Armadillo-IoT Gateway G3
Product Manual - INTL

AGX3150

Version 1.0.0
2018/02/19

Atmark Techno, Inc [http://www.atmark-techno.com]

Armadillo Official Site [http://armadillo.atmark-techno.com]

Armadillo-IoT Gateway G3Product Manual - INTL

Atmark Techno, Inc

Version 1.0.0
2018/02/19

Table of Contents

List of Figures

List of Tables

Chapter 1. Precautions

1.1. Safety Precautions

In order to use this product safely, please take special note of the following precautions.

- Be sure to read all product manuals and related documentation before using this product. Please use this product correctly and safely making sure to follow all usage precautions.
- When operating or extending this product in a way not described in the product manuals, please do so safely and on your own responsibility after having fully understood the materials on our web site and any other technical information.
- Please do not install this product in a place with a lot of water, moisture, dust or soot. This could cause a fire, product failure or electric shock.
- Some parts of this product generate heat and can reach high temperatures. Depending on the surrounding temperature and on how this product is handled, this may cause burns. Please do not touch the electronic components or the surrounding area while the product is powered on or before it has cooled down after being powered off.
- When using this product in the development of devices or systems to original specifications, please carry out the design and development after having thoroughly read and fully understood the product manuals and related materials, the technical information offered on our web site and related device data sheets. Also, please carry out full tests beforehand in order to provide and maintain reliability and safety.
- This product is not intended for uses that require extremely high reliability and safety in terms of functionality and accuracy (such as medical equipment, traffic control systems, combustion control systems, safety equipment and so on). If this product is used in these kinds of equipment, devices or systems, this company will not be held responsible in any way for any accident resulting in injury or death, fire or damage and so on.
- This product uses semiconductor components designed for generic electronics equipment such as office automation equipment, communications equipment, measurement equipment and machine tools. It is possible that a foreign noise or surge may cause this product to malfunction or fail. To ensure there will be no risk to life, the body or property in the event of malfunction or failure, be sure to take all possible measures in regard to device safety design, such as using protection circuits like limit switches or fuse breakers, or system redundancy, and to only use the device after taking measures to ensure sufficient reliability and safety.

- Please do not use products with Wireless LAN functionality in places near medical devices such as heart pacemakers and hearing aids, automatic control equipment such as fire alarms and automatic doors, microwave ovens, advanced electronic equipment or televisions and radios, or near "Premises Radio Stations" for "Mobile Body Identification" or "Specified Low Power Radio Stations". The radio waves emitted by this product may cause these types of devices to malfunction.

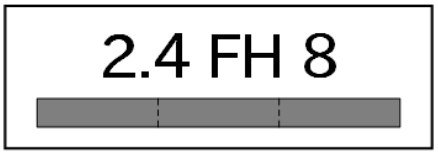
1.2. Handling Precautions

Please pay attention to the following points when handling this product in order to avoid causing any irreversible damage.

Areas Easily Damaged	The inter-board connectors can be easily damaged. Please be careful not to damage them by handling them with too much force.
Modifications to this Product	Please take note that any modifications[1] made to this product are not covered under warranty. Also, please ensure to undertake a full operational check of this product before carrying out any modifications or mounting connectors[2].
Attaching and Detaching Connectors While Powered On	Apart from the hot-pluggable interfaces (LAN, SD/SDIO and USB), do not under any circumstances attach or detach connectors or cards while power is supplied to this product or peripheral circuits.
Static Electricity	As CMOS devices are used in this product it can be damaged by static electricity. When opening this product be careful not to do so in low humidity conditions, while also making sure it is done in an environment with anti-static measures established, like using anti-static mats, anti-static measures for workers such as conductive shoes and human body earthing and countermeasures against discharges from equipment. Also, when storing this product avoid using plastic bags and containers that are prone to static electricity and store them in conductive bags and containers or racks and so on.
Latch-up	Excessive noise or a surge from the power supply or input/output, or sharp voltage fluctuations can lead to the CMOS devices incorporated in the board causing a latch-up. Once the latch-up occurs, this situation continues until the power supply is disconnected, and therefore can damage the devices. Measures such as adding a protection circuit to noise-susceptible input/output lines or not sharing the power supply with devices that can be the cause of noise are highly recommended.
Physical Stress	Please avoid strong physical stress such as drops or other impacts.
Use Location Restrictions	Using this product close to a TV and radio may cause radio disturbance.
Radio Wave Related Precautions (2.4GHz band radio)	In the case that any function using radio waves in the 2.4GHz band (such as wireless LAN) affects automatic control electronic equipment such as automatic doors, please stop using it immediately.

[1] With the exception of adding connectors to unmounted interfaces.
[2] When mounting connectors, please ensure to apply masking, avoid solder residue coming in contact with surrounding parts and avoid creating solder balls.

This wireless device (AEH-AR9462) uses the 2.4GHz band. All of the band is used, and the band used by mobile identification devices can be avoided. The modulation systems used are DS-SS and OFDM, and the estimated interference distance is 40 meters or less.



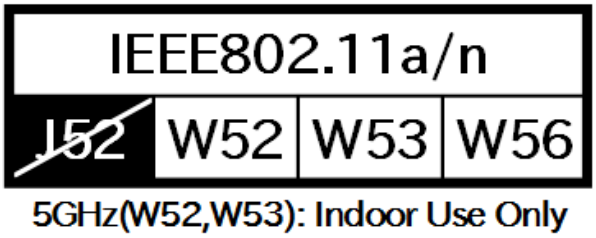
This wireless device (AEH-AR9462) uses the 2.4GHz band. All of the band is used, and the band used by mobile identification devices can be avoided. The modulation system used is FH-SS, and the estimated interference distance is 80 meters or less.

Radio Wave Related Precautions (5GHz band radio)

This wireless device (AEH-AR9462) uses the 5GHz band.

W52 and W53 outdoor use is prohibited under radio law.

Please do not use AP mode in W53, W56 as it has not received construction design certification as of June 2016.



Radio Wave Related Precautions (3G)

This wireless device (PDS6) performs 3G wireless communication.

Please do not use the 3G wireless communication function near implantable medical devices such as cardiac pacemakers or defibrillators (within about 15 cm).

1.3. Software Usage Precautions

About Software Contained in this Product

The source code of the Linux-compatible software preinstalled in the default shipping state of this product is provided as open source, unless otherwise individually specified (including written, electronic data, or oral notification). For redistribution and so on, please do so at your own risk according to the license notice included in each source code file. In addition, the software and documentation contained in this product are provided “AS IS”. The customer is required to assume the responsibility of only using this product after having fully considered and tested its suitability to the intended purpose and use. There is no guarantee of fitness for a particular purpose, reliability, correctness and no guarantee of any outcomes resulting from the use of this product.

Middleware and various other software solutions offered for Armadillo branded products in cooperation with partners and so on is all individually licensed. Please refer to the readme file etc included in each piece of software for the redistribution rights and so on. For other bundled software, please contact the provider.

The following software preinstalled in the default shipping state of this product is not open source software.

- Board info capture tool (get_board_info)

1.4. Write Prohibited Regions

The data stored by the i.MX 7 Dual electric fuse (e-Fuse) is used by the software contained in this product. Please do not write to these regions as the product may stop operating correctly. Purposefully writing to these regions voids the product warranty.

1.5. Electromagnetic Interference

This device is Class B Information Technology Equipment. Although this device is intended for use in a home environment, if this device is used close to a radio or television receiver it may cause radio interference. Please use it correctly in accordance with the instruction manual.

In order to comply with the technical standards of VCCI, it is necessary to supply power from the DC jack (CON14) with an AC adapter.

The add-on modules have not been subjected to the VCCI compliance check test as single modules. It has been confirmed that they meet VCCI technical standards when connected to Armadillo-IoT. Please use the add-on modules connected to the add-on interface of Armadillo-IoT.

1.6. Warranty

As laid out in the Product Warranty Policy which is provided with this product or available on our web site, the main board of this product is covered by a one year replacement warranty from time of purchase. Please note that the other included goods and software are not covered by the warranty.

Product warranty policy http://www.atmark-techno.com/support/warranty-policy

1.7. Exporting

- When exporting this product, it is the responsibility of the exporter to follow all export-related laws and regulations of Japan and related foreign countries, and carry out all required procedures.
- Our product may not be exported or sold etc. to nations, organizations, corporations or individuals under sanctions or trade suspension by Japan and other related countries.
- Our product and related technology may not be used for the purpose of development of weapons of mass destruction, for the purpose of military use or other military related uses, or in devices which have had their production, use, sale or procurement prohibited by national or overseas law or regulations.

1.8. Trademarks

- Armadillo is a registered trademark of Atmark Techno, Inc. All other company names, product names and related trademarks are the property of their respective owners. ™ and ® marks are omitted.
- The SD, SDHC, SDXC, microSD, microSDHC, microSDXC and SDIO logos are trademarks of SD-3C, LLC.

Chapter 2. Product Overview

2.1. Product Features

2.1.1. What is Armadillo

"Armadillo" is an embedded system platform brand with ARM core processors and Linux support. Armadillo brand products have the following features.

- ARM processors with power-saving design

  ARM based processor boards with power saving designs that operate between one to several watts, they generate little heat and don't require a fan.

- Small Size

  The CPU boards are mostly the size of a business card. A lineup of ultra-small modules including a CPU module about one third the size of a business card and also a wireless LAN module are also available.

- Linux Preinstalled as the Standard Operating System

  Linux as the default operating system provides access to rich software resources and proven stability. Source code is provided as open source.

- Development Environment

  "Atmark Techno Development Environment" (ATDE) is provided free of charge as the development environment for Armadillo. ATDE is a virtual machine image for VMware. The image is based on a Linux desktop environment and has GNU cross development tools and other necessary tools preinstalled. Using ATDE reduces the work involved with setting up a development environment like preparing a development PC and installing required tools.

2.1.2. What is Armadillo-IoT Gateway

The Armadillo-IoT Gateway is a platform for easily and quickly developing gateways for IoT / M2M and is based on the proven Armadillo embedded platform. It combines a high degree of flexibility, ease of development and robustness as an embedded device in a well-balanced manner, and is suitable for those who want to develop original commercial IoT gateways in a timely manner according to market needs.

Expansion with Add-on Modules

Two expansion interfaces are available allowing optional add-on modules to be connected. Add-on modules for wired connections such as RS232C / RS422 / RS485, digital input/output and analog input, and also those supporting energy-saving wireless communication standards such as EnOcean and Wi-SUN are available.

Also, as the interface standards for add-on modules are published, original modules can be developed. As various requirements can be met by developing just an add-on module, it is possible to reduce the overall development time and cost compared to when developing a full CPU board from scratch.

Mobile Communication (3G) Support

3G communication modules can be equipped for mobile communication. Various mobile carriers offer dedicated line plans for Armadillo-IoT, so 3G support capability can be introduced immediately.

Linux Based Software Stack

As Linux is preinstalled as the standard operating system, various software resources, starting with open source software, can be utilized. Also, as languages like Ruby and Oracle Java are supported, it is possible to develop software without using the C/C++.

Cloud Support

A software stack which includes components like a MQTT client and is highly suited to use with cloud systems is preinstalled. In addition, as the edge agents from various cloud services have already been ported to Armadillo-IoT it is simple to develop a system that works with the cloud.

2.2. Product Lineup

The lineup of add-on modules is as follows.

Table 2.1. Add-on Module Lineup

Name	Product Number
Armadillo-IoT RS232C Add-on Module RS00	OP-AGA-RS00-00
Armadillo-IoT Isolated RS232C/422/485 Add-on Module RS01	OP-AGA-RS01-00
Armadillo-IoT Isolated RS485 Add-on Module RS02	OP-AGA-RS02-00
Armadillo-IoT RN4020 Add-on Module BT00	OP-AGA-BT00-00
Armadillo-IoT EnOcean Add-on Module EN00[a]	OP-AGA-EN00-00
Armadillo-IoT Wi-SUN Add-on Module WS00	OP-AGA-WS00-00
Armadillo-IoT Isolated Digital I/O / Analog Input Add-on Module DA00	OP-AGA-DA00-00

[a]Scheduled to be released

2.3. Specifications

The main specifications of Armadillo-IoT Gateway G3 are as follows.

Table 2.2. Specifications

Product Number	AGX3150
Processor	NXP Semiconductors i.MX 7 Dual
	ARM Cortex-A7 x 2 - Instruction / data cache 32KByte/32KByte - L2 Cache 512KByte - Internal SRAM 256KByte - Media processing engine (NEON) - Thumb code (16 bit instruction set) support ARM Cortex-M4 - Instruction / data cache 16KByte/16KByte
System Clock	CPU Core Clock (ARM Cortex-A7): 996MHz CPU Core Clock(ARM Cortex-M4): 200MHz DDR Clock: 533MHz Oscillation Clock: 32.768kHz, 24MHz
RAM	DDR3L: 1GByte 32bit bus
ROM	QSPI NOR Flash Memory: 8MByte eMMC: about 3.8GB (about 3.6GiB))[a]
LAN (Ethernet)	RJ-45 x 1 1000BASE-T/100BASE-TX/10BASE-T, AUTO-MDIX
Wireless LAN	WLAN + BT Combo Module: AEH-AR9462 IEEE 802.11a/b/g/n

Product Number	AGX3150
ellular Module	3G module: Gemalto PDS6[b] MicroSIM slot x 1
Serial (UART)	3.3V CMOS x 1
SD/MMC	SD slot x 1
USB	USB 2.0 Host x 1 (High Speed)
Calendar Clock	Real-time Clock Power Supply Connector For External Backup[c]
Add-on Module[d]	None
Switch	User switch x 3, reset switch x 1
LED	User LED x 4, 3G LED x 1
Power Supply Voltage	DC 8V to 26.4V
Power Consumption (reference)[e]	3.6W approx. (standby), 4.5W approx. (during communication)[f]
Operating Temperature[g][h]	-10 to 50℃[i]
Dimensions	155.8 x 125.8 x 47.0mm (including flange)

[a]Size for SLC use. SLC is set at the factory.

[b]3G mobile communication MicroSIM card sold separately.

[c]Batteries are not included.

[d]Two add-on modules can be equipped.

[e]Power consumption is measured when LAN, USB and serial connectors are connected to external equipment. Consumption of externally connected equipment is not included.

[f]Power consumption varies depending on the radio wave environment during communication.

[g]This temperature is when power consumption is limited, such as by stopping communication of the 3G module at high temperature.

[h]With no condensation.

[i]The operating temperature the board by itself (without a case) is -20°C to 70°C.

2.4. Armadillo-IoT Gateway Exterior

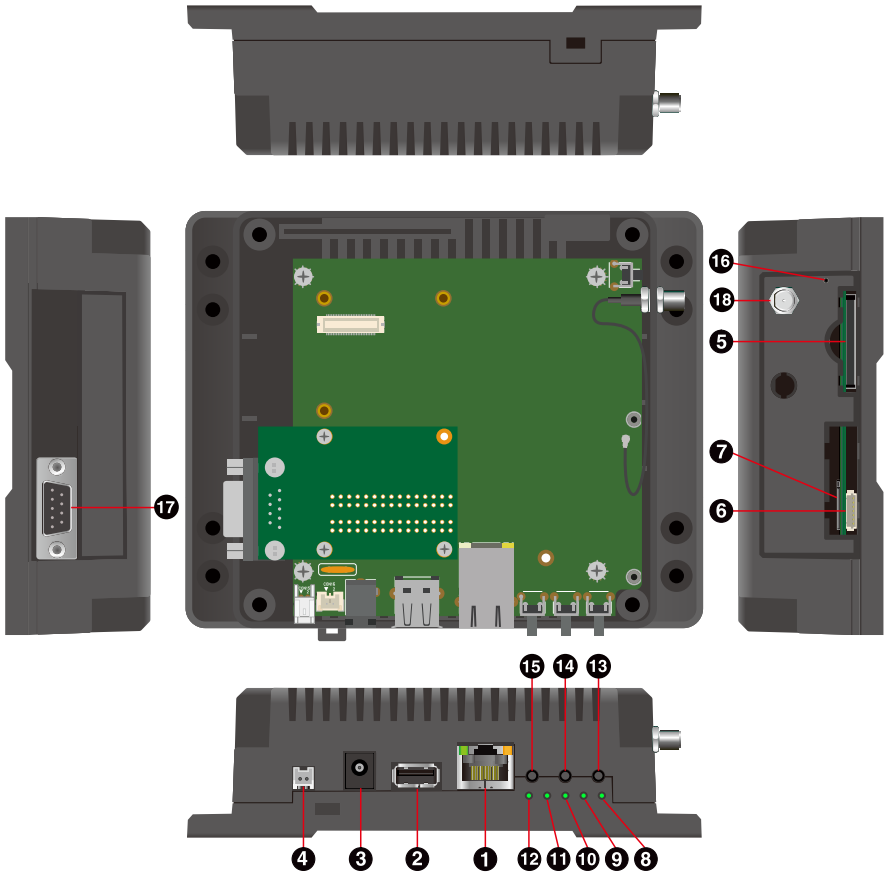


Figure 2.1. Armadillo-IoT Gateway Exterior

Table 2.3. Part Names and Functions

Number	Name	Description
1	LAN Connector	Used to connect a LAN cable.
2	USB Connector	Used to connect devices USB memory.
3	Power Connector 1	Used to connect the included AD adapter.
4	Power Connector 2	Used to connect power cables other than the included AD adapter.
5	SD slot	Used to connect SD cards.
6	Debug Serial Connector	Used to connect the included USB serial converter.
7	MicroSIM Slot	Used to connect a MicroSIM card.
8	User LED1	A green LED which users can freely assign functions to.
9	User LED2	
10	User LED3	
11	User LED4	
12	WWAN LED	Green LED that indicates the communication status of the 3G module.
13	User Switch 1	A tact switch which users can freely assign functions to.
14	User Switch 2	
15	User Switch 3	
16	Reset Switch	Tact switch for resets.
17	Serial Connector	Used to connect the serial cross cable.
18	Antenna Connector	Used to connect the supplied antenna.

2.5. Block Diagram

The block diagram of the Armadillo-IoT gateway is as follows.

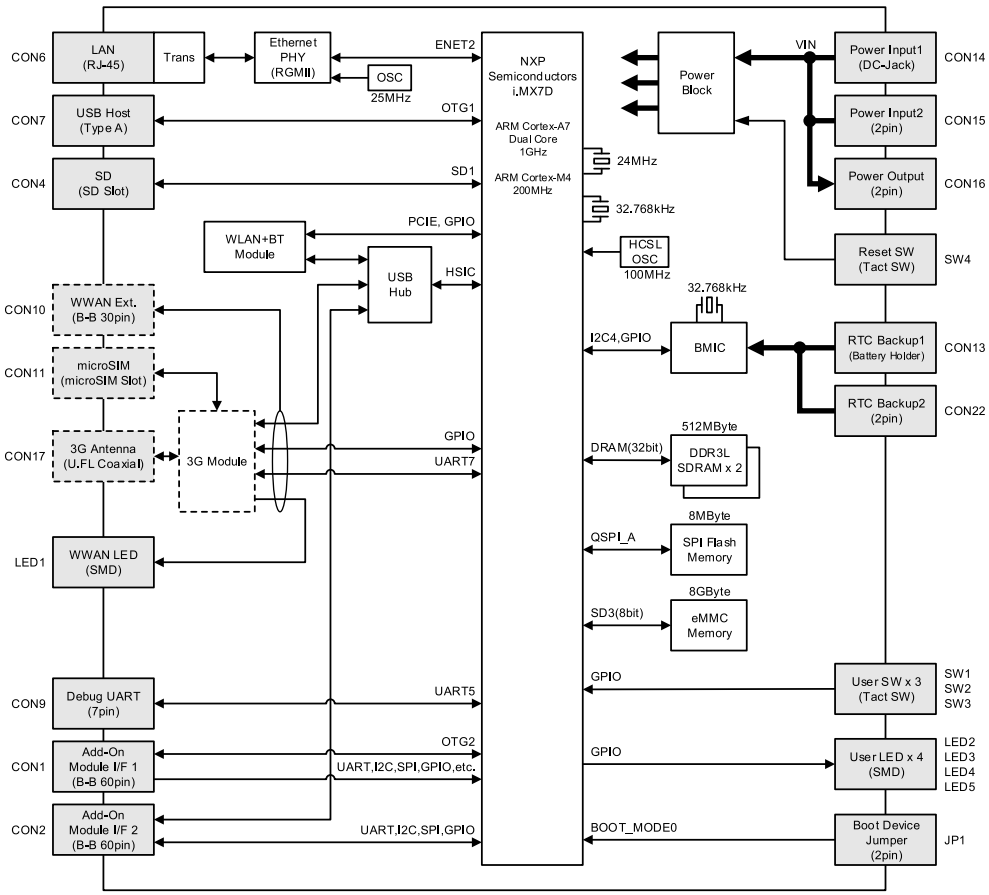


Figure 2.2. Armadillo-IoT Gateway Block Diagram[1]

2.6. Software Make-up

The following explains about the makeup of the software that runs on Armadillo-IoT.

The software available for Armadillo-IoT is shown in Table 2.4, “Software Available for Armadillo-IoT”.

Table 2.4. Software Available for Armadillo-IoT

Software	Description
U-Boot	The bootloader. In the factory default state the bootloader image is located in the QSPI Flash memory, but it can also be placed on the SD card.
Linux Kernel	A Linux kernel in the uImage format can be used. In the factory default state the Linux kernel image is located in the eMMC, but it can also be placed on the SD card by using bootloader functionality.

[1]For the dotted line blocks, whether they are mounted or not depends on the product model. For the model with the 3G module, the 3G module, CON11 and CON17 are equipped and CON10 is not equipped. For the model without the 3G module, CON10 is equipped and the 3G module, CON11 and CON17 are not equipped.

Software	Description
Debian GNU/ Linux	A Linux distribution created by the Debian Project. With its package management system, it is easy to add a wealth of software packages provided by the Debian Project. In the factory default state, the Debian GNU/Linux root file system is located in eMMC, but it can also be placed in a storage device supported by Linux kernel such as an SD card.

The memory map of the Armadillo-IoT QSPI Flash memory is shown in Table 2.5, “QSPI Flash Memory Memory Map”.

Table 2.5. QSPI Flash Memory Memory Map

Physical Address	Size	Description
0x00000000 | 0x001003FF	Approx. 1MByte	U-Boot bootloader image
0x00100400 | 0x001403FF	256 KBytes	License information
0x00140400 | 0x007FFFFF	Approx. 6.7 MBytes	Reserved area

The memory map of Armadillo-IoT's eMMC is shown below.

Table 2.6. eMMC Memory Map - When Using Installation Disk v20160928 or Later

Partition	Size	Description
1	32 MBytes	Linux kernel image / Device tree blob
2	Approx 3.4 GBytes	Debian GNU/Linux
3	128 MBytes	Recovery image

Table 2.7. eMMC Memory Map - When Using Installation Disk v20160705 or Earlier

Partition	Size	Description
1	32 MBytes	Linux kernel image / Device tree blob
2	Approx. 3.5 GBytes	Debian GNU/Linux
3	32 MBytes	Recovery image

Chapter 3. Before Turning on Armadillo

3.1. Preparation

Before using Armadillo, please prepare the following as required.

Work PC	A PC that runs either Linux or Windows which has a network interface and at least one USB port. Please refer to Section 3.2, “Setup of Environment for Development and Operational Checks” and set up the development and operational check environment on the work PC.
Network Environment	Please connect the Armadillo and Work PC so that they can communicate via a network.
SD Card	Used when checking the operation of the SD slot.
USB Memory	Used when checking USB operation.
MicroSIM (UIM Card) and APN Information	These are used to confirm 3G operation. A contract with a telecommunications carrier is required. In order to check the operation of SMS, a MicroSIM (UIM card) with SMS function is required.
Software to Extract tar.xz Format Files	Used when creating the development and operational check environment. On Linux, tar[1] can be used for the extraction. For Windows, 7-Zip and Lhaz etc can be used. 7-Zip is included on the development DVD.

3.2. Setup of Environment for Development and Operational Checks

A VMware virtual machine data image is provided to allow easier software development and operational checks of Atmark Techno products. The VMware virtual machine data image is referred to as ATDE (Atmark Techno Development Environment). To start ATDEm, VMware virtualization software is used. The ATDE image data is compressed in the tar.xz format. Please extract it with the tools suitable for your environment.

In addition to VMware, Oracle VM VirtualBox is also a well know virtualization program. Oracle VM VirtualBox has the following features.

- Provided under GPL v2 (General Public License version 2)[2]
- Supports VMware virtual disk (.vmdk) files

It is possible to start ATDE with Oracle VM and use it as the software development environment,

The Atmark Techno products supported differ depending on the ATDE version. The ATDE compatible with this product is ATDE6 v20160321 or later.

ATDE6 is based on Debian GNU/Linux 8 (codename jessie) and the cross development tools necessary for software development and performing the operational checks of Armadillo-IoT Gateway are preinstalled.

[1] Specify the `Jxf` options to expand tar.xz format files.
[2] PUEL (VirtulBox Personal Use and Evaluation License) may be applied up to version 3.x.

3.2.1. ATDE6 Setup

3.2.1.1. Installing VMware

In order to use ATDE6, VMware must be installed on the work PC. Please refer to the VMware web page (http://www.vmware.com/) and install the VMware product that most suits the purpose of use. Also, as the image data of ATDE6 is compressed in the tar.xz format, please extract it with a tool suitable for your environment.

VMware offers a number of products, from free versions for non-commercial use to paid commercial use versions. Separate licenses and end-user license agreements (EULA) exist for each product, so please choose the most appropriate product after having fully checked these.

In order to avoid VMware and ATDE6 not working correctly, please check the following items from the VMware documentation.

- Hardware requirements of the host system
- Software requirements of the host system
- Processor requirements of the guest operating system

The documentation can be obtained from the VMware web site (http://www.vmware.com/).

3.2.1.2. Obtaining the ATDE6 Archive

The ATDE6 archive can be obtained from the Armadillo site (http://armadillo.atmark-techno.com) or the development set DVD.

The version of ATDE6 supporting this product is v20160321 or later.

ATDE6 may not work properly depending on the operating environment of the work PC (hardware, VMware, ATDE6 supported architecture, etc). Please refer to the documentation for the VMware software you are using on the VMware web site (http://www.vmware.com/) and check the operating environment.

3.2.1.3. Extracting the ATDE6 Archive

Extract the ATDE6 archive. The ATDE6 archive is a compressed file in the tar.xz format.

The extraction method for Windows is shown in Procedure 3.1, “Extracting the ATDE6 Archive on Windows”, and the method for Linux is shown in Procedure 3.2, “Extracting the tar.xz Format File on Linux”.

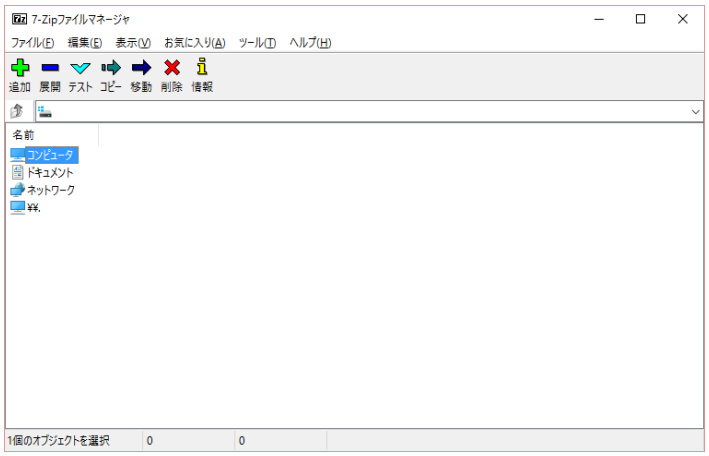
Procedure 3.1. Extracting the ATDE6 Archive on Windows

1. **Installing 7-Zip**

   Install 7-Zip. 7-Zip can be obtained from the 7-Zip website (http://sevenzip.sourceforge.jp) or the DVD included in the development set.

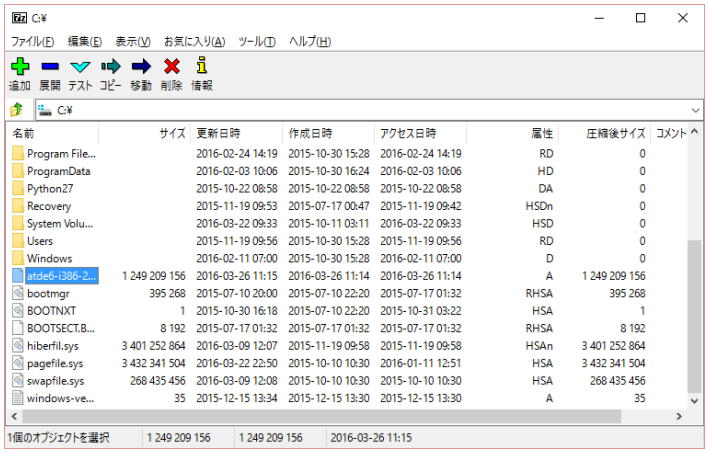
2. **Starting 7-Zip**

   Start 7-Zip.



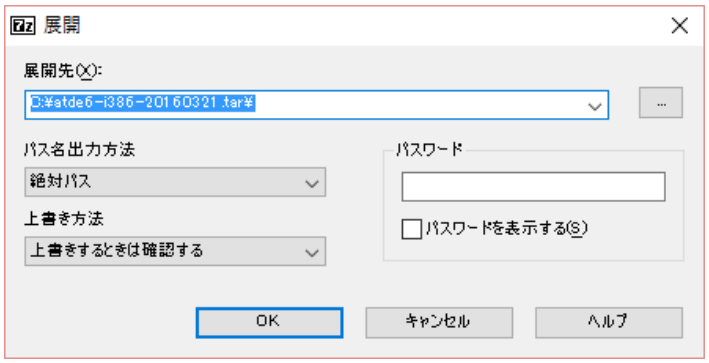
3. **Selecting the xz Compressed File**

   Extract the xz compressed file into a file in the tar format. Select the tar.xz file and click "Extract".



4. **Specifying the xz Compressed File Extraction Destination**

   Specify where to extract the file to and click "OK".



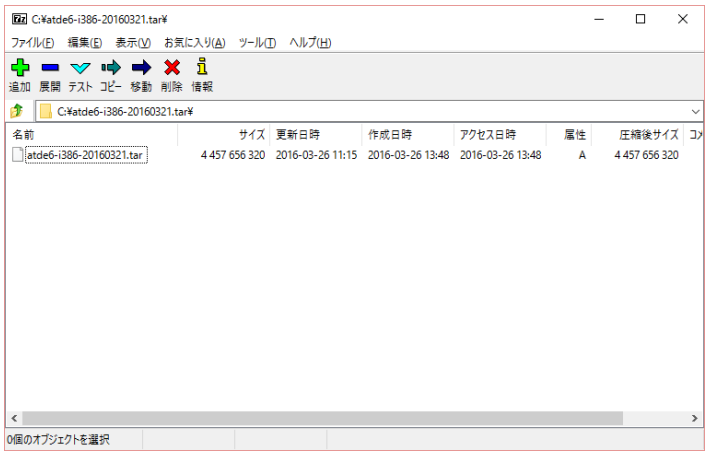
5. **Extracting the xz Compressed File**

The extraction begins.

6. **Selecting the tar Archive File**

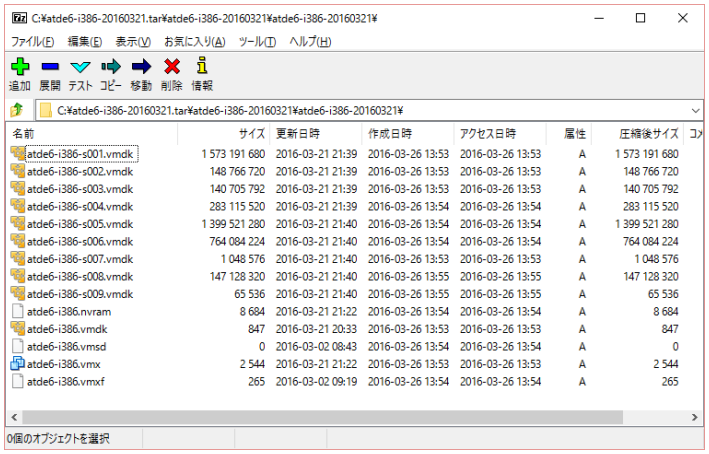
Once the extraction of the xz compressed file has completed, a file in the tar format is created.

Output the ATDE6 data image from the tar archive file with the same procedure as used for the tar archive file. Select the tar format file, click "Extract", specify the "Extracted location", and click "OK".



7. **Confirming Completion of the Extraction**

When extraction of the tar archive file is finished, the extraction of the ATDE6 archive is complete. The ATDE6 data image is output to the folder specified with "Extracted location".



Procedure 3.2. Extracting the tar.xz Format File on Linux

1. **Extracting the tar.xz Compressed File**

   Use the tar command with the `Jxf` options to extract the tar.xz compressed file.

```
[PC ~]$ tar Jxf atde6-i386-[version].tar.xz
```

2. **Confirming Completion of the Extraction**

   When extraction of the tar.xz compressed file is finished, extraction of the ATDE 6 archive is complete. The ATDE6 data image is output to the `atde6-i386-[version]` directory.

```
[PC ~]$ ls atde6-i386-[version]/
atde6-i386.nvram      atde6-i386-s005.vmdk  atde6-i386.vmdk
atde6-i386-s001.vmdk  atde6-i386-s006.vmdk  atde6-i386.vmsd
atde6-i386-s002.vmdk  atde6-i386-s007.vmdk  atde6-i386.vmx
atde6-i386-s003.vmdk  atde6-i386-s008.vmdk  atde6-i386.vmxf
atde6-i386-s004.vmdk  atde6-i386-s009.vmdk
```

3.2.1.4. Starting ATDE6

ATDE6 can be started by opening the virtual machine configuration (.vmx) file in the directory where you extracted the ATDE6 archive. Users which can be used to log on to ATDE6 are shown in Table 3.1, “Usernames and Passwords”[3].

Table 3.1. Usernames and Passwords

Username	Password	Permissions
atmark	**atmark**	General user
root	**root**	Privileged user

ATDE can be used more effectively by increasing the memory size and number of processors assigned to it. For details on how to change the hardware settings of the virtual machine, refer to the documentation for the VMware software you are using on the VMware site (http://www.vmware.com/).

[3]It is not possible to log in with the GUI as the privileged user.

3.2.2. Using Removable Devices

VMware supports the use of removable devices (such as USB devices and DVDs) with the guest operating system (ATDE). Depending on the device, these devices cannot be used simultaneously by the host operating system (the operating system running VMware) and the guest operating system. In order to use such a device in the guest operating system, it must be connected to the guest operating system.

For details on how to use removable devices, please refer to the documentation for the VMware software you are using on the VMware site (http://www.vmware.com/).

In order to perform the operational checks of Armadillo-IoT, the devices shown in Table 3.2, “Removable Devices Used for Operational Checks” must be connected to the guest operating system.

Table 3.2. Removable Devices Used for Operational Checks

Device	Device Name
USB serial converter	Future Devices FT232R USB UART
Work PC physical serial port	Serial port

3.2.3. Starting the Command Line Terminal (GNOME Terminal)

In ATDE6, we start a command line terminal that provides a CUI (Character-based User Interface) environment. Various commands to be run on ATDE6 are input and executed with the command line terminal. While there are several types of command line terminals, here we start the GNOME terminal installed by default in the GNOME desktop environment.

To start the GNOME terminal, please select "Terminal" from the menu at the top left of the desktop as shown in Figure 3.1, “Starting the GNOME Terminal”.

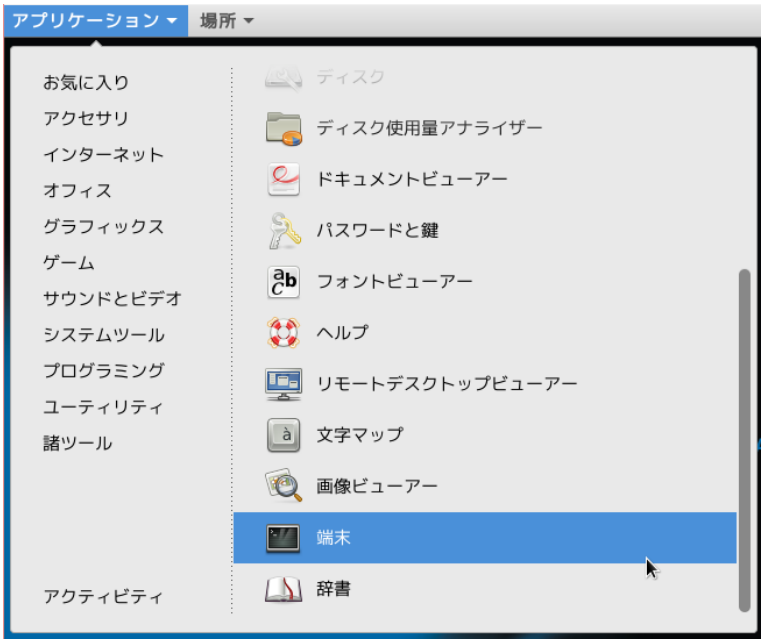


Figure 3.1. Starting the GNOME Terminal

A window opens as shown in Figure 3.2, “GNOME Terminal Window”.

Figure 3.2. GNOME Terminal Window

3.2.4. Using Serial Communication Software (minicom)

Set the serial communication settings of the serial communication software (minicom) as shown in Table 3.3, “Serial Communication Configuration”. Also, please set the width of the terminal used to start minicom to 80 or more characters. The display may become disordered when entering commands if the width is less than 80 characters.

Table 3.3. Serial Communication Configuration

Item	Configuration
Baud rate	115,200bps
Data Length	8-bit
Stop Bit	1-bit
Parity	None
Flow Control	None

To start minicom configuration, please do as shown in Figure 3.3, “Configuring minicom”. After configuration is completed, save it to the default setting (dfl) and exit.

```
[PC ~]$ LANG=C minicom --setup
```

Figure 3.3. Configuring minicom

To start minicom, please do as shown inFigure 3.4, “Starting minicom”.

```
[PC ~]$ LANG=C minicom --wrap --device /dev/ttyUSB0
```

Figure 3.4. Starting minicom

Depending on the environment, the device file name may differ from that in the examples in this document and for example may be `/dev/ttyS0` or `/dev/ttyUSB1`.

To quit minicom, first enter Ctrl+a followed by the q key. After that, when the following is displayed, move the cursor to "Yes" and press Enter to quit minicom.

```
                                +-----------------------+
                                | Leave without reset?  |
                                |     Yes       No      |
                                +-----------------------+
```

Figure 3.5. Minicom Quit Confirmation

Enter Ctrl+a followed by the z key to display minicom's command help.

3.3. Interface Layout

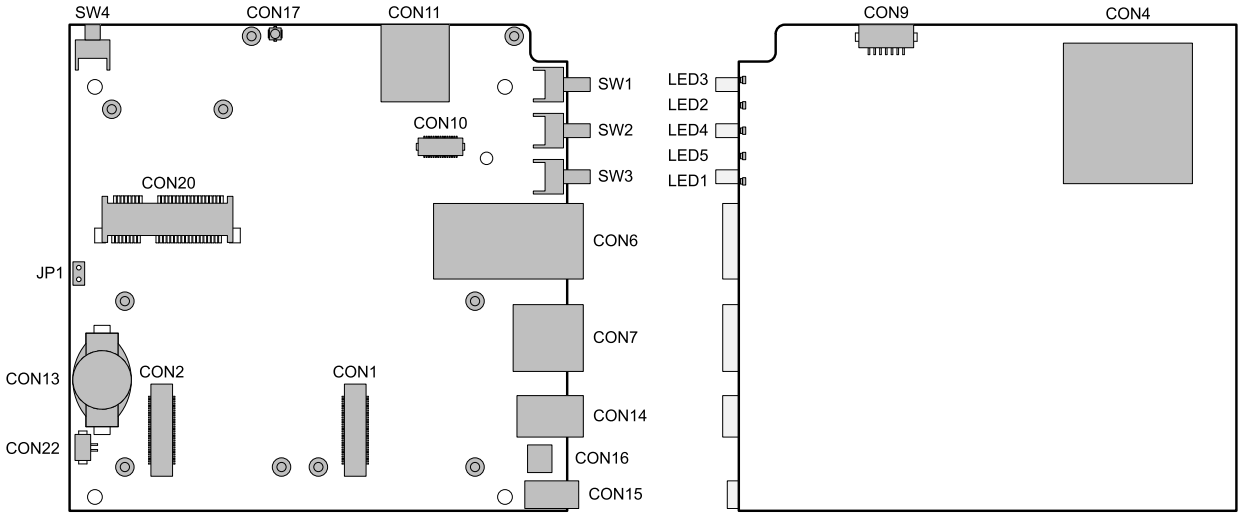


Figure 3.6. Interface Layout Diagram

Table 3.4. Interfaces

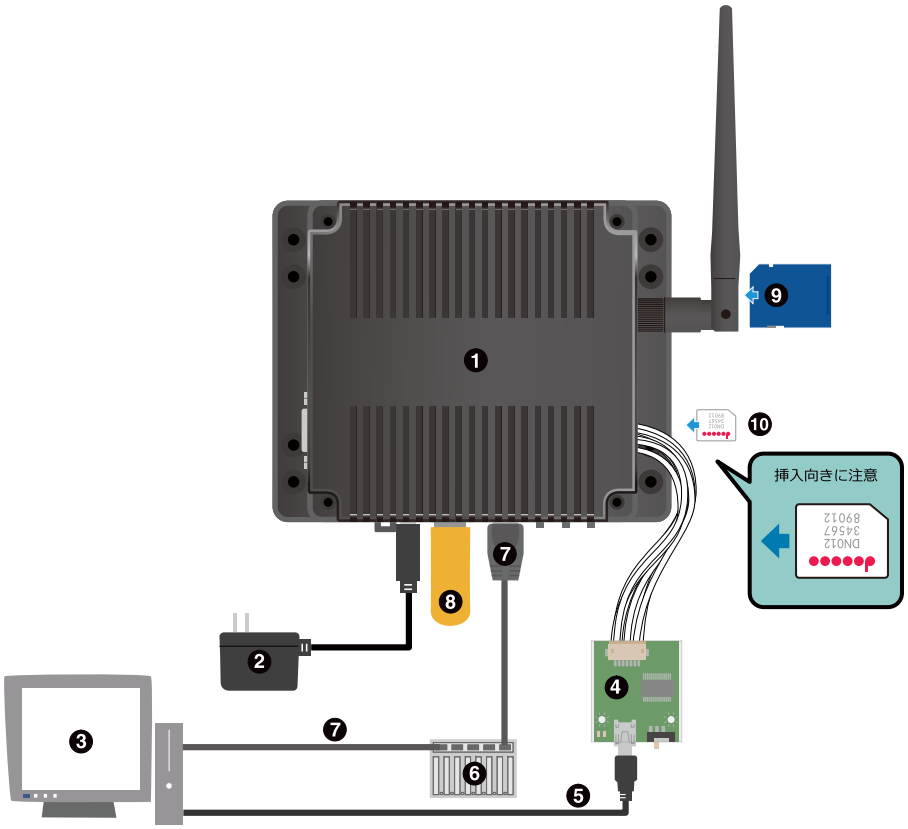
Part Number	Interface Name	Form	Notes
CON1	Add-on Interface 1	Board-to-board connector 60 pin (0.5mm pitch)	Connection cycles: 40 times[a]
CON2	Add-on Interface 2	Board-to-board connector 60 pin (0.5mm pitch)	Connection cycles: 40 times[a]
CON4	SD Interface	SD slot	
CON6	LAN Interface	RJ-45 Connector	
CON7	USB Host Interface	Type A connector	
CON9	Debug Serial Interface	Pin header 7 pin (1.25mm pitch)	Connection cycles: 40 times[a]
CON10	WWAN Expansion Interface	Board-to-board connector 30 pin (0.5mm pitch)	Connection cycles: 50 times[a]
CON11	MicroSIM Interface	MicroSIM Slot	
CON13	RTC Backup Interface 1	Battery box	Compatible battery: CR1220 etc

Part Number	Interface Name	Form	Notes
CON14	Power Input Interface 1	DC jack	Plug: Inner diameter 2.1mm, outer diameter 5.5mm
CON15	Power Input Interface 2	Pin header 2 pin (2mm pitch)	
CON16	Power Output Interface	Pin header 2 pin (2mm pitch)	
CON17	3G Antenna Interface	Small coaxial connector	Connection cycles: 30 times[a]
CON20	WLAN Interface	PCI Express Mini Card connector	
CON22	RTC Backup Interface 2	Pin header 2 pin (1.25mm pitch)	Connection cycles: 20 times[a]
JP1	Boot Device Configuration Jumper	Pin header 2 pin (2.54mm pitch)	
SW1	User Switch 1	Tact switch	
SW2	User Switch 2	Tact switch	
SW3	User Switch 3	Tact switch	
SW4	Reset Switch	Tact switch	
LED1	WWAN LED	LED (green, surface mounted)	
LED2	User LED2	LED (green, surface mounted)	
LED3	User LED1	LED (green, surface mounted)	
LED4	User LED3	LED (green, surface mounted)	
LED5	User LED4	LED (green, surface mounted)	

[a]The connection life cycle count is a guideline at the time of product shipment, and it is not guaranteed to be the actual number of times the connection/disconnection can be performed.

3.4. Connections

The following shows an example of connections between Armadillo-IoT Gateway and peripheral devices.



1. Armadillo-IoT Gateway
2. AC adapter (12V)[4]
3. Work PC
4. USB serial converter[4]
5. USB2.0 cable (A-miniB type)[4]
6. LAN hub
7. LAN cable
8. USB Memory
9. SD Card
10. MicroSIM card

Figure 3.7. Armadillo-IoT Gateway Connection Example

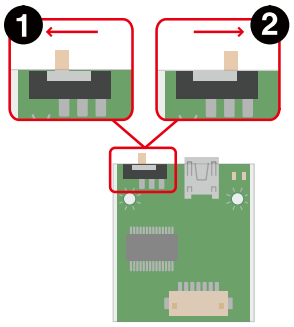
When using the AC adapter, connect the DC plug of the AC adapter to the Armadillo-IoT first and then plug the AC plug into the outlet.

[4]Armadillo-IoT Gateway development set accessories

When power cycling while using either the standard AC adapter (OP-AC12V2-00) or the wide temperature range AC adapter (OP-AC12V3-00)[4], wait at least 3 seconds before turning the power back on.

3.5. Slide Switch Configuration

The bootloader startup mode can be changed by operating the slide switch of the USB serial converter.



1 The bootloader will be in maintenance mode.[5]

2 The bootloader will be in auto boot mode.[6]

Figure 3.8. Slide Switch Configuration

3.6. Using the vi Editor

The vi editor is a text editor that is installed by default on Armadillo. In this document the vi editor is used for editing the Armadillo's configuration files.

Unlike the text editors such as gedit and emacs installed in ATDE, a major feature of the vi editor is that it has operating modes. It has a command mode and an input mode. All characters entered in command mode are treated as commands. In input mode, characters can be entered as text.

Although the command examples shown in this chapter are written to be executed in ATDE, they can be executed on Armadillo in a similar way.

3.6.1. Starting vi

To start vi, enter the following command.

```
[PC ~]# vi [file]
```

Figure 3.9. Starting vi

When the path of the filename is specified with *file*, the file will be edited (and the file will be created if *file* does not exist). vi starts in the command mode state.

[5]The bootloader command prompt starts.
[6]The operating system automatically boots.

3.6.2. Character Input

To enter characters, it is necessary to switch from command mode to input mode. To make this switch, enter the command shown in Table 3.5, “Commands to Enter Input Mode”. After entering input mode, characters are input as-is when their key is pressed.

Table 3.5. Commands to Enter Input Mode

Command	Operation
i	Start character input from the place where the cursor is located
a	Start character input from after the cursor

To return from the input mode to the command mode, press the ESC key. If you ever lose track of the current mode, just press the ESC key and return to the command mode once to avoid confusion.

Turning Off Japanese Conversion Functionality

When entering a vi command please turn off the ATDE Japanese input system (Mozc). Turning the Japanese input system on and off can be done with half-width / full-width key.

The starting position of character input when the "i" and "a" commands are entered is shown in Figure 3.10, “Explanation of Commands for Switching to Input Mode”.

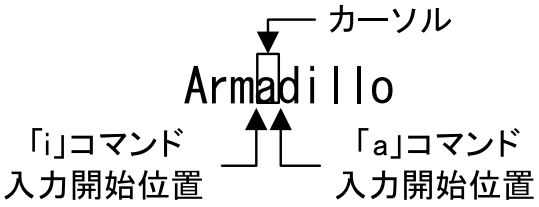


Figure 3.10. Explanation of Commands for Switching to Input Mode

Delete Characters in vi

Depending on the environment of the console, characters may not be deleted when using the BS (Backspace) key, with the "^ H" characters entered instead. In this case, delete characters using the command described in Section 3.6.4, “Deleting Characters”.

3.6.3. Moving the Cursor

The cursor can be moved with the arrow keys, but it is also possible to move the cursor by entering the commands shown in Table 3.6, “Commands to Move the Cursor” in command mode.

Table 3.6. Commands to Move the Cursor

Command	Operation
h	Move one character to the left

Command	Operation
j	Move one character down
k	Move one character up
l	Move one character to the right

3.6.4. Deleting Characters

To delete a character, enter the command shown in Table 3.7, “Commands to Delete Characters” in command mode.

Table 3.7. Commands to Delete Characters

Command	Operation
x	Delete the character under the cursor
dd	Delete the current line

The characters deleted when the "x" and "dd" commands are entered are shown in Figure 3.11, “Explanation of Commands for Deleting Characters”.

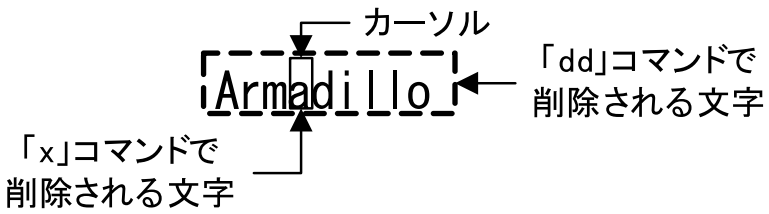


Figure 3.11. Explanation of Commands for Deleting Characters

3.6.5. Saving and Quitting

The commands for saving the file and quitting are shown inTable 3.8, “Commands for Saving and Quitting”.

Table 3.8. Commands for Saving and Quitting

Command	Operation
:q!	Quit without saving changes
:w *[file]*	Save with the filename specified in *file*
:wq	Save by overwriting the file and then quit

Commands for saving and quitting start with ":" (colon). When entering the ":" key the cursor will move to the bottom of the screen and the entered command will be displayed. After entering the command, press Enter to execute the command.

Chapter 4. Startup and Shutdown

4.1. Startup

When connecting power to Armadillo-IoT G3, the start mode changes depending on the slide switch on the USB serial converter. For details, refer to Section 3.5, “Slide Switch Configuration”. This section shows an example when maintenance mode is selected. When auto boot mode is selected, the startup completes without requiring any input.

```
U-Boot 2014.04-at2 (Jun 12 2016 - 17:46:27)

CPU:   Freescale i.MX7D rev1.1 at 792 MHz
CPU:   Temperature: can't get valid data!
Reset cause: POR
I2C:   ready
DRAM:  512 MiB
MMC:   FSL_SDHC: 0, FSL_SDHC: 1
SF: Detected N25Q64 with page size 256 Bytes, erase size 4 KiB, total 8 MiB
In:    serial
Out:   serial
Err:   serial
Found PFUZE300! deviceid 0x30, revid 0x11
Board Type: Armadillo-IoT G3(0a000000)
Revison: 0002
S/N: 2000
DRAM: 00001d05
XTAL: 00
Net:   FEC0
=>
```

Figure 4.1. Log Messages Immediately After Power On

To boot the Linux system, enter the "boot" command as shown below. When the command is executed, the bootloader starts the Linux system and the Linux boot log is displayed in the serial communication software.

```
=> boot
mmc1(part 0) is current device
mmc1(part 0) is current device
reading boot.scr
** Unable to read file boot.scr **
reading uImage
9301216 bytes read in 1560 ms (5.7 MiB/s)
Booting from mmc ...
reading armadillo_iotg_g3.dtb
52708 bytes read in 26 ms (1.9 MiB/s)
## Booting kernel from Legacy Image at 80800000 ...
   Image Name:   Linux-3.14.38-at2
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    9301152 Bytes = 8.9 MiB
   Load Address: 80008000
```

```
   Entry Point:  80008000
   Verifying Checksum ... OK
## Flattened Device Tree blob at 83000000
   Booting using the fdt blob at 0x83000000
   Loading Kernel Image ... OK
   Using Device Tree in place at 83000000, end 8300fde3

Starting kernel ...

Booting Linux on physical CPU 0x0
Linux version 3.14.38-at2 (ohsawa@pc-sienta) (gcc version 4.6.3 (Debian
4.6.3-14atmark1) ) #383 SMP PREEMPT Sun Jun 12 14:33:00 JST 2016
CPU: ARMv7 Processor [410fc075] revision 5 (ARMv7), cr=10c53c7d
CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing instruction cache
Machine model: Atmark-Techno Armadillo-IoT Gateway G3 Board
cma: CMA: reserved 320 MiB at 8c000000
Memory policy: Data cache writealloc
PERCPU: Embedded 8 pages/cpu @8bb35000 s8256 r8192 d16320 u32768
Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 130048
Kernel command line: console=ttymxc4,115200 root=/dev/mmcblk2p2 rootwait rw
PID hash table entries: 2048 (order: 1, 8192 bytes)
Dentry cache hash table entries: 65536 (order: 6, 262144 bytes)
Inode-cache hash table entries: 32768 (order: 5, 131072 bytes)
Memory: 172316K/524288K available (8487K kernel code, 504K rwdata, 6948K rodata,
2540K init, 440K bss, 351972K reserved, 0K highmem)
Virtual kernel memory layout:
    vector  : 0xffff0000 - 0xffff1000   (   4 kB)
    fixmap  : 0xfff00000 - 0xfffe0000   ( 896 kB)
    vmalloc : 0xa0800000 - 0xff000000   (1512 MB)
    lowmem  : 0x80000000 - 0xa0000000   ( 512 MB)
    pkmap   : 0x7fe00000 - 0x80000000   (   2 MB)
    modules : 0x7f000000 - 0x7fe00000   (  14 MB)
      .text : 0x80008000 - 0x80f1af20   (15436 kB)
      .init : 0x80f1b000 - 0x81196040   (2541 kB)
      .data : 0x81198000 - 0x81216360   ( 505 kB)
       .bss : 0x8121636c - 0x81284724   ( 441 kB)
SLUB: HWalign=64, Order=0-3, MinObjects=0, CPUs=2, Nodes=1
Preemptible hierarchical RCU implementation.
        RCU restricting CPUs from NR_CPUS=4 to nr_cpu_ids=2.
RCU: Adjusting geometry for rcu_fanout_leaf=16, nr_cpu_ids=2
NR_IRQS:16 nr_irqs:16 16
Switching to timer-based delay loop
sched_clock: 32 bits at 3000kHz, resolution 333ns, wraps every 1431655765682ns
Architected cp15 timer(s) running at 8.00MHz (phys).
sched_clock: 56 bits at 8MHz, resolution 125ns, wraps every 2147483648000ns
Ignoring duplicate/late registration of read_current_timer delay
Console: colour dummy device 80x30
Calibrating delay loop (skipped), value calculated using timer frequency.. 6.00
BogoMIPS (lpj=30000)
pid_max: default: 32768 minimum: 301
Mount-cache hash table entries: 1024 (order: 0, 4096 bytes)
Mountpoint-cache hash table entries: 1024 (order: 0, 4096 bytes)
CPU: Testing write buffer coherency: ok
/cpus/cpu@0 missing clock-frequency property
/cpus/cpu@1 missing clock-frequency property
CPU0: thread -1, cpu 0, socket 0, mpidr 80000000
Setting up static identity map for 0x8080e648 - 0x8080e6a0
CPU1: Booted secondary processor
```

```
CPU1: thread -1, cpu 1, socket 0, mpidr 80000001
Brought up 2 CPUs
SMP: Total of 2 processors activated (12.00 BogoMIPS).
CPU: All CPU(s) started in SVC mode.
devtmpfs: initialized
VFP support v0.3: implementor 41 architecture 2 part 30 variant 7 rev 5
pinctrl core: initialized pinctrl subsystem
regulator-dummy: no parameters
NET: Registered protocol family 16
DMA: preallocated 256 KiB pool for atomic coherent allocations
cpuidle: using governor ladder
cpuidle: using governor menu
Use WDOG1 as reset source
GPIO line 13 (MCU_INTB) hogged as input
GPIO line 43 (SVEN) hogged as output/high
GPIO line 85 (3G_VUSB_IN) hogged as output/high
GPIO line 113 (3G_IGT_3.3) hogged as output/high
syscon 30340000.iomuxc-gpr: regmap [mem 0x30340000-0x3034ffff] registered
syscon 30350000.ocotp-ctrl: regmap [mem 0x30350000-0x3035ffff] registered
syscon 30360000.anatop: regmap [mem 0x30360000-0x3036ffff] registered
vdd1p0d: 800 <--> 1200 mV at 1000 mV
vdd1p2: 1100 <--> 1300 mV
syscon 30390000.src: regmap [mem 0x30390000-0x3039ffff] registered
DDR type is DDR3!
prom_parse: Bad cell count for /regulators0/regulator@0
hw-breakpoint: found 5 (+1 reserved) breakpoint and 4 watchpoint registers.
hw-breakpoint: maximum watchpoint size is 8 bytes.
imx7d-pinctrl 302c0000.iomuxc-lpsr: initialized IMX pinctrl driver
imx7d-pinctrl 30330000.iomuxc: Invalid fsl,pins property in node /soc/aips-
bus@30000000/iomuxc@30330000/imx7d-sdb/uart1grp
imx7d-pinctrl 30330000.iomuxc: Invalid fsl,pins property in node /soc/aips-
bus@30000000/iomuxc@30330000/imx7d-sdb/uart2grp
imx7d-pinctrl 30330000.iomuxc: Invalid fsl,pins property in node /soc/aips-
bus@30000000/iomuxc@30330000/imx7d-sdb/ecspi1grp
imx7d-pinctrl 30330000.iomuxc: Invalid fsl,pins property in node /soc/aips-
bus@30000000/iomuxc@30330000/imx7d-sdb/ecspi2grp
imx7d-pinctrl 30330000.iomuxc: initialized IMX pinctrl driver
MU is ready for cross core communication!
bio: create slab <bio-0> at 0
mxs-dma 33000000.dma-apbh: initialized
USB_OTG1_VBUS: 5000 mV
VDD_SD1: 3300 mV
vgaarb: loaded
i2c-core: driver [max17135] using legacy suspend method
i2c-core: driver [max17135] using legacy resume method
SCSI subsystem initialized
usbcore: registered new interface driver usbfs
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
30800000.aips-bus:usbphy_nop1 supply vcc not found, using dummy regulator
30800000.aips-bus:usbphy_nop2 supply vcc not found, using dummy regulator
i2c i2c-0: IMX I2C adapter registered
gpio_bmic 3-0014: version: 1.0
bmic_regulator 3-0016: version: 1.0
BMIC_VREF: 1710 <--> 3600 mV at 2980 mV
i2c i2c-3: IMX I2C adapter registered
Linux video capture interface: v2.00
pps_core: LinuxPPS API ver. 1 registered
```

```
pps_core: Software ver. 5.3.6 - Copyright 2005-2007 Rodolfo Giometti
<giometti@linux.it>
PTP clock support registered
MIPI CSI2 driver module loaded
Advanced Linux Sound Architecture Driver Initialized.
Bluetooth: Core ver 2.18
NET: Registered protocol family 31
Bluetooth: HCI device and connection manager initialized
Bluetooth: HCI socket layer initialized
Bluetooth: L2CAP socket layer initialized
Bluetooth: SCO socket layer initialized
cfg80211: Calling CRDA to update world regulatory domain
armadillo_iotg_addon addon: No add-on expansion board detected at CON1.
armadillo_iotg_addon addon: No add-on expansion board detected at CON2.
Switched to clocksource arch_sys_counter
NET: Registered protocol family 2
TCP established hash table entries: 4096 (order: 2, 16384 bytes)
TCP bind hash table entries: 4096 (order: 3, 32768 bytes)
TCP: Hash tables configured (established 4096 bind 4096)
TCP: reno registered
UDP hash table entries: 256 (order: 1, 8192 bytes)
UDP-Lite hash table entries: 256 (order: 1, 8192 bytes)
NET: Registered protocol family 1
RPC: Registered named UNIX socket transport module.
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
RPC: Registered tcp NFSv4.1 backchannel transport module.
imx rpmsg driver is registered.
Bus freq driver module loaded
futex hash table entries: 512 (order: 3, 32768 bytes)
VFS: Disk quotas dquot_6.5.2
Dquot-cache hash table entries: 1024 (order 0, 4096 bytes)
squashfs: version 4.0 (2009/01/31) Phillip Lougher
NFS: Registering the id_resolver key type
Key type id_resolver registered
Key type id_legacy registered
jffs2: version 2.2. (NAND) c 2001-2006 Red Hat, Inc.
fuse init (API version 7.22)
msgmni has been set to 976
io scheduler noop registered
io scheduler deadline registered
io scheduler cfq registered (default)
imx-sdma 30bd0000.sdma: no event needs to be remapped
imx-sdma 30bd0000.sdma: loaded firmware 4.1
imx-sdma 30bd0000.sdma: initialized
pfuze100-regulator 3-0009: Full layer: 1, Metal layer: 1
pfuze100-regulator 3-0009: FAB: 0, FIN: 0
pfuze100-regulator 3-0009: pfuze3000 found.
SW1A: 700 <--> 1475 mV at 1100 mV
SW1B: 700 <--> 1475 mV at 1000 mV
SW2: 1500 <--> 1850 mV at 1800 mV
SW3: 900 <--> 1650 mV at 1350 mV
SWBST: 5000 <--> 5150 mV at 5000 mV
VSNVS: 1000 <--> 3000 mV at 3000 mV
VREFDDR: 750 mV
VLDO1: 1800 <--> 3300 mV at 1800 mV
VLDO2: 800 <--> 1550 mV at 1500 mV
VCCSD: 2850 <--> 3300 mV at 3300 mV
```

```
V33: 2850 <--> 3300 mV at 3300 mV
VLDO3: 1800 <--> 3300 mV at 3300 mV
VLDO4: 1800 <--> 3300 mV at 3300 mV
30a70000.serial: ttymxc4 at MMIO 0x30a70000 (irq = 62, base_baud = 1500000) is a IMX
console [ttymxc4] enabled
serial: Freescale lpuart driver
imx sema4 driver is registered.
[drm] Initialized drm 1.1.0 20060810
[drm] Initialized vivante 1.0.0 20120216 on minor 0
brd: module loaded
loop: module loaded
fsl-quadspi 30bb0000.qspi: Unsupported cmd 0x65
fsl-quadspi 30bb0000.qspi: Unsupported cmd 0x61
fsl-quadspi 30bb0000.qspi: Unsupported cmd 0x65
fsl-quadspi 30bb0000.qspi: n25q064 (8192 Kbytes)
3 ofpart partitions found on MTD device 30bb0000.qspi
Creating 3 MTD partitions on "30bb0000.qspi":
0x000000000000-0x000000100000 : "bootloader"
0x000000100000-0x000000140000 : "license"
0x000000140000-0x000000800000 : "reserved"
fsl-quadspi 30bb0000.qspi: QuadSPI SPI NOR flash driver
CAN device driver interface
30bf0000.ethernet supply phy not found, using dummy regulator
pps pps0: new PPS source ptp0
libphy: fec_enet_mii_bus: probed
fec 30bf0000.ethernet eth0: registered PHC device 0
PPP generic driver version 2.4.2
usbcore: registered new interface driver cdc_ether
usbcore: registered new interface driver cdc_eem
usbcore: registered new interface driver sierra_net
usbcore: registered new interface driver qmi_wwan
ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
ehci-pci: EHCI PCI platform driver
ehci-mxc: Freescale On-Chip EHCI Host driver
usbcore: registered new interface driver cdc_acm
cdc_acm: USB Abstract Control Model driver for USB modems and ISDN adapters
usbcore: registered new interface driver cdc_wdm
usbcore: registered new interface driver usb-storage
usbcore: registered new interface driver usbserial
usbcore: registered new interface driver usbserial_generic
usbserial: USB Serial support registered for generic
usbcore: registered new interface driver ftdi_sio
usbserial: USB Serial support registered for FTDI USB Serial Device
usbcore: registered new interface driver option
usbserial: USB Serial support registered for GSM modem (1-port)
usbcore: registered new interface driver sierra
usbserial: USB Serial support registered for Sierra USB modem
usbcore: registered new interface driver usb_serial_simple
usbserial: USB Serial support registered for zio
usbserial: USB Serial support registered for funsoft
usbserial: USB Serial support registered for flashloader
usbserial: USB Serial support registered for vivopay
usbserial: USB Serial support registered for moto_modem
usbserial: USB Serial support registered for hp4x
usbserial: USB Serial support registered for suunto
usbserial: USB Serial support registered for siemens_mpi
usb3503 3-0008: switched to HUB mode
usb3503 3-0008: usb3503_probe: probed in hub mode
```

```
30b10200.usbmisc supply vbus-wakeup not found, using dummy regulator
30b20200.usbmisc supply vbus-wakeup not found, using dummy regulator
30b30200.usbmisc supply vbus-wakeup not found, using dummy regulator
ci_hdrc ci_hdrc.0: EHCI Host Controller
ci_hdrc ci_hdrc.0: new USB bus registered, assigned bus number 1
ci_hdrc ci_hdrc.0: USB 2.0 started, EHCI 1.00
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 1 port detected
30b30000.usb supply vbus not found, using dummy regulator
ci_hdrc ci_hdrc.1: EHCI Host Controller
ci_hdrc ci_hdrc.1: new USB bus registered, assigned bus number 2
ci_hdrc ci_hdrc.1: USB 2.0 started, EHCI 1.00
hub 2-0:1.0: USB hub found
hub 2-0:1.0: 1 port detected
mousedev: PS/2 mouse device common for all mice
input: 30370000.snvs-pwrkey as /devices/soc/30000000.aips-bus/30370000.snvs-
pwrkey/input/input0
snvs_pwrkey 30370000.snvs-pwrkey: i.MX snvs powerkey probed
i2c-core: driver [isl29023] using legacy suspend method
i2c-core: driver [isl29023] using legacy resume method
bmic_rtc 3-0011: version: 1.0
bmic_rtc 3-0011: rtc core: registered bmic_rtc as rtc0
snvs_rtc 30370034.snvs-rtc-lp: rtc core: registered 30370034.snvs-rtc-l as rtc1
i2c /dev entries driver
IR NEC protocol handler initialized
IR RC5(x) protocol handler initialized
IR RC6 protocol handler initialized
IR JVC protocol handler initialized
IR Sony protocol handler initialized
IR RC5 (streamzap) protocol handler initialized
IR SANYO protocol handler initialized
IR MCE Keyboard/mouse protocol handler initialized
usbcore: registered new interface driver uvcvideo
USB Video Class driver (1.1.1)
i2c-core: driver [mag3110] using legacy suspend method
i2c-core: driver [mag3110] using legacy resume method
bmic_thermal 3-0013: version: 1.0
imx2-wdt 30280000.wdog: timeout 10 sec (nowayout=0)
Bluetooth: HCI UART driver ver 2.2
Bluetooth: HCI H4 protocol initialized
Bluetooth: HCI BCSP protocol initialized
Bluetooth: HCIATH3K protocol initialized
usbcore: registered new interface driver bcm203x
usbcore: registered new interface driver btusb
usbcore: registered new interface driver ath3k
sdhci: Secure Digital Host Controller Interface driver
sdhci: Copyright(c) Pierre Ossman
sdhci-pltfm: SDHCI platform and OF driver helper
mmc0: no vqmmc regulator found
mmc0: SDHCI controller on 30b40000.usdhc [30b40000.usdhc] using ADMA
sdhci-esdhc-imx 30b60000.usdhc: could not get ultra high speed state, work on
normal mode
mmc2: no vqmmc regulator found
mmc2: no vmmc regulator found
mmc2: SDHCI controller on 30b60000.usdhc [30b60000.usdhc] using ADMA
usb 2-1: new high-speed USB device number 2 using ci_hdrc
caam 30900000.caam: Instantiated RNG4 SH0
mmc2: BKOPS_EN bit is not set
```

```
caam 30900000.caam: Instantiated RNG4 SH1
caam 30900000.caam: device ID = 0x0a160300 (Era 8)
caam 30900000.caam: job rings = 3, qi = 0
mmc2: new high speed DDR MMC card at address 0001
mmcblk2: mmc2:0001 Q1J55L 3.56 GiB
mmcblk2boot0: mmc2:0001 Q1J55L partition 1 2.00 MiB
mmcblk2boot1: mmc2:0001 Q1J55L partition 2 2.00 MiB
mmcblk2rpmb: mmc2:0001 Q1J55L partition 3 4.00 MiB
 mmcblk2: p1 p2 p3
 mmcblk2boot1: unknown partition table
 mmcblk2boot0: unknown partition table
hub 2-1:1.0: USB hub found
hub 2-1:1.0: 3 ports detected
caam algorithms registered in /proc/crypto
caam_jr 30901000.jr0: registering rng-caam
snvs-secvio 30370000.caam-snvs: violation handlers armed - non-secure state
usbcore: registered new interface driver usbhid
usbhid: USB HID core driver
usbcore: registered new interface driver r8712u
bmic_adc 3-0012: version: 1.0
coresight-tmc 30086000.etr: TMC initialized
coresight-tmc 30084000.etf: TMC initialized
coresight-tpiu 30087000.tpiu: TPIU initialized
coresight-funnel 30083000.funnel: FUNNEL initialized
coresight-funnel 30041000.funnel: FUNNEL initialized
coresight-replicator replicator: REPLICATOR initialized
coresight-etm3x 3007c000.etm: ETM initialized
coresight-etm3x 3007d000.etm: ETM initialized
usbcore: registered new interface driver snd-usb-audio
NET: Registered protocol family 26
nf_conntrack version 0.5.0 (7812 buckets, 31248 max)
ipip: IPv4 over IPv4 tunneling driver
gre: GRE over IPv4 demultiplexor driver
ip_gre: GRE over IPv4 tunneling driver
ip_tables: (C) 2000-2006 Netfilter Core Team
TCP: cubic registered
Initializing XFRM netlink socket
NET: Registered protocol family 10
mip6: Mobile IPv6
ip6_tables: (C) 2000-2006 Netfilter Core Team
sit: IPv6 over IPv4 tunneling driver
ip6_gre: GRE over IPv6 tunneling driver
NET: Registered protocol family 17
NET: Registered protocol family 15
can: controller area network core (rev 20120528 abi 9)
NET: Registered protocol family 29
can: raw protocol (rev 20120528)
can: broadcast manager protocol (rev 20120528 t)
can: netlink gateway (rev 20130117) max_hops=1
Bluetooth: RFCOMM TTY layer initialized
Bluetooth: RFCOMM socket layer initialized
Bluetooth: RFCOMM ver 1.11
Bluetooth: BNEP (Ethernet Emulation) ver 1.3
Bluetooth: BNEP filters: protocol multicast
Bluetooth: BNEP socket layer initialized
Bluetooth: HIDP (Human Interface Emulation) ver 1.2
Bluetooth: HIDP socket layer initialized
8021q: 802.1Q VLAN Support v1.8
```

```
Key type dns_resolver registered
cpu cpu0: dev_pm_opp_get_opp_count: device OPP not found (-19)
imx6q-pcie 33800000.pcie: PCI host bridge to bus 0000:00
pci_bus 0000:00: root bus resource [io  0x1000-0x10000]
pci_bus 0000:00: root bus resource [mem 0x40000000-0x4fefffff]
pci_bus 0000:00: root bus resource [bus 00-ff]
PCI: bus0: Fast back to back transfers disabled
PCI: bus1: Fast back to back transfers disabled
pci 0000:00:00.0: BAR 0: assigned [mem 0x40000000-0x400fffff]
pci 0000:00:00.0: BAR 14: assigned [mem 0x40100000-0x401fffff]
pci 0000:00:00.0: BAR 15: assigned [mem 0x40200000-0x402fffff pref]
pci 0000:00:00.0: BAR 6: assigned [mem 0x40300000-0x4030ffff pref]
pci 0000:01:00.0: BAR 0: assigned [mem 0x40100000-0x4017ffff 64bit]
pci 0000:01:00.0: BAR 6: assigned [mem 0x40200000-0x4020ffff pref]
pci 0000:00:00.0: PCI bridge to [bus 01]
pci 0000:00:00.0:   bridge window [mem 0x40100000-0x401fffff]
pci 0000:00:00.0:   bridge window [mem 0x40200000-0x402fffff pref]
pcieport 0000:00:00.0: Signaling PME through PCIe PME interrupt
pci 0000:01:00.0: Signaling PME through PCIe PME interrupt
PCI: enabling device 0000:01:00.0 (0140 -> 0142)
ieee80211 phy0: Atheros AR9462 Rev:2 mem=0xa0b00000, irq=157
VDD_SD1: disabling
regulator-dummy: disabling
imx mcc test is registered.
input: gpio-keys as /devices/gpio-keys/input/input1
bmic_rtc 3-0011: setting system clock to 1970-01-01 00:00:18 UTC (18)
ALSA device list:
  No soundcards found.
usb 2-1.3: new full-speed USB device number 3 using ci_hdrc
Warning: unable to open an initial console.
Freeing unused kernel memory: 2540K (80f1b000 - 81196000)
systemd-udevd[160]: starting version 215
random: systemd-udevd urandom read with 27 bits of entropy available
usb 2-1.3: string descriptor 0 read error: -22
usb 2-1.3: USB disconnect, device number 3
EXT4-fs (mmcblk2p2): mounted filesystem with ordered data mode. Opts: (null)
systemd[1]: systemd 215 running in system mode. (+PAM +AUDIT +SELINUX +IMA
+SYSVINIT +LIBCRYPTSETUP +GCRYPT +ACL +XZ -SECCOMP -APPARMOR)
usb 2-1.3: new full-speed USB device number 4 using ci_hdrc
systemd[1]: Detected architecture 'arm'.

Welcome to Debian GNU/Linux 8 (jessie)!

systemd[1]: Set hostname to <armadillo-iotg>.
systemd[1]: /etc/mtab is not a symlink or not pointing to /proc/self/mounts. This
is not supported anymore. Please make sure to replace this file by a symlink to
avoid incorrect or misleading mount(8) output.
usb 2-1.3: string descriptor 0 read error: -22
systemd[1]: Cannot add dependency job for unit display-manager.service, ignoring:
Unit display-manager.service failed to load: No such file or directory.
systemd[1]: Expecting device dev-ttymxc4.device...
         Expecting device dev-ttymxc4.device...
systemd[1]: Starting Forward Password Requests to Wall Directory Watch.
systemd[1]: Started Forward Password Requests to Wall Directory Watch.
systemd[1]: Starting Remote File Systems (Pre).
[  OK  ] Reached target Remote File Systems (Pre).
systemd[1]: Reached target Remote File Systems (Pre).
systemd[1]: Starting Encrypted Volumes.
```

```
[  OK  ] Reached target Encrypted Volumes.
systemd[1]: Reached target Encrypted Volumes.
systemd[1]: Starting Dispatch Password Requests to Console Directory Watch.
systemd[1]: Started Dispatch Password Requests to Console Directory Watch.
systemd[1]: Starting Paths.
[  OK  ] Reached target Paths.
systemd[1]: Reached target Paths.
usb 2-1.2: new high-speed USB device number 5 using ci_hdrc
systemd[1]: Starting Arbitrary Executable File Formats File System Automount Point.
[  OK  ] Set up automount Arbitrary Executable File Formats F...utomount Point.
systemd[1]: Set up automount Arbitrary Executable File Formats File System
Automount Point.
systemd[1]: Starting Swap.
[  OK  ] Reached target Swap.
systemd[1]: Reached target Swap.
systemd[1]: Starting Root Slice.
[  OK  ] Created slice Root Slice.
systemd[1]: Created slice Root Slice.
systemd[1]: Starting User and Session Slice.
[  OK  ] Created slice User and Session Slice.
systemd[1]: Created slice User and Session Slice.
systemd[1]: Starting /dev/initctl Compatibility Named Pipe.
[  OK  ] Listening on /dev/initctl Compatibility Named Pipe.
systemd[1]: Listening on /dev/initctl Compatibility Named Pipe.
systemd[1]: Starting Delayed Shutdown Socket.
[  OK  ] Listening on Delayed Shutdown Socket.
systemd[1]: Listening on Delayed Shutdown Socket.
systemd[1]: Starting Journal Socket (/dev/log).
[  OK  ] Listening on Journal Socket (/dev/log).
systemd[1]: Listening on Journal Socket (/dev/log).
systemd[1]: Starting udev Control Socket.
[  OK  ] Listening on udecdc_acm 2-1.2:1.0: ttyACM0: USB ACM device
v Control Socket.
cdc_acm 2-1.2:1.2: ttyACM1: USB ACM device
systemd[1]: Listening on udev Control Socket.
systemd[1]: Starting udev Kernel Socket.
[  OK  ] Listening on udecdc_acm 2-1.2:1.4: ttyACM2: USB ACM device
v Kernel Socket.
cdc_acm 2-1.2:1.6: ttyACM3: USB ACM device
systemd[1]: Listening on udev Kernel Socket.
systemd[1]: Starting Journal Socket.
[  OK  ] Listening on Journal Socket.
cdc_acm 2-1.2:1.11: ttyACM4: USB ACM device
cdc_acm 2-1.2:1.13: ttyACM5: USB ACM device
systemd[1]: Listening on Journal Socket.
systemd[1]: Starting System Slice.
[  OK  ] Created slice System Slice.
systemd[1]: Created slice System Slice.
systemd[1]: Starting system-getty.slice.
[  OK  ] Created slice system-getty.slice.
systemd[1]: Created slice system-getty.slice.
systemd[1]: Starting system-serial\x2dgetty.slice.
[  OK  ] Created slice system-serial\x2dgetty.slice.
systemd[1]: Created slice system-serial\x2dgetty.slice.
systemd[1]: Starting Increase datagram queue length...
         Starting Increase datagram queue length...
systemd[1]: Starting Load Kernel Modules...
         Starting Load Kernel Modules...
```

```
systemd[1]: Started Create list of required static device nodes for the current
kernel.
systemd[1]: Mounting Debug File System...
         Mounting Debug File System...
systemd[1]: Starting udev Coldplug all Devices...
         Starting udev Coldplug all Devices...
systemd[1]: Starting Create Static Device Nodes in /dev...
         Starting Create Static Device Nodes in /dev...
systemd[1]: Mounted Huge Pages File System.
systemd[1]: Mounted POSIX Message Queue File System.
systemd[1]: Started Set Up Additional Binary Formats.
systemd[1]: Starting Slices.
[  OK  ] Reached target Slices.
systemd[1]: Reached target Slices.
systemd[1]: Starting Remount Root and Kernel File Systems...
         Starting Remount Root and Kernel File Systems...
systemd[1]: Expecting device dev-mtdblock1.device...
         Expecting device dev-mtdblock1.device...
[  OK  ] Mounted Debug File System.
systemd[1]: Mounted Debug File System.
[  OK  ] Started Increase datagram queue length.
systemd[1]: Started Increase datagram queue length.
[  OK  ] Started Load Kernel Modules.
systemd[1]: Started Load Kernel Modules.
[  OK  ] Started Create Static Device Nodes in /dev.
systemd[1]: Started Create Static Device Nodes in /dev.
[  OK  ] Started Remount Root and Kernel File Systems.
systemd[1]: Started Remount Root and Kernel File Systems.
[  OK  ] Started udev Coldplug all Devices.
systemd[1]: Started udev Coldplug all Devices.
systemd[1]: Starting Various fixups to make systemd work better on Debian...
         Starting Various fixups to make systemd work better on Debian...
systemd[1]: Starting Load/Save Random Seed...
         Starting Load/Save Random Seed...
systemd[1]: Starting udev Kernel Device Manager...
         Starting udev Kernel Device Manager...
systemd[1]: Starting Local File Systems (Pre).
[  OK  ] Reached target Local File Systems (Pre).
systemd-udevd[248]: starting version 215
systemd[1]: Reached target Local File Systems (Pre).
systemd[1]: Starting Local File Systems.
[  OK  ] Reached target Local File Systems.
systemd[1]: Reached target Local File Systems.
systemd[1]: Starting Create Volatile Files and Directories...
         Starting Create Volatile Files and Directories...
systemd[1]: Starting Remote File Systems.
[  OK  ] Reached target Remote File Systems.
systemd[1]: Reached target Remote File Systems.
systemd[1]: Mounting FUSE Control File System...
         Mounting FUSE Control File System...
systemd[1]: Mounting Configuration File System...
         Mounting Configuration File System...
systemd[1]: Starting Apply Kernel Variables...
         Starting Apply Kernel Variables...
systemd[1]: Starting Syslog Socket.
[  OK  ] Listening on Syslog Socket.
systemd[1]: Listening on Syslog Socket.
systemd[1]: Starting Journal Service...
```

```
          Starting Journal Service...
[  OK  ] Started Journal Service.
systemd[1]: Started Journal Service.
          Starting Trigger Flushing of Journal to Persistent Storage...
[  OK  ] Mounted Configuration File System.
[  OK  ] Mounted FUSE Control File System.
[  OK  ] Started udev Kernel Device Manager.
[  OK  ] Started Various fixups to make systemd work better on Debian.
[  OK  ] Started Load/Save Random Seed.
[  OK  ] Started Create Volatile Files and Directories.
[  OK  ] Started Apply Kernel Variables.
random: nonblocking pool is initialized
[  OK  ] Found device /dev/ttymxc4.
[  OK  ] Found device /dev/mtdblock1.
systemd-journald[270]: Received request to flush runtime journal from PID 1
[  OK  ] Started Trigger Flushing of Journal to Persistent Storage.
[  OK  ] Created slice system-systemd\x2drfkill.slice.
          Starting Load/Save RF Kill Switch Status of rfkill0...
          Starting Load/Save RF Kill Switch Status of rfkill1...
          Starting Trigger Flushing of Journal to Persistent Storage...
          Mounting /opt/license...
[  OK  ] Reached target Sound Card.
          Starting Update UTMP about System Boot/Shutdown...
          Starting LSB: Raise network interfaces....
          Starting Copy rules generated while the root was ro...
[  OK  ] Mounted /opt/license.
[  OK  ] Started Load/Save RF Kill Switch Status of rfkill0.
[  OK  ] Started Load/Save RF Kill Switch Status of rfkill1.
[  OK  ] Started Copy rules generated while the root was ro.
systemd-journald[270]: Received request to flush runtime journal from PID 1
[  OK  ] Started Trigger Flushing of Journal to Persistent Storage.
[  OK  ] Started Update UTMP about System Boot/Shutdown.
[  OK  ] Started LSB: Raise network interfaces..
[  OK  ] Reached target Network.
[  OK  ] Reached target Network is Online.
[  OK  ] Reached target System Initialization.
[  OK  ] Listening on Avahi mDNS/DNS-SD Stack Activation Socket.
[  OK  ] Listening on D-Bus System Message Bus Socket.
[  OK  ] Reached target Sockets.
[  OK  ] Reached target Timers.
[  OK  ] Reached target Basic System.
          Starting Bluetooth service...
          Starting Modem Manager...
          Starting Regular background program processing daemon...
[  OK  ] Started Regular background program processing daemon.
          Starting Restore /etc/resolv.conf if the system cras...s shut down....
          Starting Lighttpd Daemon...
          Starting Network Manager...
          Starting /etc/rc.local Compatibility...
          Starting Login Service...
          Starting LSB: exim Mail Transport Agent...
          Starting Avahi mDNS/DNS-SD Stack...
          Starting D-Bus System Message Bus...
[  OK  ] Started D-Bus System Message Bus.
[  OK  ] Started Avahi mDNS/DNS-SD Stack.
[  OK  ] Started Bluetooth service.
[  OK  ] Reached target Bluetooth.
          Starting System Logging Service...
```

```
          Starting Permit User Sessions...
[  OK  ] Started Restore /etc/resolv.conf if the system crash...was shut down..
[  OK  ] Started Permit User Sessions.
[  OK  ] Started System Logging Service.
[  OK  ] Started Login Service.
          Starting Hostname Service...
          Starting Authenticate and Authorize Users to Run Privileged Tasks...
[  OK  ] Started /etc/rc.local Compatibility.
[  OK  ] Started Hostname Service.
          Starting change status LED...
          Starting Getty on tty1...
[  OK  ] Started Getty on tty1.
          Starting Serial Getty on ttymxc4...
[  OK  ] Started Serial Getty on ttymxc4.
[  OK  ] Reached target Login Prompts.
[  OK  ] Started Lighttpd Daemon.
[  OK  ] Started change status LED.
[  OK  ] Started Authenticate and Authorize Users to Run Privileged Tasks.
[  OK  ] Started Modem Manager.
[  OK  ] Started Network Manager.
fec 30bf0000.ethernet eth0: Freescale FEC PHY driver [Vitesse VSC8501]
(mii_bus:phy_addr=30bf0000.etherne:00, irq=-1)
IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
          Starting WPA supplicant...
[  OK  ] Started WPA supplicant.
[  OK  ] Started LSB: exim Mail Transport Agent.
[  OK  ] Reached target Multi-User System.
[  OK  ] Reached target Graphical Interface.
          Starting Update UTMP about System Runlevel Changes...
[  OK  ] Started Update UTMP about System Runlevel Changes.

Debian GNU/Linux 8 armadillo-iotg ttymxc4

armadillo-iotg login:
```

Figure 4.2. Boot Log

The startup status of Armadillo-IoT can be checked with the user LEDs. Therefore, the startup status of Linux can be confirmed even if turning on power without connecting the USB serial converter to Armadillo-IoT.

Boot State	User LED1	User LED2	User LED3	User LED4
Bootloader is in maintenance mode	On	Off	Off	Off
While Linux is booting	On	On	Off	Off
Once Linux has completed booting	On	Off	Off	Off

For the location of the user LEDs, see Figure 5.41, “User LEDs Location”.

4.2. Login

The login prompt will be displayed once the boot has completed. The usernames shown in Table 4.1, “Serial Console Login Username and Password” can be used to login.

Table 4.1. Serial Console Login Username and Password

Username	Password	Permissions
root	root	root user
atmark	atmark	general user

When connecting the device for the first time, please make sure to change the default password by following the procedure below.

1. Login as root

   Change the default password.

```
[armadillo ~]# passwd
Enter new UNIX password: # Enter new password here
Retype new UNIX password: # Re-enter
```

2. Login as atmark

   Change the default password.

```
[armadillo ~]$ passwd
Enter new UNIX password: # Enter new password here
Retype new UNIX password: # Re-enter
[armadillo ~]$
```

With Armadillo-IoT G3 being a device that can be connected to a network, the security risk will be extremely high if used with the default password. It is strongly recommended to change the password to one with high security strength, and then manage the password appropriately thereafter.

4.3. Managing Debian Users

1. Creating a User

   Here we create a user named guest as an example.

```
[armadillo ~]# adduser guest
Adding user `[user_name]' ...
Adding new group `guest' (1001) ...
Adding new user `guest' (1001) with group `guest' ...
Creating home directory `/home/guest' ...
Copying files from `/etc/skel' ...
Enter new UNIX password: # Enter password here
Retype new UNIX password: # Re-enter
```

```
passwd: password updated successfully
Changing the user information for guest
Enter the new value, or press ENTER for the default
        Full Name []: # press ENTER
        Room Number []: # press ENTER
        Work Phone []: # press ENTER
        Home Phone []: # press ENTER
        Other []: # press ENTER
Is the information correct? [Y/n] # press ENTER
```

2. Changing the Password

   Here we change the password of the guest user as an example.

```
[armadillo ~]# passwd guest
Enter new UNIX password: # Enter new password here
Retype new UNIX password: # Re-enter
```

3. Permitting Use of sudo

   Here we give permission to use sudo to the guest user as an example. For details on how to use vi, please refer to Section 3.6, “Using the vi Editor”.

```
[armadillo ~]# visudo
...
# User privilege specification
root    ALL=(ALL:ALL) ALL
guest   ALL=(ALL:ALL) ALL # Add this line
...
```

4. Deleting a User

   Here we delete the guest user as an example.

```
[armadillo ~]# userdel guest
```

To delete the home directory at the same time, use the "r" option.

```
[armadillo ~]# userdel -r guest
```

4.4. Shutdown

To perform a safe shutdown, execute the command shown below, wait until the "System halted." message is displayed and then disconnect the power supply.

```
[armadillo ~]# halt
        Starting Synchronise Hardware Clock to System Clock..[  OK  ] Stopped
```

```
target Bluetooth.
[  OK  ] Stopped target Sound Card.
         Stopping User Manager for UID 0...
         Stopping WPA supplicant...
         Stopping Authenticate and Authorize Users to Run Privileged Tasks...
         Stopping Bluetooth service...
[  OK  ] Stopped target Graphical Interface.
[  OK  ] Stopped target Multi-User System.
         Stopping Modem Manager...
         Stopping Regular background program processing daemon...
         Stopping Network Manager...
         Stopping Lighttpd Daemon...
IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
[  OK  ] Stopped target Login Prompts.
         Stopping Getty on tty1...
         Stopping Serial Getty on ttymxc4...
         Stopping LSB: exim Mail Transport Agent...
         Stopping Avahi mDNS/DNS-SD Stack...
         Stopping D-Bus System Message Bus...
         Stopping System Logging Service...
[  OK  ] Stopped Bluetooth service.
[  OK  ] Stopped Modem Manager.
[  OK  ] Stopped Regular background program processing daemon.
[  OK  ] Stopped Network Manager.
[  OK  ] Stopped Avahi mDNS/DNS-SD Stack.
[  OK  ] Stopped D-Bus System Message Bus.
[  OK  ] Stopped System Logging Service.
[  OK  ] Stopped Lighttpd Daemon.
[  OK  ] Stopped Authenticate and Authorize Users to Run Privileged Tasks.
[  OK  ] Stopped Getty on tty1.
[  OK  ] Stopped Serial Getty on ttymxc4.
[  OK  ] Stopped WPA supplicant.
[  OK  ] Stopped User Manager for UID 0.
[  OK  ] Unmounted /opt/license.
[  OK  ] Stopped LSB: exim Mail Transport Agent.
[  OK  ] Stopped target Network is Online.
         Stopping Login Service...
[  OK  ] Removed slice user-0.slice.
[  OK  ] Removed slice system-serial\x2dgetty.slice.
[  OK  ] Removed slice system-getty.slice.
         Stopping /etc/rc.local Compatibility...
[  OK  ] Stopped /etc/rc.local Compatibility.
         Stopping Permit User Sessions...
[  OK  ] Stopped target Network.
[  OK  ] Stopped Login Service.
[  OK  ] Stopped Permit User Sessions.
[  OK  ] Stopped target Remote File Systems.
[  OK  ] Stopped target Remote File Systems (Pre).
[  OK  ] Stopped target Basic System.
[  OK  ] Stopped target Slices.
[  OK  ] Stopped target Paths.
[  OK  ] Stopped target Timers.
[  OK  ] Stopped target Sockets.
[  OK  ] Closed Avahi mDNS/DNS-SD Stack Activation Socket.
[  OK  ] Closed Syslog Socket.
[  OK  ] Closed D-Bus System Message Bus Socket.
[  OK  ] Stopped target System Initialization.
         Stopping Load/Save RF Kill Switch Status of rfkill1...
```

```
          Stopping Load/Save RF Kill Switch Status of rfkill0...
          Stopping Update UTMP about System Boot/Shutdown...
[  OK  ] Stopped target Encrypted Volumes.
          Stopping Apply Kernel Variables...
[  OK  ] Stopped Apply Kernel Variables.
          Stopping Load Kernel Modules...
[  OK  ] Stopped Load Kernel Modules.
          Stopping LSB: Raise network interfaces....
[  OK  ] Stopped target Swap.
[  OK  ] Removed slice User and Session Slice.
[  OK  ] Stopped Load/Save RF Kill Switch Status of rfkill1.
[  OK  ] Stopped Load/Save RF Kill Switch Status of rfkill0.
[  OK  ] Stopped Update UTMP about System Boot/Shutdown.
[  OK  ] Stopped LSB: Raise network interfaces..
          Stopping Load/Save Random Seed...
          Stopping Create Volatile Files and Directories...
[  OK  ] Stopped Create Volatile Files and Directories.
[  OK  ] Stopped target Local File Systems.
          Unmounting /run/user/0...
[  OK  ] Removed slice system-systemd\x2drfkill.slice.
[  OK  ] Started Synchronise Hardware Clock to System Clock.
[  OK  ] Stopped Load/Save Random Seed.
[  OK  ] Unmounted /run/user/0.
[  OK  ] Reached target Unmount All Filesystems.
[  OK  ] Stopped target Local File Systems (Pre).
          Stopping Create Static Device Nodes in /dev...
[  OK  ] Stopped Create Static Device Nodes in /dev.
          Stopping Remount Root and Kernel File Systems...
[  OK  ] Stopped Remount Root and Kernel File Systems.
[  OK  ] Reached target Shutdown.
systemd-shutdown[1]: Sending SIGTERM to remaining processes...
systemd-journald[267]: Received SIGTERM from PID 1 (systemd-shutdow).
systemd-shutdown[1]: Sending SIGKILL to remaining processes...
systemd-shutdown[1]: Unmounting file systems.
systemd-shutdown[1]: Unmounting /sys/fs/fuse/connections.
systemd-shutdown[1]: Unmounting /sys/kernel/config.
systemd-shutdown[1]: Unmounting /sys/kernel/debug.
EXT4-fs (mmcblk2p2): re-mounted. Opts: (null)
EXT4-fs (mmcblk2p2): re-mounted. Opts: (null)
EXT4-fs (mmcblk2p2): re-mounted. Opts: (null)
systemd-shutdown[1]: All filesystems unmounted.
systemd-shutdown[1]: Deactivating swaps.
systemd-shutdown[1]: All swaps deactivated.
systemd-shutdown[1]: Detaching loop devices.
systemd-shutdown[1]: All loop devices detached.
systemd-shutdown[1]: Detaching DM devices.
systemd-shutdown[1]: All DM devices detached.
systemd-shutdown[1]: Halting system.
imx2-wdt 30280000.wdog: Device shutdown: Expect reboot!
reboot: System halted
```

Figure 4.3. Shutdown

Disconnecting the power supply while data is being written to storage may result in the file system or the data being damaged. Be sure to unmount storage first before disconnecting the power supply.

Chapter 5. Operational Checks

5.1. Before Performing the Operational Checks

There is a possibility that the image file written to the Flash memory in the factory default state is not the latest version. The latest image file can be downloaded from the Armadillo site. It is recommended to write the latest image file first.

For details on rewriting the image file, see Chapter 8, Rewriting Image Files.

5.2. Network

The following explains about network configuration and applications that use the network.

5.2.1. Supported Networks

Armadillo-IoT can connect to multiple types of networks. The available network types and their corresponding network device names used in Linux is shown below.

Table 5.1. Networks and Network Devices

Network	Network Device	Notes
Wired LAN	eth0	
Wireless LAN	wlan0	AEH-AR9462-LX
3G	ttyACM3	Gemalto PDS6

5.2.2. Network Configuration

On Armadillo-IoT Gateway G3, NetworkManager is used to configure network interfaces just like as with other standard Linux systems. By default NetworkManager automatically ups eth0 (Ethernet) and obtains network configuration with DHCP.

NetworkManager manages all network settings as connections. Connections describe "How to connect to the network" and "How to create the network" and are saved in `/etc/NetworkManager/system-connections/`. Also, while it is possible to save multiple connections for each device, only one connection can be activated at one time.

While NetworkManager also supports configuration using the traditional `/etc/network/interfaces` file, this document focuses on the use of **`nmcli`**.

5.2.2.1. About nmcli

nmcli is a command line tool for operating NetworkManager.

The format for nmcli is shown in Figure 5.1, “nmcli Command Format”. From this, it can be seen that nmcli commands are entered with a format based on objects (OBJECT) and executing commands (COMMAND) on those objects. It's also possible to see that each object has help associated with it.

```
nmcli [ OPTIONS ] OBJECT { COMMAND | help }
```

Figure 5.1. nmcli Command Format

See **`man nmcli`** for more information on each object.

In addition to nmcli, the user-friendly nmtui is also installed on Armadillo-IoT, but it is not covered in this document.

5.2.3. Basic Usage of nmcli

This section explains the basic usage of the **nmcli**.

5.2.3.1. List of Connections

To check the list of registered connections, run the following command.[1]

```
[armadillo ~]# nmcli connection
NAME                UUID                                  TYPE            DEVICE
Wired connection 1  64e2e184-ede4-4cc6-ab70-0713d7cb0f0b  802-3-ethernet  eth0
```

Figure 5.2. List of Connections

5.2.3.2. Enabling and Disabling Connections

To enable a connection, run the following command.

```
[armadillo ~]# nmcli connection up [ID]
```

Figure 5.3. Enabling a Connection

To disable a connection, run the following command.

```
[armadillo ~]# nmcli connection down [ID]
```

Figure 5.4. Disabling a Connection

5.2.3.3. Creating a Connection

To create a connection, run the following command.

```
[armadillo ~]# nmcli connection add con-name [ID] \
type [type] ifname [interface name]
```

Figure 5.5. Creating a Connection

Enter the connection name (arbitary) for *[ID]*, ethernet or wifi for *[type]*, and interface name (device) for *[interface name]*. The specific connection creation method is explained in the chapter for each device.

[1] More detailed information can be displayed with the **nmcli connection show *[ID]*** command.

A connection file is created with the name *[ID]* under `/etc/NetworkManager/system-connections/`. It is also possible to edit this with vi to modify the connection.

5.2.3.4. Deleting a Connection

To delete a connection, run the following command.

```
[armadillo ~]# nmcli connection delete [ID]
```

Figure 5.6. Deleting a Connection

The connection file under `/etc/NetworkManager/system-connections/>` is also deleted at the same time.

5.2.3.5. Modifying a Connection

The following introduces specific ways to modify connections.

When wireless LAN or 3G configuration is edited with the `nmcli connection modify` command, the passphrase information is reset. When making the edits, please also set a passphrase at the same time.

For details on how to set a passphrase for wireless LAN, see Section 5.2.5, “Wireless LAN”.

For setting a passphrase for 3G, refer to Section 5.2.6.4.3, “Notes on Altering 3G Connection Settings”.

Please consult a network administrator for help with network connections.

5.2.3.5.1. Using a Static Address

An example of setting the content of Table 5.2, “Static IP Address Configuration Example” is shown in Figure 5.7, “Static IP Address Configuration”.

Table 5.2. Static IP Address Configuration Example

Item	Configuration
IP Address	192.0.2.10
Mask Length	24
Default Gateway	192.0.2.1

```
[armadillo ~]# nmcli connection modify [ID] \
ipv4.method manual ipv4.addresses "192.0.2.10/24 192.0.2.1"
```

Figure 5.7. Static IP Address Configuration

5.2.3.5.2. Using DHCP

An example of a DHCP configuration is shown in Figure 5.8, “DHCP Configuration”.

```
[armadillo ~]# nmcli connection modify [ID]\
ipv4.method auto -ipv4.addresses "192.0.2.10/24 192.0.2.1"
```

Figure 5.8. DHCP Configuration

As like with **`-ipv4.addresses`**, a set property can be deleted by adding **`'-'`** to the beginning of the property name. Conversely, a property can be added by specifying **`'+'`**.

5.2.3.5.3. Specifying a DNS Server

An example of specifying a DNS server is shown in Figure 5.9, “Specifying a DNS Server”.

```
[armadillo ~]# nmcli connection modify [ID] ipv4.dns 192.0.2.10
```

Figure 5.9. Specifying a DNS Server

5.2.3.6. Applying Connection Modifications

When having modified a currently enabled connection, make sure to re-enable the connection.

```
[armadillo ~]# nmcli connection down [ID]
[armadillo ~]# nmcli connection up [ID]
```

Figure 5.10. Applying Connection Modifications

5.2.3.7. List of Devices

To check the device list (device name, type, status, valid connection), run the following command.[2]

```
[armadillo ~]# nmcli device
DEVICE    TYPE      STATE         CONNECTION
eth0      ethernet  connected     Wired connection 1
ttyACM3   gsm       disconnected  --
wlan0     wifi      disconnected  --
gre0      gre       unmanaged     --
gretap0   gretap    unmanaged     --
ip6gre0   ip6gre    unmanaged     --
ip6tnl0   ip6tnl    unmanaged     --
tunl0     ipip      unmanaged     --
lo        loopback  unmanaged     --
sit0      sit       unmanaged     --
ip6_vti0  vti6      unmanaged     --
```

Figure 5.11. List of Devices

5.2.3.8. Connecting Devices

To connect a device, run the following command.

```
[armadillo ~]# nmcli device connect [ifname]
```

Figure 5.12. Connecting Devices

To connect a device, a valid connection for the device is required. If the message "Error: neither a valid connection nor device given" is displayed, check if a valid connection exists by, for example, using the **`nmcli connection`** command.

5.2.3.9. Disconnecting Devices

To disconnect a device, run the following command.

```
[armadillo ~]# nmcli device disconnect [ifname]
```

Figure 5.13. Disconnecting Devices

5.2.4. Wired LAN

This section explains how to use wired LAN.

5.2.4.1. Creating a Wired LAN Interface (eth0) Connection

To create for a wired LAN interface connection, run the following command.

[2] **`nmcli device`** and **`nmcli device status`** are equivalent.

Also, more detailed information can be displayed with **`nmcli device show`**.

```
[armadillo ~]# nmcli connection add type ethernet ifname eth0
Connection 'ethernet-eth0' (ac491d33-9647-4096-8b91-5c7abcf5850d) successfully
added.
```

Figure 5.14. Creating a Wired LAN Interface (eth0) Connection

5.2.4.2. Changing the Network Settings of Wired LAN

For configuring the network, see Section 5.2.3.5, “Modifying a Connection”. After modifying the connection, make sure to apply the modifications by referring to Section 5.2.3.6, “Applying Connection Modifications”.

5.2.4.3. Confirming a Wired LAN Connection

Confirm that normal communication is possible on the wired LAN. If the configuration has changed, be sure to re-enable the interface.

Perform a PING with another network device on the same network. In the following example, it is assumed that the network device has an IP address of "192.0.2.20".

```
[armadillo ~]# ping 192.0.2.20
```

Figure 5.15. Ping Confirmation of Wired LAN

If any connections other than wired LAN are enabled, the wired LAN may not be used for network communication. In order to ensure that the wired LAN connection is used for the confirmation, disable the other connections in advance.

5.2.5. Wireless LAN

This section explains how to use the wireless LAN module installed on Armadillo-IoT.

As an example, connect to a WPA2-PSK(AES) access point. For details on how to connect to access points other than WPA2-PSK(AES), please refer to **`man nm-settings`**. In the explanation below, the ESSID of the access point is shown as *`[essid]`* and the passphrase as *`[passphrase]`*.

5.2.5.1. Connecting to a Wireless LAN Access Point

To connect to a wireless LAN access point, run the following command.

```
[armadillo ~]# nmcli device wifi connect [essid] password [passphrase]
```

Figure 5.16. Connecting to a Wireless LAN Access Point

5.2.5.2. Creating a Wireless LAN (wlan0) Connection

The Section 5.2.5.1, “Connecting to a Wireless LAN Access Point” method is easy but only supports DHCP. Therefore, in order to use a static IP with the wireless LAN or to make detailed network settings, a connection must be created.

To create a wireless LAN (wlan0 connection, run the following command.

```
[armadillo ~]# nmcli connection add type wifi ifname wlan0 ssid [essid] ❶
Connection 'wifi-wlan0' (d3cbb49d-b843-4dbf-94d5-7e7275449e8a) successfully added.
[armadillo ~]# nmcli connection modify wifi-wlan0 \ ❷
802-11-wireless-security.key-mgmt wpa-psk \ ❸
802-11-wireless-security.psk [passphrase]
```

Figure 5.17. Creating a Wireless LAN (wlan0) Connection

❶ Specify wifi for the type to create a wireless LAN connection.

❷ Set the encryption key management method to wpa-psk.

❸ Set a passphrase.

Depending on the access point being connected to, the following message may be displayed and you may not be able to connect to the access point.

```
wlan0: authenticate with 00:3a:9d:42:cc:92
wlan0: send auth to 00:3a:9d:42:cc:92 (try 1/3)
wlan0: authenticated
wlan0: associate with 00:3a:9d:42:cc:92 (try 1/3)
wlan0: RX AssocResp from 00:3a:9d:42:cc:92 (capab=0x431
status=0 aid=1)
wlan0: associated
cfg80211: Calling CRDA to update world regulatory domain
cfg80211: World regulatory domain updated:
cfg80211:  DFS Master region: unset
cfg80211:   (start_freq - end_freq @ bandwidth),
(max_antenna_gain, max_eirp)
cfg80211:   (2402000 KHz - 2472000 KHz @ 40000 KHz), (N/A,
2000 mBm)
cfg80211:   (2457000 KHz - 2482000 KHz @ 40000 KHz), (N/A,
2000 mBm)
cfg80211:   (2474000 KHz - 2494000 KHz @ 20000 KHz), (N/A,
2000 mBm)
cfg80211:   (5170000 KHz - 5250000 KHz @ 80000 KHz), (N/A,
2000 mBm)
cfg80211:   (5250000 KHz - 5330000 KHz @ 80000 KHz), (N/A,
2000 mBm)
cfg80211:   (5490000 KHz - 5730000 KHz @ 160000 KHz), (N/A,
2000 mBm)
cfg80211:   (5735000 KHz - 5835000 KHz @ 80000 KHz), (N/A,
2000 mBm)
cfg80211:   (57240000 KHz - 63720000 KHz @ 2160000 KHz), (N/
A, 0 mBm)
cfg80211: Calling CRDA for country: JP
cfg80211: Regulatory domain changed to country: JP
cfg80211:  DFS Master region: JP
cfg80211:   (start_freq - end_freq @ bandwidth),
(max_antenna_gain, max_eirp)
cfg80211:   (2402000 KHz - 2482000 KHz @ 40000 KHz), (N/A,
2000 mBm)
cfg80211:   (2474000 KHz - 2494000 KHz @ 20000 KHz), (N/A,
```

```
2000 mBm)
cfg80211:   (4910000 KHz - 4990000 KHz @ 40000 KHz), (N/A,
2300 mBm)
cfg80211:   (5030000 KHz - 5090000 KHz @ 40000 KHz), (N/A,
2300 mBm)
cfg80211:   (5170000 KHz - 5250000 KHz @ 80000 KHz), (N/A,
2000 mBm)
cfg80211:   (5250000 KHz - 5330000 KHz @ 80000 KHz), (N/A,
2000 mBm)
cfg80211:   (5490000 KHz - 5710000 KHz @ 160000 KHz), (N/A,
2300 mBm)
```

5.2.5.3. Changing Wireless Network Settings

For configuring the network, refer to Section 5.2.3.5, “Modifying a Connection”. When modifying the connection, also set a passphrase by referring to Figure 5.17, “Creating a Wireless LAN (wlan0) Connection”. After making the modifications, make sure to apply them by referring to Section 5.2.3.6, “Applying Connection Modifications”.

5.2.5.4. Confirming Wireless LAN Connection

Confirm that it is possible to communicate properly over wireless LAN.

Perform a PING with another network device on the same network. In the following example, it is assumed that the network device has an IP address of "192.0.2.20".

```
[armadillo ~]# ping 192.0.2.20
```

Figure 5.18. Ping Confirmation of Wireless LAN

If any connections other than wireless LAN are enabled, the wireless LAN may not be used for network communication. In order to ensure that the wireless LAN connection is used for the confirmation, disable the other connections in advance.

5.2.6. 3G

The following explains how to use the "Gemalto 3G Communication Module PDS6" 3G module installed in Armadillo-IoT.

The "Gemalto 3G Communication Module PDS6" 3G module has completed the DoCoMo interoperability test.

5.2.6.1. Before Configuring 3G Data Communication

In order to use 3G data communication a contract with a telecommunications carrier is needed. Prepare the MicroSIM (UIM card) and APN information provided by the carrier at the time of contract.

Make sure that the Armadillo-IoT is powered off before installing the MicroSIM (UIM card).

This product has a MicroSIM slot.

Using a NanoSIM card with a SIM adapter or using a standard size SIM card cut to MicroSIM size may cause damaged to the MicroSIM slot. If this product is broken by using cards in this way it will not be covered by warranty even within the warranty period.

Insert the microSIM (UIM card) with its notch facing the direction opposite to that of insertion and with the marked side facing up.

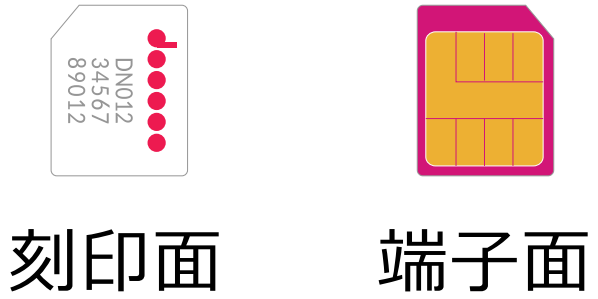


Figure 5.19. microSIM

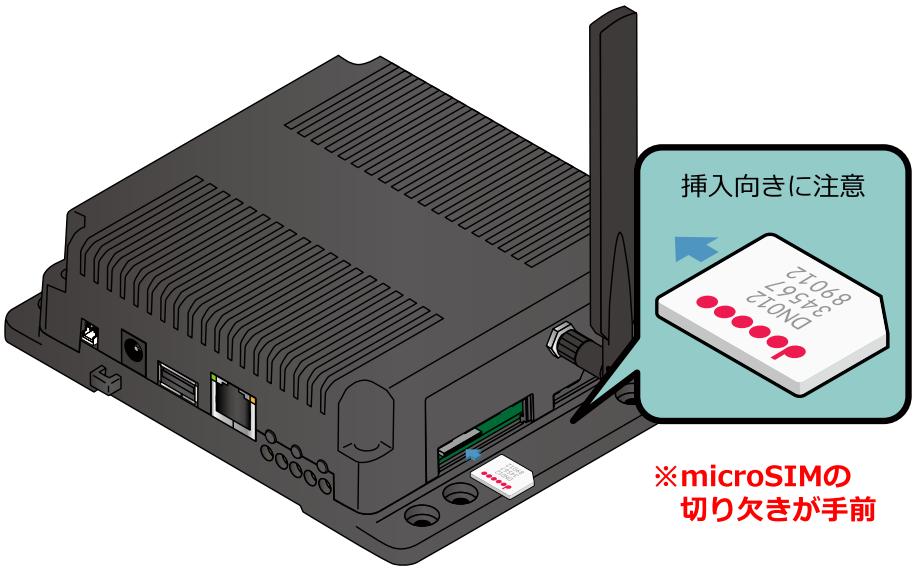


Figure 5.20. Installing MicroSIM

The following information is necessary to configure the APN.

- APN

- Username
- Password
- Authentication method (PAP or CHAP)
- PDP Type (only IP is supported)

5.2.6.2. Creating a 3G Connection

An example of setting the content of Table 5.3, “APN Information Configuration Example” is shown in Figure 5.21, “Creating a 3G Connection”.

Table 5.3. APN Information Configuration Example

Item	Configuration
APN	*[apn]*
Username	*[user]*
Password	*[password]*

```
[armadillo ~]# nmcli connection add type gsm \
ifname ttyACM3 apn [apn] user [user] password [password]
Connection 'gsm-ttyACM3' (a9e51a2d-bbee-443f-80ba-07b65c3097e8) successfully
added.
```

Figure 5.21. Creating a 3G Connection

After creating the connection, the data connection will be done automatically at startup.

5.2.6.3. Establishing a 3G Data Connection

To establish a data connection without rebooting immediately after creating the 3G connection or changing its settings, execute the command shown in Figure 5.22, “3G Data Connection”.

```
[armadillo ~]# nmcli connection up gsm-ttyACM3
Connection successfully activated (D-Bus active path: /org/freedesktop/
NetworkManager/ActiveConnection/5)
```

Figure 5.22. 3G Data Connection

5.2.6.4. Confirming a 3G Connection

Confirm that it is possible to communicate properly over 3G.

Perform a PING with Atmark Techno's web server. If an internet connection is not available because of the use of a VPN connection and so on, please use a network device on the local network instead.

```
[armadillo ~]# ping www.atmark-techno.com
```

Figure 5.23. Ping Confirmation of 3G

If any connections other than 3G are enabled, 3G may not be used for network communication. In order to ensure that the 3G connection is used for the confirmation, disable the other connections in advance.

Even if a connection is created following Figure 5.21, “Creating a 3G Connection”, there are times when it may not be possible to enable the connection. If the connection can not be enabled even if the configured APN information is correct, it may be possible to activate the connection after first resetting PDS6 as shown below.

Procedure 5.1. Resetting PDS6

1. Quit NetworkManager and ModemManager.

```
[armadillo ~]#service NetworkManager stop
[armadillo ~]#service ModemManager stop
```

2. To reset PDS6, connect to /dev/ttyACM0 and enter an AT command.

```
[armadillo ~]#cu -l /dev/ttyACM0 -s 115200
Connected.
AT^SMSO ❶
^SMSO: MS OFF

OK

^SHUTDOWN
usb 2-1.2: USB disconnect, device number 5
cu: Got hangup signal

Disconnected.
[armadillo ~]#usb 2-1.2: new high-speed USB device
number 6 using ci_hd
rc
cdc_acm 2-1.2:1.0: ttyACM0: USB ACM device
cdc_acm 2-1.2:1.2: ttyACM1: USB ACM device
cdc_acm 2-1.2:1.4: ttyACM2: USB ACM device
cdc_acm 2-1.2:1.6: ttyACM3: USB ACM device
cdc_acm 2-1.2:1.11: ttyACM4: USB ACM device
cdc_acm 2-1.2:1.13: ttyACM5: USB ACM device
```

❶ Run the AT command.

3. Start NetworkManager and ModemManager.

```
[armadillo ~]#service ModemManager start
[armadillo ~]#service NetworkManager start
```

5.2.6.4.1. Ending a 3G Data Connection

Terminate the 3G reconnection service before terminating the data communication with the nmcli command. If data communication is terminated without terminating the 3G reconnection service, the data connection will be restarted by that service.

Stop the 3G reconnection service, and then end the data communication.

```
[armadillo ~]# systemctl stop connection-recover.service
[armadillo ~]# nmcli connection down gsm-ttyACM3
```

Figure 5.24. Ending Data Communication

5.2.6.4.2. Restarting a 3G Data Connection

Start the data communication.

```
[armadillo ~]# nmcli connection up gsm-ttyACM3
[armadillo ~]# systemctl start connection-recover.service
```

Figure 5.25. Starting Data Communication

5.2.6.4.3. Notes on Altering 3G Connection Settings

When wireless LAN or 3G configuration is edited with the `nmcli connection modify` command, the password information is reset. Execute the following command to set the password again each time.

```
[armadillo ~]# nmcli connection modify gsm-ttyACM3 gsm.password [password]
```

Figure 5.26. Setting a 3G Passphrase with the nmcli connection modify Command

5.2.6.5. 3G Reconnection Service

The 3G reconnection service periodically monitors the status of the 3G data connection and reconnects it when disconnection is detected. It is supported in atmark-x1-base v1.5.0-1 or later (root file system: debian-jessie-armhf_aiotg3l_20170419.tar.gz or later).

5.2.6.5.1. Service Specifications

When a MicroSIM is inserted and a valid NetworkManager 3G connection has been configured, it monitors the state of the connection once every 120 seconds.

If the connection is disabled it is judged to be in a disconnected state and the connection is enabled. If the connection is enabled, a PING is executed to a specific destination. If the PING results in an error, it is judged to be in the disconnected state, and reconnection is performed by disabling and enabling the connection.

If reconnection by disabling and enabling the connection fails more than once, it is judged that the 3G module is not operating normally and the power supply of the 3G module is cycled off and on on order to reconnect.

5.2.6.5.2. Factory Settings

It is enabled in the default factory state, and the service starts automatically at system startup. The ping destination is "8.8.8.8" by default. Please change the configuration file (/etc/connection-recover/gsm-ttyACM3_connection-recover.conf) appropriately for your environment.

5.2.6.5.3. Stop and Start

To stop the 3G reconnection service, run the following command.

```
[armadillo ~]# systemctl stop connection-recover.service
```

Figure 5.27. Stopping the 3G Reconnection Service

To start the 3G reconnection script, run the following command.

```
[armadillo ~]# systemctl start connection-recover.service
```

Figure 5.28. Starting the 3G Reconnection Service

5.2.6.6. ModemManager

ModemMamager and mmcli are explained here.

Apart from NetworkManager which manages the network, ModemManager which manages the modem is installed on Armadillo-IoT. ModemManager operates mobile broadband devices (such as 3G modules) and manages their connection status.

By using the **`mmcli`** command line tool of ModemManager, it is possible to acquire the 3G communication signal strength and SIM card information (telephone number, IMEI etc). Please refer to **`man mmcli`** for usage details of mmcli.

5.2.6.6.1. Obtaining a List of Recognized Modems

To obtain a list of recognized modems, run the following command.

```
[armadillo ~]# mmcli -L

Found 1 modems:
    /org/freedesktop/ModemManager1/Modem/0 [Cinterion] PDS6
```

Figure 5.29. Obtaining a List of Recognized Modems

5.2.6.6.2. Obtaining Modem Information

To obtain the status of the modem, run the following command.

```
[armadillo ~]# mmcli -m 0

/org/freedesktop/ModemManager1/Modem/0 (device id
'8e393ace639c5e82057ef21094a9b4639ca0a060')
  -------------------------
  Hardware |   manufacturer: 'Cinterion'
           |          model: 'PDS6-J'
           |       revision: 'REVISION 03.001'
           |      supported: 'gsm-umts'
           |        current: 'gsm-umts'
           |   equipment id: '353088070004351'
  -------------------------
  System   |         device: '/sys/devices/soc/30800000.aips-bus/30b30000.usb/
ci_hdrc.1/usb2/2-1/2-1.2'
           |        drivers: 'cdc_acm'
           |         plugin: 'Cinterion PDS'
           |   primary port: 'ttyACM3'
           |          ports: 'ttyACM3 (at), ttyACM5 (at), ttyACM0 (at)'

(省略)
```

Figure 5.30. Obtaining Modem Information

In order to obtain modem information a MicroSIM must be inserted. Please make sure a MicroSIM is correctly inserted.

5.2.6.6.3. Obtaining MicroSIM Information

To obtain MicroSIM information, run the following command.

```
[armadillo ~]# mmcli -m 0
(abbreviated)
-------------------------
SIM     |          path: '/org/freedesktop/ModemManager1/SIM/[number]' # [number]
is used in the next command
(abbreviated)
[armadillo ~]# mmcli -i [number]
SIM '/org/freedesktop/ModemManager1/SIM/0'
  -------------------------
  Properties |          imsi : 'XXXXXXXXXXXXXXX'
             |            id : 'XXXXXXXXXXXXXXXXXXX'
             |   operator id : 'XXXXX'
             | operator name : 'XXXXXXXXXX'
```

Figure 5.31. Obtaining MicroSIM Information

5.2.6.6.4. Obtaining Communication Line Information

To obtain communication line information, run the following command.

```
[armadillo ~]# mmcli -m 0
(abbreviated)
-------------------------
Bearers  |          paths: '/org/freedesktop/ModemManager1/Bearer/[number]' #
[number] is used in the next command
[armadillo ~]# mmcli -b [number]
Bearer '/org/freedesktop/ModemManager1/Bearer/0'
  -------------------------
  Status              |   connected: 'yes'
                      |   suspended: 'no'
                      |   interface: 'ttyACM3'
                      |  IP timeout: '20'
  -------------------------
  Properties          |         apn: 'XXXXXXXXXX'
                      |     roaming: 'allowed'
                      |     IP type: 'none'
                      |        user: 'XXXX'
                      |    password: 'XXXX'
                      |      number: '*99#'
                      | Rm protocol: 'unknown'
  -------------------------
  IPv4 configuration |   method: 'ppp'
                      |  address: 'unknown'
                      |   prefix: '0'
                      |  gateway: 'unknown'
                      |      DNS: none
  -------------------------
  IPv6 configuration |   method: 'unknown'
```

Figure 5.32. Obtaining Communication Line Information

5.2.7. Configuration Example with NetworkManager

The following introduces a configuration example with NetworkManager based on Figure 5.33, “Network Structure Diagram”.

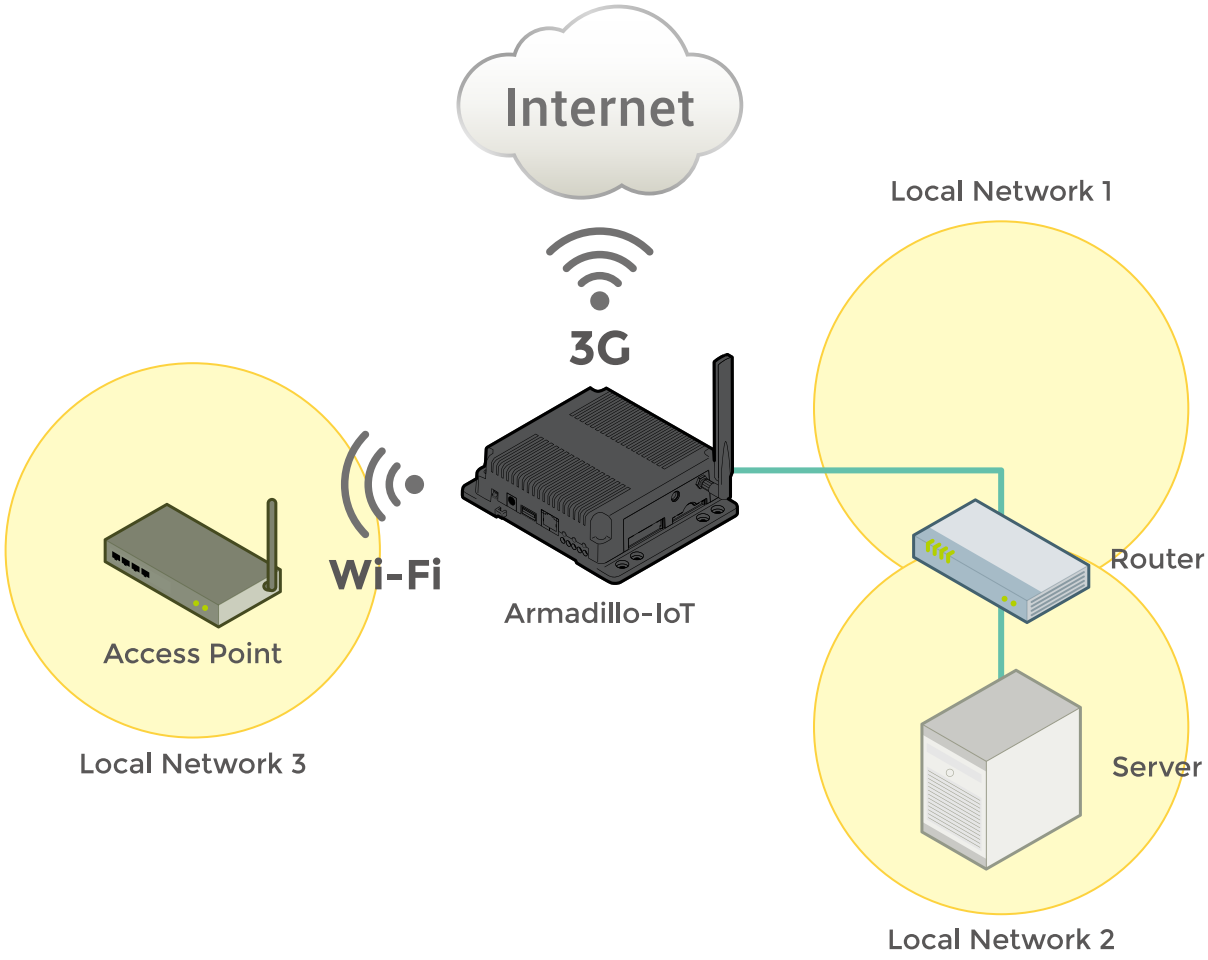


Figure 5.33. Network Structure Diagram

Table 5.4. Network Address Information

Node Name	Network Device	IP Address	Network Address
Armadillo	eth0	192.168.0.2	192.168.0.0/24
	wlan0	172.16.xxx.xxx[a]	172.16.0.0/16[a]
	ttyACM3[b]	xxx.xxx.xxx.xxx[a]	xxx.xxx.xxx.xxx[a]
Router	eth0	192.168.0.1	192.168.0.0/24
	eth1	192.168.10.1	192.168.10.0/24
Server	eth0	192.168.10.2	192.168.10.0/24
Access Point	eth0	172.16.0.1	172.16.0.0/16

[a] Automatically acquired with DHCP

[b] The network device of the 3G Module

5.2.7.1. Network Configuration Procedure

The network setup procedure when configuring the network shown in Figure 5.33, “Network Structure Diagram” is as follows. Note that the created connection is saved in `/etc/NetworkManager/system-connections/` and is activated even after restarting Armadillo.

Procedure 5.2. Network Configuration Procedure

1. Make sure that the state of eth0, ttyACM3 and wlan0 is disconnected.

```
[armadillo ~]#nmcli device
DEVICE    TYPE      STATE         CONNECTION
eth0      ethernet  disconnected  --
ttyACM3   gsm       disconnected  --
wlan0     wifi      disconnected  --
gre0      gre       unmanaged     --
gretap0   gretap    unmanaged     --
ip6gre0   ip6gre    unmanaged     --
ip6tnl0   ip6tnl    unmanaged     --
tunl0     ipip      unmanaged     --
lo        loopback  unmanaged     --
sit0      sit       unmanaged     --
ip6_vti0  vti6      unmanaged     --
```

   If any are in a state other than disconnected, change each to the disconnected state by following Table 5.5, “Making the Device State disconnected”.

Table 5.5. Making the Device State disconnected

Device State	Procedure
unmanaged	Check that the device settings are not included in `/etc/network/interfaces`. Remove them if they are.
unavailable	Check that the LAN cable is not disconnected. Connect the cable if it is.
connecting	A connection using the device is being enabled. Disable it by referring to Figure 5.4, “Disabling a Connection”.
connected	A connection using the device is enabled. Disable it by referring to Figure 5.4, “Disabling a Connection”.

2. Configure the wireless LAN (wlan0).

```
[armadillo ~]#nmcli connection add type wifi ifname wlan0 ssid [essid] ❶
[armadillo ~]#nmcli connection modify wifi-wlan0 ipv4.never-default yes
❷
[armadillo ~]#nmcli connection modify wifi-wlan0 \
802-11-wireless-security.key-mgmt wpa-psk \
802-11-wireless-security.psk [passphrase] ❸
[armadillo ~]#nmcli connection down wifi-wlan0 ❹
[armadillo ~]#nmcli connection up wifi-wlan0 ❺
```

   ❶ Create the wireless LAN (wlan0) connection.

   ❷ Disable the default gateway of the connection of wireless LAN (wlan0) connection.

   ❸ Set the encryption key management method to wpa-psk and set a passphrase.

   ❹ In order to apply the changes, first disable wireless LAN (wlan0) connection.

   ❺ Enable the wireless LAN (wlan0) connection.

3. Configure the wired LAN interface (eth0).

```
[armadillo ~]#nmcli connection add type ethernet ifname eth0  ❶
[armadillo ~]#nmcli connection modify ethernet-eth0 ipv4.method manual \
ipv4.addresses "192.168.0.2/24"  ❷
[armadillo ~]#nmcli connection modify ethernet-eth0 \
ipv4.routes "192.168.10.0/24 192.168.0.1"  ❸

[armadillo ~]#nmcli connection modify ethernet-eth0 ipv4.never-default
yes  ❹
[armadillo ~]#nmcli connection down ethernet-eth0  ❺
[armadillo ~]#nmcli connection up ethernet-eth0  ❻
```

❶ Create the wired LAN interface (eth0) connection.

❷ Specify a static IP address for the wired LAN interface (eth0) connection.

❸ Add the routing information to the wired LAN interface (eth0) connection.

❹ Disable the default gateway of the wired LAN interface (eth0) connection.

❺ In order to apply the changes, first disable the wired LAN (eth0) connection.

❻ Enable the wired LAN (eth0) connection.

4. Configure 3G (ttyACM3)

```
[armadillo ~]#nmcli connection add type gsm ifname ttyACM3 apn [apn] user
[user] password [password]  ❶
```

❶ Create the 3G (ttyACM3) connection.

5. Make sure that the state of eth0, ttyACM3, ppp0 and wlan0 is connected.

```
[armadillo ~]#nmcli device
DEVICE    TYPE      STATE      CONNECTION
eth0      ethernet  connected  ethernet-eth0
ttyACM3   gsm       connected  gsm-ttyACM3
ppp0      unknown   connected  ppp0
wlan0     wifi      connected  wifi-wlan0
gre0      gre       unmanaged  --
gretap0   gretap    unmanaged  --
ip6gre0   ip6gre    unmanaged  --
ip6tnl0   ip6tnl    unmanaged  --
tunl0     ipip      unmanaged  --
lo        loopback  unmanaged  --
sit0      sit       unmanaged  --
ip6_vti0  vti6      unmanaged  --
```

6. Check the routing table.

```
[armadillo ~]#route
Kernel IP routing table
```

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         xxx.xxx.xxx.xxx 0.0.0.0          UG    1024   0        0 ppp0
link-local      *               255.255.0.0      U     1000   0        0 wlan0
172.16.0.0      *               255.255.0.0      U     0      0        0 wlan0
192.168.0.0     *                255.255.255.0   U     0      0        0 eth0
192.168.10.0    192.168.0.1      255.255.255.0   UG    1      0        0
eth0
```

5.2.8. Firewall

A simple firewall is active on Armadillo. To display its configuration content execute the command shown in Figure 5.34, “iptables”.

```
[armadillo ~]# iptables --list
```

Figure 5.34. iptables

5.2.9. Network Applications

The following explains about the network applications that can be used in the factory default image.

It is assumed that the network configuration of ATDE and Armadillo is the default state. If the network configuration has been modified please adjust for that as needed.

5.2.9.1. HTTP Server

An HTTP server is active on Armadillo. When accessing the URL of Armadillo (http://[IP address of Armadillo]/) from a web browser on a PC such as ATDE, the top page (index.html) of lighttpd is displayed.

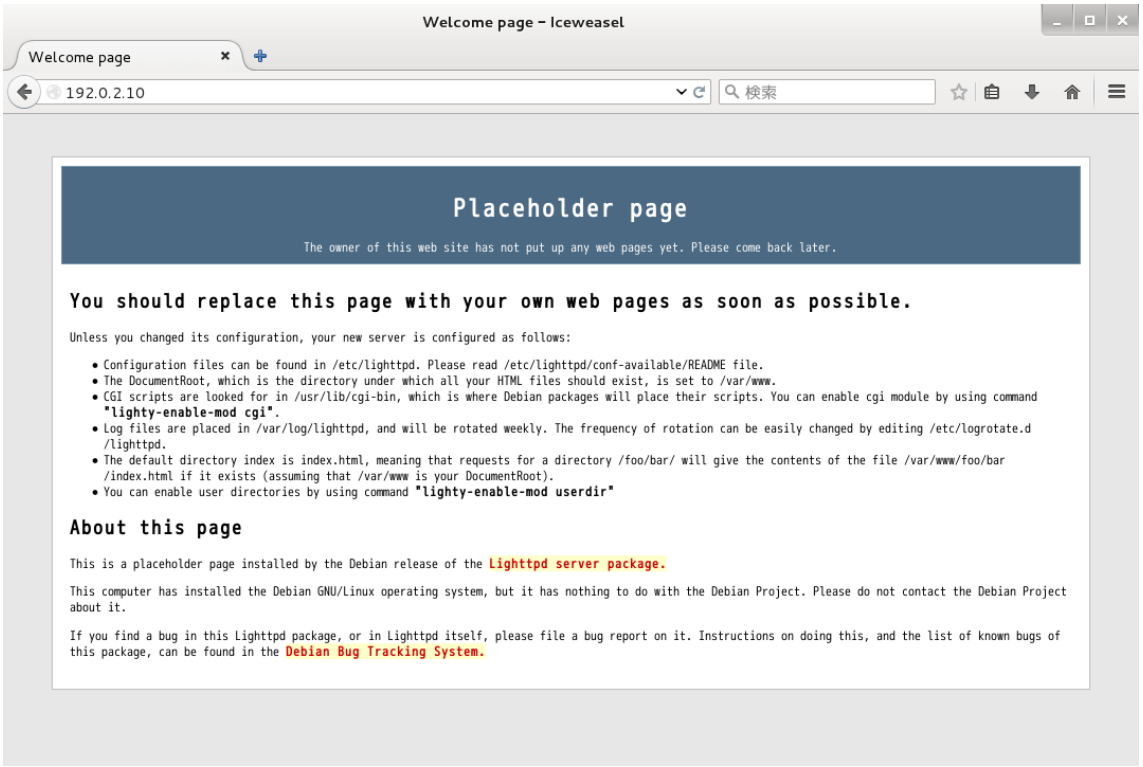


Figure 5.35. Armadillo Top Page

5.3. Storage

The following devices are available for use as storage on Armadillo-IoT.

Table 5.6. Storage Devices

Device Type	Disk Device	First Partition	Interfaces
SD/SDHC/SDXC Cards	`/dev/mmcblk*`[a]	`/dev/mmcblk*p1`	`SD Interface (CON4)`
USB Flash Memory	`/dev/sd*`[b]	`/dev/sd*1`	`USB Host Interface (CON7)`

[a] When microSD/microSDHC/microSDXC cards are connected, they are labeled as mmcblk0, then mmcblk1 and so on in the order that they are detected.

[b] When connecting multiple USB memory devices using a USB hub, they are labeled as sda, then sdb and so on in the order that they are detected.

5.3.1. Using Storage

The following explains how to use storage by using an SDHC card as an example.SD/SDHC/SDXC cards are referred to as SD cards wherever the operations can be applied to any of them.

When using an SDXC/microSDXC card, it is necessary to format it in advance by referring to Section 5.3.2, “Changing and Formatting Storage Partitions”. This is because the Linux kernel cannot handle the exFAT file system. Normally, SDXC/microSDXC cards which have just been purchased are formatted with the exFAT file system.

On Linux all accessible files and directories are brought together in one tree structure. Adding the file system of a storage device to this tree structure is called mounting. The mount command used to perform this mounting.

The typical format of the mount command is as follows.

```
mount [-t fstype] device dir
```

Figure 5.36. mount Command Format

The file system type is specified for `fstype` following the `-t` option[3]. For the FAT32 file system use `vfat`[4], and for the EXT4 file system use `ext4`.

The device filename of the storage device is specified for `device`. For partition one of the SD card this will be `/dev/mmcblk0p1` and for partition two it will be `/dev/mmcblk0p2`.

The directory where the file system on the storage device is to be mounted is specified for `dir`.

With the SDHC card inserted in the SD slot, execute the command shown in Figure 5.37, “Mounting Storage” to mount the SDHC card file system on the /mnt directory. Files in the SD card become visible under the /mnt directory.

```
[armadillo ~]# mount -t vfat /dev/mmcblk0p1 /mnt
```

Figure 5.37. Mounting Storage

In order to safely remove storage it must be unmounted. The umount command is used to perform the unmounting. The directory where the device to unmount is mounted is specified as its option.

```
[armadillo ~]# umount /mnt
```

Figure 5.38. Unmounting Storage

5.3.2. Changing and Formatting Storage Partitions

Normally, SDHC cards and USB memory just purchased have one partition and are formatted with the FAT32 file system.

If you want to change the partition configuration, use the fdisk command. An example of using the fdisk command to divide the partition of an SD card configured with just that one partition into two partitions is shown in Figure 5.39, “Altering Partitions with the fdisk Command”. After deleting the existing partition, two new primary partitions are created. 100Mbyte is allocated to the first partition and the remaining capacity to the second partition. The first partition becomes /dev/mmcblk0p1 and the second /dev/mmcblk0p2. For details on how to use the fdisk command, please refer to the man page etc.

```
[armadillo ~]# fdisk /dev/mmcblk0

The number of cylinders for this disk is set to 62528.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)
```

[3]Specifying the file system type is optional. If omitted, the mount command guesses the type. This guess is not necessarily appropriate, so if the file system type is known in advance, please explicitly specify it.
[4]Normally, SDHC cards just purchased are formatted with the FAT32 file system.

```
Command (m for help): d
Selected partition 1

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-62528, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-62528, default 62528): +100M

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (3054-62528, default 3054):
Using default value 3054
Last cylinder or +size or +sizeM or +sizeK (3054-62528, default 62528):
Using default value 62528

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
 mmcblk0: p1 p2
 mmcblk0: p1 p2
Syncing disks.
```

Figure 5.39. Altering Partitions with the fdisk Command

To format a storage device with the FAT32 file system, use the mkfs.vfat command. Likewise, for EXT3 use the mkfs.ext3 command and for EXT4 use the mkfs.ext 4 command. The command example for formatting partition one of the SD card with the EXT4 file system is shown below.

```
[armadillo ~]# mkfs.ext4 /dev/mmcblk0p1
```

Figure 5.40. Creating a EXT4 File System

5.4. LED

As the Armadillo-IoT's LEDs are connected to GPIO they can be controlled in software.

As the device driver is implemented as an LED class, the LEDs can be controlled with files under the LED class directory. The LED class directories and their corresponding LEDs are shown below.

Table 5.7. LEDs and LED Class Directories

LED Class Directory	Interfaces	Default Trigger
/sys/class/leds/led1/	User LED1	default-on
/sys/class/leds/led2/	User LED2	default-on
/sys/class/leds/led3/	User LED3	none

LED Class Directory	Interfaces	Default Trigger
/sys/class/leds/led4/	User LED4	none

The locations of the user LEDs seen from the exterior of Armadillo-IoT are shown below.

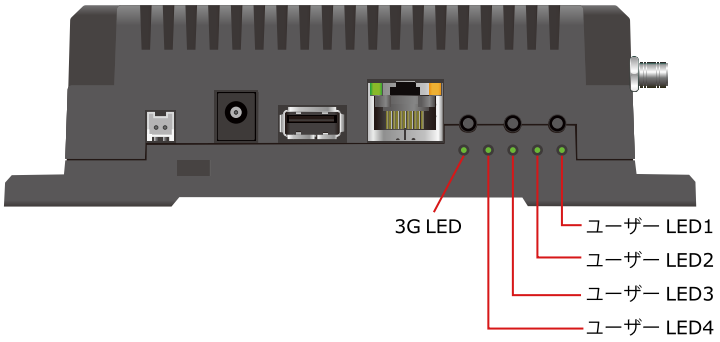


Figure 5.41. User LEDs Location

In the following explanation, the LED class directory representing an arbitrary LED is written as `/sys/class/leds/[LED]`.

5.4.1. Activating and Deactivating LEDs

An LED can be turned on and off by writing a value to the brightness file under the LED class directory. Valid values to write to brightness are between 0 and 255.

Writing a value other than 0 to brightness turns the LED on.

```
[armadillo ~]# echo 1 > /sys/class/leds/[LED]/brightness
```

Figure 5.42. Turning on an LED

Since the LEDs on Armadillo-IoT do not have brightness control functionality only two states, 0 (off) and 1-255 (on), can be specified.

Writing 0 to brightness turns the LED off.

```
[armadillo ~]# echo 0 > /sys/class/leds/[LED]/brightness
```

Figure 5.43. Turning off an LED

The state of an LED can be obtained by reading brightness.

```
[armadillo ~]# cat /sys/class/leds/[LED]/brightness
0
```

Figure 5.44. Displaying the State of an LED

5.4.2. Using Triggers

A trigger to turn an LED on and off can be set by writing a value to the trigger file under the LED class directory. Valid values to write to the trigger file are shown below.

Table 5.8. trigger Types

Configuration	Description
none	No trigger is set.
mmc0	Act as the SD Interface (CON4) access lamp.
mmc2	Act as the eMMC access lamp.
timer	Blink on and off at certain intervals. After setting this trigger, delay_on and delay_off files appear under the LED class directory which can be used to set the on and off times in millisecond increments.
heartbeat	Blink on and off like a heart beat.
default-on	Used mainly from the Linux kernel. The LED will turn on.

When the following command is run, the LED will repeatedly turn on for two seconds and off for one second.

```
[armadillo ~]# echo timer > /sys/class/leds/[LED]/trigger
[armadillo ~]# echo 2000 > /sys/class/leds/[LED]/delay_on
[armadillo ~]# echo 1000 > /sys/class/leds/[LED]/delay_off
```

Figure 5.45. Specifying timer for the LED Trigger

The current trigger of the LED can be obtained by reading the trigger file. The value enclosed in [] is the current trigger.

```
[armadillo ~]# cat /sys/class/leds/[LED]/trigger
[none] rc-feedback nand-disk mmc0 mmc2 timer oneshot heartbeat backlight gpio de
fault-on rfkill0 phy0rx phy0tx phy0assoc phy0radio phy0tpt rfkill1
```

Figure 5.46. Displaying the LED Trigger

5.5. RTC

Armadillo-IoT uses the RTC functionality of the Board Management IC.

To retain the time even if the power is turned off, an external battery (for example: CR1220) can be connected to the RTC backup interface (CON13).

5.5.1. Setting the Time on the RTC

There are two types of time on Linux: the system clock managed by the Linux kernel and the hardware clock managed by the RTC. In order to set the time in the RTC, first set the system clock. Following that, make the hardware clock match the system clock.

The system clock is set using the date command. For the date command argument, specify the time to be set in the format [MMDDhhmmCCYY.ss]. The meaning of each field of the time format is as follows.

Table 5.9. Time Format Fields

Field	Meaning
MM	Months
DD	Days (in month)
hh	Hours
mm	Minutes
CC	First two digits of the year (optional)
YY	Last two digits of the year (optional)
ss	Seconds (optional)

An example of setting it to 12:34:56 on June 2, 2015 is shown below.

```
[armadillo ~]#date ❶
Sat Jan  1 09:00:00 JST 2000
[armadillo ~]#date 060212342015.56 ❷
Tue Jun  2 12:34:56 JST 2015
[armadillo ~]#date ❸
Tue Jun  2 12:34:57 JST 2015
```

❶ Display the current system clock.

❷ Set the system clock.

❸ Confirm that the system clock has been set correctly.

Figure 5.47. Setting System Clock

If there is a time server on the network to which Armadillo-IoT is connected, the system clock can be set using the NTP (Network Time Protocol) client.

```
[armadillo ~]#ntpdate [NTP SERVER]
2 Jun 12:34:56 ntpdate[742]: adjust time server x.x.x.x
offset 0.004883
 sec
[armadillo ~]#date
Tue Jun  2 12:34:57 JST 2015
```

After setting the system clock, use the hwclock command to set the hardware clock.

```
[armadillo ~]#hwclock ❶
Sat Jan  1 00:00:00 2000  0.000000 seconds
[armadillo ~]#hwclock --utc --systohc ❷
[armadillo ~]#hwclock --utc ❸
Tue Jun  2 12:35:08 2015  -0.897934 seconds
```

❶ Display the current hardware clock.

❷ Set the hardware clock in Coordinated Universal Time (UTC).

❸ Confirm that the hardware clock has been set correctly in UTC.

Figure 5.48. Setting the Hardware Clock

5.6. User Switches

The device driver of the Armadillo-IoT user switches is implemented as an input device. Push and release events of the user switches can be obtained from the device file of the input device.

The input device file of the user switches and the event code corresponding to each switch are shown below.

Table 5.10. Input Device Files and Event Codes

User Switches	Input Device File	Event Code
SW1	/dev/input/event1	2 (KEY_1)
SW2		3 (KEY_2)
SW3		4 (KEY_3)

Input devices are indexed in the order in which they are detected. If an input device has been added by connecting a USB device etc., the index of the device file may be different.

5.6.1. Confirming Events

Here the `evtest` command is used to check the push and release events of the user switch. To stop evtest, enter Ctrl+c.

```
[armadillo ~]# evtest /dev/input/event1
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100
Input device name: "gpio-keys"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 2 (KEY_1)
    Event code 3 (KEY_2)
    Event code 4 (KEY_3)
Properties:
Testing ... (interrupt to exit)
Event: time 1458887649.091957, type 1 (EV_KEY), code 3 (KEY_2), value 1 ❶
Event: time 1458887649.091957, -------------- EV_SYN ------------
Event: time 1458887650.311954, type 1 (EV_KEY), code 3 (KEY_2), value 0 ❷
Event: time 1458887650.311954, -------------- EV_SYN ------------
                 :
[armadillo ~]#
```

❶ Displayed when a SW2 button push event is detected.

❷ Displayed when a SW2 button release event is detected.

Figure 5.49. User Switch: Event Confirmation

5.7. Temperature Sensor

The Armadillo-IoT temperature sensor uses the i.MX 7Dual's TEMPMON (Temperature Monitor).

5.7.1. Obtaining the Temperature

By reading the value from the `/sys/class/thermal/thermal_zone1/temp` file, it is possible to obtain the measured temperature of the i.MX 7 Dual.

```
[armadillo ~]# cat /sys/class/thermal/thermal_zone1/temp
50000 ❶
```

❶ Temperature is displayed in units of milli °C. In this example it shows 50.000 °C.

Figure 5.50. Obtaining the measurement temperature of the i.MX 7 Dual

5.8. AD Converter

Armadillo-IoT G3 can acquire the power supply voltage and the voltage of the external battery connected to the RTC backup interface (CON13) with the AD converter function of the BMIC (Board Management IC).

5.8.1. Obtaining the Voltage

The power supply voltage is divided and input to the AD converter. In order to acquire the power supply voltage, it is necessary to first acquire the input voltage to the AD converter. Since the voltage of the external battery is not divided, the input voltage of the AD converter will be the voltage of the external battery as-is.

The AD converter is implemented as an IIO (Industrial I/O) device. The input voltage can be calculated from the file under `/sys/bus/iio/devices/iio:device0/` directory.

The IIO device will be named iio:deviceN (N is a sequential number from '0') in the order in which the devices are detected. IIO devices can be identified from the IIO device name. The IIO device name of the BMIC's AD converter is "3-0012".

```
[armadillo ~]# cat /sys/bus/iio/devices/iio:device0/name
3-0012
```

The input voltage to the AD converter can be calculated from the AD conversion value and the minimum input voltage variation.

```
[Input voltage (mV) to the AD converter] = [in_voltage_raw] × [in_voltage_scale]
```

Figure 5.51. Calculation Formula of Input Voltage to the AD Converter

The files under `/sys/bus/iio/devices/iio:device0/` which are needed for calculating the input voltage are shown below.

Table 5.11. Files Needed for Calculating Input Voltage

File	Description
in_voltage0_raw	AD conversion value of single end input CH0 (power supply voltage)
in_voltage1_raw	AD conversion value of single end input CH1 (external battery voltage)
in_voltage_scale	Minimum input voltage variation of single end input

As an example, the following describes how to obtain the power supply voltage.

```
[armadillo ~]# cat /sys/bus/iio/devices/iio:device0/in_voltage0_raw
1766
[armadillo ~]# cat /sys/bus/iio/devices/iio:device0/in_voltage_scale
0.714111328
```

Figure 5.52. Obtaining the Input Voltage to the AD Converter

In example Figure 5.52, “Obtaining the Input Voltage to the AD Converter”, you can see that the input voltage to the AD converter is about 1.261 V (1766 x 0.714111328 [mV]).

The calculation formula for obtaining the power supply voltage from the input voltage to the AD converter is shown below.

```
[Power supply voltage (mV)] = [Input voltage to AD converter] x (200 + 24) ÷ 24
```

Figure 5.53. Calculation Formula of the Power Supply Voltage

Taking Figure 5.52, “Obtaining the Input Voltage to the AD Converter” as an example, from the AD converter input voltage of 1.261V it can be determined that the power supply voltage is about 11.770V.

The power supply voltage can be displayed using the `awk` command as below.

```
[armadillo ~]# adin_raw=`cat /sys/bus/iio/devices/
iio:device0/in_voltage0_raw`
[armadillo ~]# adin_scale=`cat /sys/bus/iio/devices/
iio:device0/in_voltage_scale`
[armadillo ~]# echo $adin_raw $adin_scale | awk '{printf ("%d
\n",$1*$2*(200+24)/24)}'
11770
```

5.8.2. Monitoring Power Supply Voltage

The `vintrigger` command can be used to execute an arbitrary command when the power supply voltage reaches a specified voltage.

`vintrigger` cannot be run multiple times at once.

The `vintrigger` command help is as follows.

```
[armadillo ~]# vintrigger
Usage: vintrigger -o|-u VOLTAGE [-i INTERVAL] [COMMAND ARGS]
Options:
  -o, --over=VOLTAGE
      Execute the program COMMAND when the detected voltage is equal
      to or over the VOLTAGE[mV].
  -u, --under=VOLTAGE
      Execute the program COMMAND when the detected voltage is equal
      to or under the VOLTAGE[mV].
VOLTAGE: Range: 0 - 28980

  -i, --interval=INTERVAL
      Compare with Vin to the VOLTAGE at INTERVAL second intervals.
INTERVAL: Range: 0 - 4294967295 (Default: 60)

  -h, --help
      Print usage(this message) and exit.
  -v, --version
      Print version information and exit.
```

Figure 5.54. vintrigger Command Help

The following shows an example of monitoring the power supply voltage at intervals of 30 seconds and turning on LED2 when the power supply voltage drops below 11000mV (11V).

```
[armadillo ~]# vintrigger -u 11000 -i 30 echo 1 > /sys/class/leds/led2/brightness
```

Figure 5.55. vintrigger Command Example

The `vintrigger` command log is output to the `/var/log/messages` file.

```
[armadillo ~]# cat /var/log/messages
                  :
Jul  1 09:38:52 armadillo-iotg vintrigger[812]: waiting for
an under range alert (11000 mV).❶
Jul  1 09:38:52 armadillo-iotg vintrigger[812]: exceeded the
limit. executing command.❷
```

❶ Waiting until it falls below the specified voltage (11000mV).

❷ The command is executed as the specified voltage has been reached.

5.9. Armadillo-IoT RS232C Add-on Module RS00

The Armadillo-IoT RS232C Add-on Module RS00 (hereafter referred to as the RS232C add-on module) has one RS232C level serial port. Since the device driver of the serial port of the RS232C add-on module is implemented as a TTY device, it can be controlled from the TTY device file.

The add-on interface that connects the RS232C add-on module and the corresponding TTY device file is shown below.

Table 5.12. Add-on Interfaces and TTY Device Files

Add-on Interface	TTY Device File
CON1	/dev/ttymxc0
CON2	/dev/ttymxc1

The RS232C Add-on Module is connected to CON1 in the development set's factory default state.

The add-on interface to which the RS232C add-on module is connected can be seen in the Linux kernel boot log. When connected to CON1, the output is as follows.

```
Atmark Techno RS232C board detected at CON1(Rev 2,
SerialNumber=xxxx).
```

When using the Linux kernel 3.14.38-at2 or earlier (kernel image uImage-x1-v 2.00 or earlier), the maximum baud rate that can be used is 1500000 [Baud] due to software restrictions.

5.10. Armadillo-IoT Isolated RS232C/422/485 Add-on Module RS01

The Armadillo-IoT Isolated RS232C/422/485 Add-on Module RS01 (hereafter referred to as the isolated serial add-on module) has one electrically isolated RS232C/RS422/RS485 serial port on it. Since the device driver of the serial port of the isolated serial add-on module is implemented as a TTY device, it can be controlled from the TTY device file.

The add-on interface that connects the isolated serial add-on module and the corresponding TTY device file is shown below.

Table 5.13. Add-on Interfaces and TTY Device Files

Add-on Interface	TTY Device File
CON1	/dev/ttymxc0
CON2	/dev/ttymxc1

The add-on interface to which the isolated serial add-on module is connected can be seen in the Linux kernel boot log. When connected to CON1, the output is as follows.

```
Atmark Techno RS485/RS422/RS232C board detected at CON1(Rev
2, SerialNumber=xxxx).
```

When using the Linux kernel 3.14.38-at2 or earlier (kernel image uImage-x1-v 2.00 or earlier), the maximum baud rate that can be used is 1500000 [Baud] due to software restrictions.

5.10.1. Changing RS422/RS485 Communication Settings

Before turning on Armadillo-IoT, if the isolated serial add-on module SW1.1 is set to OFF, the RS485 configuration of the TTY device is automatically activated.

Do not change the setting of the insulated serial add-on module SW1.1 after turning on the Armadillo-IoT. Doing so may cause damage.

RS485 settings that can be altered and the default values when automatically activated are shown in Table 5.14, “RS485 Setting and Default Values”. Flags shows the logical sum of each bit.

Table 5.14. RS485 Setting and Default Values

Configuration		Description	Default Value
flags	ENABLED (bit0)	0: RS485 disabled 1: RS485 enabled	1
	RTS_ON_SEND (bit1)	0: RTS (Driver Enable) is low during data transmission 1: RTS (Driver Enable) is high during data transmission	1
	RTS_AFTER_SEND (bit2)	0: RTS (Driver Enable) is low outside of data transmission 1: RTS (Driver Enable) is high outside of data transmission	0
	RX_DURING_TX (bit4)	0: Half duplex 1: Full duplex	0
delay_rts_before_send		Delay time before transmission (ms)	0
delay_rts_after_send		Delay time after transmission (ms)	0

Please do not change the default values of RTS_ON_SEND and RTS_AFTER_SEND of "flags". If changed, data transmission will no longer be possible.

It is not possible to use a TTY device with RS485 enabled as a console.

The RS485 settings can be changed with an application program or Linux kernel startup options.

For details on how to create an application program, refer to the document (`Documentation/serial/serial-rs485.txt`) included in the Linux kernel source code.

For the Linux kernel startup options, RS485 is set with the following option specifiers.

Table 5.15. Setting RS485 from Linux kernel Boot Options

Option Specifier	Description
imx.rs485_uart1=	Specify RS485 settings for UART1 (ttymxc0) connected to CON1.
imx.rs485_uart2=	Specify RS485 settings for UART2 (ttymxc1) connected to CON2.

The format of the RS485 settings is as follows.

```
<flags>,<delay_rts_before_send>,<delay_rts_after_send>
```

As an example, to set the RS485 settings of the Isolated Serial Add-on Module connected to CON2 to full duplex communication, start up in Maintenance mode and run the following commands.

```
=>setenv optargs imx.rs485_uart2=0x13,0,0
=>saveenv
```

5.11. Armadillo-IoT Wi-SUN Add-on Module WS00

The Armadillo-IoT Wi-SUN Add-on Module WS00 (hereafter referred to as Wi-SUN add-on module) is equipped with ROHM BP35A1.

The Wi-SUN add-on module can be controlled from the TTY device file using ASCII commands. The add-on interface that connects the Wi-SUN add-on module and the corresponding TTY device file are shown below.

Table 5.16. Add-on Interfaces and TTY Device Files

Add-on Interface	TTY Device File
CON1	/dev/ttymxc0
CON2	/dev/ttymxc1

The add-on interface to which the Wi-SUN add-on module is connected can be seen in the Linux kernel boot log. When connected to CON1, the output is as follows.

```
Atmark Techno Wi-SUN board detected at CON1(Rev 2,
SerialNumber=xxxx).
```

5.11.1. Obtaining Configuration Information

As an example of controlling the Wi-SUN add-on module, obtain the configuration information of BP35A1.

The procedure for obtaining the configuration information of BP35A1 equipped on the Wi-SUN add-on module connected to the add-on interface (CON1) is shown below.

Procedure 5.3. Obtaining Configuration Information

1. Run the `cu` command and connect to `/dev/ttymxc0`. The baudrate is 115,200 bps.

```
[armadillo ~]$ cu -l /dev/ttymxc0 -s 115200
Connected.
```

2. Running the SKINFO command results in the configuration information of BP35A1 being displayed.

```
SKINFO
EINFO FE80:0000:0000:0000:021D:1290:0004:0FBE 001D129000040FBE 21 FFFF
FFFE
OK
```

3. To quit `cu`, enter "~." (a tilde "~" followed by a dot ".").

```
Disconnected.
[armadillo ~]$
```

To obtain the configuration information with cu, atmark-x1-base v1.1.0 or later must be used. It can be used in either of the following ways.

- Run the following command to update atmark-x1-base to the latest version

```
[armadillo ~]$ apt-get update
[armadillo ~]$ apt-get upgrade
```

- Use the debian-jessie-armhf_aiotg3_20160705.tar.gz or later root file system
- Use a root file system created using x1-debian-builder v1.1.0 or later

For other ASCII commands and detailed information on BP35A1 please refer to ROHM documentation.

"ROHM Sub-GHz Series" Support Page Document Download | Semiconductor ROHM

http://micro.rohm.com/jp/download_support/wi-sun

5.12. Armadillo-IoT Isolated RS485 Add-on Module RS02

The Armadillo-IoT Isolated RS485 Add-on Module RS02 (hereafter referred to as the isolated RS485 add-on module) has one electrically isolated RS422/RS485 serial port on it. Since the device driver for the serial port of the isolated RS485 add-on module is implemented as a TTY device, it can be controlled from the TTY device file.

The add-on interface that connects the isolated RS485 add-on module and the corresponding TTY device file are shown below.

Table 5.17. Add-on Interfaces and TTY Device Files

Add-on Interface	TTY Device File
CON1	/dev/ttymxc0
CON2	/dev/ttymxc1

The add-on interface to which the isolated RS485 add-on module is connected can be seen in the Linux kernel boot log. When connected to CON1,the output is as follows.

```
Atmark Techno RS485 board detected at CON1(Rev 1,
SerialNumber=xxxx).
```

When using the Linux kernel 3.14.38-at2 or earlier (kernel image uImage-x1-v 2.00 or earlier), the maximum baud rate that can be used is 1500000 [Baud] due to software restrictions.

5.12.1. Changing RS422/RS485 Communication Settings

The TTY device RS485 settings are applied automatically.

RS485 settings that can be altered and the default values when automatically activated are shown in Table 5.14, “RS485 Setting and Default Values”. Flags shows the logical sum of each bit.

Table 5.18. RS485 Setting and Default Values

Configuration		Description	Default Value
flags	ENABLED (bit0)	0: RS485 disabled 1: RS485 enabled	1
	RTS_ON_SEND (bit1)	0: RTS (Driver Enable) is low during data transmission 1: RTS (Driver Enable) is high during data transmission	1
	RTS_AFTER_SEND (bit2)	0: RTS (Driver Enable) is low outside of data transmission 1: RTS (Driver Enable) is high outside of data transmission	0
	RX_DURING_TX (bit4)	0: Half duplex 1: Full duplex	0
delay_rts_before_send		Delay time before transmission (ms)	0
delay_rts_after_send		Delay time after transmission (ms)	0

Please do not change the default values of RTS_ON_SEND and RTS_AFTER_SEND of "flags". If changed, data transmission will no longer be possible.

It is not possible to use a TTY device with RS485 enabled as a console.

The RS485 settings can be changed with an application program or Linux kernel startup options.

For details on how to create an application program, refer to the document (`Documentation/serial/serial-rs485.txt`) included in the Linux kernel source code.

For the Linux kernel startup options, RS485 is set with the following option specifiers.

Table 5.19. Setting RS485 from Linux kernel Boot Options

Option Specifier	Description
imx.rs485_uart1=	Specify RS485 settings for UART1 (ttymxc0) connected to CON1.
imx.rs485_uart2=	Specify RS485 settings for UART2 (ttymxc1) connected to CON2.

The format of the RS485 settings is as follows.

```
<flags>,<delay_rts_before_send>,<delay_rts_after_send>
```

As an example, to set the RS485 configuration of the isolated serial add-on module connected to CON2 to full duplex communication, start up in maintenance mode and execute the following command.

```
=>setenv optargs imx.rs485_uart2=0x13,0,0
=>saveenv
```

5.13. Armadillo-IoT Isolated Digital I/O / Analog Input Add-on Module DA00

The Armadillo-IoT Isolated Digital I/O and Analog Input Add-on Module DA00 (hereafter referred to as the isolated I/O add-on module) consists of two electrically isolated digital input ports, two digital output ports and two 0 to 5V analog input ports.

The digital I/O device driver for the isolated I/O add-on module is implemented as a GPIO device, and the analog input device driver is implemented as a IIO (Industrial I/O) device.

Table 5.20, “Add-on Interfaces and GPIO Class Directories” shows the add-on interface that connects the isolated IO add-on module and the corresponding GPIO class directory. The IIO device is iio:deviceN (N is a sequential number start from '0') in the order in which the devices are detected.

Table 5.20. Add-on Interfaces and GPIO Class Directories

Add-on Interface	Port	GPIO Class Directory
CON1	Digital Output 1	/sys/class/gpio/gpio89
	Digital Output 2	/sys/class/gpio/gpio90
	Digital Input 1	/sys/class/gpio/gpio141
	Digital Input 2	/sys/class/gpio/gpio140
CON2	Digital Output 1	/sys/class/gpio/gpio106
	Digital Output 2	/sys/class/gpio/gpio107
	Digital Input 1	/sys/class/gpio/gpio181
	Digital Input 2	/sys/class/gpio/gpio180

The add-on interface to which the isolated I/O add-on module is connected can be seen in the Linux kernel boot log. When connected to CON2, the output is as follows.

```
Atmark Techno DI/DO/AD board detected at CON2 (Rev 1,
SerialNumber=xxxx).
```

5.13.1. Setting the Digital Output State

The output state can be set by writing a value to the value file under the GPIO class directory. "0" indicates open circuit, "1" indicates closed circuit.

An example of setting the digital output of the isolated IO add-on module connected to the add-on interface (CON2) to open is shown below.

```
[armadillo ~]# echo 0 > /sys/class/gpio/gpio106/value
```

Figure 5.56. Changing the Digital Output State

5.13.2. Getting the Digital Input State

The input state can be obtained by reading the value from the value file under the GPIO class directory. "0" represents a connection to GND_ISO. "1" represents either open or the application of 3.15V or more.

An example of obtaining the state of the digital input of the isolated IO add-on module connected to the add-on interface (CON2) is shown below.

```
[armadillo ~]# cat /sys/class/gpio/gpio181/value
1
```

Figure 5.57. Getting the Digital Input State

5.13.3. Getting the Analogue Input Voltage

The input voltage can be calculated from the files under the `/sys/bus/iio/devices/iio:device1/` directory.

The IIO devices are iio:deviceN (N is a sequential number from '0') in the order in which the devices are detected. IIO devices can be identified from the IIO device name. The IIO device name of the AD converter on the isolated IO add-on module is "mcp3202".

```
[armadillo ~]# cat /sys/bus/iio/devices/iio:device1/name
mcp3202
```

The input voltage to the AD converter can be calculated from the AD conversion value and the minimum input voltage variation.

```
[Input Voltage (mV) to AD Converter] = [AD Conversion Value] × [Minimum Input
Voltage Variation]
```

Figure 5.58. Calculation Formula of Input Voltage to the AD Converter

The files under the `/sys/bus/iio/devices/iio:device1/` directory required for calculating the input voltage are shown below.

Table 5.21. Files Needed for Calculating Input Voltage

File	Description
in_voltage0_raw	AD conversion value of single end input CH 0
in_voltage1_raw	AD conversion value of single end input CH 1
in_voltage_scale	Minimum input voltage variation of single end input
in_voltage0-voltage1_raw	AD conversion value of the pseudo differential input
in_voltage-voltage_scale	Minimum input voltage variation of pseudo differential input

An example of calculating the input voltage to the single end input CH0 is shown below.

```
[armadillo ~]# cat /sys/bus/iio/devices/iio:device1/in_voltage0_raw
2048
[armadillo ~]# cat /sys/bus/iio/devices/iio:device1/in_voltage_scale
1.220703125
```

Figure 5.59. Obtaining the Input Voltage to the AD Converter

In the Figure 5.59, “Obtaining the Input Voltage to the AD Converter” example, the input voltage to the single end input CH0 is 2.5 V (2048 x 1.220703125 [mV]).

The power supply voltage can be displayed using the `awk` command as below.

```
[armadillo ~]# adin_raw=`cat /sys/bus/iio/devices/
iio:device1/in_voltage0_raw`
[armadillo ~]# adin_scale=`cat /sys/bus/iio/devices/
iio:device1/in_voltage_scale`
[armadillo ~]# echo $adin_raw $adin_scale | awk '{printf
("%d",$1*$2)}'
2500
```

Chapter 6. Bootloader Specifications

This chapter explains about the boot modes and available functions of the bootloader.

6.1. Bootloader Boot Modes

When the bootloader starts up, it switches to one of two modes depending on the state of the slide switch of the USB serial converter. For details about the USB serial converter slide switch, refer to Section 3.5, “Slide Switch Configuration”.

Table 6.1. Bootloader Boot Modes

Boot Mode Type	Slide Switch	Description
Maintenance mode	Outside	The U-Boot command prompt starts from which various configuration is possible.
Auto boot mode	Inside	The Linux kernel is automatically booted after power on.

When a USB serial converter is not connected, auto boot mode is used and the Linux kernel boots.

6.2. Bootloader Functions

In the maintenance mode of U-Boot it is possible to perform configuration such as Linux Kernel Boot Options.

Useful commands that can be used in maintenance mode are shown in Table 6.2, “Useful Maintenance Mode Commands List”.

Table 6.2. Useful Maintenance Mode Commands List

Command	Description
boot	Used to boot the operating system
bdinfo	Displays hardware information
md mm nm mw cp cmp	Used for simple memory access
printenv setenv saveenv	Used to set environment variables. Operating system boot options can be set with environment variables.
crc32	Used to display the checksum of a memory area
version	Displays bootloader version

Help for each command can be displayed as shown in Figure 6.1, “Displaying U-Boot Command Help”.

```
=> help [command]
```

Figure 6.1. Displaying U-Boot Command Help

6.2.1. How to Specify the Linux Kernel Image and Device Tree Blob

For the bootloader loading of the operating system, it is possible to use a Linux kernel image and device tree blob saved in either the eMMC or an SD card.

To specify the device storing the file use the environment variable "mmcdev", and to specify the partition number use the environment variable "mmcpart".

The Linux kernel image is saved with the filename "uImage". The device tree blob is saved with the filename "armadillo_iotg_g3.dtb".

Table 6.3, “mmcdev Configuration Values and Boot Devices” shows the relationship between the values that can be set with "mmcdev" and the boot device.

Table 6.3. mmcdev Configuration Values and Boot Devices

Configuration Value	Boot Device
0	SD card (connected to CON4)
1	eMMC

When specifying partition 1 of eMMC, please do as shown inFigure 6.2, “Booting from a Linux Kernel Image Stored on Partition 1 of eMMC”.

```
=> setenv mmcdev 1
=> setenv mmcpart 1
```

Figure 6.2. Booting from a Linux Kernel Image Stored on Partition 1 of eMMC

The default values of "mmcdev" and "mmcpart" differ between the bootloader for QSPI and the bootloader for SD. Table 6.4, “Bootloader Types and Default Values for mmcdev and mmcpart” shows the relationship between the type of bootloader and the default values.

Table 6.4. Bootloader Types and Default Values for mmcdev and mmcpart

Bootloader Type	Bootloader Filename	mmcdev Default Value	mmcpart Default Value
For QSPI	u-boot-x1-at*.bin	1 (eMMC)	1
For SD	u-boot-x1-sd-at*.bin	0 (SD)	1

6.2.2. Specifying the Root File System

Devices on which a root file system is stored can be specified with the environment variable "mmcroot".

When specifying partition 2 of eMMC, do as shown in Figure 6.3, “Specifying the Root File System Stored in Partition 2 of eMMC”.

```
=> setenv mmcroot /dev/mmcblk2p2
```

Figure 6.3. Specifying the Root File System Stored in Partition 2 of eMMC

The default value of "mmcroot" differs between the bootloader for QSPI and the bootloader for SD. Table 6.5, “Bootloader Types and mmcroot Default Values” shows the relationship between the type of bootloader and the default value.

Table 6.5. Bootloader Types and mmcroot Default Values

Bootloader Type	Bootloader Filename	mmcroot Default Value
For QSPI	u-boot-x1-v*.*.*.bin	/dev/mmcblk2p2 (eMMC partition 2)
For SD	u-boot-x1-sd-v*.*.*.bin	/dev/mmcblk0p2(SD partition 2)

6.2.3. Saving Environment Variables

Environment variables can be saved with the "saveenv" command. If the power of Armadillo-IoT is turned off without performing the save, the environment variables set with setenv will be lost.

When using the QSPI bootloader environment variables are saved on the QSPI Flash. When using the SD bootloader, environment variables are saved on the SD.

To restore all the environment variables to default values, see Figure 6.4, “Restoring All Environment Variables to Default Values”.

```
=> env default -a
=> saveenv
```

Figure 6.4. Restoring All Environment Variables to Default Values

6.2.4. Linux Kernel Boot Options

6.2.4.1. Typical Linux Kernel Boot Options

The Linux kernel has various startup options. For details, refer to the Linux reference books and the documentation (Documentation/kernel-parameters.txt) included in the Linux kernel source code.

In this section, typical startup options that can be used with Armadillo-IoT are shown in Table 6.6, “Linux Kernel Boot Options Examples”.

Table 6.6. Linux Kernel Boot Options Examples

Option Specifier	Description
console=	Specify the initial console which outputs boot log etc. In the following example ttymxc1 is specified for the console and 115200 for the baudrate. `console=ttymxc1,115200`

root=	Specify the device on which the root file system is stored.
	For the device, specify the device as recognized by the Linux kernel.
	The initrd is set as the root file system as shown in the following example.
	`root=/dev/ram0`
	When storing the root file system on an SD card, specify the SD card's device file. In the following example, the second partition of the microSD card is specified for the device.
	`root=/dev/mmcblk0p2`

rootwait	Delay the mounting of the root file system until the device specified with "root=" becomes available.
mem	Specify the amount of memory available to the Linux kernel. Specify this when part of RAM is to be used as dedicated memory.

6.2.4.2. Setting Linux Kernel Boot Options

Linux kernel boot options can be specified with the "mmcargs" environment variable.

The default value of "mmcargs" is set to the following value.

```
setenv mmcargs setenv bootargs console=${console},${baudrate} root=${mmcroot} $
{optargs}
```

By default, the console is set to the environment variable "console", the baudrate of the console is set to the environment variable "baudrate", and the root file system is set to the environment variable "mmcroot".

When wanting to add Linux kernel boot options, it is convenient to use the "optargs" environment variable.

Next, as an example, how to set the amount of memory available to the Linux kernel to 384M is shown in Figure 6.5, “Setting the Available Memory Amount to 384M”.

```
=> setenv optargs mem=384M
=> saveenv
=> printenv optargs
mem=384M
```

Figure 6.5. Setting the Available Memory Amount to 384M

Chapter 7. Build Procedure

This chapter describes the procedure for creating an image the same as the factory default image.

The source code to use is included in the development set DVD. The latest source code can be downloaded from the Armadillo site. As over time new functionality is added and defects are fixed, we recommend checking if versions newer than that on the DVD have been released or not and then using the latest version.

Armadillo Site - Armadillo-IoT Gateway Documents / Downloads

http://armadillo.atmark-techno.com/armadillo-iot-g3/downloads

The development process involves working with basic libraries, applications and system configuration files. While all files are altered only under the working directory, in order to ensure the PC operating system is not inadvertently damaged from any mistakes made during development please perform all work as a **general user** and not a root user.

7.1. Building the Bootloader

This section explains the procedure for creating an image file from the source code of the bootloader "U-Boot".

Procedure 7.1. Building the Bootloader

1. **Preparing Source Code**

   Prepare and extract the U-Boot source code archive.

```
[PC ~]$ ls
uboot_2014.04-at1.0.0.tar.gz
[PC ~]$ tar xf uboot_2014.04-at[version].tar.gz
[PC ~]$ ls
uboot_2014.04-at[version]  uboot_2014.04-at[version].tar.gz
```

2. **Applying Default Configuration**

   Enter the U-Boot directory and apply the default configuration for Armadillo-IoT Gateway G3. In this example, an image for Flash memory boots is created. Specify `x1_config` as the default config. To create an SD boot image, specify `x1_sd_config`.

```
[PC ~]$ cd uboot_2014.04-at[version]
[PC ~/uboot_2014.04-at[version]]$ make ARCH=arm x1_config
```

3. **Building**

   The **`make`** command is used to perform the build.

```
[PC ~/uboot_2014.04-at[version]]$ make CROSS_COMPILE=arm-linux-gnueabihf-
```

4. **Confirming the creation of the Image File**

   When the build finishes an image file is created in the U-Boot directory.

```
[PC ~/uboot_2014.04-at[version]]$ ls u-boot-x1.bin
u-boot-x1.bin
```

7.2. Building the Linux Kernel

This section explains the procedure for creating an image file from the Linux kernel source code and initramfs archive.

Files required for the build

linux-3.14-x1-at[version].tar.gz
initramfs_x1-[version].cpio.gz

Procedure 7.2. Building the Linux Kernel

1. **Extracting the Archive**

   Extract the Linux kernel source code archive.

```
[PC ~]$ls
initramfs_x1-[version].cpio.gz linux-3.14-x1-at[version].tar.gz
[PC ~]$tar xf linux-3.14-x1-at[version].tar.gz
[PC ~]$ls
initramfs_x1-[version].cpio.gz linux-3.14-x1-at[version]  linux-3.14-x1-
at[version].tar.gz
```

2. **Creating a Symbolic Link to the Initramfs Archive**

   Move to the Linux kernel directory and create a symbolic link to the initramfs archive.

```
[PC ~]$cd linux-3.14-x1-at[version]
[PC ~/linux-3.14-x1-at[version]]$ln -s ../initramfs_x1-[version].cpio.gz
initramfs_x1.cpio.gz
```

3. **Configuration**

   Perform the configuration.

```
[PC ~/linux-3.14-x1-at[version]]$make ARCH=arm x1_defconfig
```

4. **Building**

   To do the build, run the following commands.

```
[PC ~/linux-3.14-x1-at[version]]$ make ARCH=arm CROSS_COMPILE=arm-linux-
gnueabihf-
[PC ~/linux-3.14-x1-at[version]]$ make ARCH=arm CROSS_COMPILE=arm-linux-
gnueabihf- LOADADDR=0x80008000 uImage
```

5. **Confirming the creation of the Image File**

   When the build finishes, image files (the Linux kernel and DTB) are created under the arch/arm/boot/directory and arch/arm/boot/dts/ directory.

```
[PC ~/linux-3.14-x1-at[version]]$ ls arch/arm/boot/uImage
uImage
[PC ~/linux-3.14-x1-at[version]]$ ls arch/arm/boot/dts/
armadillo_iotg_g3.dtb
armadillo_iotg_g3.dtb
```

7.3. Building a Debian GNU/Linux Root File System

The following shows how to build a Debian GNU/Linux root file system using x1-debian-builder.

x1-debuan-builder is a tool that can be used on Linux running on a PC such as ATDE6 to build armhf architecture Debian GNU/Linux root file systems for Armadillo-IoTG3.

Depending on how it is used, after starting Armadillo-IoT G3 the first time the ssh secret key, operation log, shell command history, the configuration files associated with the hardware UUID and so on are generated on the root file system. If the root file system is then copied as-is to another Armadillo-IoT G3 , there is a possibility that behavior differences may occur due to key leakage or UUID mismatch. Therefore it is recommended to use x1-debuan-builder and build a new root file system for use in mass production etc.

7.3.1. Creating the Factory Default Root File System Archive

The procedure for building the root file system archive of the factory default state is as follows. As packages are obtained over the internet it will depending on the connection speed, but will generally take around 40 minutes.

```
[ATDE ~]$sudo apt-get update && sudo apt-get install qemu-user-static
[ATDE ~]$tar xf x1-debian-builder-[VERSION].tar.gz
[ATDE ~]$cd x1-debian-builder-[VERSION]
[ATDE ~]$sudo ./build.sh
```

Figure 7.1. Procedure for Creating the Factory Default Root File System Archive

7.3.2. Creating a Customized Root File System Archive

The root file system can be modified by altering the files under x1-debian-builder-*[VERSION]*/aiotg3_resources and executing the build.sh script.

7.3.2.1. Adding Files and Directories

All files and directories under aiotg3_resources except for the resources directory are copied directly to the root file system. Both the UID and GID of the files become "root".

7.3.2.2. Changing Packages

The packages to be installed on the root file system can be customized by altering aiotg3_resources/resources/packages.

One package name is written per line. Please use package names in the form that can be correctly used as the argument to "apt-get install" on Armadillo-IoT G3.

If an incorrect package name is specified, the following error message is displayed in the build log and the archive is generated without the package.

```
E: Unable to locate package XXXXX
```

Figure 7.2. Error Message when Incorrect Package Name Specified

Other packages that the package depends on are automatically installed by the apt command without needing to specify them. Also, packages that form the base of Debian GNU/Linux such as apt and dpkg are also automatically installed.

The packages include the lua and ruby interpreters and a web server (lighttpd). If they are not required just delete each package's line.

For packages such as openssh-server that automatically generate private keys when installed, as a general rule do not add them to the packages and instead use "apt-get install" to install them separately after starting Armadillo.

If openssh-server is added to packages, it will be possible to log in to all Armadillo which the built root file system archive is written to using a single public key. If intentionally using the same secret key for multiple Armadillo, understand that it can lead to vulnerabilities and take appropriate measures before using it.

Chapter 8. Rewriting Image Files

This chapter explains the procedure for rewriting the image file written to Armadillo-IoT G3's internal storage (eMMC and QSPI flash memory).

The bootloader image files used in this chapter are included on the development set DVD. The latest versions of the files can be downloaded from the "Armadillo site". As over time new functionality is added and defects are fixed, we recommend checking if versions newer than that on the DVD have been released or not and then using the latest version.

Armadillo Site - Armadillo-IoT G3 Documents / Downloads

http://armadillo.atmark-techno.com/armadillo-iot-g3/downloads

8.1. Using the Installation Disc

Using the installation disc, all of the images on the internal storage can be rewritten at once. They can be used as a recovery method if Armadillo stops booting due to a software issue.

As all image files saved in the internal storage are overwritten, any data or applications already saved there will be deleted.

When rewriting only a specific image, please refer to Section 8.2, “Rewriting Only Specific Image Files”.

The installation disk is created with ATDE. The file used to create the installation disk is shown below.

Table 8.1. File Used for Creating Installation Disk

File	File Name
Install disk image	install_disk_sd_*[version]*.img

8.1.1. Creating an Installation Disk

1. Please prepare an SD card with a 512MB or larger capacity.
2. Connect the SD card to ATDE. For details, refer toSection 3.2.2, “Using Removable Devices”.
3. Unmount the SD card if it is currently mounted.

```
[PC ~]$mount
(abbreviated)
/dev/sdb1 on /media/atmark/B18A-3218 type vfat
(rw,nosuid,nodev,relatime,uid=1000,gid=1000,fmask=0022,dmask=0077,codepag
e=437,iocharset=utf8,shortname=mixed,showexec,utf8,flush,errors=remount-
ro,uhelper=udisks2)
[PC ~]$sudo umount /dev/sdb1
```

4. Write the installation disk image to the SD card.

```
[PC ~]$sudo dd if=install_disk_sd_[version].img of=/dev/sdb bs=4M
94+1 records input
94+1 records output
397410304 bytes (397 MB) copied, 45.8441 seconds、 8.7 MB/sec
[PC ~]$sync
```

8.1.2. Running the Installation

1. Make sure the Armadillo is powered off. If the power is on, turn it off. Also, be sure to remove the battery if one is connected to Section 11.10, “CON13 RTC Backup Interface 1”.

2. Check the slide switch of the USB serial converter. Make sure that the slide switch is set to side "1" of Figure 3.8, “Slide Switch Configuration”.

3. Use the installation disk to perform an SD boot. Insert the installation disk into the SD slot (CON4) and set JP1 to short.

4. When the Armadillo is powered on while holding User Switch 1 down, the bootloader will start up from the SD card and the following log will be displayed. When the log is displayed, release User Switch 1. For the location of User Switch 1, see Section 2.4, “Armadillo-IoT Gateway Exterior”.

```
U-Boot 2014.04-at2 (Jun 12 2016 - 17:47:04)

CPU:   Freescale i.MX7D rev1.1 at 792 MHz
CPU:   Temperature: can't get valid data!
Reset cause: POR
I2C:   ready
DRAM:  512 MiB
MMC:   FSL_SDHC: 0, FSL_SDHC: 1
In:    serial
Out:   serial
Err:   serial
Found PFUZE300! deviceid 0x30, revid 0x11
Board Type: Armadillo-IoT G3(0a000000)
Revison: 0002
S/N: 2000
DRAM: 00001d05
XTAL: 00
Net:   FEC0
Normal Boot
=>
```

5. When the "boot" command is executed as follows, the installation starts and the eMMC and QSPI are re-written automatically.

```
=> boot
mmc0 is current device
mmc0 is current device
reading boot.scr
** Unable to read file boot.scr **
```

```
reading uImage
9301216 bytes read in 501 ms (17.7 MiB/s)
Booting from mmc ...
reading armadillo_iotg_g3.dtb
52708 bytes read in 20 ms (2.5 MiB/s)
## Booting kernel from Legacy Image at 80800000 ...
   Image Name:   Linux-3.14.38-at2
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    9301152 Bytes = 8.9 MiB
   Load Address: 80008000
   Entry Point:  80008000
   Verifying Checksum ... OK
## Flattened Device Tree blob at 83000000
   Booting using the fdt blob at 0x83000000
   Loading Kernel Image ... OK
   Using Device Tree in place at 83000000, end 8300fde3

Starting kernel ...
: (abbreviated)
*+*+* Recovery Start!! *+*+*
```

6. The following message will be displayed and installation is complete once automatically halted.

```
*+*+* Recovery Completed!! *+*+*

System is going down for system reboot now.

Starting local stop scripts.
Syncing all filesystems: done
Unmounting all filesystems: done
The system is going down NOW!
Sent SIGTERM to all processes
Sent SIGKILL to all processes
Requesting system halt
reboot: System halted
```

After installation is complete, please set JP1 to open.

8.2. Rewriting Only Specific Image Files

Only certain image files can be rewritten when Armadillo-IoT G3 is running.

The image files and their corresponding destinations are shown below.

Table 8.2. Image Files and their Destinations

Name	File Name	Storage	Device File
Bootloader Image	u-boot-x1-*[version]*.bin	QSPI Flash memory	/dev/mtdblock0
Linux Kernel Image	uImage-x1-*[version]*	eMMC	/dev/mmcblk2p1
Device Tree Blob	armadillo_iotg_g3-*[version]*.dtb		/dev/mmcblk2p1
Debian GNU/Linux Root File System	debian-jessie-armhf_aiotg3_*[version]*.tar.gz		/dev/mmcblk2p2

8.2.1. Rewriting the BootLoader Image

The method of rewriting the bootloader image is shown below. It is done by writing the image file directly to the MTD block device.

```
[armadillo ~]#dd if=u-boot-x1-[version].bin of=/dev/mtdblock0 ❶
282+1 records in
282+1 records out
288816 bytes (289 kB) copied, 5.4582 s, 52.9 kB/s
[armadillo ~]$sync
```

❶ Write the bootloader image at the beginning of the MTD block device.

8.2.2. Rewriting the Linux Kernel Image

The method for rewriting the Linux kernel image is shown below.

```
[armadillo ~]#mount -t vfat /dev/mmcblk2p1 /mnt ❶
[armadillo ~]#cp uImage-x1-[version] /mnt/uImage ❷
[armadillo ~]#umount /mnt ❸
```

❶ Mount the eMMC's first partition to the `/mnt/` directory.

❷ Copy the Linux kernel image to the `/mnt/` directory.

❸ Unmount the first partition of the eMMC mounted to the `/mnt/` directory.

8.2.3. Rewriting the DTB

The method of rewriting the DTB is shown below.

```
[armadillo ~]#mount -t vfat /dev/mmcblk2p1 /mnt ❶
[armadillo ~]#cp armadillo_iotg_g3-[version].dtb /mnt/armadillo_iotg_g3.dtb ❷
[armadillo ~]#umount /mnt ❸
```

❶ Mount the eMMC's first partition to the `/mnt/` directory.

❷ Copy the DTB to the `/mnt/` directory.

❸ Unmount the first partition of the eMMC mounted to the `/mnt/` directory.

8.2.4. Rewriting the Root File System

The procedure for rewriting the root file system on the eMMC is shown below.

Procedure 8.1. Rewriting the Root File System on eMMC

1. An SD boot must be performed in order to rewrite the root file system on eMMC. For details on how to create a boot disk and execute an SD boot, see Chapter 9, Utilizing SD Booting.

2. Prepare the Debian GNU/Linux root file system archive.

```
[armadillo ~]#ls
debian-jessie-armhf_aiotg-g3_[version].tar.gz
```

3. Recreate the root file system within the second partition of the eMMC.

```
[armadillo ~]# mkfs.ext4 /dev/mmcblk2p2  ❶
mke2fs 1.42.12 (29-Aug-2014)
/dev/mmcblk2p2 contains a ext4 file system
-last mounted on /root on Thu Jan  1 09:00:07 1970
Proceed anyway? (y,n) y  ❷
...(abbreviated)...

[armadillo ~]# mount -t ext4 /dev/mmcblk2p2 /mnt  ❸
[armadillo ~]# tar zxf debian-jessie-armhf_aiotg-g3l_[version].tar.gz -C /
mnt  ❹
[armadillo ~]# umount /mnt  ❺
```

-

❶ Recreate the file system on the second partition of eMMC.

❷ Enter y and then ENTER.

❸ Mount the second partition of the eMMC to the `/mnt/` directory.

❹ Extract the root file system archive to the `/mnt/` directory.

❺ Unmount the second partition of the eMMC mounted ot the `/mnt/` directory.

Chapter 9. Utilizing SD Booting

This chapter shows the procedure for starting Armadillo directly from an SD card (hereafter referred to as "SD booting"). By utilizing SD booting, the system image can be changed by replacing the SD card. In order to execute the procedure described in this chapter, an SD card with a capacity of 2GB or more is required. While in the procedure for SD booting below Debian GNU/Linux 8 (codename jessie) is used as an example, it is also possible to SD boot other operating systems.

When booting with SD, the settings of the bootloader are saved to the SD card.

Work on the SD card is done with ATDE and therefore the SD card must be connected to ATDE. For details, please refer to Section 3.2.2, “Using Removable Devices”.

When an SD card is connected to ATDE, it will be mounted automatically to the `/media/` directory. In order to execute the procedure described in this chapter, first unmount the SD card as follows.

```
[PC ~]$ mount
(abbreviated)
/dev/sdb1 on /media/52E6-5897 type ext2
(rw,nosuid,nodev,relatime,uid=1000,gid=1000,fmask=0022,dmask=0077,codepage=cp437,
iocharset=utf8,shortname=mixed,showexec,utf8,flush,errors=remount-
ro,uhelper=udisks)
[PC ~]$ sudo umount /dev/sdb1
```

Figure 9.1. Unmounting an Automatically Mounted SD Card

The bootloader image files used in this chapter are included on the development set DVD. The latest versions of the files can be downloaded from the "Armadillo site". As over time new functionality is added and defects are fixed, we recommend checking if versions newer than that on the DVD have been released or not and then using the latest version.

Armadillo Site - Armadillo-IoT G3 Documents / Downloads

http://armadillo.atmark-techno.com/armadillo-iot-g3/downloads

9.1. Boot Disk Creation

The boot disk is created with ATDE. The file used to create the boot disk is shown below.

Table 9.1. Files Used to Create Boot Disk

File	File Name
Bootloader Image for SD Boots	u-boot-x1-sd-*[version]*.bin

The procedure to create the boot disk in Table 9.2, “Boot Disk Configuration Example” is shown in Procedure 9.1, “Boot Disk Creation Example”.

Table 9.2. Boot Disk Configuration Example

Partition Number	Partition Size	File System	Description
1	128MByte	FAT32	The bootloader image for SD boots is placed here.
2	All remaining	ext4	An ext4 file system is first created to allow the creation of the root file system.

Procedure 9.1. Boot Disk Creation Example

1. Obtain the bootloader image file for SD boots.

```
[PC ~]$ls
u-boot-x1-sd-[version].bin
```

There are two types of bootloader image files.

Storage Location	Image File
SD Card	u-boot-x1-sd-[version].bin
Flash Memory	u-boot-x1-[version].bin

2. Create two primary partitions on the SD card.

```
[PC ~]$sudo fdisk /dev/sdb ❶

Welcome to fdisk (util-linux 2.25.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): o ❷
Created a new DOS disklabel with disk identifier 0x2b685734.

Command (m for help): n ❸
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): ❹

Using default response p.
Partition number (1-4, default 1): ❺
First sector (2048-7761919, default 2048): ❻
Last sector, +sectors or +size{K,M,G,T,P} (2048-7761919, default
7761919): +128M ❼

Created a new partition 1 of type 'Linux' and of size 128 MiB.

Command (m for help): n ❽
Partition type
```

```
   p   primary (1 primary, 0 extended, 3 free)
   e   extended (container for logical partitions)
Select (default p):  ❾

Using default response p.
Partition number (2-4, default 2):  ❿
First sector (264192-7761919, default 264192):  ⓫
Last sector, +sectors or +size{K,M,G,T,P} (264192-7761919, default
7761919):  ⓬

Created a new partition 2 of type 'Linux' and of size 3.6 GiB.

Command (m for help): t  ⓭
Partition number (1,2, default 2): 1  ⓮
Hex code (type L to list all codes): b  ⓯

If you have created or modified any DOS 6.x partitions, please see the
fdisk documentation for additional information.
Changed type of partition 'Linux' to 'W95 FAT32'.

Command (m for help): w  ⓰
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

[PC ~]$
```

❶ Start operations on the partition table of the SD card. If USB memory and so on is connected, the device file of the SD card may be for example sdc or sdd and different to that in this example.

❷ Create a new empty DOS partition table.

❸ Add a new partition.

❹ Since the default value (p: primary) is specified for the partition type, just press enter.

❺ Since the default value (1) is specified for the partition number, just press enter.

❻ Since the default value (the beginning of usable sectors) is used for the start sector, just press enter.

❼ Specify 128MByte worth for the last cylinder.

❽ Add a new partition.

❾ Since the default value (p: primary) is specified for the partition type, just press enter.

❿ Since the default value (2) is specified for the partition number, just press enter.

⓫ Since the default value (the sector following the last sector of the first partition) is used for the start sector, just press enter.

⓬ Since the default value (end sector) is used for the final sector, just press enter.

⓭ Change the system type of the partition.

⓮ Specify the first partition.

⓯ Specify 0xb (Win95 FAT32) as the system type of the partition.

⓰ Write the changes to the SD card.

3. Please display the partition list and confirm that two partitions have been created.

```
[PC ~]$sudo fdisk -l /dev/sdb

Disk /dev/sdb: 3.7 GiB, 3974103040 bytes, 7761920 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x2b685734

Device     Boot  Start     End Sectors  Size Id Type
/dev/sdb1         2048  264191  262144  128M  b W95 FAT32
/dev/sdb2       264192 7761919 7497728  3.6G 83 Linux
```

4. Create a file system on each partition.

```
[PC ~]$sudo mkfs.vfat -F 32 /dev/sdb1  ❶
mkfs.fat 3.0.27 (2014-11-12)
[PC ~]$sudo mkfs.ext4 /dev/sdb2  ❷
mke2fs 1.42.12 (29-Aug-2014)
Creating filesystem with 937216 4k blocks and 234320 inodes
Filesystem UUID: AAAAAAAA-BBBB-CCCC-DDDD-EEEEEEEEEEEE
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

[PC ~]$
```

❶ Create a FAT32 file system on the first partition.

❷ Create an ext4 file system on the second partition.

5. Write the bootloader image file for SD boots to the SD card.

```
[PC ~]$ls
u-boot-x1-sd-[version].bin
[PC ~]$sudo dd if=u-boot-x1-sd-[version].bin of=/dev/sdb bs=1k seek=1
[PC ~]$sync
```

9.2. Creating a Root File System

The root file system is created on the boot disk prepared in Section 9.1, “Boot Disk Creation”.

A Debian GNU/Linux root file system can be created. The file used to build the root file system is shown below.

Table 9.3. File Used for Creating Root File System

Linux Distribution	File Name	File Explanation
Debian GNU/Linux	debian-jessie-armhf_aiotg-g3_*[version]*.tar.gz	Root file system archive of Debian GNU/Linux 8 (codename jessie) for the ARM (armhf) architecture

9.2.1. Creating a Debian GNU/Linux Root File System

The procedure for building a root file system from the Debian GNU/Linux root file system archive is shown below.

Procedure 9.2. Creating a Root File System from the Debian GNU/Linux Root File System Archive

1. Prepare the Debian GNU/Linux root file system archive.

```
[PC ~]$ ls
debian-jessie-armhf_aiotg-g3_[version].tar.gz
```

2. Create the root file system on the second partition of the boot disk.

```
[PC ~]$ mkdir sd ❶
[PC ~]$ sudo mount -t ext4 /dev/sdb2 sd ❷
[PC ~]$ sudo tar zxf debian-jessie-armhf_aiotg-g3_[version].tar.gz -C sd
❸
[PC ~]$ sudo umount sd ❹
[PC ~]$ rmdir sd ❺
```

❶ Create the `sd/` directory to mount the SD card.

❷ Mount the second partition to the `sd/` directory.

❸ Extract the root file system archive to the `sd/` directory.

❹ Unmount the second partition mounted to the `sd/` directory.

❺ Delete the `sd/` directory.

Removing the SD card from the work PC before the unmounting is completed may damage the data on the SD card.

9.3. Storing the Linux Kernel Image and DTB

Place the Linux kernel image and DTB (Device Tree Blob) created in Section 9.1, “Boot Disk Creation” on the boot disk. The following files are used. Hereafter, DTB (Device Tree Blob) is referred to as DTB.

Table 9.4. Files Used to Create Boot Disk

File	File Name
Linux Kernel Image	uImage-x1-*[version]*
DTB	armadillo_iotg_g3-*[version]*.dtb

When placing the Linux kernel image and DTB on the SD card, the following conditions must be met. If these conditions are not met, the bootloader may not be able to detect the Linux kernel image or DTB.

Table 9.5. Conditions Under which the Bootloader can Detect the Linux Kernel

Item	Condition
File System	FAT32
Compressed Format	Uncompressed
Linux Kernel Image File Name	`uImage`
DTB Filename	`armadillo_iotg_g3.dtb`

The procedure for storing the Linux kernel image and DTB on the boot disk is as follows.

Procedure 9.3. Storing the Linux Kernel Image and DTB

1. Prepare the Linux kernel image and DTB.

```
[PC ~]$ls
uImage-x1-[version]  armadillo_iotg_g3-[version].dtb
```

2. Store the Linux kernel image on the first partition of the boot disk.

```
[PC ~]$mkdir sd ❶
[PC ~]$sudo mount -t vfat /dev/sdb1 sd ❷
[PC ~]$sudo cp uImage-x1-[version] sd/uImage ❸
[PC ~]$sudo cp armadillo_iotg_g3-[version].dtb sd/armadillo_iotg_g3.dtb
❹
[PC ~]$sudo umount sd ❺
[PC ~]$rmdir sd ❻
```

❶ Create the `sd/` directory to mount the SD card.

❷ Mount the first partition to the `sd/` directory.

❸ Copy the Linux kernel image to the `sd/` directory.

❹ Copy the DTB to the `sd/` directory.

❺ Unmount the first partition mounted to the `sd/` directory.

❻ Delete the `sd/` directory.

Removing the SD card from the work PC before the unmounting is completed may damage the data on the SD card.

9.4. Performing an SD Boot

This section explains how to boot from the boot disk created in Section 9.1, “Boot Disk Creation”.

Perform the following preparation before powering on Armadillo.

1. Connect the boot disk to the SD slot (CON4).
2. Set JP1 to short.

After preparation is complete, SD booting can be done by turning on the power. If the SD boot is successful, executing the saveenv command will display the message shown in Figure 9.2, “saveenv Messages for SD Boot”. Check that the save destination of the environment variables is "MMC".

```
=>saveenv
Saving Environment to MMC...
Writing to MMC(0)... done
=>
```

Figure 9.2. saveenv Messages for SD Boot

Please disable the write protection (switch) of the SD card. SD booting cannot be performed normally if the SD card cannot be written to.

Chapter 10. Electrical Specifications

10.1. Absolute Maximum Ratings

Table 10.1. Absolute Maximum Ratings

Item	Symbol	Min.	Max.	Unit	Notes
Power Supply Voltage	VIN	-0.3	26.4	V	
Input / Output Voltage	VI,VO	-0.3	OVDD+0.3	V	OVDD=VCC_3.3V_IO,VCC_3.3V @CON1,CON2,CON9,CON10
		-0.3	3.63	V	USB_HS_DP,USB_HS_DM @CON1,CON2,CON10
Operating Temperature Range[a][b]	Topr	-10	50	℃	With no condensation

[a]Board only (without a case): -20°C. to 70°C.

[b]This temperature is when power consumption is limited, such as by stopping communication of the 3G module at high temperature.

Absolute maximum ratings are values that should not be exceeded under any use or test conditions, even just momentarily. Please factor in a margin against these values during use.

10.2. Recommended Operating Conditions

Table 10.2. Recommended Operating Conditions

Item	Symbol	Min.	Typ.	Max.	Unit	Notes
Power Supply Voltage	VIN	8	12	26.4	V	
Ambient Operating Temperature[a]	Ta	-10	25	50	℃	With no condensation

[a]This temperature is when power consumption is limited, such as by stopping communication of the 3G module at high temperature.

When using the standard AC adapter (OP-AC12V2-00) or the wide temperature range AC adapter (OP-AC12V3-00)[1] when re-powering, wait at least 3 seconds before turning on the power.

When turning on the power using a power supply unit other than the above AC adapter, please do so with a monotonically increasing supply. Also, when re-powering, make sure to first turn the power off (5V or less for at least 10ms) and then turn it on again.

[1]Armadillo-IoT Gateway development set accessories

10.3. Input / Output Interface Electrical Specifications

Table 10.3. Input / Output Interface Power Supply Electrical Specifications

Item	Symbol	Min.	Typ.	Max.	Unit	Notes
Power Supply Voltage	VCC_5V CON1_USB_VBUS CON2_USB_VBUS	4.75	5	5.25	V	
	VCC_3.3V VCC_3.3V_IO	3.135	3.3	3.465	V	

Table 10.4. Input / Output Interface Electrical Specifications (OVDD = VCC_3.3V, VCC_3.3V_IO)[a]

Item	Symbol	Min.	Max.	Unit	Notes
High-level output voltage	VOH	0.8xOVDD	OVDD	V	IOH = -1.8mA, -3.6mA, -7.2mA, -10.8mA
Low-level output voltage	VOL	0	0.2×OVDD	V	IOL = 1.8mA, 3.6mA, 7.2mA, 10.8mA
High-level input voltage	VIH	0.7×OVDD	OVDD+0.3	V	
Low-level input voltage	VIL	-0.3	0.3×OVDD	V	
Input leakage current (no Pull-up/Pull-down)	IIN	-5	5	μA	
Pull-up resistance (5K_PU)	-	4.8	5.3	kΩ	
Pull-up resistance (47K_PU)	-	45.8	49.8	kΩ	
Pull-up resistance (100K_PU)	-	101	105	kΩ	
Pull-up resistance (100K_PD)	-	101	108	kΩ	

[a] Excluding pin 9 (LED1_CTRL) of the WWAN expansion interface

Table 10.5. WWAN Expansion Interface Pin 9 (LED1_CTRL) Electrical Specifications

Item	Symbol	Min.	Max.	Unit	Notes
High-level input voltage	VIH	1.4	5.25	V	
Low-level input voltage	VIL	-0.3	0.3	V	

10.4. Power Supply Circuit Structure

The structure of the power supply circuit of Armadillo-IoT is as follows. The input voltage from the power input interface 1 (CON 14) or the power input interface 2 (CON 15) is converted to each required voltage by the power supply IC and supplied to the internal circuits and each interface. Please design the connection of any external equipment and power supplies so as not to exceed the maximum output current value of each interface and switching regulator (DC-DC).

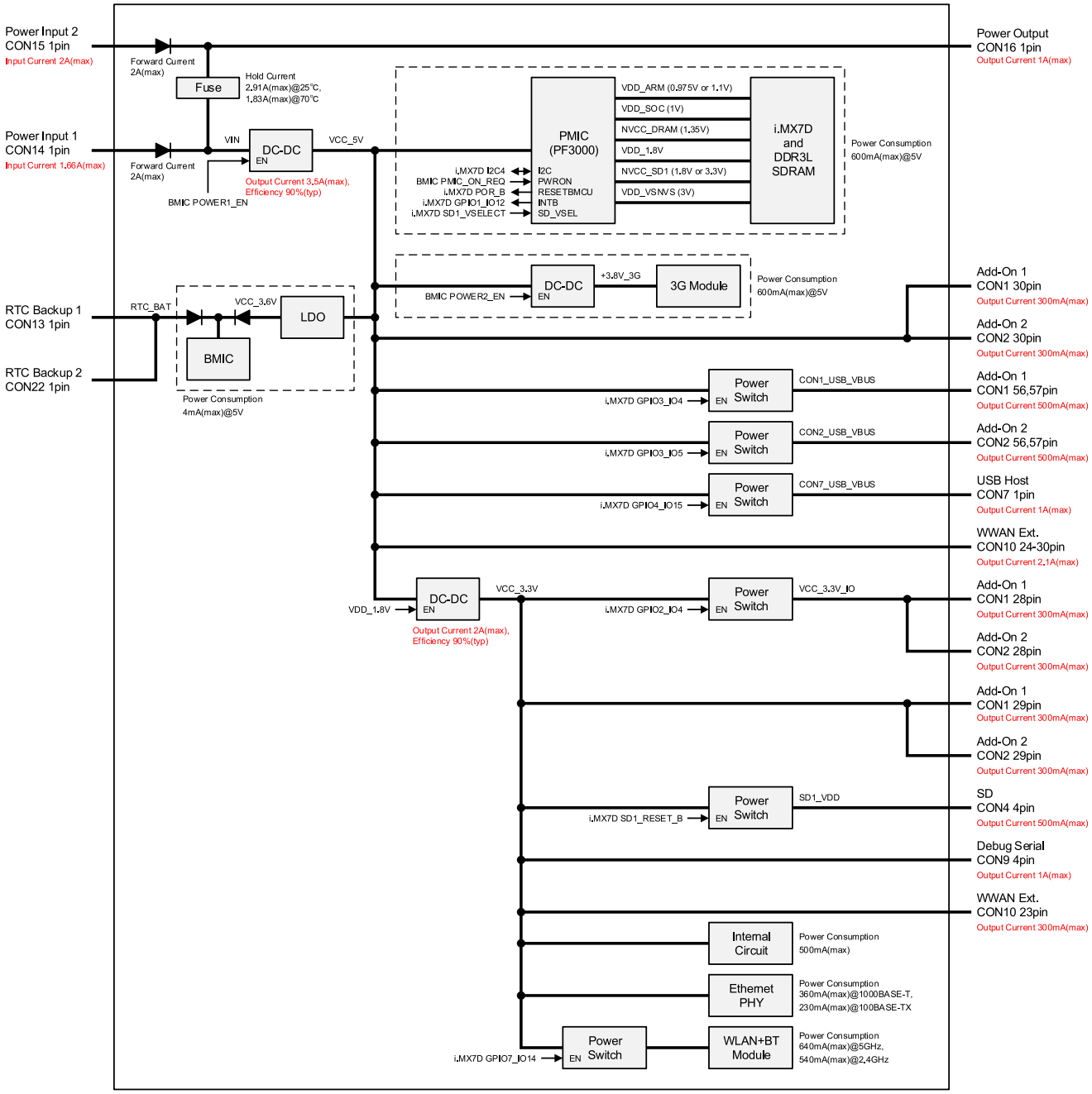


Figure 10.1. Power Supply Circuit Structure

Chapter 11. Interface Specifications

The following explains the interface specifications of Armadillo-IoT.

11.1. Interface Layout

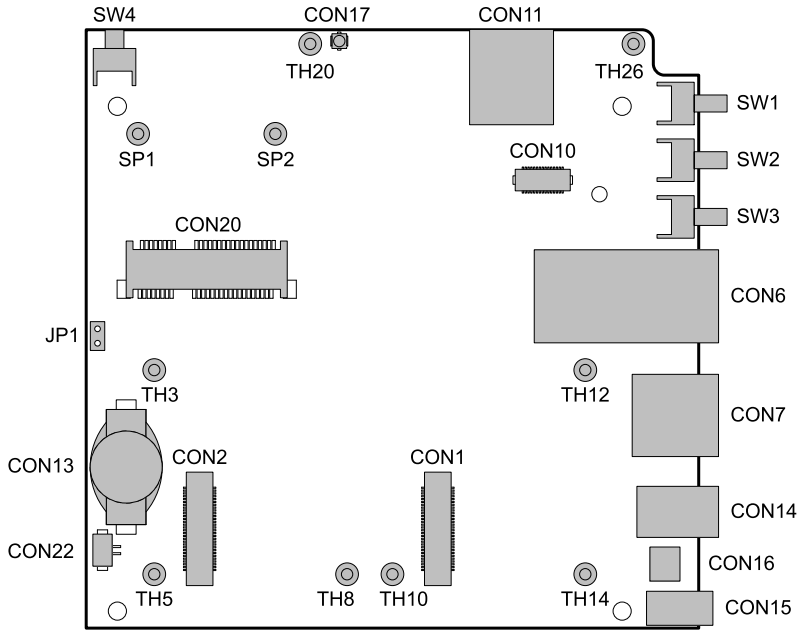


Figure 11.1. Armadillo-IoT Interface Layout (Side A)

Table 11.1. Armadillo-IoT Interface List (Side A)

Part Number	Interface Name	Product Number	Maker
CON1	Add-on Interface 1	DF17(4.0)-60DS-0.5V(57)	HIROSE ELECTRIC
CON2	Add-on Interface 2	DF17(4.0)-60DS-0.5V(57)	HIROSE ELECTRIC
CON6	LAN Interface	9771-8813-S3L6T1	SUN JUN ELECTRONICS
CON7	USB Host Interface	UBA-4R-D14T-4D(LF)(SN)	J.S.T. Mfg.
CON10[a]	WWAN Expansion Interface	DF12(3.0)-30DP-0.5V(86)	HIROSE ELECTRIC
CON11[a]	MicroSIM Interface	CIM-J78	MITSUMI
CON13	RTC Backup Interface 1	SMTU1220-LF	Renata SA
CON14	Power Input Interface 1	PJ-102AH	CUI
CON15	Power Input Interface 2	S02B-PASK-2(LF)(SN)	J.S.T. Mfg.
CON16	Power Output Interface	B02B-PASK(LF)(SN)	J.S.T. Mfg.
CON17[a]	3G Antenna Interface	U.FL-R-SMT-1	HIROSE ELECTRIC
CON20	WLAN Interface	MM60-52B1-E1-R650	Japan Aviation Electronics Industry
CON22	RTC Backup Interface 2	DF13C-2P-1.25V (21)	HIROSE ELECTRIC
SW1	User Switch 1	SKHHLRA010	ALPS ELECTRIC
SW2	User Switch 2	SKHHLRA010	ALPS ELECTRIC
SW3	User Switch 3	SKHHLRA010	ALPS ELECTRIC
SW4	Reset Switch	SKHHLUA010	ALPS ELECTRIC
JP1	Boot Device Configuration Jumper	A2-2PA-2.54DSA(71)	HIROSE ELECTRIC

Part Number	Interface Name	Product Number	Maker
TH3	Stud for Add-on Module	KRB-2008	Hirosugi-Keiki
TH5			
TH8			
TH10			
TH12			
TH14			
TH20[a]	Stud for WWAN Module	TH-1.6-3.0-M2	Mac-Eight
TH26[a]			
SP1	Stud for Wireless LAN Module	NT4R1600	Japan Aviation Electronics Industry
SP2			

[a]Some models are not equipped with these parts.

The is no guarantee that the parts installed have the part numbers shown in Table 11.1, “Armadillo-IoT Interface List (Side A)”. Please check the parts specifications and change history table which can be downloaded from the Atmark Techno Users' site for the parts on your product.

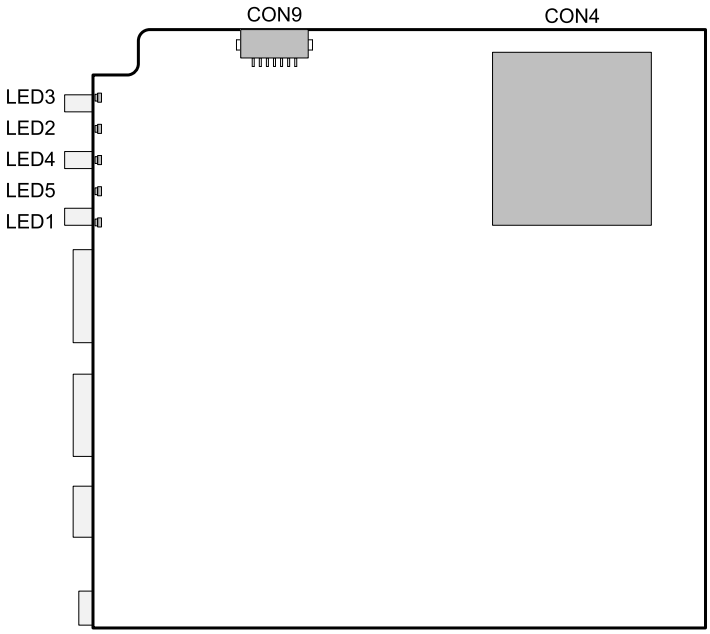


Figure 11.2. Interface Layout (Side B)

Table 11.2. Armadillo-IoT Interface List (Side B)

Part Number	Interface Name	Product Number	Maker
CON4	SD Interface	DM1B-DSF-PEJ(82)	HIROSE ELECTRIC
CON9	Debug Serial Interface	DF13A-7P-1.25H(51)	HIROSE ELECTRIC
LED1	WWAN LED	SML-A12P8T	ROHM
LED2	User LED2	SML-A12P8T	ROHM
LED3	User LED1	SML-A12P8T	ROHM
LED4	User LED3	SML-A12P8T	ROHM
LED5	User LED4	SML-A12P8T	ROHM

The is no guarantee that the parts installed have the part numbers shown in Table 11.2, “Armadillo-IoT Interface List (Side B)”. Please check the parts specifications and change history table which can be downloaded from the Atmark Techno Users' site for the parts on your product.

11.2. CON1 Add-on Interface 1

CON1 is an expansion interface. It connects to signal lines of the i.MX7 Dual which offer multiple functions (multiplexing), and functions such as USB, UART, SPI, I2C, SD, CAN, PWM and GPIO can be interfaced to.

Mounted Connector DF17(4.0)-60DS-0.5V(57)/HIROSE ELECTRIC

Matching Connector Example DF17(4.0)-60DP-0.5V(57)/HIROSE ELECTRIC

Allowable Current 0.3A (per one terminal)

Table 11.3. CON1 Signals

Pin Number	Pin Name	I/O	Description
1	GND	Power	Power (GND)
2	GND	Power	Power (GND)
3	GPIO3_IO0	In/Out	Expansion I/O, connected to LCD_CLK pin of i.MX7 Dual Shared connection with pin 41 of CON2
4	GPIO3_IO1	In/Out	Expansion I/O, connected to LCD_ENABLE pin of i.MX7 Dual Shared connection with pin 40 of CON2
5	GPIO3_IO2	In/Out	Expansion I/O, connected to LCD_HSYNC pin of i.MX7 Dual Shared connection with pin 39 of CON2
6	GPIO3_IO3	In/Out	Expansion I/O, connected to LCD_VSYNC pin of i.MX7 Dual Shared connection with pin 38 of CON2
7	GPIO4_IO2	In/Out	Expansion I/O, connected to UART2_RXD pin of i.MX7 Dual Shared connection with pin 32 of CON2
8	GPIO4_IO3	In/Out	Expansion I/O, connected to UART2_TXD pin of i.MX7 Dual Shared connection with pin 34 of CON2
9	GPIO4_IO4	In/Out	Expansion I/O, connected to UART3_RXD pin of i.MX7 Dual Shared connection with pin 36 of CON2
10	GPIO4_IO5	In/Out	Expansion I/O, connected to UART3_TXD pin of i.MX7 Dual Shared connection with pin 37 of CON2
11	GPIO4_IO6	In/Out	Expansion I/O, connected to UART3_RTS pin of i.MX7 Dual Shared connection with pin 35 of CON2
12	GPIO4_IO7	In/Out	Expansion I/O, connected to UART3_CTS pin of i.MX7 Dual Shared connection with pin 50 of CON2
13	GPIO4_IO10	In/Out	Expansion I/O, connected to I2C2_SCL pin of i.MX7 Dual Shared connection with pin 24 of CON2
14	GPIO4_IO11	In/Out	Expansion I/O, connected to I2C2_SDA pin of i.MX7 Dual Shared connection with pin 25 of CON2
15	GPIO5_IO11	In/Out	Expansion I/O, connected to SD2_RESET_B pin of i.MX7 Dual Shared connection with pin 33 of CON2
16	GPIO6_IO19	In/Out	Expansion I/O, connected to SAI2_TXFS pin of i.MX7 Dual Shared connection with pin 46 of CON2
17	GPIO6_IO20	In/Out	Expansion I/O, connected to SAI2_TXC pin of i.MX7 Dual Shared connection with pin 47 of CON2

Pin Number	Pin Name	I/O	Description
18	GPIO6_IO22	In/Out	Expansion I/O, connected to SAI2_TXD pin of i.MX7 Dual Shared connection with pin 49 of CON2
19	GPIO6_IO21	In/Out	Expansion I/O, connected to SAI2_RXD pin of i.MX7 Dual Shared connection with pin 48 of CON2
20	GPIO4_IO8	In/Out	Expansion I/O, connected to I2C1_SCL pin of i.MX7 Dual Shared connection with pin 20 of CON2
21	GPIO4_IO9	In/Out	Expansion I/O, connected to I2C1_SDA pin of i.MX7 Dual Shared connection with pin 21 of CON2
22	GPIO4_IO12	In/Out	Expansion I/O, connected to I2C3_SCL pin of i.MX7 Dual Shared connection with pin 42 of CON2
23	GPIO4_IO13	In/Out	Expansion I/O, connected to I2C3_SDA pin of i.MX7 Dual Shared connection with pin 43 of CON2
24	GPIO3_IO25	In/Out	Expansion I/O, connected to LCD_DATA20 pin of i.MX7 Dual
25	GPIO3_IO26	In/Out	Expansion I/O, connected to LCD_DATA21 pin of i.MX7 Dual
26	GND	Power	Power (GND)
27	GND	Power	Power (GND)
28	VCC_3.3V_IO	Power	Power (VCC_3.3V_IO)
29	VCC_3.3V	Power	Power (VCC_3.3V)
30	VCC_5V	Power	Power (VCC_5V)
31	DETECT_CON1	Out	Connected to GND with 1kΩ resistor
32	GPIO7_IO12	In/Out	Expansion I/O, connected to ENET1_TX_CLK pin of i.MX7 Dual
33	GPIO7_IO13	In/Out	Expansion I/O, connected to ENET1_RX_CLK pin of i.MX7 Dual
34	GPIO7_IO7	In/Out	Expansion I/O, connected to ENET1_TD1 pin of i.MX7 Dual
35	GPIO7_IO2	In/Out	Expansion I/O, connected to ENET1_RD2 pin of i.MX7 Dual
36	GPIO7_IO8	In/Out	Expansion I/O, connected to ENET1_TD2 pin of i.MX7 Dual
37	GPIO7_IO3	In/Out	Expansion I/O, connected to ENET1_RD3 pin of i.MX7 Dual
38	GPIO7_IO0	In/Out	Expansion I/O, connected to ENET1_RD0 pin of i.MX7 Dual
39	GPIO7_IO1	In/Out	Expansion I/O, connected to ENET1_RD1 pin of i.MX7 Dual
40	GPIO4_IO1	In/Out	Expansion I/O, connected to UART1_TXD pin of i.MX7 Dual
41	GPIO4_IO0	In/Out	Expansion I/O, connected to UART1_RXD pin of i.MX7 Dual
42	GPIO7_IO4	In/Out	Expansion I/O, connected to ENET1_RX_CTL pin of i.MX7 Dual
43	GPIO7_IO5	In/Out	Expansion I/O, connected to ENET1_RXC pin of i.MX7 Dual
44	GPIO7_IO6	In/Out	Expansion I/O, connected to ENET1_TD0 pin of i.MX7 Dual
45	GPIO7_IO10	In/Out	Expansion I/O, connected to ENET1_TX_CTL pin of i.MX7 Dual
46	GPIO7_IO11	In/Out	Expansion I/O, connected to ENET1_TXC pin of i.MX7 Dual
47	GPIO5_IO12	In/Out	Expansion I/O, connected to SD2_CLK pin of i.MX7 Dual
48	GPIO5_IO13	In/Out	Expansion I/O, connected to SD2_CMD pin of i.MX7 Dual
49	GPIO5_IO14	In/Out	Expansion I/O, connected to SD2_DATA0 pin of i.MX7 Dual
50	GPIO7_IO9	In/Out	Expansion I/O, connected to ENET1_TD3 pin of i.MX7 Dual
51	GPIO5_IO15	In/Out	Expansion I/O, connected to SD2_DATA1 pin of i.MX7 Dual
52	GPIO5_IO16	In/Out	Expansion I/O, connected to SD2_DATA2 pin of i.MX7 Dual
53	GPIO5_IO17	In/Out	Expansion I/O, connected to SD2_DATA3 pin of i.MX7 Dual
54	GND	Power	Power (GND)
55	PMIC_ONOFF	In	The power management IC on/off signal. Please input an open drain (or open collector) signal.
56	CON1_USB_VBUS	Power	Power (CON1_USB_VBUS)
57	CON1_USB_VBUS	Power	Power (CON1_USB_VBUS)
58	GND	Power	Power (GND)
59	CON1_USB_HS_DP	In/Out	CON1_USB plus side signal, connected to USB_OTG2_DP pin of i.MX7 Dual
60	CON1_USB_HS_DM	In/Out	CON1_USB minus side signal, connected to USB_OTG2_DN pin of i.MX7 Dual

There are a number of add-on modules available which can be connected.

There are many signals which connect to both CON1 and CON2. Please be careful of possible signal conflicts when designing an expansion board.

11.3. CON2 Add-on Interface 2

CON2 is an expansion interface. It connects to signal lines of the i.MX7 Dual which offer multiple functions (multiplexing), and functions such as USB, UART, SPI, I2C, and GPIO can be interfaced to.

Mounted Connector	DF17(4.0)-60DS-0.5V(57)/HIROSE ELECTRIC
Matching Connector Example	DF17(4.0)-60DP-0.5V(57)/HIROSE ELECTRIC
Allowable Current	0.3A (per one terminal)

Table 11.4. CON2 Signals

Pin Number	Pin Name	I/O	Description
1	GND	Power	Power (GND)
2	GND	Power	Power (GND)
3	NC	-	Not Connected
4	NC	-	Not Connected
5	NC	-	Not Connected
6	NC	-	Not Connected
7	NC	-	Not Connected
8	NC	-	Not Connected
9	NC	-	Not Connected
10	NC	-	Not Connected
11	NC	-	Not Connected
12	NC	-	Not Connected
13	NC	-	Not Connected
14	NC	-	Not Connected
15	NC	-	Not Connected
16	NC	-	Not Connected
17	NC	-	Not Connected
18	NC	-	Not Connected
19	NC	-	Not Connected
20	GPIO4_IO8	In/Out	Expansion I/O, connected to I2C1_SCL pin of i.MX7 Dual Shared connection with pin 20 of CON1
21	GPIO4_IO9	In/Out	Expansion I/O, connected to I2C1_SDA pin of i.MX7 Dual Shared connection with pin 21 of CON1
22	NC	-	Not Connected
23	NC	-	Not Connected

Pin Number	Pin Name	I/O	Description
24	GPIO4_IO10	In/Out	Expansion I/O, connected to I2C2_SCL pin of i.MX7 Dual Shared connection with pin 13 of CON1
25	GPIO4_IO11	In/Out	Expansion I/O, connected to I2C2_SDA pin of i.MX7 Dual Shared connection with pin 14 of CON1
26	GND	Power	Power (GND)
27	GND	Power	Power (GND)
28	VCC_3.3V_IO	Power	Power (VCC_3.3V_IO)
29	VCC_3.3V	Power	Power (VCC_3.3V)
30	VCC_5V	Power	Power (VCC_5V)
31	DETECT_CON2	Out	Connected to VCC_3.3 V_IO with 1kΩ resistor
32	GPIO4_IO2	In/Out	Expansion I/O, connected to UART2_RXD pin of i.MX7 Dual Shared connection with pin 7 of CON1
33	GPIO5_IO11	In/Out	Expansion I/O, connected to SD2_RESET_B pin of i.MX7 Dual Shared connection with pin 15 of CON1
34	GPIO4_IO2	In/Out	Expansion I/O, connected to UART2_TXD pin of i.MX7 Dual Shared connection with pin 8 of CON1
35	GPIO4_IO6	In/Out	Expansion I/O, connected to UART3_RTS pin of i.MX7 Dual Shared connection with pin 11 of CON1
36	GPIO4_IO4	In/Out	Expansion I/O, connected to UART3_RXD pin of i.MX7 Dual Shared connection with pin 9 of CON1
37	GPIO4_IO5	In/Out	Expansion I/O, connected to UART3_TXD pin of i.MX7 Dual Shared connection with pin 10 of CON1
38	GPIO3_IO3	In/Out	Expansion I/O, connected to LCD_VSYNC pin of i.MX7 Dual Shared connection with pin 6 of CON1
39	GPIO3_IO2	In/Out	Expansion I/O, connected to LCD_HSYNC pin of i.MX7 Dual Shared connection with pin 5 of CON1
40	GPIO3_IO1	In/Out	Expansion I/O, connected to LCD_ENABLE pin of i.MX7 Dual Shared connection with pin 4 of CON1
41	GPIO3_IO0	In/Out	Expansion I/O, connected to LCD_CLK pin of i.MX7 Dual Shared connection with pin 3 of CON1
42	GPIO4_IO12	In/Out	Expansion I/O, connected to I2C3_SCL pin of i.MX7 Dual Shared connection with pin 22 of CON1
43	GPIO4_IO13	In/Out	Expansion I/O, connected to I2C3_SDA pin of i.MX7 Dual Shared connection with pin 23 of CON1
44	NC	-	Not Connected
45	NC	-	Not Connected
46	GPIO6_IO19	In/Out	Expansion I/O, connected to SAI2_TXFS pin of i.MX7 Dual Shared connection with pin 16 of CON1
47	GPIO6_IO20	In/Out	Expansion I/O, connected to SAI2_TXC pin of i.MX7 Dual Shared connection with pin 17 of CON1
48	GPIO6_IO21	In/Out	Expansion I/O, connected to SAI2_RXD pin of i.MX7 Dual Shared connection with pin 19 of CON1
49	GPIO6_IO22	In/Out	Expansion I/O, connected to SAI2_TXD pin of i.MX7 Dual Shared connection with pin 18 of CON1
50	GPIO4_IO7	In/Out	Expansion I/O, connected to UART3_CTS pin of i.MX7 Dual Shared connection with pin 12 of CON1
51	NC	-	Not Connected
52	NC	-	Not Connected
53	NC	-	Not Connected
54	GND	Power	Power (GND)
55	PMIC_ONOFF	In	The power management IC on/off signal. Please input an open drain (or open collector) signal.
56	CON2_USB_VBUS	Power	Power (CON2_USB_VBUS)
57	CON2_USB_VBUS	Power	Power (CON2_USB_VBUS)
58	GND	Power	Power (GND)

Pin Number	Pin Name	I/O	Description
59	CON2_USB_HS_DP	In/Out	CON2_USB plus side signal, connected to i.MX7 Dual USB HSIC controller via USB HUB controller (port 1)
60	CON2_USB_HS_DM	In/Out	CON2_USB minus side signal, connected to i.MX7 Dual USB HSIC controller via USB HUB controller (port 1)

There are a number of add-on modules available which can be connected.

There are many signals which connect to both CON1 and CON2. Please be careful of possible signal conflicts when designing an expansion board.

11.4. CON4 SD Interface

CON 4 is an SD interface compatible with UHS-I (SDR104, maximum clock frequency: 196.36MHz). The SD signal lines are connected to the SD/MMC controller (uSDHC1) on the i.MX7 Dual.

The power (SD_VDD) supplied to the SD card can be controlled with the SD1_RESET_B pin of the i.MX7 Dual. Power is supplied during high level output and power is turned off with low level output.

Mounted Connector DM1B-DSF-PEJ(82)/HIROSE ELECTRIC

Table 11.5. CON4 Signals

Pin Number	Pin Name	I/O	Description
1	SD_DAT3	In/Out	SD data bus (bit 3), connected to SD1_DATA3 pin of i.MX7 Dual
2	SD_CMD	In/Out	SD command / response, connected to SD1_CMD pin of i.MX7 Dual
3	GND	Power	Power (GND)
4	SD_VDD	Power	Power (SD_VDD)
5	CLK	Out	SD clock, connected to SD1_CLK pin of i.MX7 Dual
6	GND	Power	Power (GND)
7	SD_DAT0	In/Out	SD data bus (bit 0), connected to SD1_DATA0 pin of i.MX7 Dual
8	SD_DAT1	In/Out	SD data bus (bit 1), connected to SD1_DATA1 pin of i.MX7 Dual
9	SD_DAT2	In/Out	SD data bus (bit 2), connected to SD1_DATA2 pin of i.MX7 Dual

Table 11.6. CON4 Card Detection and Write Protect Detection

Item	Description
Card detection	Connected to SD1_CD_B pin of i.MX7 Dual (Low: card inserted, High: card removed)
Write protect detection	Connected to SD1_WP pin of i.MX7 Dual (Low: write enabled, High: write disabled)

Please do not apply excessive external force to the connector as it may cause it to break or be damaged.

11.5. CON6 LAN Interface

CON6 is a LAN interface compatible with 10BASE-T / 100BASE-TX / 1000BASE-T. Ethernet cables of category 5e or higher can be connected. AUTO-MDIX is supported, automatically detecting a straight cable or crossover cable and switching the transmission / reception terminals as necessary.

The signal lines are connected to the Ethernet MAC (ENET2) of the i.MX7 Dual via the Ethernet PHY (VSC8501XML-03/Microsemi).

Mounted Connector 9771-8813-S3L6T1/SUN JUN ELECTRONICS

Table 11.7. CON6 Signals (10BASE-T/100BASE-TX)

Pin Number	Pin Name	I/O	Description
1	TX+	In/Out	Transmit data +
2	TX-	In/Out	Transmit data -
3	RX+	In/Out	Receive data +
4	-	-	
5	-	-	
6	RX-	In/Out	Receive data -
7	-	-	
8	-	-	

Table 11.8. CON6 Signals (1000BASE-T)

Pin Number	Pin Name	I/O	Description
1	TRD0+	In/Out	Transmit / receive data 0 +
2	TRD0-	In/Out	Transmit / receive data 0 -
3	TRD1+	In/Out	Transmit / receive data 1 +
4	TRD2+	In/Out	Transmit / receive data 2 +
5	TRD2-	In/Out	Transmit / receive data 2 -
6	TRD1-	In/Out	Transmit / receive data 1 -
7	TRD3+	In/Out	Transmit / receive data 3 +
8	TRD3-	In/Out	Transmit / receive data 3 -

Table 11.9. LAN Connector LEDs

Name	Status	Description
LINK_ACTIVITY_LED	Off	Link not established
	On (yellow)	Link established
	Blinking (yellow)	Link established and transmitting / receiving data
SPEED_LED	Off	Connected at 10Mbps
	On (green)	Connected at 100Mbps
	On (amber)	Connected at 1000Mbps

11.6. CON7 USB Host Interface

CON7 is a USB 2.0 host interface. The signal lines are connected to the USB controller (OTG1) on the i.MX7 Dual.

The power (CON7_USB_VBUS) supplied to the USB device can be controlled with the I2C4_SDA (GPIO4_IO15) pin of the i.MX7 Dual. Power is supplied with a high level output and power is turned off with a low level output.

Data Transmission Mode

- High Speed (480Mbps)
- Full Speed (12Mbps)
- Low Speed (1.5Mbps)

Mounted Connector UBA-4R-D14T-4D/J.S.T. Mfg.

Table 11.10. CON7 Signals

Pin Number	Pin Name	I/O	Description
1	CON7_USB_VBUS	Power	Power (CON7_USB_VBUS)
2	CON7_USB_HS_DM	In/Out	USB minus side signal, connected to USB_OTG1_DN pin of i.MX7 Dual
3	CON7_USB_HS_DP	In/Out	USB plus side signal, connected to USB_OTG1_DP pin of i.MX7 Dual
4	GND	Power	Power (GND)

11.7. CON9 Debug Serial Interface

CON9 is a serial interface for debugging. It is connected to the UART controller (UART5) on the i.MX7 Dual.

Pin 6 of CON6 is connected to the GPIO1_09 pin of the i.MX7 Dual. Maintenance mode is entered on a low level input and operating system auto boot mode on a high level input.

Mounted Connector DF13A-7P-1.25H(51)/HIROSE ELECTRIC

Matching Connector Example DF13-7S-1.25C/HIROSE ELECTRIC (Housing)
DF13-2630SCFA/HIROSE ELECTRIC (Contact)

Signal Level 3.3V CMOS

Allowable Current 1A (per terminal)

Table 11.11. CON9 Signals

Pin Number	Pin Name	I/O	Description
1	DEBUG_UART_RXD	In	Data receive, connected to GPIO1_IO06 pin of i.MX7 Dual
2	GND	Power	Power (GND)
3	DEBUG_UART_TXD	Out	Data transmit, connected to GPIO1_IO07 pin of i.MX7 Dual
4	VCC_3.3V	Power	Power (VCC_3.3V)
5	DEBUG_UART_CTS	In	Clear To Send, connected to GPIO1_IO05 pin of i.MX7 Dual
6	BOOTLOADER_EN_B	In	Boot mode setting, connected to GPIO1_IO09 pin of i.MX7 Dual (Low: Maintenance mode, High: Operating system auto boot mode)
7	DEBUG_UART_RTS	Out	Request To Send, connected to GPIO1_IO04 pin of i.MX 7 Dual

11.8. CON10 WWAN Expansion Interface

CON 10 is a WWAN expansion interface.

The USB signal lines are connected to the USB HSIC controller via the USB HUB controller (port 2), and the UART signal lines are connected to the UART controller (UART7) of the i.MX7 Dual.

Mounted Connector DF12(3.0)-30DP-0.5V(86)/HIROSE ELECTRIC

Matching Connector Example DF12(3.0)-30DS-0.5V(86)/HIROSE ELECTRIC

Allowable Current 0.3A (per one terminal)

Table 11.12. CON10 Signals

Pin Number	Pin Name	I/O	Description
1	GND	Power	Power (GND)
2	GND	Power	Power (GND)
3	GND	Power	Power (GND)
4	GND	Power	Power (GND)
5	GND	Power	Power (GND)
6	GND	Power	Power (GND)
7	GND	Power	Power (GND)
8	-	-	Reserved
9	LED1_CTRL[a]	In	LED1 control, connected to LED1 (Low: off, High: on)
10	GPIO4_IO19	In/Out	General-purpose I/O, connected to ECSPI1_SS0 pin of i.MX7 Dual
11	GPIO4_IO16	In/Out	General-purpose I/O, connected to ECSPI1_SCLK pin of i.MX7 Dual
12	WWAN_UART_RTS	Out	Request To Send, connected to ECSPI2_SS0 pin of i.MX 7 Dual
13	WWAN_UART_CTS	In	Clear To Send, connected to ECSPI2_MISO pin of i.MX7 Dual
14	WWAN_UART_TXD	Out	Data transmit, connected to ECSPI2_MOSI pin of i.MX7 Dual
15	WWAN_UART_RXD	In	Data receive, connected to ECSPI2_MOSI pin of i.MX7 Dual
16	GND	Power	Power (GND)
17	WWAN_USB_HS_DM	In/Out	WWAN_USB minus side signal, connected to USB HSIC controller of i.MX7 Dual via USB HUB controller (port 2)
18	WWAN_USB_HS_DP	In/Out	WWAN_USB plus side signal, connected to USB HSIC controller of i.MX7 Dual via USB HUB controller (port 2)
19	GND	Power	Power (GND)
20	WWAN_PWR_ON	Out	WWAN module power control, connected to ECSPI1_MOSI pin of i.MX7 Dual
21	WWAN_RESET	Out	WWAN module reset, connected to ECSPI1_MISO pin of i.MX7 Dual
22	-	-	Reserved
23	VCC_3.3V	Power	Power (VCC_3.3V)
24	VCC_5V	Power	Power (VCC_5V)
25	VCC_5V	Power	Power (VCC_5V)
26	VCC_5V	Power	Power (VCC_5V)
27	VCC_5V	Power	Power (VCC_5V)
28	VCC_5V	Power	Power (VCC_5V)
29	VCC_5V	Power	Power (VCC_5V)
30	VCC_5V	Power	Power (VCC_5V)

[a]LED1_CTRL can also be controlled with +1.8V. For details, refer to Table 10.5, “WWAN Expansion Interface Pin 9 (LED1_CTRL) Electrical Specifications”.

11.9. CON11 MicroSIM Interface

CON11 is a MicroSIM interface for the 3G Module (PDS6 / Gemalto).

Table 11.13. CON11 Signals

Pin Number	Pin Name	I/O	Description
1	GND	Power	Power (GND)
2	SIM_VCC	Power	SIM power, connected to CCVCC pin of 3G module
3	SIM_RST	Out	SIM reset, connected to CCRST pin of 3G module
4	SIM_CLK	Out	SIM clock, connected to CCCLK pin of 3G module
5	SIM_VPP	-	Not Connected
6	SIM_I/O	In	SIM data, connected to CCIO pin of 3G module

CON11 does not support hot swapping. Please remove the SIM card after having turned off the power of this product.

11.10. CON13 RTC Backup Interface 1

CON13 is a backup interface for real-time clock functionality. Please use it when you want to retain time data while power is off for an extended period.

Mounted Connector SMTU1220-LF/Renata SA

Compatible Battery Example CR1220, BR1220, etc.

Table 11.14. CON13 Signals

Pin Number	Pin Name	I/O	Description
1	RTC_BAT1	Power	Power supply input for backup of real-time clock
2	GND	Power	Power (GND)

The average monthly gap of the real-time clock differs depending on the RTC version in the board information. The relationship between the versions and the average monthly difference (reference value) at an ambient temperature of 25°C is shown below.

Table 11.15. RTC Versions and Average Monthly Difference (Reference Value) at 25°C Ambient Temperature

RTC Version	Average Monthly Difference at 25°C Ambient Temperature
1.0	Approx. 300 seconds
1.1 or later	Approx 90 seconds

The RTC version can be checked with the following command.

```
[armadillo ~]# dmesg | grep bmic_rtc
[    1.639111] bmic_rtc 3-0011: version: 1.1
```

Even if the RTC version is 1.0, you can update to RTC version 1.1 or later by applying installation disk v20161026_iotg3 or later (install_disk_sd_20161026_iotg3.img or later). For details on how to use the installation disc, please refer to Section 8.1, “Using the Installation Disc”.

11.11. CON14 Power Input Interface 1

CON14 is a power supply interface. AC adapters with the same polarity mark as shown in Figure 11.3, “AC Adapter Polarity Mark” may be used.

Mounted Connector PJ-102AH/CUI

Table 11.16. CON14 Signals

Pin Number	Pin Name	I/O	Description
1	VIN	Power	Power input (VIN)
2	GND	Power	Power (GND)
3	GND	Power	Power (GND)

Figure 11.3. AC Adapter Polarity Mark

When using CON14 do not supply power from CON15 at the same time. It may cause damage.

When using the AC adapter, connect the DC plug of the AC adapter to the Armadillo-IoT first and then plug the AC plug into the outlet.

When using the standard AC adapter (OP-AC12V2-00) or the wide temperature range AC adapter (OP-AC12V3-00)[1] when re-powering, wait at least 3 seconds before turning on the power.

When turning on the power using a power supply unit other than the above AC adapter, please do so with a monotonically increasing supply. Also, when re-powering, make sure to first turn the power off (5V or less for at least 10ms) and then turn it on again.

11.12. CON15 Power Input Interface 2

CON15 is a power supply interface.

Mounted Connector	S02B-PASK-2(LF)(SN)/J.S.T. Mfg.
Matching Connector	PAP-02V-S/J.S.T. Mfg. (Housing)
Example	SPHD-001T-P0.5/J.S.T. Mfg. (Contact)

Table 11.17. CON15 Signals

Pin Number	Pin Name	I/O	Description
1	VIN	Power	Power input (VIN)
2	GND	Power	Power (GND)

[1] Armadillo-IoT Gateway development set accessories

When using CON15 do not supply power from CON14 at the same time. It may cause damage.

When using this connector to turn on the power supply, please start it with a monotonically increasing supply. Also, when re-powering, make sure to first turn the power off (5V or less for at least 10ms) and then turn it on again.

11.13. CON16 Power Output Interface

CON16 is a power output interface.

Mounted Connector	B02B-PASK(LF)(SN)/J.S.T. Mfg.
Matching Connector Example	PAP-02V-S/J.S.T. Mfg. (Housing) SPHD-001T-P0.5/J.S.T. Mfg. (Contact)

Table 11.18. CON16 Signals

Pin Number	Pin Name	I/O	Description
1	VOUT	Power	Power (VOUT)
2	GND	Power	Power (GND)

11.14. CON17 3G Antenna Interface

CON17 is an interface for the 3G module (PDS6 / Gemalto) antenna.

Mounted Connector	U.FL-R-SMT-1/HIROSE ELECTRIC

When connecting the antenna cable to the antenna terminal, please be careful not to apply excessive force it may cause damage.

11.15. CON20 WLAN Interface

CON20 is an interface for the Wireless LAN Module (AEH-AR9462 / VoxMicro).

The PCI Express signal lines are connected to the PCI Express PHY (PCIe_PHY) on the i.MX7 Dual, and the USB signal lines are connected via the USB HUB controller (port 3) to the USB HSIC controller of the i.MX7 Dual.

The power (WLAN_VDD) supplied to the wireless LAN module can be controlled with the ENET1_CRS (GPIO7_IO14) pin on the i.MX7 Dual. Power is supplied with a high level output and power is turned off with a low level output.

Mounted Connector	MM60-52B1-E1-R650/Japan Aviation Electronics Industry

Table 11.19. CON20 Signals

Pin Number	Pin Name	I/O	Description
1	-	-	Reserved
2	WLAN_VDD	Power	Power (WLAN_VDD)
3	-	-	Reserved
4	GND	Power	Power (GND)
5	BT_DISABLE_L	Out	BT enable / disable signal, connected to SAI1_RXFS pin of i.MX7 Dual (Low: BT disabled, High: BT enabled)
6	-	-	Reserved
7	CLKREQ_L	In	Reference clock request, connected to SAI1_RXC pin of i.MX7 Dual
8	NC	-	Not Connected
9	GND	Power	Power (GND)
10	NC	-	Not Connected
11	REFCLK-	Out	Differential reference clock (-), connected to PCIE_REFCLKOUT_N pin of i.MX7 Dual
12	NC	-	Not Connected
13	REFCLK+	Out	Differential reference clock (+), connected to PCIE_REFCLKOUT_P pin of i.MX7 Dual
14	NC	-	Not Connected
15	GND	Power	Power (GND)
16	NC	-	Not Connected
17	NC	-	Not Connected
18	GND	Power	Power (GND)
19	NC	-	Not Connected
20	W_DISABLE_L	Out	Wireless LAN enable / disable signal, connected to SAI1_TXC pin of i.MX7 Dual (Low: Wireless LAN disabled, High: Wireless LAN enabled)
21	GND	Power	Power (GND)
22	PERST_L	Out	Basic reset signal, connected to SAI1_RXD pin of i.MX7 Dual (Low: Reset state, High: Reset release)
23	PERn0	In	Differential receiver (-), connected to PCIE_RX_N pin of i.MX7 Dual
24	WLAN_VDD	Power	Power (WLAN_VDD)
25	PERp0	In	Differential receiver (+), connected to PCIE_RX_P pin of i.MX7 Dual
26	GND	Power	Power (GND)
27	GND	Power	Power (GND)
28	-	-	Reserved
29	GND	Power	Power (GND)
30	-	-	Reserved
31	PETn0	Out	Differential transmitter (-), connected to PCIE_TX_N pin of i.MX7 Dual
32	-	-	Reserved
33	PETp0	Out	Differential transmitter (+), connected to PCIE_TX_P pin of i.MX7 Dual
34	GND	Power	Power (GND)
35	GND	Power	Power (GND)
36	WLAN_USB_HS_DM	In/Out	Minus side signal of WLAN_USB, connected to USB HSIC controller on i.MX7 Dual via USB HUB controller (port 3)
37	GND	Power	Power (GND)
38	WLAN_USB_HS_DP	In/Out	Plus side signal of WLAN_USB, connected to USB HSIC controller of i.MX7 Dual via USB HUB controller (port 3)
39	WLAN_VDD	Power	Power (WLAN_VDD)
40	GND	Power	Power (GND)
41	WLAN_VDD	Power	Power (WLAN_VDD)
42	NC	-	Not Connected
43	GND	Power	Power (GND)
44	-	-	Reserved
45	NC	-	Not Connected
46	-	-	Reserved

Pin Number	Pin Name	I/O	Description
47	NC	-	Not Connected
48	-	-	Reserved
49	NC	-	Not Connected
50	GND	Power	Power (GND)
51	NC	-	Not Connected
52	WLAN_VDD	Power	Power (WLAN_VDD)

11.16. CON22 RTC Backup Interface 2

CON22 is an external backup interface for real-time clock functionality. Please use it when you want to retain time data while power is off for an extended period.

Mounted Connector	DF13C-2P-1.25V(21)/HIROSE ELECTRIC
Matching Connector Example	DF13-2S-1.25C/HIROSE ELECTRIC (Housing) DF13-2630SCF/HIROSE ELECTRIC (Contact)
Allowable Current	1A (per terminal)
Compatible Battery Example	CR2032 WK11/Hitachi Maxell[2] etc

Table 11.20. CON22 Signals

Pin Number	Pin Name	I/O	Description
1	RTC_BAT2	Power	Real-time clock external backup power in
2	GND	Power	Power (GND)

The average monthly gap of the real-time clock differs depending on the RTC version in the board information. The relationship between the versions and the average monthly difference (reference value) at an ambient temperature of 25°C is shown below.

Table 11.21. RTC Versions and Average Monthly Difference (Reference Value) at 25°C Ambient Temperature

RTC Version	Average Monthly Difference at 25°C Ambient Temperature
1.0	Approx. 300 seconds
1.1 or later	Approx 90 seconds

The RTC version can be checked with the following command.

```
[armadillo ~]# dmesg | grep bmic_rtc
[    1.639111] bmic_rtc 3-0011: version: 1.1
```

Even if the RTC version is 1.0, you can update to RTC version 1.1 or later by applying installation disk v20161026_iotg3 or later (in-

[2]For more information, please contact an Armadillo dealer.

stall_disk_sd_20161026_iotg3.img or later). For details on how to use the installation disc, please refer to Section 8.1, “Using the Installation Disc”.

11.17. JP1 Boot Device Setting Jumper

JP1 is the boot device setting jumper.

Mounted Connector A2-2PA-2.54DSA(71)/HIROSE ELECTRIC

Table 11.22. JP1 Signals

Pin Number	Pin Name	I/O	Description
1	VCC_3.3V	Power	Power (VCC_3.3V)
2	SDBOOT_EN	In	Boot device setting, connected to BOOT_MODE0 pin of i.MX7 Dual (Low: SPI flash memory boot, High: SD boot)

Table 11.23. Jumper Function

Part Number	Function	Operation
JP1	Boot device configuration	Open: Runs the bootloader in the SPI flash memory. Short: Runs the bootloader in the SD card inserted in CON4.

11.18. SW1 to SW3 User Switches

SW1, SW2 and SW3 are tactile switches that can be used freely by the user.

Table 11.24. User Switch Connections

Part Number	Description
SW1	Connected to GPIO1_IO02 pin of i.MX7 Dual, PTB5/IRQ_12 pin of BMIC (On: Low, Off: High)
SW2	Connected to LCD_DAT 17 pin of i.MX7 Dual (On: Low, Off: High)
SW3	Connected to LCD_DAT18 pin of i.MX7 Dual (On: Low, Off: High)

11.19. SW4 Reset Switch

SW4 is a tact switch for resets.

Table 11.25. Reset Switch Connections

Part Number	Description
SW4	External Reset (On: Reset state, Off: Reset release)

When resetting with SW4, press and hold for more than one second in order to ensure the reset occurs.

11.20. LED1 WWAN LED

LED1 is a surface-mounted green LED indicating the communication status of the 3G module (PDS 6-J / Gemalto).[3]

[3]On models without the 3G Module, this is connected to the WWAN expansion interface (CON10).

Table 11.26. WWAN LED Connections

Part Number	Description
LED1	Connected to STATUS pin of 3G module or pin 9 of CON10 (Low: Off、High: On)

11.21. LED2 to LED5 User LEDs

LED2, LED3, LED4 and LED5 are surface mounted green LEDs that can be used freely by the user.

Table 11.27. User LED Connections

Part Number	Description
LED2	Connected to LCD_DATA05 pin of i.MX7 Dual (Low: Off、High: On)
LED3	Connected to LCD_DATA06 pin of i.MX7 Dual (Low: Off、High: On)
LED4	Connected to LCD_DATA07 pin of i.MX7 Dual (Low: Off、High: On)
LED5	Connected to LCD_DATA08 pin of i.MX7 Dual (Low: Off、High: On)

Revision History

Version	Date	Description
1.0.0	2017/10/10	• Initial Release

Armadillo-IoT Gateway G3Product Manual - INTL
Version 1.0.0
2018/02/19