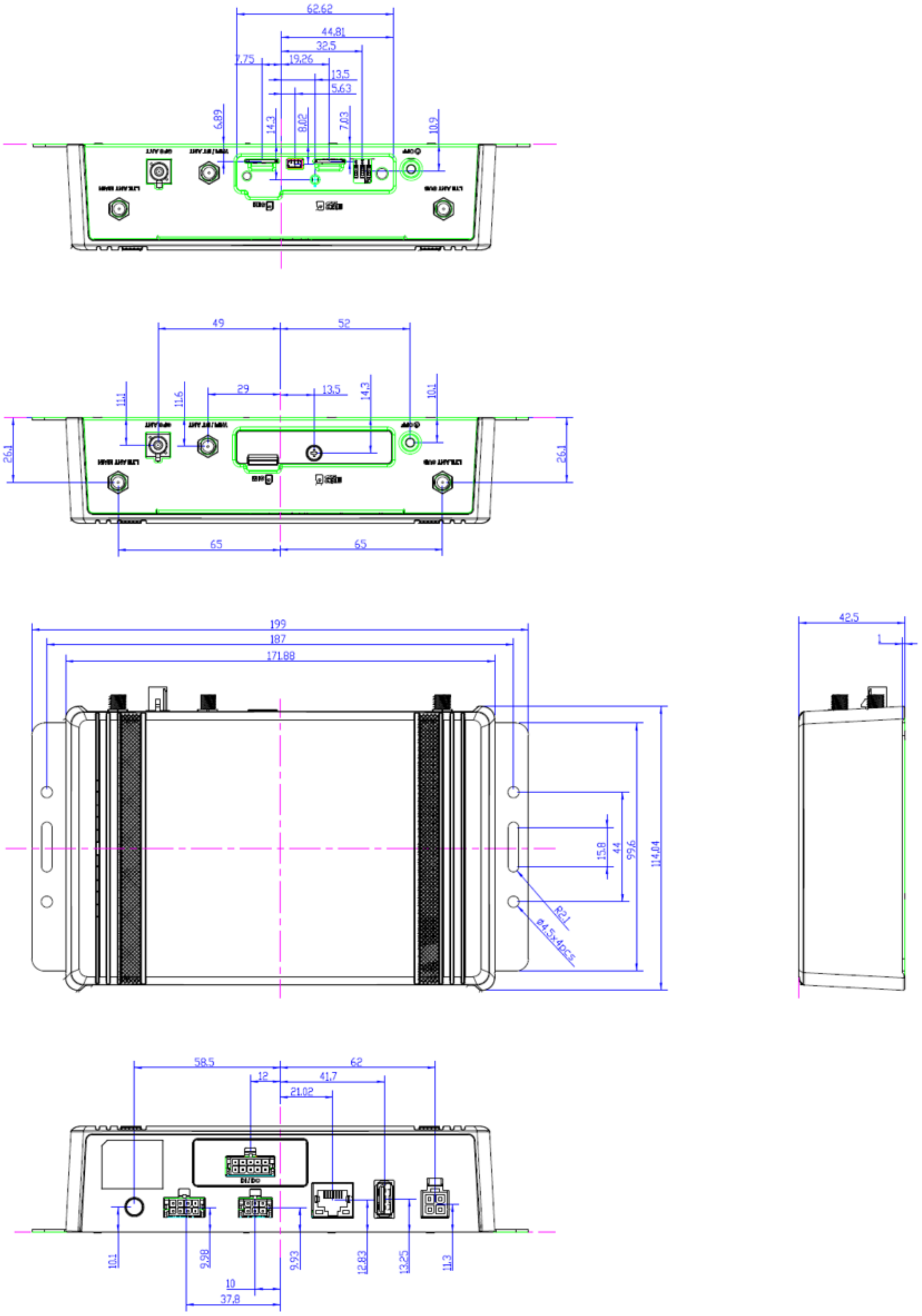


Fig. 2.1-1 Dimensional Drawings of the Housing

2.2. Table of Specifications

The main specifications of the CONEXIOBlackBear are shown in “Table 2.2-1 Table of Specifications”.

Table 2.2-1 Table of Specifications

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 ±10kV contact discharge and ± 10kV air discharge ESD testing (Criterion C)	
Certification	Compliant with Telec, VCCI, CE, FCC, IC and NCC (Up to LTE module chip certification, CONEXIOBlackBear 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 ± 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

The CONEXIOBlackBear is available as part of the Indoor Set, which is for indoor use, and the Commercial Set, which is for automotive use. The only differences between these two sets are the accessories. (See “Accessories in Table 2.2-1 Table of Specifications”.)

In addition, the CONEXIOBlackBear is sold as the “J Model”, “A Model” and “E Model,” depending on the country or region. With the exception of the LTE module frequencies and WiFi channels, the configuration of the CONEXIOBlackBear is the same for the models of all countries and regions, and therefore, the developed system can be used on all models.
The model numbers corresponding to each of the models are shown in “Table 2.2-2 Numbers by Model”.

Table 2.2-2 Numbers by Model

Model	Country/Region	Indoor/Automotive	Model No.
J Model	Japan and Taiwan	Indoor	MS-5776-JC-H Indoor
		Automotive	MS-5776-JC-H Commercial
A Model	U.S. and Canada	Indoor	MS-5776-A-H Indoor
		Automotive	MS-5776-A-H Commercial
E Model	Europe and Southeast Asian countries (*)	Indoor	MS-5776-E-H Indoor
		Automotive	MS-5776-E-H Commercial

(*) For the countries and regions supported, see “Countries and regions supported or expected to be supported” by accessing the following URL.

https://conexio-iot.jp/serviceproduct/blackbear_faq.html

While this Instruction Manual provides information based on the “J Model”, which is intended for use in Japan and Taiwan, as mentioned above, all models use the same operating system.

As each model is certified for the relevant country or region, be sure to select the correct model that is certified in the country or region in which it will be used. (See also “8.5. Overseas Use”.)

2.3. Block Diagram

The block diagram of the CONEXIOBlackBear is shown in “Fig. 2.3-1 Block Diagram”.

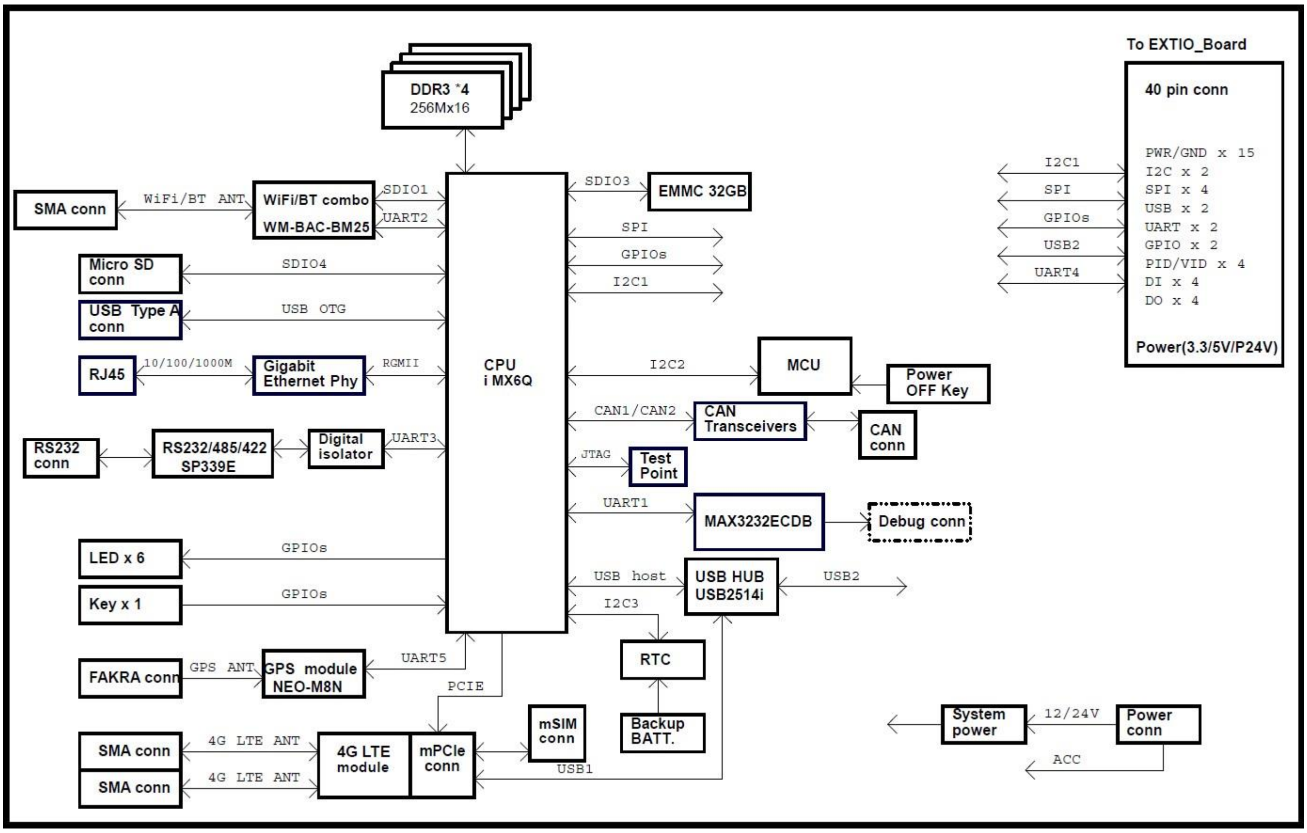
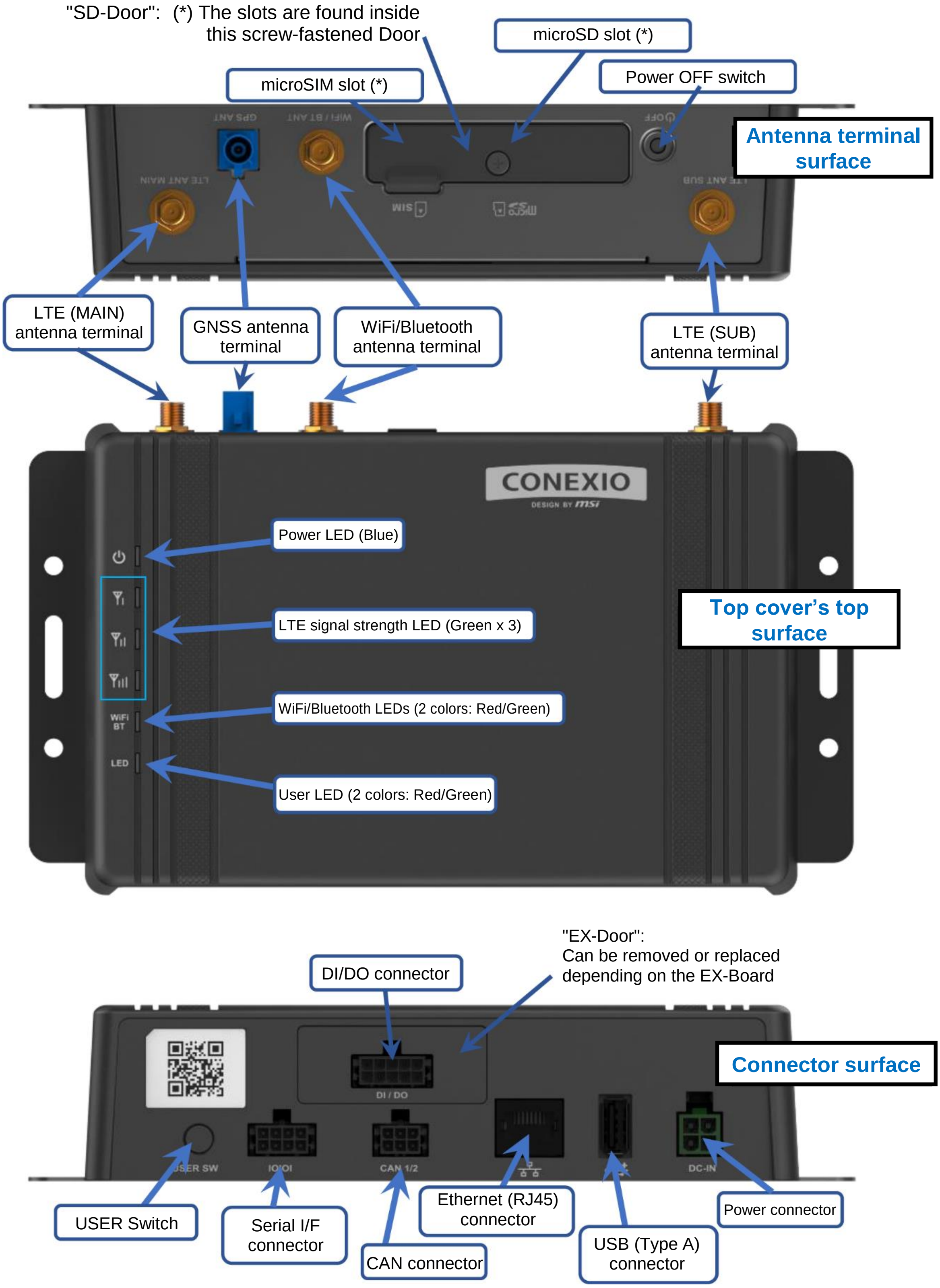


Fig. 2.3-1 Block Diagram

2.4. Interface Layout Diagram

The interface layout of the CONEXIOBlackBear is shown in “Fig. 2.4-1 Interface Layout”.



"SD-Door": (*) The slots are found inside this screw-fastened Door

"EX-Door":
Can be removed or replaced depending on the EX-Board

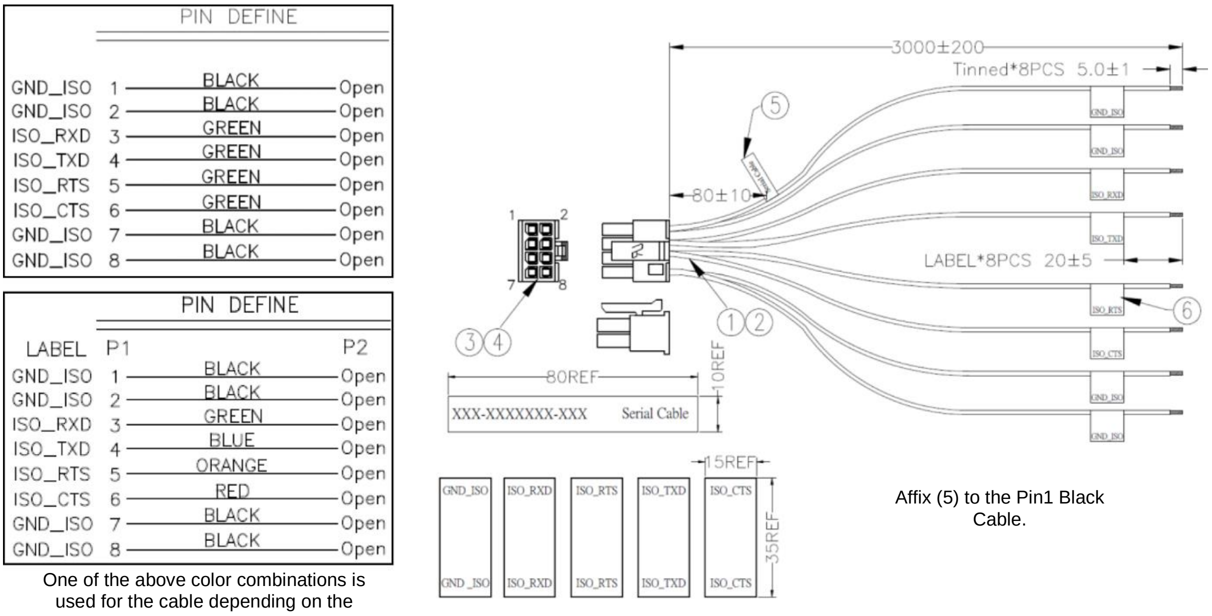
Fig. 2.4-1 Interface Layout

2.5. Accessories

The CONEXIOBlackBear is available as part of the “Indoor Set” or the “Automotive Set (Commercial Set)”, and some of the included parts differ between the two sets. The following sections separately explain the CONEXIOBlackBear parts included in both sets and those that are exclusive for each set.
Note that the “serial cable” and “DI/DO cable” are available in two cable colors depending on the timing of the shipment. Note that the two cable colors are written alongside each other for these cables.

2.5.1. Parts included in both the Indoor and Automotive sets

The following four items are included in both the “Indoor Set” and “Automotive Set”: “Serial cable”, “DI/DO cable”, “CAN cable” and “Holeless EX-Door”. Fig. 2.5-1 through Fig. 2.5-4 show the details of these included parts.



PIN DEFINE

GND_ISO	1	BLACK	Open
GND_ISO	2	BLACK	Open
ISO_RXD	3	GREEN	Open
ISO_TXD	4	GREEN	Open
ISO_RTS	5	GREEN	Open
ISO_CTS	6	GREEN	Open
GND_ISO	7	BLACK	Open
GND_ISO	8	BLACK	Open

PIN DEFINE

LABEL	P1		P2
GND_ISO	1	BLACK	Open
GND_ISO	2	BLACK	Open
ISO_RXD	3	GREEN	Open
ISO_TXD	4	BLUE	Open
ISO_RTS	5	ORANGE	Open
ISO_CTS	6	RED	Open
GND_ISO	7	BLACK	Open
GND_ISO	8	BLACK	Open

One of the above color combinations is used for the cable depending on the timing of the shipment.

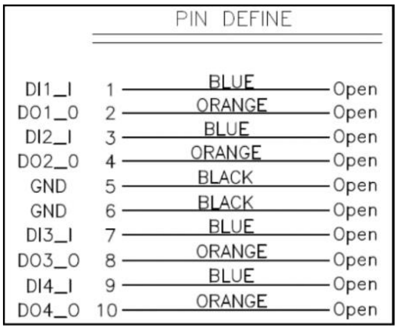
Affix (5) to the Pin1 Black Cable.

Pin No.	RS-232C		RS-485 Half Duplex		RS-485/RS-422 Full Duplex	
	Signal	I/O	Signal	I/O	Signal	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

* Leave the pins in the blank fields and those not used for GND_ISO unconnected.

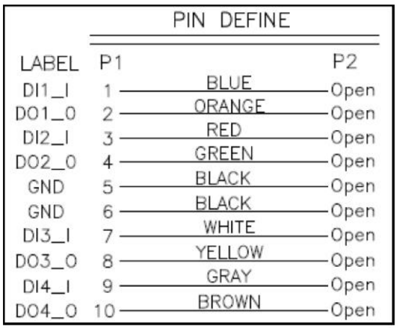
(1)	UL1007 20AWG(21/0.178)TIN PLATED COPPER WIRE BLACK INSULATION(-40°C~+85°C) Ø1.8mm L=3010mm (RoHS)	4pcs
(2)	UL1007 20AWG(21/0.178)TIN PLATED COPPER WIRE GREEN INSULATION(-40°C~+85°C) Ø1.8mm L=3010mm (RoHS)	4pcs
(3)	HOUSING:(WSTP/N:PD08-I30002) 2*4P/M PH:3.0mm PA66 BLACK (RoHS)	1pcs
(4)	TERMINAL: (WSTP/N:I30002PS-2) PH:3.0mm PHOSPHOR BRONZE TIN-PLATED (RoHS)	8pcs
(5)	Tedron waterproof label, 80*10 mm Black-on-silver text, Times New Roman font (RoHS)	1pcs
(6)	Tedron waterproof label, 35*15 mm Black-on-silver text, Times New Roman font (RoHS)	8pcs

Fig. 2.5-1 Serial Cable



PIN DEFINE

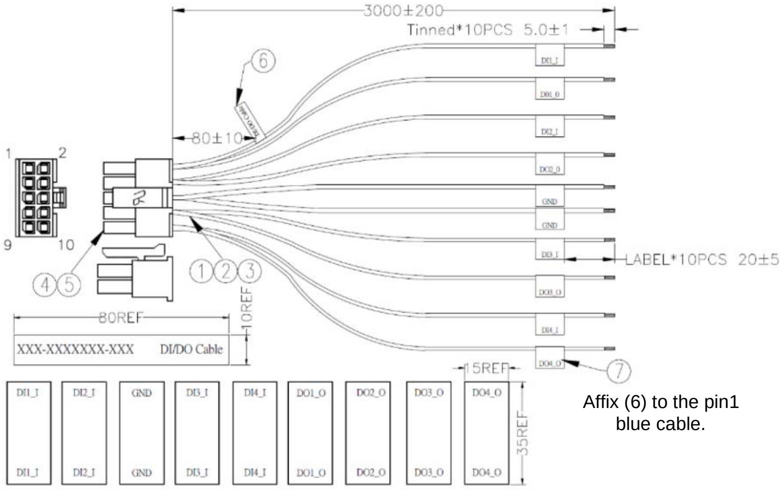
DI1_I	1	BLUE	Open
DO1_O	2	ORANGE	Open
DI2_I	3	BLUE	Open
DO2_O	4	ORANGE	Open
GND	5	BLACK	Open
GND	6	BLACK	Open
DI3_I	7	BLUE	Open
DO3_O	8	ORANGE	Open
DI4_I	9	BLUE	Open
DO4_O	10	ORANGE	Open



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

One of the above color combinations is used for the cable depending on the timing of the shipment.

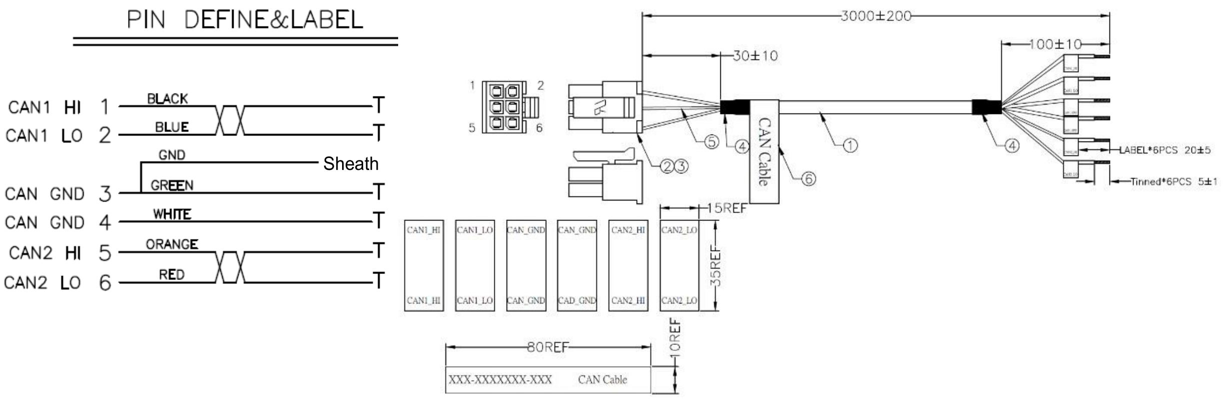


Affix (6) to the pin1 blue cable.

Pin No.	Signal	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)	Tedron waterproof label, 80*10 mm Black-on-silver text, Times New Roman font (RoHS)	1pcs
(7)	Tedron waterproof label, 35*15 mm Black-on-silver text, Times New Roman font (RoHS)	10pcs

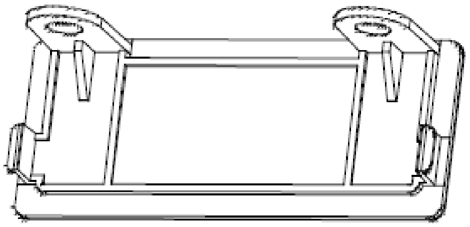
Fig. 2.5-2 DI/DO Cable



Pin No.	Signal	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)	Tedron waterproof label, 80*10 mm Black-on-silver text, Times New Roman font (RoHS)	6pcs
(7)	Tedron waterproof label, 35*15 mm Black-on-silver text, Times New Roman font (RoHS)	6pcs

Fig. 2.5-3 CAN Cable



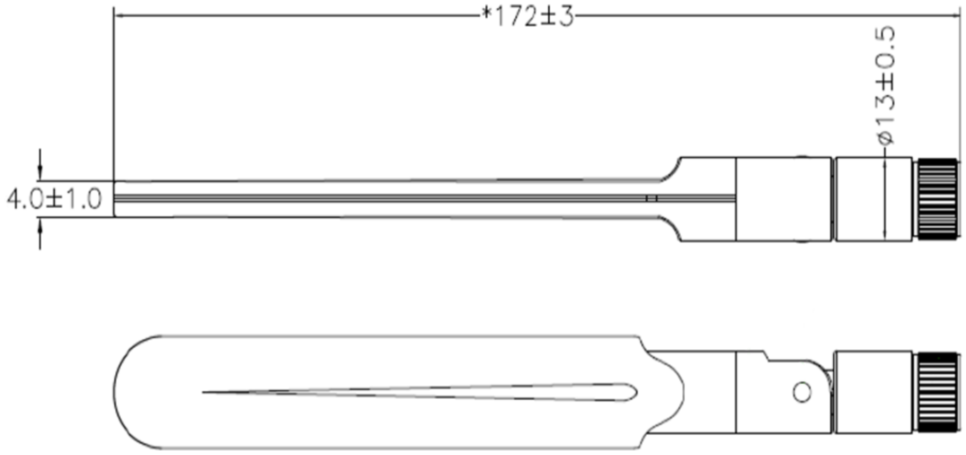
* The “EX-Door” with a hole for the “DI/DO connector” is attached standard to the CONEXIOBlackBear. When the EX-Board is removed, or on such other occasions, you can replace the standard “EX-Door” with the “Holeless EX-Door”, which is the same as the “EX-Door” but does not have a connector hole.

* When installing, take note of “8.6. Warranty”, and if necessary, find contact information in “9. Contact”.

Fig. 2.5-4 Holeless EX-Door

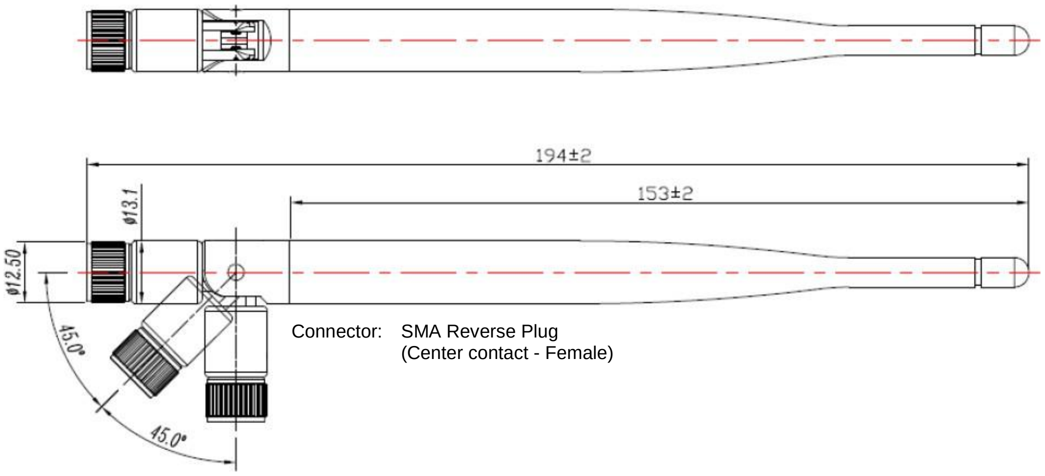
2.5.2. Indoor Set only

The following five items are included only in the “Indoor set”: Two “LTE antennas”, “WiFi/Bluetooth antenna”, “GNSS antenna”, “AC adapter” and “AC adapter conversion cable”. Fig. 2.5-5 through Fig. 2.5-9 show the details of these included parts.



Connector: SMA Normal Plug
(Center contact - Male)

Fig. 2.5-5 LTE Antenna



Connector: SMA Reverse Plug
(Center contact - Female)

Fig. 2.5-6 WiFi/Bluetooth Antenna

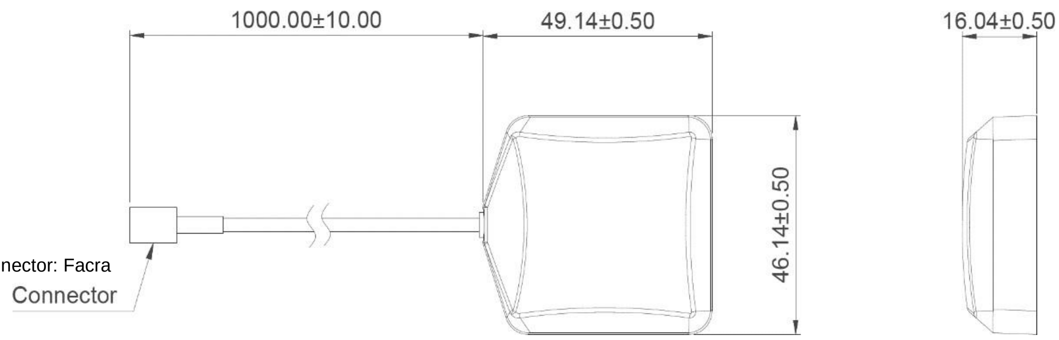
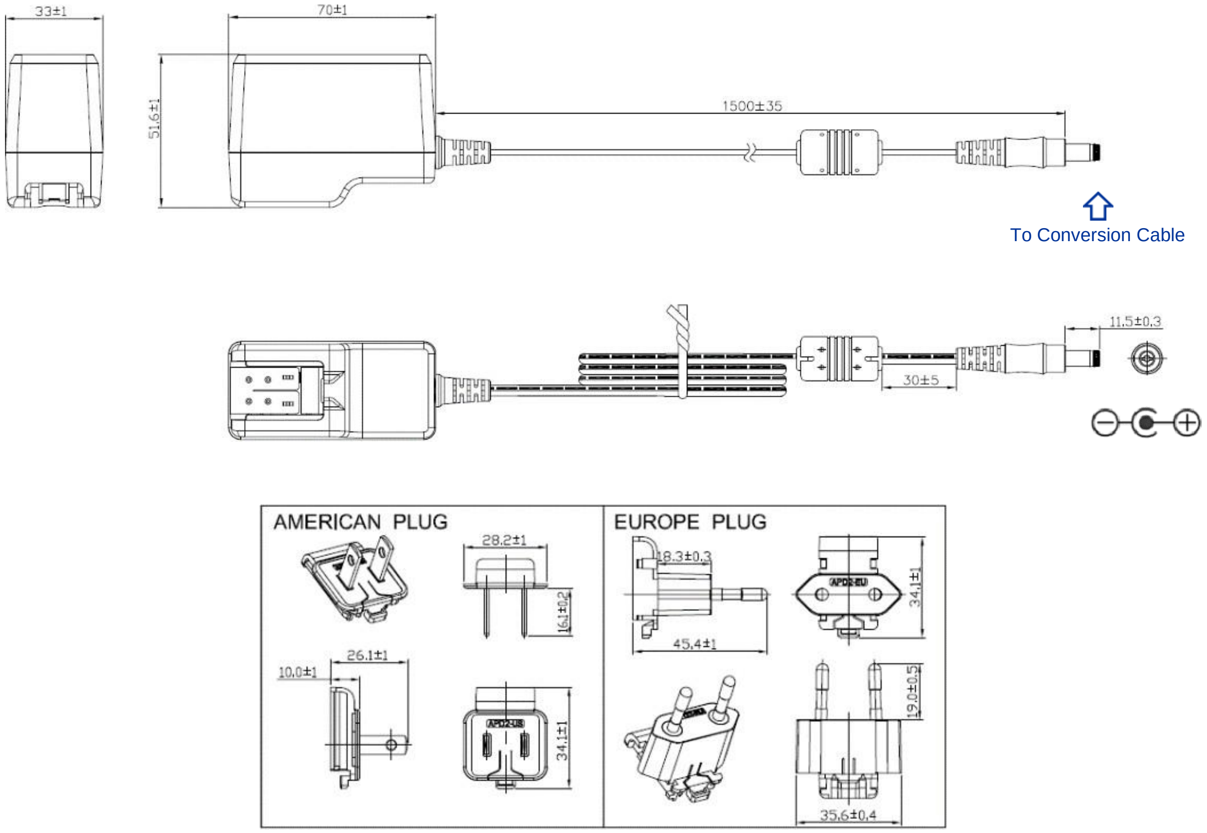


Fig. 2.5-7 GNSS Antenna

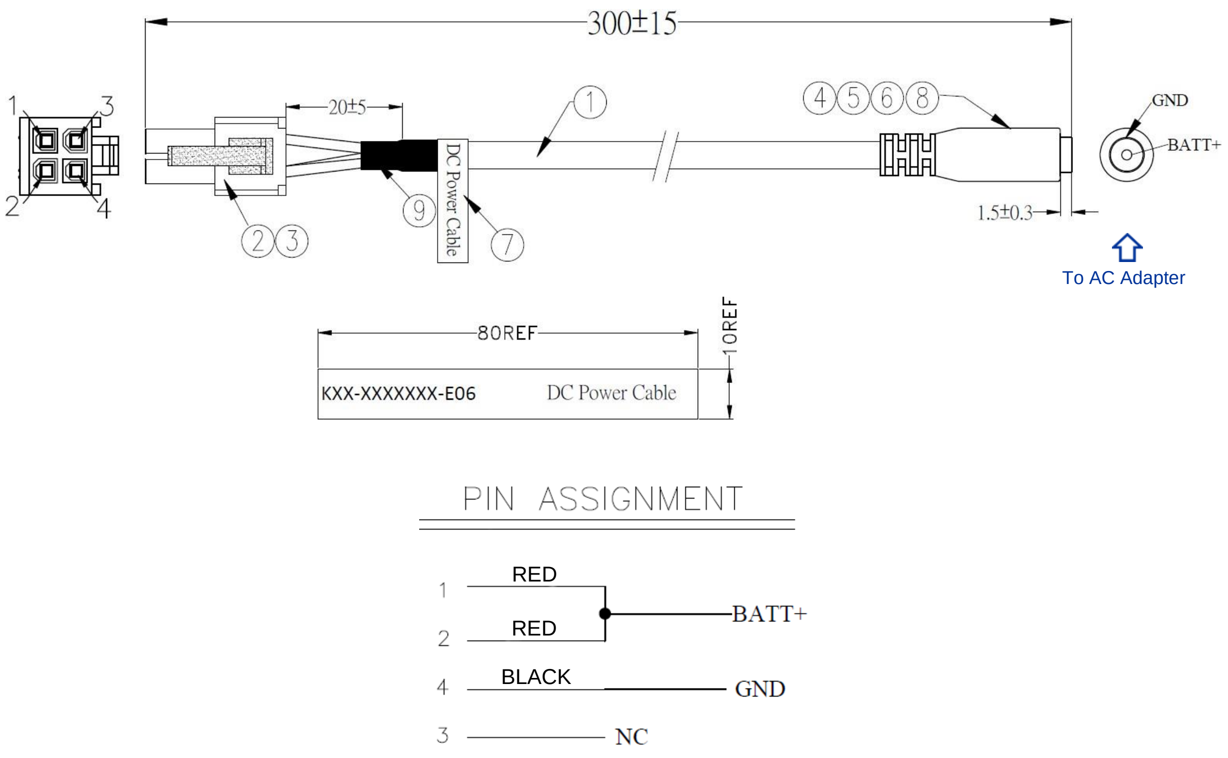


To Conversion Cable

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

* AC adapter replaces the plug to support various countries and regions.

Fig. 2.5-8 AC Adapter



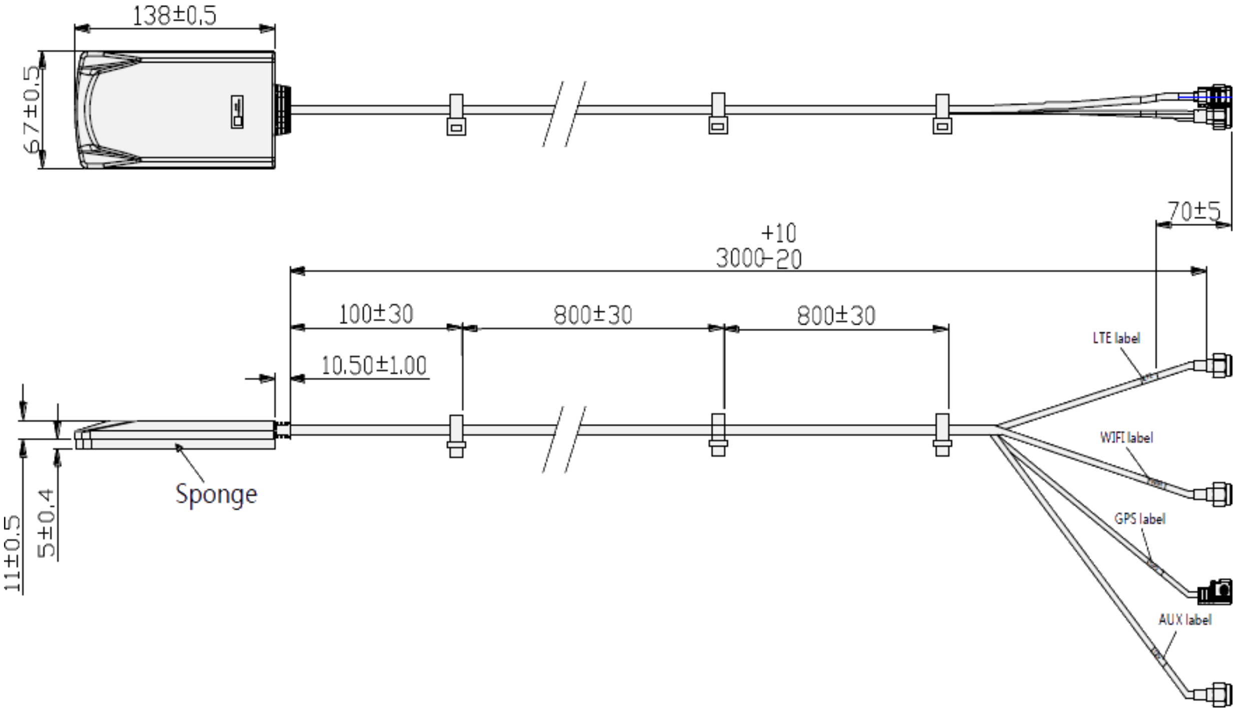
(1)	UL2464 18AWG*3C (Red*2 / Black*1) PVC Black (-40 to 105°C)(RoHS), L = 270±5 MM	1pcs
(2)	HSG:PH4.2mm 2*2P/M NYLON66 Transparent (WST P/N:P4-I42002)(RoHS) or equivalent	1pcs
(3)	TER: PH: 4.2 mm applied, 18# to 24# wire rod, brass tinned female terminal, (WST P/N: I42002BS-2)(RoHS) or equivalent	4pcs
(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)	Tedron waterproof label, 80*10 mm Black-on-silver text, Times New Roman font (RoHS)	1pcs
(8)	Black heat shrink sleeve: Φ1.5*0.13 (T) mm*6 mm EVA red phosphorus not contained (RoHS)	1pcs
(9)	Black heat shrink sleeve: Φ5.0*0.13 (T) mm*20 mm EVA red phosphorus not contained (RoHS)	1pcs

* Using this conversion cable, connect the AC adapter to the power connector of the CONEXIOBlackBear.

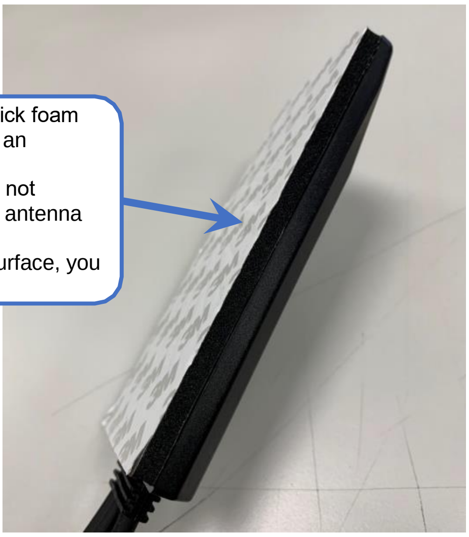
Fig. 2.5-9 AC Adapter Conversion Cable

2.5.3. Automotive Set only

The following two items are included only in the “Automotive Set”: “4-in-1 antenna/cable” and “power cable”. Fig. 2.5-10 and Fig. 2.5-11 show the details of these included parts.

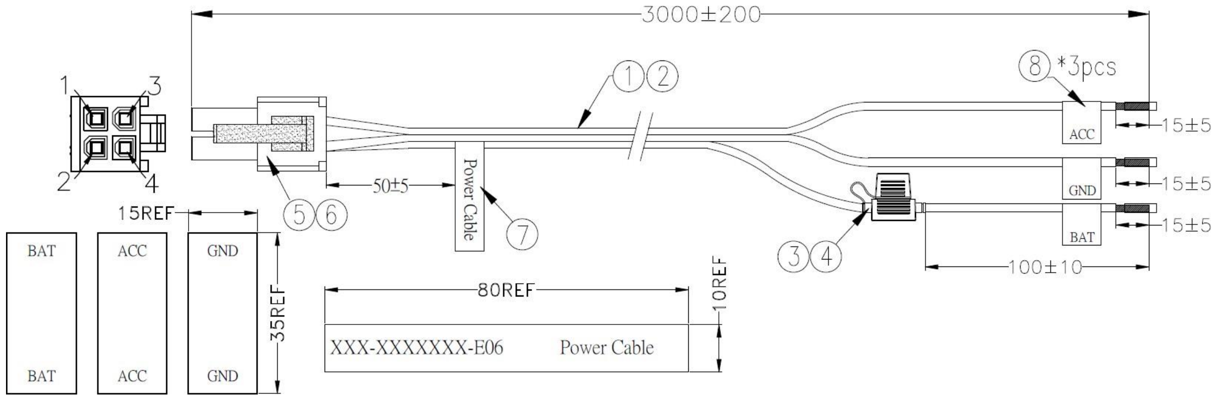


Antenna	Label	Connector
LTE MAIN	LTE	SMA (Male)
WiFi/Bluetooth	WIFI	SMA Reverse Type (Female)
GNSS	GPS	Facra-C code (Blue)
LTE SUB	AUX	SMA (Male)



The antenna’s bottom surface has a 5 mm thick foam rubber piece, whose surface is provided with an adhesive.
To maintain the antenna’s characteristics, do not remove this foam rubber, when attaching the antenna to a metal surface.
By affixing a magnet sheet to the adhesive surface, you can install the antenna on a steel surface.

Fig. 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)	Electronics cable: UL1007 18AWG (34/0.178), tinned copper OD: 2.1mm, PVC insulation (-40 to 105°C) Black (RoHS), L = 2990±5 mm	1pcs
(2)	Electronics cable: UL1007 18AWG (34/0.178), tinned copper OD: 2.1mm, PVC insulation (-40 to 105°C) Red (RoHS), L = 2990±5 mm	1pcs
(3)	Fuse holder: Black + Electronics cable (AVS0.8, yellow, L1 = 2870 mm / L2 = 100 mm), all (RoHS)	1pcs
(4)	Blade fuse: 32 V, 5 A, amber (P/N: MIN-N5) (RoHS)	1pcs
(5)	HSG:PH4.2mm 2*2P/M NYLON66, transparent (WST P/N: P4-I42002) (RoHS) or equivalent	1pcs
(6)	TER: PH: 4.2 mm applied, 18# to 24# wire rod, brass tinned female terminal, (WST P/N: I42002BS-2) (RoHS) or equivalent	4pcs
(7)	Tedron waterproof label, 80*10 mm Black-on-silver text, Times New Roman font (RoHS)	1pcs
(8)	Tedron waterproof label, 35*15 mm Black-on-silver text, Times New Roman font (RoHS)	3pcs

Fig. 2.5-11 Power Cable

2.6. OS and Major Preinstalled Software

The OS and preinstalled software of the CONEXIOBlackBear are shown in “Table 2.6-1 OS and Major Preinstalled Software”.
BusyBox is used as the main utility tool. For the use of open-source software (OSS), take note of the information provided in “8.4. Notes on Open-Source Software”.

Table 2.6-1 OS and Major Preinstalled Software

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	1.9.5p2
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	1.32.10
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. Using the CONEXIOBlackBear (Preparation, Startup, Exit)

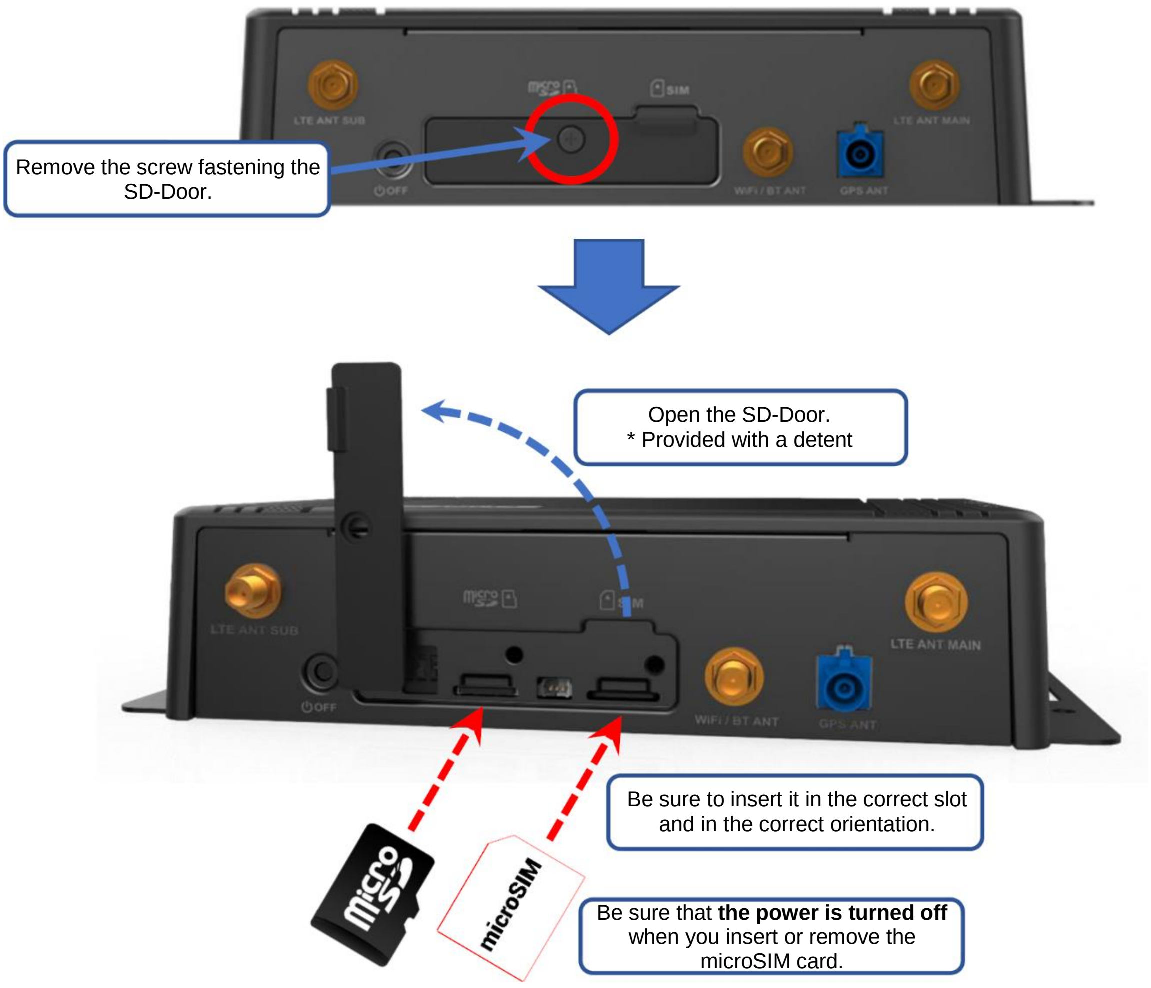
This chapter provides the instructions for basic operation, including preparation before starting up the CONEXIOBlackBear, turning the power on, exiting from the Linux system, and turning the power off.

3.1. Insertion of microSIM and microSD Cards

Remove the screw fastening the “SD-Door” shown in “Fig. 3.1-1 Insertion of microSIM and microSD Cards”, and if necessary, insert microSIM and microSD cards. Make sure to insert each of them in the correct slot and in the correct orientation by using the marks printed above the “SD-Door” for reference. In addition, be careful not to insert them with unnecessary force, as this may damage the slot. It is especially important that the power is turned off before inserting or removing the microSIM card to prevent damage or malfunction.

After confirming that they are reliably inserted, fasten the “SD-Door” with the screw.

If you are using a SIM card adapter, use it at your own risk. Take plenty of caution to avoid damaging the SIM slot, such as by using one with fall protection. If damaged due to the use of a SIM card adapter or the insertion of an object other than a microSIM card, the SIM slot will not be covered by the product warranty policy for free-of-charge service.



Remove the screw fastening the SD-Door.

Open the SD-Door.
* Provided with a detent

Be sure to insert it in the correct slot and in the correct orientation.

Be sure that **the power is turned off** when you insert or remove the microSIM card.

Fig. 3.1-1 Insertion of microSIM and microSD Cards

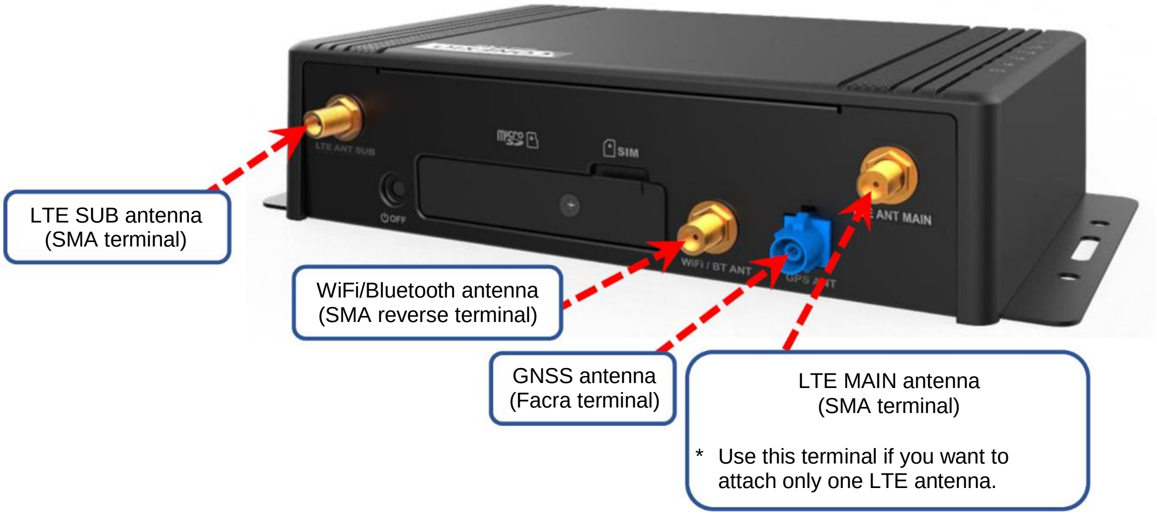
3.2. Attaching the Antennas

Attach the LTE, WiFi/Bluetooth, and GNSS antennas to the antenna terminals provided on the antenna terminal surface shown in “Fig. 2.4-1 Interface Layout” as necessary.

Be sure to use the correct antenna terminal. Both the LTE antenna terminals and the WiFi/Bluetooth antenna terminal are SMA types, but only the SMA terminal of the WiFi/Bluetooth antenna is a reverse type. The GNSS antenna terminal (“GPS ANT” lettered on the housing) is a Facra terminal type.

For the Indoor Set, if you want to attach only one LTE antenna instead of two, attach it to the antenna terminal indicated “LTE ANT MAIN”. Note that attaching only one antenna is not recommended because it is highly likely that characteristics will not be maintained.

For the connection of the automotive 4-in-1 antenna, see the label on the cable.
(The cable for the “LTE SUB antenna” is indicated “AUX”.)



LTE SUB antenna
(SMA terminal)

WiFi/Bluetooth antenna
(SMA reverse terminal)

GNSS antenna
(Facra terminal)

LTE MAIN antenna
(SMA terminal)

* Use this terminal if you want to attach only one LTE antenna.

Fig. 3.2-1 Attaching the Antennas

3.3. Cable Connection

Connect the Ethernet cable, serial cable, DI/DO cable, CAN cable, and USB cable to the connectors provided on the “connector surface” shown in “Fig. 2.4-1 Interface Layout” as necessary.

The serial cable, DI/DO cable, and CAN cable are provided with stoppers on their terminals. Insert them all the way into place and ensure that the stopper is fitted with the connector’s catch on the CONEXIOBlackBear side.

For the Indoor Set, use the conversion cable to connect the AC adapter to the CONEXIOBlackBear. You may also want to connect the AC adapter conversion cable, but keep in mind that turning on the power causes the CONEXIOBlackBear to start up as described in “3.4. Startup (Power Turn-ON/Login/Time Setup/RTC)”. Be careful not to turn on the power unintentionally (deliver power to the CONEXIOBlackBear by connecting the AC adapter to the power supply) until you are ready to start it up.

For the Automotive Set, connect the power cables as follows: the yellow cable to the battery’s positive terminal; the black cable to the battery’s negative terminal; and the red cable to the ACC or IG power terminal. The yellow cable has a 5A fuse at its midpoint.

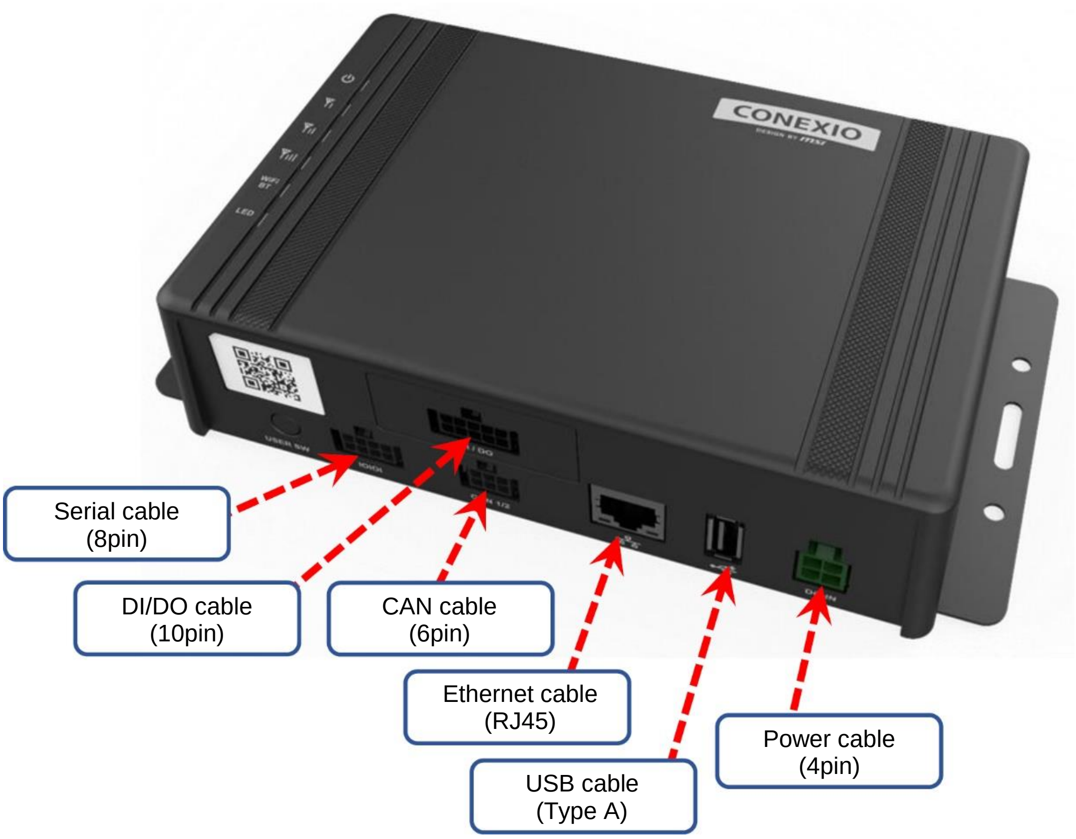


Fig. 3.3-1 Cable Connection

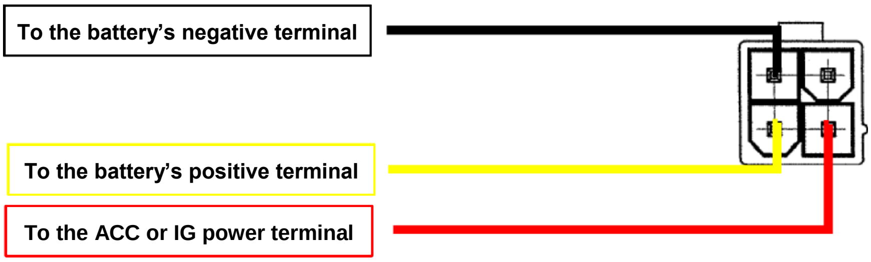

To the battery’s negative terminal
To the battery’s positive terminal
To the ACC or IG power terminal

Fig. 3.3-2 Automotive Power Cable Connection

3.4. Startup (Power Turn-ON/Login/Time Setup/RTC)

3.4.1. Power ON

Table 3.4-1 Instructions for turning the power on to startup

Applicable set	Instructions for turning the power on
Indoor Set (AC adapter conversion cable used)	Deliver power by connecting the AC adapter.
Automotive Set (automotive power cables used)	Turn the ACC/IG power ON.

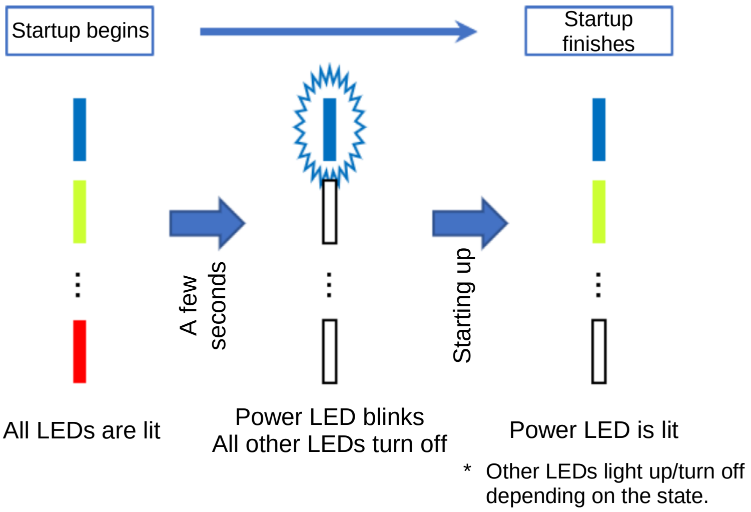


Fig. 3.4-1 LED Indication on Startup

Just turning on the power automatically starts the CONEXIOBlackBear and starts up the system and installed applications.

To turn on the Indoor Set (AC adapter conversion cable used), deliver power by connecting the AC adapter. To turn on the Automotive Set (automotive power cables used), turn on the ACC/IG power after connecting the automotive power cables to the battery or other power supply.

After the CONEXIOBlackBear is turned on, all LEDs on the “top cover’s top surface” shown in “Fig. 2.4-1 Interface Layout” light up. Then, the power LED starts blinking and all LEDs other than the power LED turns off in a few seconds.
The power LED keeps blinking while the CONEXIOBlackBear is being started up, changing to a solid light after the CONEXIOBlackBear has been started up. After the power LED has changed to a solid light, the CONEXIOBlackBear is ready for use.

As mentioned above, you can start using the CONEXIOBlackBear by simply turning it on if you just want to use the preconfigured information or preinstalled applications.

3.4.2. Login

The user name used for the first login after the system is started up is “root” (a user with superuser privileges) by default. **At the first login after factory shipment, you are prompted to configure a new password.** Configure an appropriate password following the on-screen instructions.

To ensure security, please take the utmost care to properly manage passwords and set user privileges.

3.4.3. Time Setup/RTC

With the CONEXIOBlackBear’s RTC backed up by the internal battery, the configured time is maintained even when the power is turned off. The internal battery is the CR1220. It is attached to the battery holder in the housing and can be replaced.

You can check the configured time and time zone using the timedatectl command.

Example) Example of execution of the timedatectl command (“timedatectl status” gives the same result)

```
# 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
```

* The “#” at the beginning of the line represents the Linux command prompt to indicate that the line is a Linux input command. (This notation will continue to be used in this manual from this chapter onward.)

On the CONEXIOBlackBear, the local time is set to UTC by default. The timedatectl command is also used to change the time zone from UTC as shown below.

Example) Setting the time zone to JST after checking the available time zones

```
# timedatectl list-timezones                    Check the available time zones.
    • • •
# timedatectl set-timezone Asia/Tokyo           Set the time zone to JST.
# 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
```

When the CONEXIOBlackBear is connected to the network, if “Network time on” is set to “yes”, the time is configured via clock synchronization executed for the network NTP server. After clock synchronization has been completed, “NTP synchronized” changes to “yes”.

* “Network time on” shows whether to execute clock synchronization using the network NTP server. The default setting for the CONEXIOBlackBear is “yes”.
* “NTP synchronized” shows whether or not the time was configured using the NTP server.
* Clock synchronization may take several tens of seconds depending on the condition of the NTP server or the network.

The default NTP server the CONEXIOBlackBear uses is “time.google.com” (time1.google.com to time4.google.com are used). To change this, configure the NTP server to be used for “/etc/systemd/timesyncd.conf”. (NTP is for main use, and FallbackNTP is for spare use. If more than one NTP server is configured as a spare, delimit them with half-width spaces.)
To apply to NTP server settings, reboot the CONEXIOBlackBear or reboot the clock synchronization daemon using the following command.

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

It is also possible to manually configure the time without using the NTP server. Note that manual configuration of the time is not possible if both “Network time on” and “NTP synchronized” are set to “yes”. In such cases, set “Network time on” to “no” by executing the “timedatectl set-ntp” command and then configure the time using the “date” command or “timedatectl set-time” command as follows.

Example) Manually setting the time to 13:39:00 on October 25, 2019 without using the NTP server

```
# timedatectl set-ntp no                (“0” or “false” can be substituted for the argument “no”.)
# date -s "2019-10-25 13:39:00"
```

To change the time setting mode from manual to the use of the NTP server, set “Network time on” to “yes” by executing the following “timedatectl set-ntp” command.

```
# timedatectl set-ntp yes               (“1” or “true” can be substituted for the argument “yes”.)
```

After the time has been configured or corrected, configure the time on the RTC as necessary using the hwclock command as follows.

Example) Setting the local time of the system clock as the UTC of the RTC

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

* It has been extensively announced that the RTC may cause unexpected bugs in the Linux kernel if configured with a local time such as JST instead of UTC. We strongly recommend configuring the RTC to UTC.

Conversely, the UTC of the RTC can be configured as the local time of the system clock using the hwclock command as shown below.

Example) Setting the UTC of the RTC as the local time of the system clock

```
# hwclock -s
```

3.5. Suspend – Resume

3.5.1. Suspend

Table 3.5-1 Instructions for Suspend

Methods to suspend	Operation instruction
Linux command	Execute the “go-sys-suspend” command.

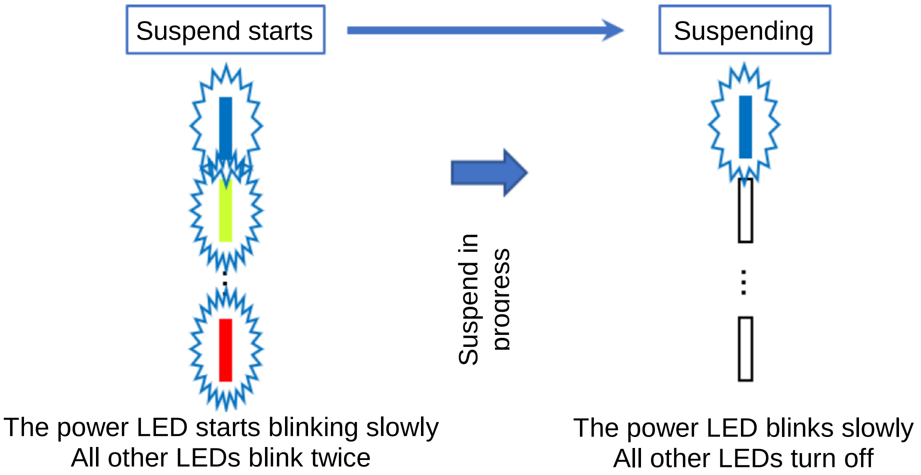

Suspend starts
Suspending
Suspend in progress
The power LED starts blinking slowly
All other LEDs blink twice
The power LED blinks slowly
All other LEDs turn off

Fig. 3.5-1 LED Indication for Suspend

The CONEXIOBlackBear can be put in “Suspend Mode” by executing the “go-sys-suspend” command (this command is unique to the CONEXIOBlackBear).

In “Suspend Mode”, all processors other than the MCU (power control processor) and the peripheral I/O devices enter a state of low power consumption while the processing status of the CONEXIOBlackBear is kept in memory.

All LEDs on the top surface of the top cover other than the power LED will blink twice at the start of “Suspend Mode”, and the power LED will continue blinking slowly during “Suspend Mode”.

3.5.2. Resume

Table 3.5-2 Instructions for Resume

Methods to resume	Operation instruction
USER SW	Press the USER SW.
SMS	Receive an SMS message via LTE.
RTC alarm	Specify the Resume timing using the rtcwake command, etc.

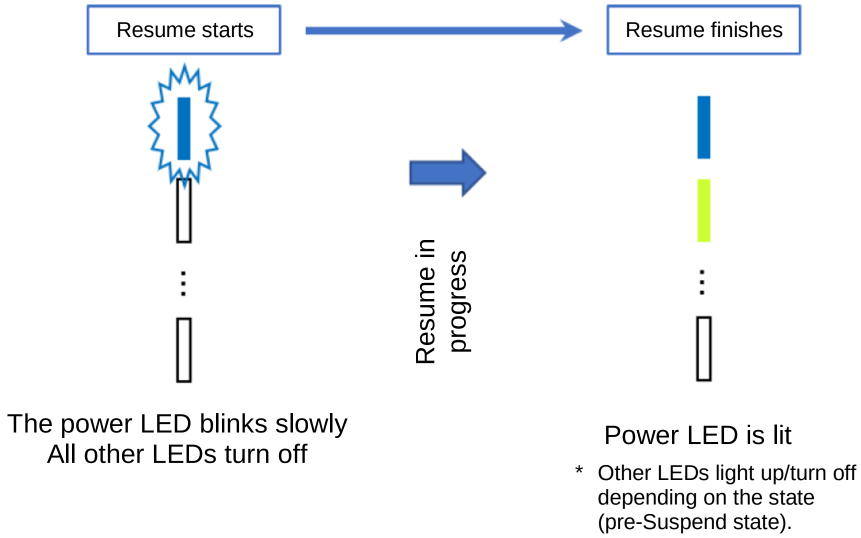

Resume starts
Resume finishes
Resume in progress
The power LED blinks slowly
All other LEDs turn off
Power LED is lit
* Other LEDs light up/turn off depending on the state (pre-Suspend state).

Fig. 3.5-2 LED Indication for Resume

To resume from “Suspend Mode”, the following three methods are available.

- Press the “USER Switch (USER SW)” shown in “Fig. 2.4-1 Interface Layout”.
- Receive SMS messages via LTE.
- Configure the Resume timing in advance using the rtcwake command, etc.

Note that SMS reception requires a subscription to a service capable of SMS communication. SMS messages are automatically received if LTE has been already connected upon completion of LTE connection setup described in “4.2. LTE”.
For details about SMS, see “4.2.2. SMS”.

From OS V1.1.5 or later, the CONEXIOBlackBear can be resumed from “Suspend Mode” with RTC alarm interrupts.
The Resume timing can be configured as shown in the following example using the “rtcwake” command, etc.
After the Resume timing has been configured, enter ”Suspend Mode” by executing the “go-sys-suspend” command.

Example) Setting Resume to be executed at 15:20:00 on May 26, 2020

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

Example) Setting Resume to be executed in 1 hour

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

* When the rtcwake command is used, specify “no” for the “-m” option to configure only the Resume timing. To enter “Suspend Mode”, be sure to use the “go-sys-suspend” command.

* The time specified for the “--date” option of the rtcwake command is configured to be the time of the system clock.
Even when the RTC remains configured to UTC and the system clock is configured to a local time (JST, etc.) other than UTC, Resume is executed at the time that was configured with the local time using the rtcwake command.

To check the configured Resume timing, execute the following command so that the results will be returned as “alrm_time” and “alrm_date”.

Example) Checking the configured RTC alarm

```
# cat /proc/driver/rtc
rtc_time        : 09:39:29
rtc_date        : 2020-05-25
alrm_time       : 15:20:00
alrm_date       : 2020-05-26
     •
     •
     •
```

The power LED finishes blinking and changes to a solid light after Resume from “Suspend Mode” finishes. After Resume finishes, the process resumes from the point where the system entered “Suspend Mode”, and all LEDs other than the power LED also return to the state they were in before the system entered “Suspend Mode”.

3.6. Termination (Power OFF/Reboot)

3.6.1. Power OFF

Table 3.6-1 Instructions for Termination to Power OFF

Methods to turn off the power	Operation instruction
Power OFF SW	Hold down for 10 seconds or more
Automotive power cables	Turn the ACC/IG power off.
Linux command	Execute the “go-sys-off” command.

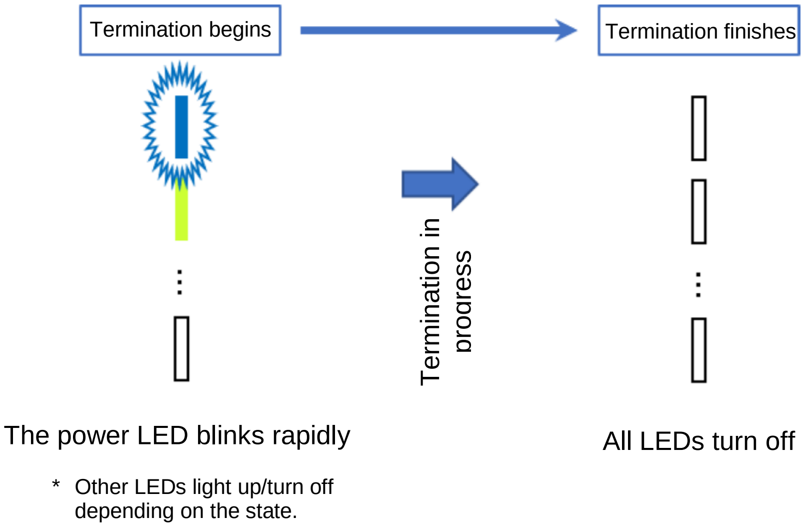

Termination begins
Termination finishes
Termination in progress
The power LED blinks rapidly
* Other LEDs light up/turn off depending on the state.
All LEDs turn off

Fig. 3.6-1 LED Indication for Exit

When the “Power OFF switch” shown in “Fig. 2.4-1 Interface Layout” is pressed for 10 seconds or more, the functions of the CONEXIOBlackBear will be terminated and the power will turn off. If the automotive power cables of the Automotive Set are used, the power can also be turned off by turning the ACC/IG power off.

The “Power OFF Switch” is embedded in the recess on the “antenna terminal surface” to prevent the power from being turned off unintentionally. For this reason, pressing the switch with your finger or fingernail may be difficult. If necessary, use a thin pin-like object to press the switch.

The power LED blinks rapidly during the termination process and shuts off when the power turns off after the CONEXIOBlackBear has completely terminated. To avoid unexpected trouble or malfunction, wait until the power LED shuts off before disconnecting the power cable or AC adapter.

When using the AC adapter, if you turn the power off but plan to turn it on again immediately afterwards, it is recommended that you disconnect the CONEXIOBlackBear from the AC adapter (by disconnecting and reconnecting either connector shown in “Fig. 2.5-9 AC Adapter Conversion Cable”).
If the power is turned off by shutting off the 100V AC supplied to the AC adapter, the current accumulated in the AC adapter needs to be discharged through the CONEXIOBlackBear. For this reason, even if supply of the 100V AC to the AC adapter is resumed, the correct power supply will be lost. For this reason, startup of the CONEXIOBlackBear will be prevented for approximately 20 seconds after the 100V AC is shut off. Be sure to wait approximately 20 seconds or more before turning on the 100V AC again after supply to the AC adapter has been shut off.

When terminating the system with a Linux command, be sure to use the “go-sys-off” command (this command is unique to the CONEXIOBlackBear). If the system is terminated using the shutdown or poweroff command, the synchronization between the CPU and MCU may not be executed properly, which may cause operation problems.

3.6.2. Reboot

Table 3.6-2 Instructions for Reboot

Methods for rebooting	Operation instruction
Linux command	Execute the “go-sys-reboot” command.

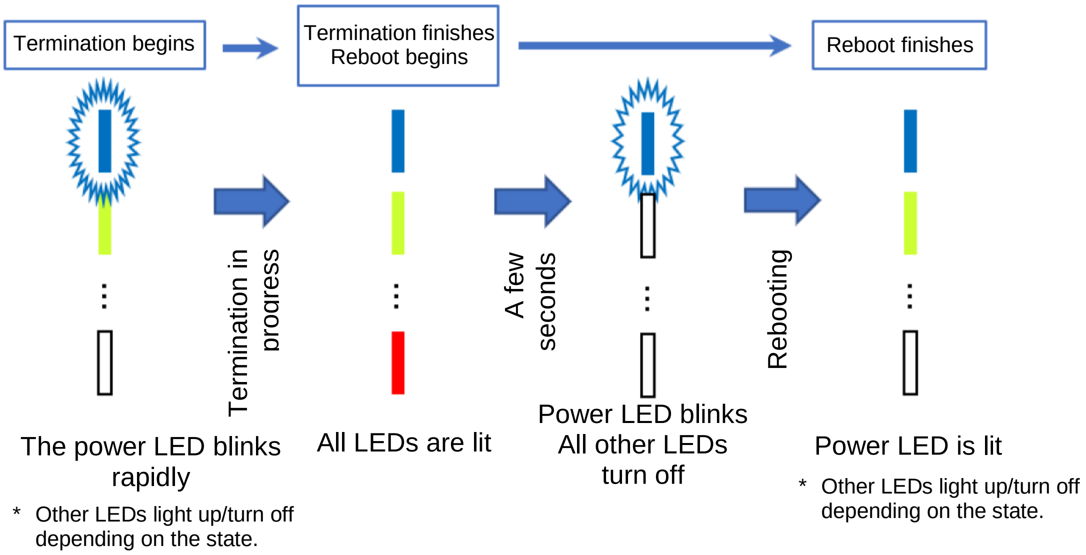


Fig. 3.6-2 LED indication for Reboot (go-sys-reboot executed)

When rebooting the system with a Linux command, be sure to use the “go-sys-reboot” command (this command is unique to the CONEXIOBlackBear). If the system is rebooted using the reboot command, the synchronization between the CPU and MCU may not be executed properly, which may cause operation problems.
Executing the “go-sys-reboot” command causes the CPU to work together with the MCU to control the power and LEDs. This allows you to reboot while you check the LEDs to see if the system is terminating or rebooting. The LED illumination pattern between blinking and solid light is a combination of the patterns for termination and startup described in the previous sections.

4. Using the Interfaces

This chapter provides basic instructions for using the CONEXIOBlackBear interfaces shown in “Fig. 2.4-1 Interface Layout”.

4.1. Ethernet

4.1.1. DHCP connection

Table 4.1-1 Ethernet Device

Device Name
eth0

Under the default settings, you only need to connect an Ethernet cable to the Ethernet connector of the CONEXIOBlackBear in order for Ethernet connection to become available for use with an IP address obtained from the DHCP. By default, the Ethernet connection name is “Wired connection 1”.

The Ethernet connection information is not saved as a file under “/etc/NetworkManager/system-connections/” under the default settings, but it can be changed and managed from Network Manager. If changes are made using Network Manager, the modified Ethernet connection information is stored under “/etc/NetworkManager/system-connections/”.

The following is an example of creating a new Ethernet connection using Network Manager with an IP address obtained from the DHCP instead of using the default Ethernet connection “Wired connection 1”. The Ethernet device that can be managed using Network Manager is eth0.

```
# 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” is an example of a newly created Ethernet connection name.
* If a new connection is added using Network Manager, the Ethernet connection information is stored under “/etc/NetworkManager/system-connections/” with the settings specified at the time of creation.

4.1.2. Static IP connection

To use an Ethernet connection with a static IP address instead of the default DHCP connection, the settings need to be changed. The following is an example that shows how to change the settings using Network Manager. When using a static IP address, it is not recommended to use the default connection name ‘Wired connection 1’. In order to avoid possible problems during reboot, use another connection name when you want to use a static IP address as the Ethernet connection for the CONEXIOBlackBear.

```
# 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” is an example of a newly created Ethernet connection name.
* “[IPADDRESS]” refers to the static IP address that you want to use. (Example: 192.168.21.117/24)
* “[GATEWAY]” refers to the IP address of the default gateway that you want to use. (Example: 192.168.21.1)

4.1.3. DHCP/DNS server function

On the CONEXIOBlackBear, dnsmasq is preinstalled as standard. This dnsmasq provides DHCP and DNS server function.

The following is an example that shows how to configure a connection that executes the DHCP/DNS server function with the Ethernet interface.

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

* “eth_shared” is an example name for the Ethernet connection that serves as the DHCP server.
* “[NETWORK]” refers to the IP address of the network that you want to use.
For example, if “192.168.9.1/24” is configured, the Ethernet IP address of the CONEXIOBlackBear will be 192.168.9.1, and the devices that are connected via Ethernet to the CONEXIOBlackBear are assigned with IP addresses on the 192.168.9.1/24 network.
* With the above setting, the CONEXIOBlackBear doubles as a DNS server via the dnsmasq function.
* iPv4 routing will also be enabled and registered with iptables, including IP masquerading to achieve Ethernet-to-LTE communication.
If LTE connection is enabled, this allows LTE connection to access the Internet from devices that have been connected to the Ethernet interface using the above setting.

4.1.4. Disabling automatic Ethernet connection

As described in “4.1.1. DHCP connection”, the CONEXIOBlackBear automatically enables Ethernet on startup by default. To disable automatic Ethernet connection on startup, use the following command.

```
# nmcli connection modify 'Wired connection 1' connection.autoconnect no
```

With the execution of the above command, Ethernet will no longer be automatically enabled and will instead be launched in a disconnected state from the next startup onward. To disconnect it immediately with no reboot after executing the above command, execute the following command.

```
# nmcli connection down 'Wired connection 1'
```

To automatically re-enable Ethernet on startup after automatic connection has been disabled as shown above, enable automatic Ethernet connection using the following command.

```
# nmcli connection modify 'Wired connection 1' connection.autoconnect yes
```

Executing the above command causes Ethernet to be automatically enabled even if reboot is performed. When automatic connection is cancelled, executing the following command enables Ethernet immediately without rebooting. Note that if reboot is performed, Ethernet will not be enabled automatically but will be launched in a disconnected state.

```
# nmcli connection up 'Wired connection 1'
```

To cause Ethernet to be automatically enabled even if reboot is performed, auto connection needs to be configured.

4.1.5. RJ45 connector

The RJ45 connector for connecting Ethernet cables has two LEDs that show the operating status of Ethernet as shown in “Fig. 4.1-1 RJ45 Connector LEDs”.
“Table 4.1-2 Link Speed LEDs” and “Table 4.1-3 Link Status LEDs” show what each LED indicates.

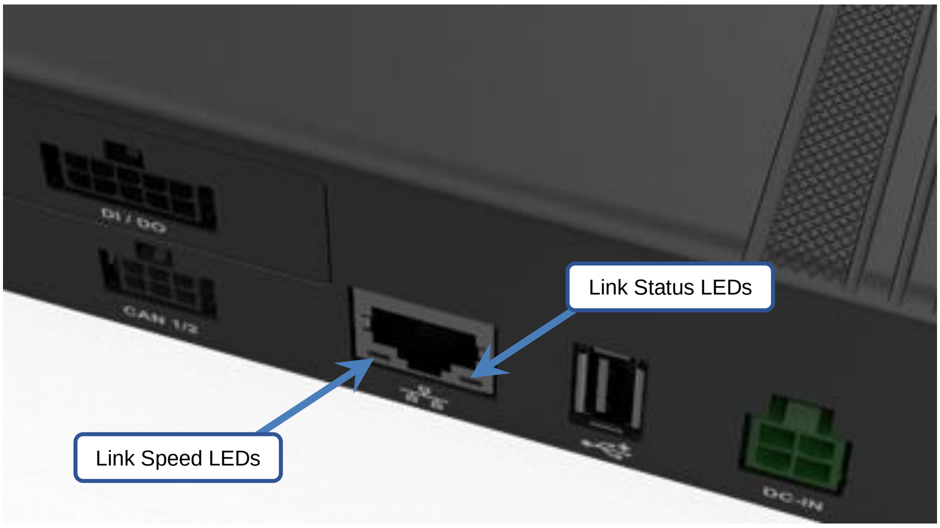


Fig. 4.1-1 RJ45 connector LEDs

Table 4.1-2 Link Speed LEDs

LED Status	Link Speed
Not lit	10 Mbps / Ethernet not connected
Lit: Green	100Mbps
Lit: Orange	1000Mbps

Table 4.1-3 Link Status LEDs

LED Status	Link Status
Not lit	Link not established
Lit: Yellow	Link established
Blinking: Yellow	Data is being sent/received

4.2. LTE･SMS

4.2.1. LTE

Table 4.2-1 LTE Module Device

Device Name
ttyUSB2

The following is an example that shows how to configure an LTE connection on the CONEXIOBlackBear using Network Manager.
The LTE module device that can be managed using Network Manager is ttyUSB2.

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

* “lte-wwan0” is an example of a name of a LTE connection to be newly configured.
* “[APN]” is the APN to which to connect.
* “[USER]” is the user name to be used for the connection to APN.
* “[PASSWORD]” is the password to be used for the connection to APN.

If PAP is specified as the type of connection authentication, configure the connection by adding an option that disables CHAP as shown below.

```
# 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
```

Where the ifconfig or ip addr command is used to check the network interface status, the LTE connection status will be shown in interface “ppp0”.

You can also check the status of the inserted SIM and LTE communication using the following Modem Manager command.

```
# mmcli -m 0
```

Normally, “/dev/ttyUSB2” is used by Network Manager and Modem Manager. Therefore, if you want to access the LTE Module while using these services, such as executing AT commands to the LTE Module, use “/dev/ttyUSB3”.

Note that if there is a mismatch between the SIM information and the connection information with the APN information (for example, if the connection information is still configured as the APN information of the SIM before replacement) after the SIM is replaced and the CONEXIOBlackBear is started up, the message “audit: type =1701...” message will be displayed. This message will not appear if you configure the correct APN information as a new connection.

When starting to use a new SIM or replacing a SIM with another SIM, it is recommended that you reboot the CONEXIOBlackBear once after setting the APN information as the new connection information to ensure consistency between the information of SIM and APN.

Table 4.2-2 LTE Enable/Disable Command

Command	Function
lte_on	Enables LTE from an application or command line (makes ttyUSB2 available). Turns the LTE module ON.
lte_off	Disables LTE from an application or command line (makes ttyUSB2 unavailable). Turns the LTE module OFF.

On OS Version 1.1.5 or later, it is possible to turn the LTE module on and off using commands. Use the “lte_off” command to turn the LTE module off, and use the “lte_on” command to turn the LTE module on again. (It takes a while to complete the execution of the “lte_off” command. Wait until the message “Power off LTE module” appears. The “lte_off” and “lte_on” commands are unique to the CONEXIOBlackBear.)

Executing “lte_off” will completely turn off the LTE Module and stop the Modem Manager, so none of the LTE communication functions, including SMS, can be used.

When the CONEXIOBlackBear is starting up or rebooting, the CPU accesses the LTE module for system configuration purposes, so the system will always be in the “lte_on” state after the startup/reboot of the CONEXIOBlackBear. Therefore, even if go-sys-reboot is executed in the “lte_off” state, the status will be “lte_on” after reboot. Moreover, before executing the “lte_off” command, be sure to wait until startup/reboot has been completed.

When go-sys-reboot is executed, the LTE Module power supply is turned off and on when rebooting, but if you want to reset only the LTE Module after startup, execute lte_off → delay → lte_on.

Note that to configure LTE connection using AT commands after shutting down the Network Manager and Modem Manager, wwan0 must be used as the LTE module device. The following is an example that shows how to connect using AT commands.

```
# 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$QCRMCALL=1,1¥r¥n' > /dev/ttyUSB2
# udhcpc -i wwan0
```

When a connection is configured as shown above, if the ifconfig or ip addr command is used to check the network interface status, the LTE connection status will be shown in “wwan0”.

4.2.2. SMS

Note that, as described in “3.5.2. Resume”, SMS transmission and reception require a subscription to a service that enables SMS communication. Moreover, SMS messages are automatically received if LTE has been already connected with the completion of LTE connection setup described in “4.2.1. LTE”.

The following shows the main SMS operations using Modem Manager.

[Viewing SMS messages in a list]

```
# 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)
```

Response example

[Viewing SMS content]

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

- * “[SMS Number]” refers to the number of the SMS message whose content is to be shown. (0 to 2 in the above list view example)
- * The sender’s telephone number and SMS content (message text) are shown as “Content”, and related information is shown as “Properties”.

[Creating SMS messages]

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

Response example

- * “[Phone Number]” refers to the recipient’s telephone number.
- * “[SMS Message]” refers to the SMS content (message text) to be sent.
- * The above message is shown for successful SMS creation.
  “[SMS Number]” refers to the number of the SMS message created. This number is used when sending the SMS message.

[Sending SMS messages]

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

- * “[SMS Number]” is the number of the SMS message to be sent. The same number shown at the time of SMS creation is used.

[Deleting SMS messages]

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

- * “[SMS Number]” is the number of the SMS message to be deleted.

4.3. WiFi

4.3.1. Connecting to the access point

Table 4.3-1 WiFi Device

Device Name
wlan0

By default, the CONEXIOBlackBear does not automatically enable the WiFi function on startup.

To enable WiFi function, execute the “wifi_on” WiFi enable command shown in “Table 4.3-2 WiFi Enable/Disable Command” (this command is unique to the CONEXIOBlackBear). By default, this WiFi enable command needs to be executed each time you turn on or reboot the CONEXIOBlackBear, even after WiFi connection has been configured using Network Manager.

To automatically enable WiFi function on startup, perform the procedure described in “4.3.4 Automatically enabling WiFi when the CONEXIOBlackBear is started up”.

Table 4.3-2 WiFi Enable/Disable Command

Command	Function
wifi_on	Enables WiFi from an application or command line (makes wlan0 available). Turns the WiFi communication module ON. Configures and enables the frequency band available for use with WiFi according to your LTE module model.
wifi_off	Disables WiFi from an application or command line (makes wlan0 unavailable). Turns the WiFi communication module OFF.

Executing the above WiFi enable command enables the wlan0 WiFi device that can be controlled using Network Manager. If a WiFi connection has been configured using Network Manager, executing the above command immediately starts the process based on the configured WiFi connection information. An example is shown below for the configuration of a WiFi connection using Network Manager (when the CONEXIOBlackBear is used as a WiFi slave).

```
# 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” is an example of a name of a WiFi connection to be newly configured.
* “[SSID]” refers to the SSID of the access point to which you want to connect.
* For “[SECURITY]”, configure a security setting that is suitable for connecting to the access point, such as “wpa-psk”.
* “[PASSPHRASE]” refers to the passphrase that you will need to connect to the access point.

To disable WiFi, execute the “wifi_off” WiFi disable command shown in “Table 4.3-2 WiFi Enable/Disable Command” (like the WiFi enable command, this command is unique to the CONEXIOBlackBear).

The WiFi connection information configured using Network Manager will continue to be stored even when the WiFi disable command is executed. If a WiFi connection has already been configured, executing the WiFi enable command enables the WiFi connection and starts the process based on the WiFi connection information.

When changing the WiFi configuration, such as when changing the settings used (2.4 GHz/5 GHz, slave/AP), start by executing the WiFi disable command, then change the settings. After completing configuration, execute the WiFi enable command.

4.3.2. 2.4 GHz Access point configuration

The CONEXIOBlackBear, which comes preinstalled with dnsmasq as standard as described in “4.1.3. DHCP/DNS server function”, also provides DHCP and DNS server functions for WiFi.

The following is a configuration example that shows how to use the CONEXIOBlackBear as a 2.4 GHz access point.

```
# 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
```

* Change the access point settings as necessary according to your operating environment.
For details on the parameters, see “https://developer.gnome.org/NetworkManager/stable/”.

* In the above example, the following settings are used.
SSID:BBtest-g
PASSPHRASE:blackbear
IP address: With 102.168.31.1 used by the CONEXIOBlackBear,
IP addresses on the 192.168.31.1/24 network are assigned.
WiFi Security:wpa-psk

* With the above setting, the CONEXIOBlackBear doubles as a DNS server via the dnsmasq function.

* The above setting also enables iPv4 routing, and registration to iptables is performed, including the IP masquerade for communication between LTE and WiFi.
If LTE connection is enabled, this allows LTE connection to access the Internet from devices that have been connected to the WiFi interface through the above setting.

Note that for OS V1.0.1 or earlier, named.service needs to have been disabled as shown below before configuring the CONEXIOBlackBear as an access point. From OS V1.0.1 or later, you do not need to disable named.service.

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

4.3.3. 5 GHz Access point configuration

The following shows a configuration example for using the CONEXIOBlackBear as a 5 GHz access point. The points to note are the same as those described in “4.3.2. 2.4 GHz Access point configuration”.

```
# 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
```

Note the available frequency bands described in “8.2.3. WiFi (wireless LAN) frequency bands (5 GHz frequencies)”. You cannot configure a valid connection if an unavailable channel is specified.

4.3.4. Automatically enabling WiFi when CONEXIOBlackBear starts up

As mentioned in previous sections, by default, the CONEXIOBlackBear does not automatically enable WiFi function on startup.

To enable WiFi function upon startup of the CONEXIOBlackBear, execute the following command. Executing this command causes the WiFi function to be automatically enabled from the next startup onward.

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

* This command can be used from OS V1.0.8 or later.

The WiFi enable command “wifi_on” described in “Table 4.3-2 WiFi Enable/Disable Command” obtains the model number information of the LTE module, and based on the information, executes the required setup. Therefore, to enable the WiFi function automatically upon startup of the CONEXIOBlackBear, note that simply adding “wifi_on” to rc.local will not work.

* In OS V1.0.3 or earlier, the “wifi_on” command does not enable the process of acquiring the LTE module model number information, etc., so “wifi_on” can be added to rc.local for this function to work.

To temporarily disable the WiFi function, which is configured to be enabled automatically upon startup, execute the previously mentioned “wifi_off” command. To enable the WiFi function again, execute the previously mentioned “wifi_on” command.

To return to the setting in which the WiFi function was not automatically enabled on startup, execute the following command. Executing this command prevents the WiFi function from being automatically enabled from the next startup onward.

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

* This command can be used from OS V1.0.8 or later.

It returns the CONEXIOBlackBear to its default state (the WiFi function is not enabled automatically on startup).

4.4. Bluetooth

Table 4.4-1 Bluetooth Device

Device Name
hci0

The CONEXIOBlackBear, which is equipped with BlueZ, provides hcitool, hciconfig, etc. The Bluetooth device that can be used with these tools is hci0.

To enable Bluetooth, execute the “bt_on” Bluetooth enable command shown in “Table 4.4-2 Bluetooth Enable/Disable Command” (this command is unique to the CONEXIOBlackBear). This command needs to be executed each time you turn on or reboot the CONEXIOBlackBear.

Executing this command enables the Bluetooth device hci0, which can be controlled with hciconfig, etc.

Table 4.4-2 Bluetooth Enable/Disable Command

Command	Function
bt_on	Enables Bluetooth from an application or the command line. (Makes hci0 available in the “hciconfig up” state.) Turns the Bluetooth module on.
bt_off	Disables Bluetooth from an application or the command line. (Changes hci0 to the “hciconfig down” state.) Changes the Bluetooth module to the Suspend state.

Moreover, using the following commands, you can check the Bluetooth device information of the CONEXIOBlackBear or scan surrounding Bluetooth devices.

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

When connecting to Bluetooth using the command line, interactive configuration of connection is possible with the execution of bluetoothctl. In this section, only an overview of an example in which bluetoothclt is used to connect is shown. (Details may vary for each device to connect to.)

```
# bluetoothctl                          Execute bluetoothclt from the command line.
  [bluetooth]# power on                 (The prompt changes.) Start up Controller.
  [bluetooth]# scan on                  Scan the surroundings to search for the device that
                                        you want to connect.
  [bluetooth]# devices                  Check the BD address of the device to which to pair.
  [bluetooth]# pair [BDaddress]         Pair the device.
  [bluetooth]# trust [BDaddress]        Trust the device.
  [bluetooth]# connect [BDaddress]      Connect to the device.
   ...
  [bluetooth]# help                     See Help.
   ...
  [bluetooth]# exit                     Finish bluetoothclt to exit interactive mode.
#
```

To disable Bluetooth (shut down connection), execute the “bt_off” Bluetooth disable command shown in “Table 4.4-2 Bluetooth Enable/Disable Command” (like the “bt_on” command, this command is unique to the CONEXIOBlackBear).

To enable Bluetooth again, execute the “bt_on” command as shown above.

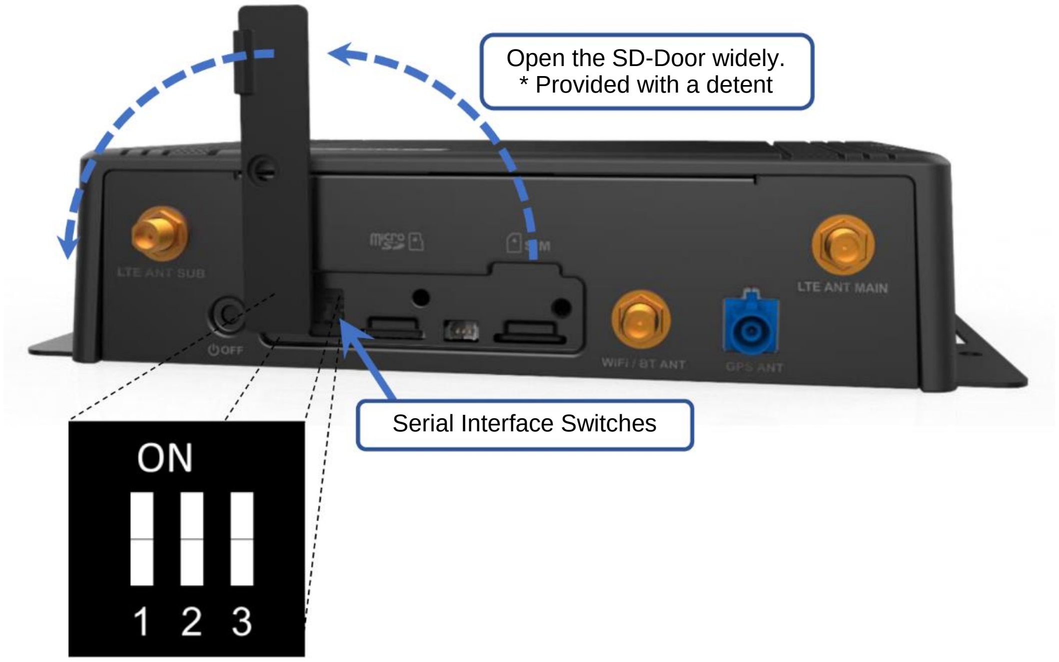
4.5. Serial

Table 4.5-1 Serial Communication Device File

Device File
/dev/ttymxc2

The device file used for serial communication of the CONEXIOBlackBear is “/dev/ttymxc2”. RS-232C, RS-485 half duplex, and RS-485/RS-422 full duplex are supported as serial interfaces. Depending on your purpose, you can switch between them using the 3P DIP switches found inside the “SD-Door” shown in “Fig. 4.5-1 Serial Interface Switches”. Switch each 3P DIP switches when the power is turned OFF.

To use the switches, open the SD-Door wide to a 180° angle.
In this Figure, the 3P DIP switches are partially hidden behind the SD-Door, which is only opened 90°.



Open the SD-Door widely.
* Provided with a detent

Serial Interface Switches

Fig. 4.5-1 Serial Interface Switches

The serial interface switch settings are shown in Table 4.5-2 and Table 4.5-3.
Use the SW1 and SW2 shown in Fig. 4.5-1 to change between RS-232C, RS-485 half duplex, and RS-485/RS-422 full duplex, and use SW3 to set up electrical termination when RS-485/RS-422 is used.

Table 4.5-2 Settings to Change Serial Interfaces

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

Table 4.5-3 Electrical Termination Settings

SW3	Electrical Termination
OFF	OPEN
ON	120Ω

The CONEXIOBlackBear is equipped with uarttest, which is a tool used to test the operation status of serial communication. As shown in the following command example, the serial communication connection can be checked with data communicated between uarttest-implemented devices or looped back.

```
# uarttest -s -e -p /dev/ttymxc2 -b 115200 -q -c -a 2000
```

Note that the following settings must be configured in the program in order to use RS-485 from the program.

```
/*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

Table 4.6-1 CAN Devices

Device Name
can0
can1

The CONEXIOBlackBear comes standard with two CAN port channels, which are can0 and can1 compliant with CAN Version 2.0B active. To enable these CAN ports, execute the following commands individually for each one that you want to enable.

Example) Enabling the can0 port

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

The CONEXIOBlackBear, which can use the Linux SocketCAN function, can directly send and receive raw CAN frames (supports the standard and extended formats). The CONEXIOBlackBear is also equipped with CAN-Utils, enabling the use of “candump”, “canecho”, “cansend”, and “cansequence”. Send and receive tests can be performed with the “cansend” and “candump” commands, respectively. Note that the SAE J1939 protocol is not supported.

Example) Outputting data to the can0 port

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

Example) Inputting data from the can1 port

```
   # candump can1
```

Note that no terminating resistors are installed on the CAN port to allow the CONEXIOBlackBear to be used at both the midpoint and the end of the CAN-BUS. (No switching function is available for electrical termination.)

4.7. DI/DO

4.7.1. DI/DO control ports

The DI/DO of the CONEXIOBlackBear is controlled using the control gpio ports shown in “Table 4.7-1 DI/DO Control Ports”. The DI ports and DO ports each come in four ports for digital input and output, respectively, and the PWR_CTL ports are provided to control pull up of the DI or DO ports.

Table 4.7-1 DI/DO Control Ports

Control Port	Function
/sys/class/gpio/gpio9	DI1
/sys/class/gpio/gpio205	DI2
/sys/class/gpio/gpio101	DI3
/sys/class/gpio/gpio171	DI4
/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

The ports shown above are not enabled on startup. To use them, write all of the above control gpio port numbers to /sys/class/gpio/export as shown below so that the ports will be enabled.

Example) Enabling the DI1 control gpio port

```
   # echo 9 > /sys/class/gpio/export
```

Next, configure the settings for the use of the DI ports, DO ports, and PWR_CTL ports. To configure the settings, write the values shown in "Table 4.7-2 Function Setup on the Ports" for the direction argument of each control gpio port.
(Write "in" for the DI ports, "out" for the DO ports, and "out" for the PWR_CTL ports.)

Table 4.7-2 Function Setup on the Ports

Control Port	Setting values for direction
/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

Example) Setting the DI2 control port as a DI port

```
   # echo "in" > /sys/class/gpio/gpio205/direction
```

Furthermore, the PWR_CTL_DI and PWR_CTL_DO ports can be set to "no-voltage contact" for the DI and DO ports, or to "pull up" to the input voltage to the CONEXIOBlackBear. To configure the settings, write the setting values shown in "Table 4.7-3 PWR_CTL Port Setting Values" as the value argument for each port.

Table 4.7-3 PWR_CTL Port Setting Values

Control Port	Setting values for value	
	0	1
/sys/class/gpio/gpio2	DI No-voltage contact	DI pull up
/sys/class/gpio/gpio5	DO No-voltage contact	DO pull up

Example) Setting to pull up the DI ports to the input voltage for the CONEXIOBlackBear

```
   # echo 1 > /sys/class/gpio/gpio2/value
```

(Note): The above setting of "no-voltage contact" or "pull up" is effective over the entire range of DI1 to DI4. It is not possible to configure each port individually. The settings for gpio5 are also effective over the entire range of DO1 to DO4.

Each DO port produces an output of "Hi" level (for DO pull up) or "Hi impedance " (for no-voltage contact) if "0" is written as the value argument of the corresponding DO control gpio port after configuring the settings as shown above, and if "1" is written, it produces an output of "Lo" level.

Example) Executing “Lo” level output on the DO3 port

```
# echo 1 > /sys/class/gpio/gpio32/value
```
→ DO3 port output: “Lo” level

Moreover, by reading the value from the value argument of each DI control gpio port, you can obtain the input status for the DI port, which is “0” in the case of “Hi” level (for no-voltage contact) or “Hi impedance” (for DI pull up), or “1” in the case of “Lo” level.

Example) Retrieving the input status for the DI4 port

```
# cat /sys/class/gpio/gpio171/value
0
```
→ Status of DI4 port input: “Hi” level/“Hi impedance”

You can check the status of the control gpio port using the following command. With this command, the levels of the control gpio ports are shown as “lo” if value is “0” and “hi” if value is “1”. Note that this notation is the reverse of the input/output levels shown for the DI/DO ports.

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

4.7.2. How to use the DI

The DI is intended for connection with an open collector output device. For example, with a device such as a laser sensor connected, its ON/OFF status can be notified to the CONEXIOBlackBear. The circuit diagram of the DI is shown in “Fig. 4.7-1 DI Circuit Diagram”.

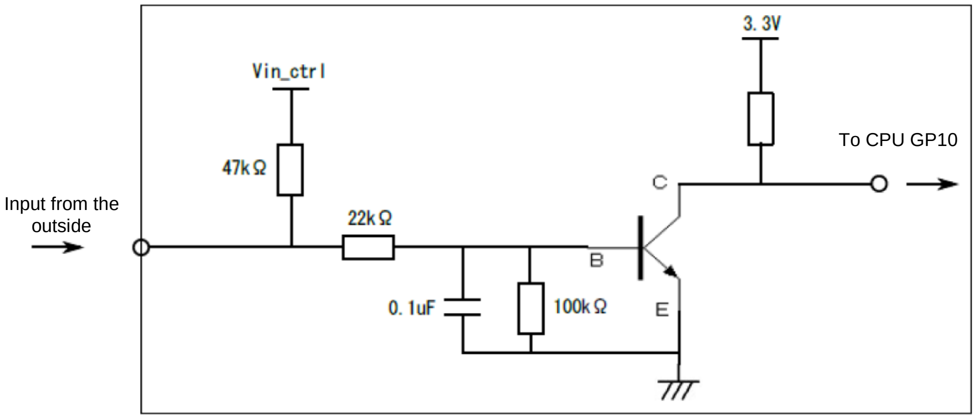


Fig. 4.7-1 DI Circuit Diagram

The left side is outside of the CONEXIOBlackBear, and the area enclosed in a square frame is DI circuit inside the CONEXIOBlackBear. Pull up is turned on or off using a FET inside the CONEXIOBlackBear. Pull up voltage Vin_ctrl takes the same voltage value as the power supply voltage of the CONEXIOBlackBear.

- When connecting with a no-voltage device (Internal pull up: ON)
  Be sure that the device to be connected has no problem meeting the following conditions.
  - When the device connected has a high impedance: A voltage of Vin_ctrl must be applied.
  - When the device connected is low: A current of Vin_ctrl divided by 47 kΩ must be applied. (Approx. 510 μA in the case of 24 V)
- When connecting with a voltage-carrying device (Internal pull up: OFF)
  It is recommended that the voltage applied to the DI is 25 V or less, which is equal to or less than the supply voltage of the CONEXIOBlackBear.

As mentioned earlier, the device to be connected to the DI is expected to be an open collector output device. Heed the following notes on similar circuit parts such as switches and relays.

- Switches: May not operate properly depending on the minimum current requirements, if any.
- Relays: The minimum applicable load for mechanical relays and the output ON voltage drop for solid-state relays may prevent normal operation.

In addition, take note that the circuit, which has resistors and capacitors for the purposes of circuit protection and noise countermeasures as shown in “Fig. 4.7-1 DI Circuit Diagram”, is not suited to the acquisition of rapid changing signals such as pulses.

4.7.3. How to use the DO

The DO is intended to communicate data, such as the status (ON/OFF, etc.) indicating the processing results of the CONEXIOBlackBear, to the devices connected via open collector output. The circuit diagram of the DO is shown in “Fig. 4.7-2 DO Circuit Diagram”.

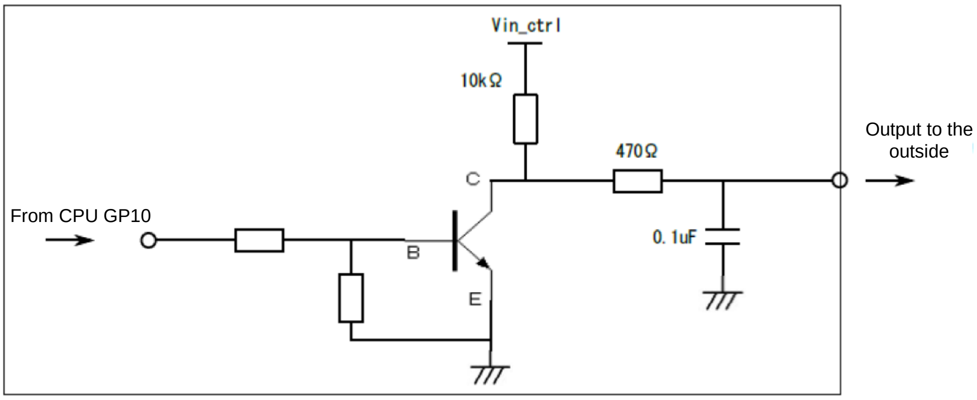


Fig. 4.7-2 DO Circuit Diagram

The right side is outside of the CONEXIOBlackBear, and the area enclosed in a square frame is DO circuit inside the CONEXIOBlackBear. As is the case for the DI, the pull up for the DO is turned on or off using a FET inside the CONEXIOBlackBear.
Pull up voltage, which has the same name (Vin_ctrl) in “Fig. 4.7-1 DI Circuit Diagram” and “Fig. 4.7-2 DO Circuit Diagram”, can be separately controlled each for DI and DO as shown in “4. DI/DO control ports”.

- When connecting with a no-voltage device (Internal pull up: ON)
  Ensure that the pull up voltage (Vin_ctrl) can be applied to the connected device and that the Hi level can be recognized with the pull up resistor constant value.

- When connecting with a voltage-carrying device (Internal pull up: OFF)
  When pulling up on the external device, we recommend a voltage of 40 V or less, which is equal to or less than the CONEXIOBlackBear power supply voltage. The lower limit of the current that a transistor can run at Low level output is about 3.7 mA.

Also, as shown in Fig. 4.7-2 DO Circuit Diagram”, because resistors are placed in series for circuit protection and the collector current of the transistor is limited, DO cannot pass a current that can directly turn on the relay coil. Therefore, when a relay is used as a connection device, note that an external circuit is required to turn on the relay coil.

4.8. GNSS

Table 4.8-1 GNSS Module Device File

Device File
/dev/ttymxc4

The CONEXIOBlackBear comes standard with the u-blox NEO-M8N as the GNSS module. The location information obtained by the GNSS module is output to the device file "/dev/ttymxc4" at intervals of 1 second as serial communication output.

Example) How to obtain location information from the GNSS module - 1

```
  # ln /dev/ttymxc4 /dev/gpsdevice
  # stty -F /dev/gpsdevice ispeed 9600 && cat < /dev/gpsdevice
```

To just simply check whether location information is output from the GNSS module, the following method can be used.

Example) How to obtain location information from the GNSS module - 2

```
  # cat /dev/ttymxc4
```

The CONEXIOBlackBear uses the NEO-M8N by default setting. This default setting uses "GPS + GLONASS + SBAS + QZSS" as the GNSS satellites to obtain location information.
Galileo can be used simultaneously with these satellites, but in order to use it, you need to configure NEO-M8N to use Galileo in addition, as in the example command below.

Example) Command example to enable "GNSS + GLONASS + Galileo + SBAS + QZSS" with the addition of Galileo

```
  # echo -e "¥xB5¥x62¥x06¥x3E¥x3C¥x00¥x00¥x00¥x20¥x07¥x00¥x08¥x10¥x00¥x01¥x00
  ¥x01¥x01¥x01¥x01¥x03¥x00¥x01¥x00¥x01¥x01¥x02¥x04¥x08¥x00¥x01¥x00¥x01¥x01¥x0
  3¥x08¥x10¥x00¥x00¥x00¥x01¥x01¥x04¥x00¥x08¥x00¥x00¥x00¥x01¥x01¥x05¥x00¥x03¥x
  00¥x01¥x00¥x01¥x01¥x06¥x08¥x0E¥x00¥x01¥x00¥x01¥x01¥x30¥xAD" > /dev/ttymxc4
   # echo -e "¥xB5¥x62¥x06¥x17¥x14¥x00¥x00¥x41¥x00¥x02¥x00¥x00¥x00¥x00¥x00¥x00
  ¥x00¥x01¥x00¥x00¥x00¥x00¥x00¥x00¥x00¥x00¥x75¥x57" > /dev/ttymxc4
```

The location information from the GNSS module is output as messages in NMEA format. "Table 4.8-2 NMEA Messages" shows the types of messages that are output under the default settings of the GNSS module.

Table 4.8-2 NMEA Messages

Output Message	Main Information
GNRMC	Location, positioning date, speed, direction
GNVTG	Speed, direction
GNGGA	Location, positioning time, the number of satellite data captures, accuracy
GNGSA	Satellite used for positioning, accuracy reduction rate
GPGSV	Satellite information (GPS, QZSS)
GLGSV	Satellite information (GLONASS)

The CONEXIOBlackBear, which is designed for the Michibiki Quasi-Zenith Satellite System (QZSS) as the satellite used to obtain location information, only supports the function of the "L1C/A: Satellite Positioning Service". The "L1S: Sub-meter Level Augmentation Service" and the "L6: Centimeter Level Augmentation Service" are not supported.

4.9. USB

Table 4.9-1 USB Control Port and Setting Value

Control Port	Setting values for direction
/sys/class/gpio/gpio174	out

Table 4.9-2 Memory Drive Device File

Device File
/dev/sda1

The USB port of the CONEXIOBlackBear is not enabled upon startup. To make the USB port available for a USB memory drive or other USB devices, execute the following command.

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

By executing this command, the a USB memory drive connected to the USB port is recognized as a "/dev/sda1" device file as shown in "Table 4.9-2 USB Memory Drive Device File". Mounting this device file to any of the directories enables you to read from and write to the USB memory drive.

Note that the "1" in "/dev/sda1" represents the first partition. If the USB memory drive has multiple partitions, each of them is recognized as a device file, appearing as "/dev/sda2" and "/dev/sda3", for example.

Note that when other types of USB devices are connected, the recognized device files vary depending on the connected USB device.

4.10. SD (Storage)

Table 4.10-1 microSD Card Device File

Device File
/dev/mmcblk3p1

The microSD slot of the CONEXIOBlackBear supports the "UHS-I SDR104" microSDXC, as well as exFAT. (Because support for exFAT is implemented using FUSE, we do not recommend using udev to mount exFAT filesystems.. The "transport endpoint not connected" error will occur.)

The microSD card inserted in the microSD slot is recognized as a "/dev/mmcblk3p1" device file. Mounting this device file to any of the directories enables you to read from and write to the microSD card.

Note that the "p1" of "/dev/mmcblk3p1" represents the first partition. If the microSD card has multiple partitions, each of them is recognized as a device file, appearing as "/dev/mmcblk3p2" and "/dev/mmcblk3p3", for example.

4.11. LED

4.11.1. USER LED

As shown in “Fig. 2.4-1 Interface Layout”, the CONEXIOBlackBear has a bicolor “USER LED” on the top surface of the top cover.
The “File System Nodes” for controlling the bicolor LED are shown in “Table 4.11-1 USER LED”.

Table 4.11-1 USER LED

USER LED	Color	File System Node
user1-led	Red	/sys/class/leds/user1-led
user2-led	Green	/sys/class/leds/user2-led

This USER LED can be controlled by configuring a value on the file under each of the File System Nodes shown in Table 4.11-1. The control methods common to USER LEDs are shown in “Table 4.11-2 USER LED Control Method”.

Table 4.11-2 USER LED Control Method

Control File	Setting Value	Control Result
brightness	0	LED OFF
	1	LED ON
trigger	none	LED OFF * Discontinues the illumination pattern configured by trigger to turn the LED off.
	timer	Default Value delay_on : 500ms delay_off : 500ms * The delay_on and delay_off values can be configured only after “timer” has been set for “trigger”. * The delay_on and delay_off values revert to their default values if any value is set for “trigger”.
	heartbeat	Default Value invert : 0 * Causes the LED to blink in a heartbeat-like pattern. * The blinking pattern inverts if “invert” is set to 1. * The “invert” value reverts to its default value if any value is set for “trigger”.
	oneshot	Default Value delay_on : 100ms delay_off : 100ms invert : 0 * Illuminates only once or shuts off according to the above settings if any value is configured for “shot”. * The delay_on, delay_off, invert, and shot values can be set only after “oneshot” has been set for “trigger”. * The illumination pattern inverts if “invert” is set to 1. * The delay_on, delay_off, and invert values revert to their default values if any value is set for “trigger”.

4.11.2. WiFi/Bluetooth LED

As shown in “Fig. 2.4-1 Interface Layout”, the CONEXIOBlackBear has a bicolor “WiFi/Bluetooth LED” on the top surface of the top cover.
The colors of the bicolor LED are as shown in “Table 4.11-3 WiFi/Bluetooth LEDs”.

Table 4.11-3 WiFi/Bluetooth LED

USER LED	Color
Bluetooth	Red
WiFi	Green

These LEDs illuminate, shut off or blink depending on the connection or communication status of each communication channel. “Table 4.11-4 WiFi/Bluetooth LED Illumination Conditions” shows the relationship between the communication channel statuses common to these LEDs and LED illumination, shutoff, and blinking.

Table 4.11-4 WiFi/Bluetooth LED Illumination Conditions

LED	State of Channel
Not lit	Not connected * Does not illuminate if the CONEXIOBlackBear is not connected to the other end/network even when the communication module is turned on.
Illuminates	Connected * Illuminates if the CONEXIOBlackBear is connected to the other end/network and has no packet in the communication channel.
Blinking	Communicating * Blinks if the CONEXIOBlackBear is communicating with the other end and has packets in the communication channel.

4.11.3. LTE signal strength LEDs

As shown in “Fig. 2.4-1 Interface Layout”, the CONEXIOBlackBear has three green “LTE signal strength LEDs” on the top surface of the top cover.
These LEDs illuminate or shut off in response to the “rssi” value, which is read at intervals of approximately 5 seconds from the LTE module SIM7600 when the CONEXIOBlackBear is connected to the LTE network. “Table 4.11-5 LTE Signal Strength LED Illumination Conditions” shows the relationship between the “rssi” value and the illumination and shutoff of the LTE signal strength LEDs. The illumination status does not respond immediately and is not accurate, so it should only be used as an approximate reference.

Table 4.11-5 LTE Signal Strength LED Illumination Conditions

rss1-led	rss2-led	rss3-led	rssi value
▯	▯	▯	-107 dBm or less
🟩	▯	▯	-105 dBm ~ -87 dBm
🟩	🟩	▯	-85 dBm ~ -67 dBm
🟩	🟩	🟩	-65 dBm or more

▯ : Not lit 🟩 : Lit

4.12. USER Switch

Table 4.12-1 USER Switch Input Device File

Input Device File
/dev/input/event0

The CONEXIOBlackBear has a USER Switch for which the usage method can be configured by the user. The pressing and releasing of the USER Switch are detected as events generated in the input device file “/dev/input/event0” for the USER Switch.

Generated events are defined as follows in “/usr/include/linux/input.h”.

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

The following events are generated if the USER Switch is pressed and released. “XXXXXXXXXX.XXXXXX” is Unixtime.

Example) Event generated when the USER Switch is pressed

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

Example) Event generated when the USER Switch is released

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

The above events can be checked using the following command.

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

The CONEXIOBlackBear is equipped with “evtest”, which can be verified by running this command. (The above examples of generated events are part of the results of executing the evtest command.)

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

4.13. CPU Temperature

The CONEXIOBlackBear includes a function to measure the temperature of its CPU. For reference purposes, this chapter explains this feature, although it is not presented as an external interface in “Fig. 2.4-1 Interface Layout”.

To obtain CPU temperature information, execute the following command. An example execution result is shown in the line following the command.

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

The example execution result “39315” shown above represents “39.315 °C”.

5. OS Update (Initialization)

There are two main ways to update the OS of the CONEXIOBlackBear. One method is an “incremental update,” and the other is a “clean installation”.

This chapter provides instructions for “clean installation,” in which the entire OS of the CONEXIOBlackBear is replaced (initialized) using a microSD card.

For details on “incremental update”, see “6. Incremental OS Update”. To learn how to update other software and firmware, see the “Simplified Programming Manual”.

5.1. How to Check the OS Version

The version information of the OS installed on the CONEXIOBlackBear is stored in “/etc/os-release”. The following is an example showing stored version information.

```
# 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)"
#
```

In the above version information example, the “1.1.5” appearing at the end of the “VERSION” line and other lines refers to the version No., which is updated with each release. When changing to a different release version of the OS, etc., check this number.
The details of the “VERSION” line are as follows. (Based on the above example of version information)

"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)

In the event that something goes wrong in the operation of the CONEXIOBlackBear, be sure to include the OS version information in the analysis request.

In addition, by executing the OS update described in “5.2. OS Update Using a microSD Card”, the firmware of the MCU, which is the power control processor, will also be updated at the same time (if there are any updates to be made). The MCU firmware can only be updated in this manner. If the MCU firmware is updated, users will be notified of it through the support site.
The version information of the MCU firmware is stored in “/proc/ms5776_mcu_ver/mcu_version”. The following is an example showing stored version information.

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

In the above version information example, the “V0.20” refers to the version No. of the MCU.

To check the version information of the OS and the MCU, reboot the CONEXIOBlackBear after the OS update.

5.2. OS Update Using a microSD Card

This chapter describes the procedure for updating the OS using a microSD card.

5.2.1. Downloading “SD_burn_EMMC”

From the server at the following URL, download the “SD_burn_EMMC-VXXX.zip” file for the version to which you want to perform update.
(XXX is replaced by the version No. that is updated with each release → Example: SD_burn_EMMC-V115.zip in the case of OS V1.1.5)

Server URL: https://www.bb-customer-support.conexiot.jp/ (*This is the URL for the CONEXIO IoT support website.)

Unzip the downloaded “SD_burn_EMMC” file and save it to a Windows PC virtual machine or Ubuntu on a Linux PC. (Install Media creation is intended to be performed on Ubuntu 18.04.)

5.2.2. Procedure for Install Media (microSD card) creation

Insert the microSD card that you want to use as Install Media. Use a microSD card with 8 GB of space or more. The Install Media cannot be created with 4 GB of space.

After insertion, check the device name of the microSD card using the “lsblk” command, etc. In this example, the name is shown as “/dev/sdX”. If the microSD card is mounted as soon as you insert it, unmount it with superuser privileges.

Go to the Ubuntu “SD_burn_EMMC” directory to which you saved the file.

```
# cd SD_burn_EMMC
```

Using superuser privileges, execute the shell script “mkSD.sh” found in the “SD_burn_EMMC” directory.

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

After the execution of “mkSD.sh” has been completed, remove the microSD card from the PC and insert it into the PC again.

After insertion, check the device name of the microSD card again using the “lsblk” command, etc. The name is again shown as “/dev/sdX” as an example. Again, if the microSD card is mounted, unmount it using superuser privileges.

Using superuser privileges, execute the shell script “updateSD.sh” found in the “SD_burn_EMMC” directory.

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

* If “Proceed anyway? (y,N)” is displayed during the process, enter “y”.

* It has been found that the steps to take at this point may vary depending on your execution environment.
If "Proceed anyway? (y,N)" is displayed but the program immediately exits with "Done.", execute "sudo . /mkSD.sh /dev/sdX" and then execute "sudo . /updateSD.sh /dev/sdX" without removing and inserting the microSD Card.

* Moreover, if the device name of the microSD card is “mmcblk0”, for example, and if the partition name generated there is not a device name + “any number from 1 to 3,” such as “mmcblk0p1” to “mmcblk0p3”, specify the entire name excluding the final number for “/dev/sdX” when executing “/updateSD.sh”. (In this example, the command would be “sudo ./updateSD.sh /dev/mmcblk0p”.)

The creation of Install Media will be completed upon execution of “updateSD.sh”. Remove the microSD card from the PC.

5.2.3. Procedure for OS update on the CONEXIOBlackBear

Insert the microSD card on which you created the Install Media into the microSD slot of the CONEXIOBlackBear.

After inserting the microSD card, turn ON the CONEXIOBlackBear while pressing the USER SW. The CONEXIOBlackBear will boot from the microSD card.

After the power is turned on, and after all the LEDs turn on, all the LEDs except the power LED will turn off, and the power LED will start blinking. At this point, lift your finger from the USER SW.

After the CONEXIOBlackBear has been booted from the microSD card, all LEDs will illuminate when rewriting of the eMMC for OS update starts. Wait a moment for OS update to complete.

After the OS update has been completed, the power LED will stay lit and all other LEDs will start blinking. After this, you can turn off the CONEXIOBlackBear. This update procedure takes approximately 11 minutes for OS V1.0.0 or later. Be sure that you do not accidentally turn off the power before the OS update has been completed.

When the OS update has been completed, the Linux console of the CONEXIOBlackBear shows a message that prompts you to “remove the microSD card and reboot the OS”.

After reboot, check the version information of the OS as instructed in “5.1. How to Check the OS Version” as necessary.

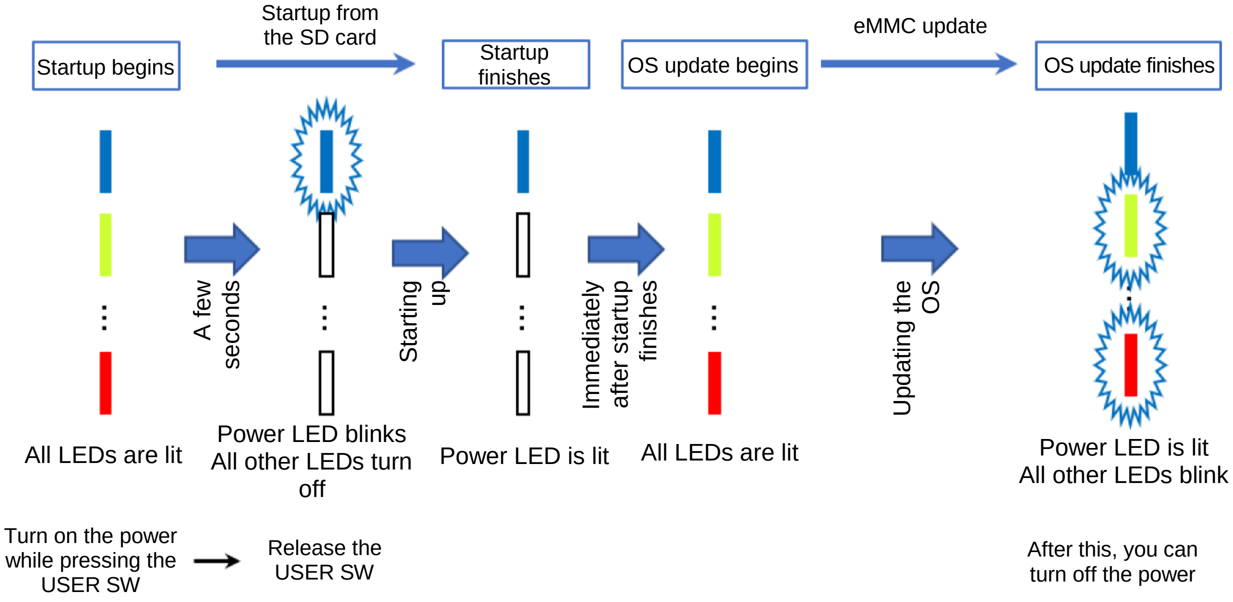


Fig. 5.2-1 LED Indication for OS Update

6. Incremental OS Update

6.1. Overview

This chapter provides the instructions for incremental updates, which are used to modify only the updated portions of the OS, unlike the update method described in “5. OS Update (Initialization)”, which replaces the entire OS.

The advantage of this method is that, unlike the “clean installation” method, it enables you to update the OS without deleting any applications that have been already installed by the user.

However, incremental update OS packages support only limited update combinations from certain OS versions to other specific OS versions. For this reason, it is important to be aware of the supported versions.

The following section describes the update procedure for an incremental update using an incremental update OS package.

6.2. Update Procedure

6.2.1. Obtaining the incremental update OS package

Incremental update OS packages are available at the CONEXIO IoT support website at the following URL.

Server URL: https://www.bb-customer-support.conexiot.jp/

From the above website, download the incremental update OS package of the version combination for your update. For example, the incremental update OS package to be used for the update from OS V1.1.5 to OS V1.1.9 is named as follows.

OS V1.1.5 → OS V1.1.9 Incremental update OS package: diffOS-V115V119.zip

After downloading the incremental update OS package from the server, transfer it to the “/tmp”, directory etc. on the CONEXIOBlackBear. You can transfer it via the network or using a microSD card.

To download an incremental update OS package directly from the CONEXIOBlackBear using the “wget” command, etc., access the directory at the following URL, where the files of the same names are stored.

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

The OS has two blocks, which are the “A block” and the “B block”, as described in “eMMC Partition Configuration” in the “Simplified Programming Manual”. Place the incremental update OS package in the block that you want to update.

The following section describes the operation procedure for incremental updates with an example from OS V1.1.5 to OS V1.1.9.

6.2.2. Performing incremental OS updates

First, go to the directory (“/tmp” in the following example) where the incremental update OS package of your desired versions (“diffOS-V115V119.zip” in the following example) is stored.

```
# cd /tmp
```

Execute the unzip command to unzip “diffOS-V115V119.zip” and extract the installation script “install_diffOS.sh” and the package components.

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

Go to the directory diffOS-V115V119 where the installation script “install_diffOS.sh” and package components that you extracted are stored.

```
# cd diffOS-V115V119
```

Execute the installation script “install_diffOS.sh” to start the incremental OS update.

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

Once script execution has started, you will be asked if you want to execute the OS update as shown above. To execute, enter “y” or “Y”. To abort the OS update, enter a different character. If a character other than “y” or “Y” is entered, the update process will be terminated, and the text “abort.” will be displayed.

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

If “y” or “Y” is entered, the incremental OS update process will start.

```
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.
     .
     .
```

If the OS version (current OS version already installed on the CONEXIOBlackBear) to which the OS incremental update is to be executed is different from the OS version to which the OS package for differential update is to be executed (OS V 1.1.5 in this example), the following message will appear, and the update process will not be executed. Check the version of the installed OS against the OS version intended for the incremental update OS package.

```
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.
```

While the OS incremental update process is being executed, a message indicating the processing content is output. If the process is successful, the following success message is output. After a countdown of approximately 10 seconds, the system will automatically restart.

```
2021-03-16 10:17:50 MS5776 OS update success.
After the countdown, reboot the MS5776 : 10
       ↓ (A countdown starts and the remaining seconds decrement.)
After the countdown, reboot the MS5776 : 0
#
       ↓
  (go-sys-reboot is executed automatically.)
```

Rebooting completes the incremental OS update process.

Due to the execution of a reboot, the connection will be temporarily disconnected if the OS incremental update process was being executed via a remote connection. However, the connection information that was configured in the CONEXIOBlackBear before the OS incremental update process will remain, so the connection can be established again after the time required for the reboot has elapsed.

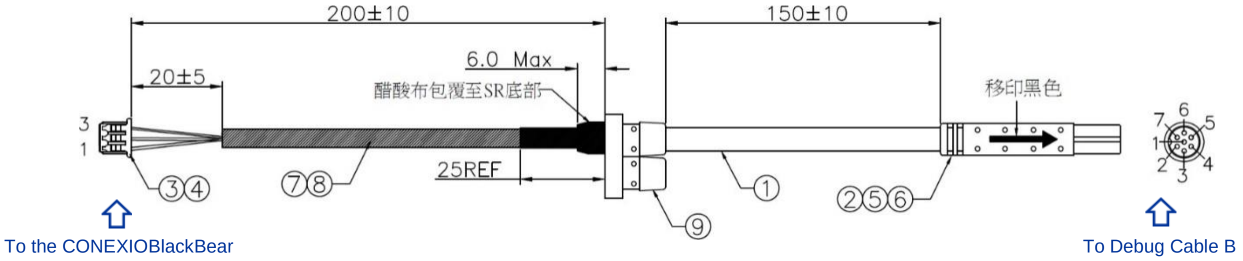
Note that, as described in “4.1.2. Static IP connection”, it is not recommend to use the connection name ‘Wired connection 1’, which is used by default, when a static IP address is configured as the Ethernet connection of the CONEXIOBlackBear to access the CONEXIOBlackBear via Ethernet. In order to avoid possible trouble during reboot, use another connection name when you want to use a static IP address as the Ethernet connection for the CONEXIOBlackBear.

7. Development Tools

7.1. Debug Cable

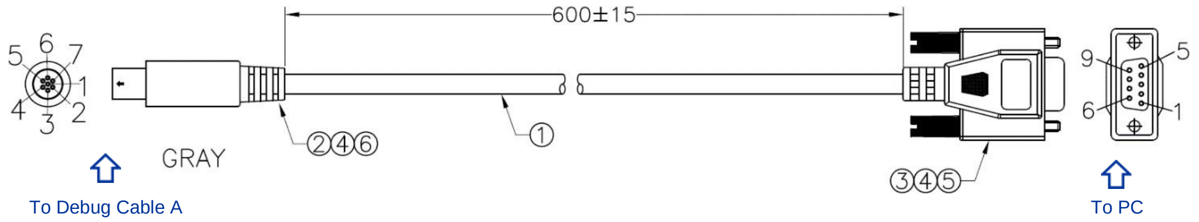
CONEXIO Corporation provides “Debug Cables,” which can be used to initialize the CONEXIOBlackBear or perform software development or operation verification on the CONEXIOBlackBear.
To acquire these “Debug Cables”, which are not included in the standard sets, contact the vendor from which you purchased the CONEXIOBlackBear, if necessary.

The “Debug Cables” are the two cables shown in “Fig. 7.1-1 Debug Cable A” and “Fig. 7.1-2 Debug Cable B”. Use these two cables by connecting the connector on the right end shown in “Fig. 7.1-1 Debug Cable A” to the one on the left end shown in “Fig. 7.1-2 Debug Cable B”. Connecting the CONEXIOBlackBear to a PC using the “Debug Cable” allows you to use the Linux console of the CONEXIOBlackBear from the PC.



(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 Terminal silver plating (SPD-07F1-01)(RoHS)	1pcs
(3)	MOLEX 51021 PH: 1.25mm PHOSPHOR BRONZE TIN-PLATED or equivalent (RoHS)	3pcs
(4)	MOLEX 51021 PH: 1.25mm 1*3P NYLON66 WHITE or equivalent (RoHS)	1pcs
(5)	INNER MOLD: PP, transparent (RoHS)	0.9g
(6)	OVER MOLD :PVC 30P GRAY(427C)(-40°C~105°C) (RoHS)	3.2g
(7)	Black acetate cloth: 25*180mm	1pcs
(8)	Black acetate cloth: 25*30mm	1pcs
(9)	SR MOLD:PVC 45P BLACK(-40°C~105°C) (RoHS)	1.7g

Fig. 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 Terminal silver plating (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

Fig. 7.1-2 Debug Cable B

7.2. Debug Port

To connect to the CONEXIOBlackBear, open the SD-Door and connect the 3P connector on the left end of “Fig. 7.1-1 Debug Cable A” to the “Debug Port” found between the microSD slot and the microSIM slot as shown in “Fig. 7.2-1 Connecting the Debug Cable to the CONEXIOBlackBear”.

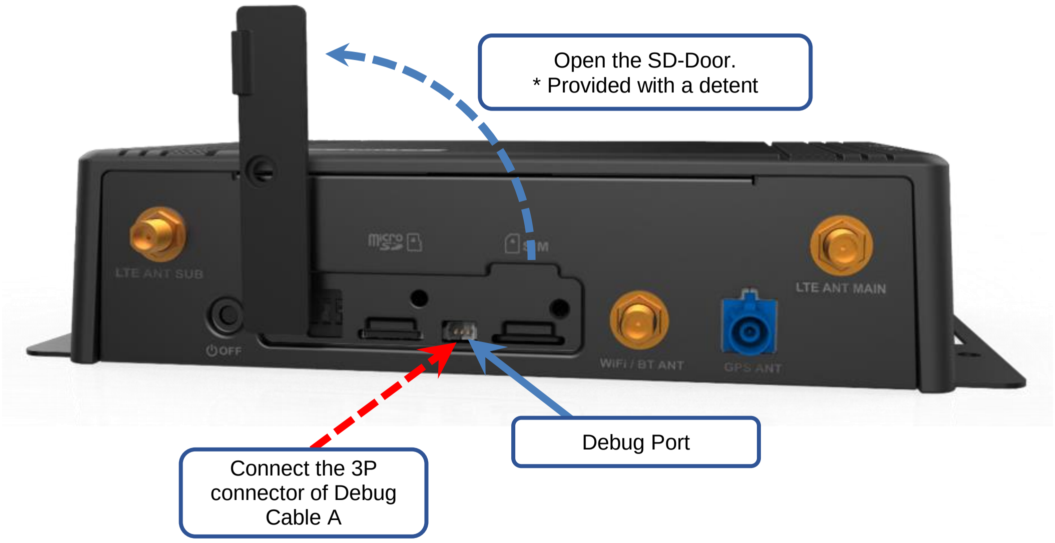


Fig. 7.2-1 Connecting the Debug Cable to the CONEXIOBlackBear

To connect to a PC, connect the RS232C connector (D-sub, 9-pin female terminal) on the right end of “Fig. 7.1-2 Debug Cable B” to the PC’s serial port or to the PC’s USB port using a commercially available USB-serial conversion cable.

The above connection allows you to operate the CONEXIOBlackBear through the Linux console screen using a terminal emulator on the PC, such as Tera Term.

The serial communication settings for the terminal emulator are shown in “Table 7.2-1 Serial Communication Settings for the Terminal Emulator”.

Table 7.2-1 Serial Communication Settings for the Terminal Emulator

Item	Setting Value
Communication speed	115,200 bps
Data length	8 bits
Parity bit	none
Stop bit	1 bit
Flow control	none

You can also log in via the network using ssh if network configuration has been completed and you know the IP address of the CONEXIOBlackBear.

8. Notes

8.1. Precautions for Safe Use

Please read the manual and the related materials and follow the instructions carefully.

The following categories describe the degree of harm or damage that can occur if the product is used incorrectly.

⚠ **Danger**	**Improper handling of the product carries a significant risk of death or serious injury.**
⚠ **Warning**	**Improper handling of the product may result in death or serious injury.**
⚠ **Caution**	**Improper handling of the product may result in injury or property damage.**

⚠ Danger

- Do not use, store, or leave the CONEXIOBlackBear in places where it will be exposed to high temperatures, contained heat, or direct sunlight. Failure to observe this instruction may result in fire, malfunction, electric shock, etc.

- The CONEXIOBlackBear is IP40 dustproof and waterproof. Use the CONEXIOBlackBear indoors, inside a car, or in an environment without exposure to excessive dust and rainwater, etc. When using the CONEXIOBlackBear outdoors, ensure that it is well protected from dust and water. Use in areas with a lot of water, dust, moisture, oil, etc., may result in fire, malfunction, electric shock, etc.

- The CONEXIOBlackBear is not intended for use in situations that require high reliability and safety (such as for medical devices, aviation devices, etc.). CONEXIO Corporation shall not be held liable for any damage caused by the use in these applications.

- The CONEXIOBlackBear uses materials manufactured for general electronic equipment. Malfunctions or failures may occur due to external noise, surges, etc. In case of malfunction or failure, safety measures should be implemented for use of this device to prevent potential loss of life or damage to the human body or to property, and the device should be used only after sufficient measures have been taken.

⚠ Warning

- Do not drop or otherwise subject the product to strong impact. If the product continues to be used after being damaged by impact, it may cause fire, malfunction or electric shock.

- The CONEXIOBlackBear is equipped with a wireless function, which may affect peripheral devices such as medical devices. Install the CONEXIOBlackBear at a sufficient distance from such devices.

- Modifying the CONEXIOBlackBear may cause fire, malfunction, electric shock, etc.

- The CONEXIOBlackBear incorporates a coin-type lithium manganese dioxide battery. If the positive and negative electrodes of the battery are short-circuited, liquid leakage, heat generation, rupture or ignition may occur.

⚠ Caution

- The CONEXIOBlackBear is intended for use in its installed state. Do not use it in an uninstalled state or while holding it with your hand.
- When the CONEXIOBlackBear is used continuously, its housing may become hot due to heat dissipation.
- The CONEXIOBlackBear uses Linux as its OS. The source code is provided as open source software unless explicitly stated otherwise. Redistribution rights should be exercised at your own risk in accordance with the form of licensing described in each source code.
- If you wish to add software other than preinstalled software, do so at your own discretion after thoroughly verifying its usage and reliability. CONEXIO Corporation shall not be held liable for any damage caused by problems resulting from additional software.
- Do not remove or insert the antennas, cables, or microSIM while the power is turned on. Failure to observe this instruction may result in malfunction or damage.
- Do not apply excessive force to the external interfaces (antenna terminals, connectors, slots, switches, etc.). Failure to observe this instruction may result in malfunction or damage.
- Do not clean the CONEXIOBlackBear using chemicals or cleaning agents containing solvents.

Other Notes

- Do not remove or damage any nameplate, label, or sticker on the CONEXIOBlackBear. Failure to observe this instruction may result in your product being excluded from warranty coverage.

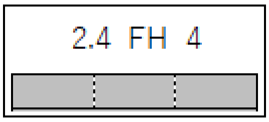
8.2. Radio-related Precautions

The frequency band in which the CONEXIOBlackBear operates is shared by microwaves and other devices used for industrial, scientific, and medical purposes, as well as on-premises radio stations and specified low-power radio stations for identification of mobile units used on factory production lines, etc., and amateur radio stations.

- Before using the CONEXIOBlackBear, ensure that there is no nearby operation of on-premises radio stations and specified low-power radio stations for identification of mobile units and amateur radio stations.
- In the unlikely event that adverse interference occurs from the CONEXIOBlackBear to any on-premises radio station for the identification of mobile units, discontinue use immediately.
- Furthermore, in the event that obstructive interference occurs from the CONEXIOBlackBear to specified low power radio stations for identification of mobile units and amateur radio stations, discontinue use immediately.

8.2.1. Bluetooth frequency band

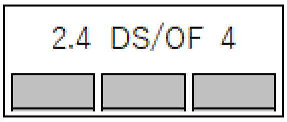
The Bluetooth function uses the following frequency band.



2.4 : Indicates that the radio equipment uses the 2,400 MHz band.
FH : Indicates that the modulation method is FH-SS.
4 : Indicates that the expected interference distance is 40 m or less.
▨▨▨ : Indicates that the equipment uses a band between 2,400 and 2,483.5 MHz and cannot avoid the operating band of mobile unit identification systems.

8.2.2. WiFi (wireless LAN) frequency band (2.4 GHz band)

WiFi (wireless LAN) function uses the following frequency band.



2.4 : Indicates that the radio equipment uses the 2,400 MHz band.
DS/OF : Indicates that the modulation method is DS-SS or OFDM.
4 : Indicates that the expected interference distance is 40 m or less.
▨▨▨ : Indicates that the equipment uses a band between 2,400 and 2,483.5 MHz and can avoid the operating band of mobile unit identification systems.

8.2.3. WiFi (wireless LAN) frequency band (5 GHz band)

In the case of domestic models mounted with the “SIM7600JC-H”, the CHs that can be used in the 5 GHz band are as follows.

5.2 GHz band (W52: 36, 40, 44, 48 ch)
5.3 GHz band (W53: 52, 56, 60, 64 ch)
5.6 GHz band (W56: 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140 ch)

* Outdoor uses in the 5 GHz band are restricted under the Radio Act For details, visit the website of the Ministry of Internal Affairs and Communications.
* Note that when using a WiFi 5 GHz access point configuration, the available frequency band for the CONEXIOBlackBear is set to the 5.2 GHz band (W52:36, 40, 44, 48 ch) only.
* If you change the frequency band to be used, be sure to comply with the relevant laws and regulations of Japan and change it at your own risk. Incorrect configuration may be subject to legal consequences

* The above specifications are for the domestic CONEXIOBlackBear models in Japan. Please contact the vendor about the frequency bands that can be used in the countries other than Japan.

8.3. Compliance with the Radio Act and the Telecommunications Business Act

The domestic CONEXIOBlackBear model incorporates the “SIM7600JC-H” as the LTE module. It is equipped with the “WM-BAC-BM25” as its WiFi/Bluetooth Module. These communication modules are approved with the “certification of conformity to technical regulations under the Radio Act” and the “accreditation of conformity to technical regulations under the Telecommunications Business Act” required for use in Japan.
The certification numbers of these communication modules are shown in “Table 8.3-1 Information on Certification/Accreditation of Conformity to Technical Regulations”.

Table 8.3-1 Information on Certification/Accreditation of Conformity to Technical Regulations

Technical Conformity	Module	Certification No.
Certification of conformity to technical regulations under the Radio Act (construction design certification)	BM25	020-180109
	SIM7600	003-180005
Accreditation of conformity to technical regulations under the Telecommunications Business Act (design certification)	BM25	D180048020
	SIM7600	DF180082003

Moreover, as a device, the CONEXIOBlackBear is has been approved for “accreditation of conformity to standards related to IoT device security regulations under the Telecommunications Business Act”.
This certification number is shown in “Table 8.3-2 Information on Accreditation of Conformity to Security Regulations”.

Table 8.3-2 Information on Accreditation of Conformity to Security Regulations

Technical Conformity	Device	Certification No.
Accreditation of conformity to security regulations under the Telecommunications Business Act	CONEXIOBlackBear	D210136001

The certification numbers and information shown in “Table 8.3-1 Information on Certification/Accreditation of Conformity to Technical Regulations” and “Table 8.3-2 Information on Accreditation of Conformity to Security Regulations” are labeled on the bottom of the CONEXIOBlackBear as shown in “Fig. 8.3-1 Information on Certification/Accreditation of Conformity to Technical Regulations - Bottom Label”. Check the label as necessary.

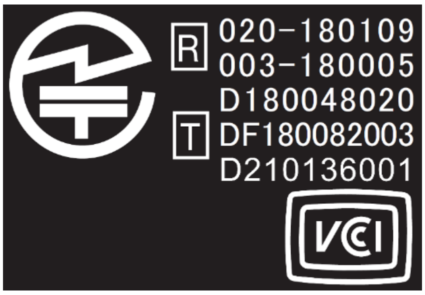


Fig. 8.3-1 Information on Certification/Accreditation of Conformity to Technical Regulations - Bottom Label

* The information above covers the domestic CONEXIOBlackBear models in Japan. For certification support in the countries other than Japan, please contact the vendor.

8.4. Notes on Open-Source Software

The factory-installed software that comes standard on the CONEXIOBlackBear includes open-source software (OSS), including those listed in “Table 2.6-1 OS and Major Preinstalled Software”.

These OSS items are licensed by their respective copyright holders, and the customers who purchase the CONEXIOBlackBear are also obligated to comply with the license terms set forth for them.

For details on these OSS licenses, see the notes in “Yocto Project—Board Support Package (BSP)” at the CONEXIO IoT support site.

If you obtain source code and modify or redistribute it in accordance with the license conditions set forth in each OSS, do so at your own risk.

These OSS items are provided on an “as-is” basis, and no warranty is provided for each OSS itself. In addition, no technical support is provided for the content of OSS source code itself.

If you modify the factory-installed software that comes standard on the CONEXIOBlackBear, it will not be covered by product warranty. Any modifications should be made at your own risk.

To obtain the original i.MX BSP, visit the NXP site at the link below.

NXP site URL:
https://www.nxp.com/design/i-mx-developer-resources/i-mx-software-and-development-tools:IMX-SW

8.5. Overseas Use

The CONEXIOBlackBear is available in models accredited as conforming to the radio-related regulations and product safety in the relevant countries and regions in addition to Japan’s “Technical Conformity Mark” and “PSE.” These models, which are equipped with an LTE module that supports the frequencies in each applicable country or region, have been certified in each of those countries and regions.

While this Instruction Manual provides information based on the domestic model equipped with the “SIM7600JC-H”, the configuration of the CONEXIOBlackBear is the same among the models of all countries and regions, except for the LTE module frequencies and WiFi channels. Therefore, the CONEXIOBlackBear can be deployed as is overseas without any redevelopment of the system, which was developed in Japan. However, be sure to select the correct model that is certified in the country or region in which the device will be used.

Access the following URL to check the latest information on the countries and regions in which the CONEXIOBlackBear has been certified.

URL: https://conexio-iot.jp/serviceproduct/blackbear_faq.html

Product warranty is not available for models other than those certified for the country or region in which they are used. Moreover, the user may be subject to legal consequences in the country or region where it was used.

When exporting the CONEXIOBlackBear, comply with the import/export laws and regulations of Japan and the relevant country or region, and take the necessary import/export procedures.

8.6. Warranty Matters

The warranty is made in accordance with the “CONEXIOBlackBear Warranty Policy” included with the CONEXIOBlackBear or provided at the following URL.

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

Note the clauses concerning the scope and period of warranty.

==

Caution

The seal shown in “Fig. 8.6-1 TAMPER SEAL” is affixed to one of the four screw holes on the bottom of the CONEXIOBlackBear. This seal is intended to prevent unauthorized disassembly or modification of the CONEXIOBlackBear. Note that the CONEXIOBlackBear will **not be covered by the warranty even within the warranty period if any damage to this seal, such as peeling or tearing, is found.**
If you need to disassemble the CONEXIOBlackBear, contact us using the information in “9. Contact”.

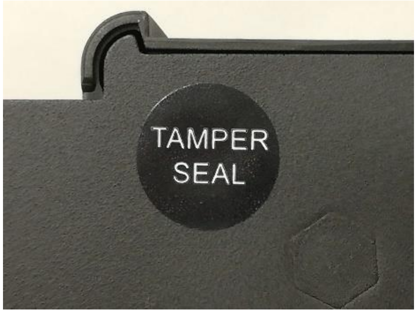


Fig. 8.6-1 TAMPER SEAL

==

9. Contact

CONEXIO Corporation
Corporate Smart Service Division
IoT Sales Section, IoT Solutions Department

7th floor, Kamiyacho Trust Tower, 4-1-1, Toranomon, Minato-ku,
Tokyo 105-6907, Japan

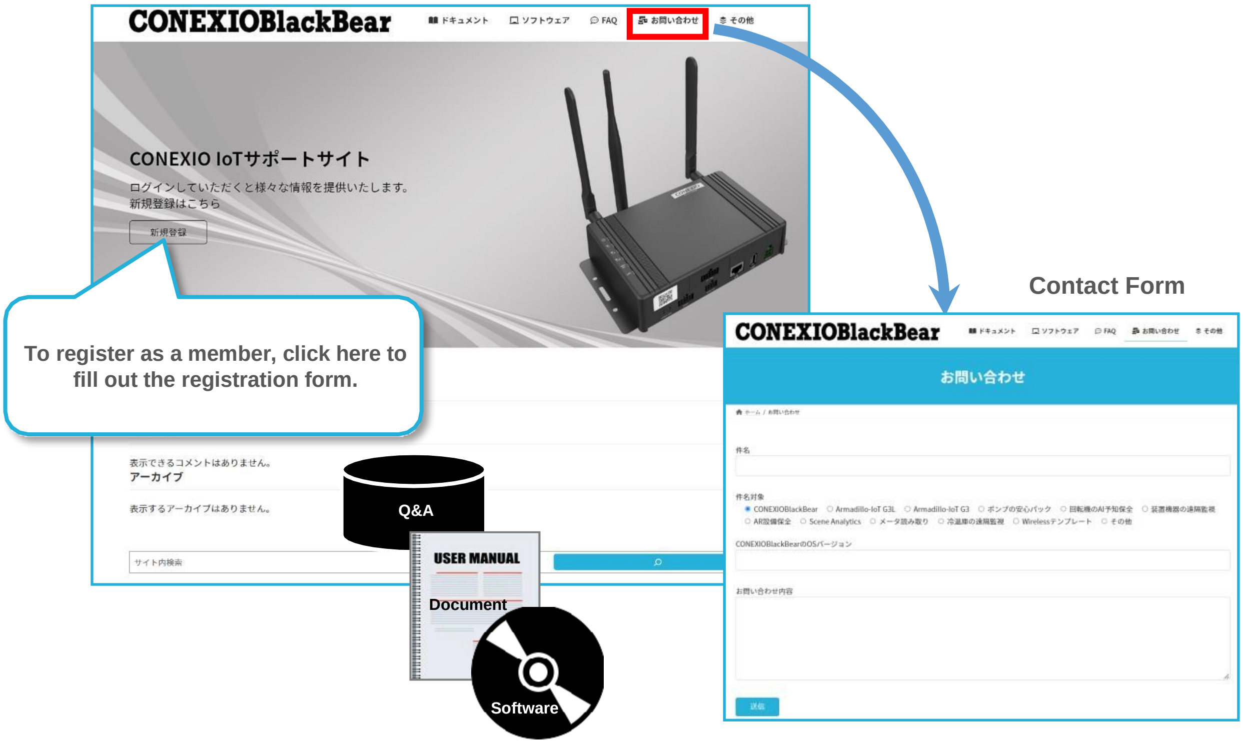
TEL.03-5408-3177

Technical inquiries can be made using the Contact Form found on the CONEXIO IoT support website at the following URL. Please make an inquiry using this form.

CONEXIO IoT support website: https://www.bb-customer-support.conexiot.jp/

* Membership registration is required to use the support website. Please register as a member to make an inquiry.

Support website homepage



CONEXIO IoTサポートサイト

ログインしていただくと様々な情報を提供いたします。
新規登録はこちら

表示できるコメントはありません。

アーカイブ

表示するアーカイブはありません。

お問い合わせ

件名

件名対象
● CONEXIOBlackBear ○ Armadillo-IoT G3L ○ Armadillo-IoT G3 ○ ポンプの安心パック ○ 回転機のAI予知保全 ○ 装置機器の遠隔監視 ○ AR設備保全 ○ Scene Analytics ○ メータ読み取り ○ 冷温庫の遠隔監視 ○ Wirelessテンプレート ○ その他

CONEXIOBlackBearのOSバージョン

お問い合わせ内容