



Developer's manual  
for amnimo G series

**AG10**



**AG20**



# Licenses and Trademarks

## Licenses

- This product uses software under an open source license, such as the GNU General Public License (GPL).  
Please visit our website for details.  
URL: <https://support.amnimo.com/>
- Warranty coverage and responsibilities  
In this product, amnimo does not offer any warranties for the operation of the open source software itself under the terms of the GPL, etc.

## Trademarks

- Proper nouns including product or company names, etc. mentioned in this manual are the trademarks or registered trademarks of their respective companies.

# Introduction

Thank you for choosing our Edge Gateway amnimo G series (hereinafter referred to as "Edge Gateway").

The amnimo Developer's Manual (hereinafter referred to as "this manual") provides information for "developers" who create applications that run on Edge Gateway. This manual assumes that you are familiar with the information in the "User's Manual (IM AMD03A01-01EN)". Please also refer to the User's Manual.

## About This Manual

### Notes on This Manual

- The contents of this manual are subject to change without notice.
- Reproduction in whole or in part of this manual is prohibited without permission.
- The information in this manual has been prepared with great care. However, if you have any doubt, questions or find any error, please contact our Customer Support.

Contact: amnimo Customer Support

E-mail: [support@amnimo.com](mailto:support@amnimo.com)

URL: <https://support.amnimo.com>

- Note that for changes to specifications, structure, and used parts where it seems that there are no problems in terms of function or performance, they may not be revised on each occasion.

## List of Manuals

- amnimo Gateway Series Developer's Manual (this manual)  
[https://amnimo.com/manual/edge\\_gw/doc/IM\\_AMD03A01-51EN\\_Edge\\_Gateway\\_Indoor\\_amnimo\\_G\\_series\\_Developers\\_manual.pdf](https://amnimo.com/manual/edge_gw/doc/IM_AMD03A01-51EN_Edge_Gateway_Indoor_amnimo_G_series_Developers_manual.pdf)
- amnimo Gateway Series CLI User's Manual  
[https://amnimo.com/manual/edge\\_gw/en/ja/cli\\_users\\_manual.pdf](https://amnimo.com/manual/edge_gw/en/ja/cli_users_manual.pdf)
- amnimo G-Series Edge Gateway User's Manual  
[https://amnimo.com/manual/edge\\_gw/en/edge\\_gw\\_users\\_manual.pdf](https://amnimo.com/manual/edge_gw/en/edge_gw_users_manual.pdf)
- amnimo G Series Edge Gateway Startup Guide  
[https://amnimo.com/manual/edge\\_gw/sg/en/edge\\_gw\\_sg.pdf](https://amnimo.com/manual/edge_gw/sg/en/edge_gw_sg.pdf)
- License terms for open source software (in Japanese)  
[https://amnimo.com/manual/edge\\_gw/doc/IM\\_AMD03A01-12EN\\_amnimo\\_GW\\_series\\_OSS\\_license.pdf](https://amnimo.com/manual/edge_gw/doc/IM_AMD03A01-12EN_amnimo_GW_series_OSS_license.pdf)

## Icons and Symbols Used in This Manual

The icons and symbols in this manual have the following meanings:

	Provides information about features and operations that should be paid special attention.
	Provides additional information about features and operations.
	Provides reference information in this manual or other documents.
	Indicates that the command operation can be performed in general user mode.
	Indicates that the command operation can be performed in administrator mode.
	Indicates that the command operation can be performed in configuration mode.

## Command Convention

The command syntax in this manual is described as follows:

Notation	Description
VALUE	<ul style="list-style-type: none"> <li>● Bold indicates a fixed value.</li> <li>● Bold italic indicates a setting parameter or keyword. Required.</li> </ul>
[ A   B ]	Select A or B. Optional.
< A   B >	Select A or B. Required.
[ 0 – 9 ]	Select a value between 0 and 9. Optional.
< 0 – 9 >	Select a value between 0 and 9. Required.
↵	Indicates a line break ([ Enter ] key input).

## Command Examples

The command examples in this manual are described as follows using color-coding according to the environment in which they are run.

### ■ Edge Gateway

Indicates console operations on Edge Gateway (native development environment).

```
gateway
```

### ■ Host machine

Indicates console operations on the host machine (cross development environment).

```
host
```

### ■ Other

Indicates common commands, syntax, and source code.

## Safety and Modification Precautions

- For the protection and safety of yourself and Edge Gateway or systems that include Edge Gateway, follow the safety instructions in this manual when handling Edge Gateway. If you do not follow these instructions, amnimo cannot guarantee safety.
- Modification to Edge Gateway without permission is strictly prohibited. Do not attempt to disassemble Edge Gateway or its accessories.
- Other details on safety precautions, including radio communications, are described in the User's Manual (IM AMD03A01-01EN). Please also refer to the User's Manual.

# Table of Contents

	Licenses and Trademarks .....	2
	Introduction .....	3
	About This Manual .....	3
	List of Manuals .....	3
	Safety and Modification Precautions .....	5
	Table of Contents .....	6
<hr/>		
1	Development Environment for Edge Gateway .....	8
	1.1 Overview .....	8
	1.1.1 Hardware specifications for Edge Gateway .....	8
	1.1.2 Software specifications for Edge Gateway .....	9
	1.2 Setup Preparation .....	10
	1.2.1 Hardware configuration .....	10
	1.2.2 Setting up Edge Gateway .....	11
	1.2.3 Preparing your PC environment .....	16
	1.3 Creating a Cross Development Environment .....	17
	1.3.1 Preparing your PC environment .....	17
	1.3.2 Executing a cross build .....	18
	1.4 Native Development Environment .....	20
	1.4.1 Preparing Edge Gateway .....	20
	1.4.2 Executing a native build .....	21
	1.5 Building an environment for a distributed build .....	23
	1.5.1 Preparing your PC environment .....	23
	1.5.2 Preparing Edge Gateway .....	24
	1.5.3 Executing a distributed build .....	25
<hr/>		
2	Firmware for Edge Gateway .....	26
	2.1 File System Configuration for Edge Gateway .....	26
	2.2 Creating a Firmware Image for Edge Gateway .....	27
	2.2.1 Installing applications and packages .....	27
	2.2.2 Creating a firmware image .....	27
<hr/>		
3	Edge Gateway Utilization .....	28
	3.1 Utilizing Hook Processing .....	28
	3.2 Service Management Using systemd .....	29
	3.2.1 Creating a unit file .....	29
	3.2.2 Control and management of the service state .....	30
	3.3 Creating a Package .....	31
	3.3.1 Preparation .....	31
	3.3.2 Creating a package .....	33

3.3.3	Installing/uninstalling packages .....	37
3.4	Linking the Digital Input (D IN) with Nx Witness .....	38
3.4.4	Setting of amnimo-dimoni.....	38
3.4.5	Setting Events of Nx Witness.....	40
<hr/>		
4	Troubleshooting.....	42
4.1	Utilizing Edge Gateway Setting Functions.....	42
4.1.1	Default configuration file.....	42
4.1.2	Backing up the configuration file .....	42
4.1.3	Managing generations of configuration files .....	43
4.1.4	Reflecting the configuration file .....	44
4.2	Troubleshooting for Cases of Startup Failure .....	45
4.2.1	About the sample case .....	45
4.2.2	Step 1. Switching the startup areas on the boot loader .....	45
4.2.3	Step 2. Check the startup area after starting Linux.....	47
4.2.4	Step 3. Synchronizing the firmware .....	48
	Revision history .....	49

# 1 Development Environment for Edge Gateway

This chapter explains the development environment for Edge Gateway.

## 1.1 Overview

Edge Gateway has an ARMv8 64-bit CPU. To develop applications that run on Edge Gateway, you must build a native development environment to build and execute programs on Edge Gateway. If you need to build programs for this CPU's architecture on your host PC, you must build a cross development environment.

This manual explains how C-based Edge Gateway application development can be achieved in a cross and native development environment.

### 1.1.1 Hardware specifications for Edge Gateway

Category	Description
CPU	Arm Cortex-A53 1000 MHz (2 cores)
RAM	2 Gbyte
NOR-FLASH	4 MByte
NAND-FLASH	512 MByte
eMMC	32 Gbyte
SSD	256 GByte to 2 TByte <sup>*1</sup>
Mobile Network <sup>*2</sup>	2G/3G/4G
Antenna connectors	SMA for 2G/3G/4G x 2 SMA for GPS x 1
GNSS	GPS (QZSS)/GLONASS/Galileo/BeiDou
PoE power supply	IEEE 802.3at
Interface	
Ethernet	Gigabit Ethernet 5 ports (of which 4 ports have switch with PoE power supply functions)
Console	RJ45
RS-232	DB9
SD card	1 slot (SDXC)
SIM slot	Micro SIM (3FF) x2 slots, eSIM x 2 <sup>*3</sup>
USB	USB2.0 Host 1 port, Type-A
DIO	Digital-in Photo coupler 4ch Digital-out PhotoMOS relay 2ch Terminal block (2-piece screw type)
Switch	Push switch x 1 (for reset) DIP switch x 4 (for settings)
LED	Two-color LEDs (PWR, ANT, MOB, ANT, ST1, ST2, ST3)
Size	177 (W) x 110 (D) x 44 (H) mm (excluding protruding sections)
Weight	Approximately 820 g
Power specification	AG10-030GA-10 : 24 VDC AG10-040GA-10 : 10.8 VDC (12 VDC-10%) to 32 VDC (24 VDC+ 20%)
Power consumption	Up to 50 W (including PoE: 40 W) Absolute maximum rating: 60 W
Operating temperature	-20°C to 60°C.
Storage temperature	-20°C to 70°C.

\* 1 Provided as an option.

\* 2 Bandwidth depends on the communication module installed.

\* 3 Please contact us if you are considering using eSIM.

## 1.1.2 Software specifications for Edge Gateway

Category	Description
Kernel	Linux (4.19)
OS	Ubuntu 18.04 LTS
Basic protocol	ARP/IPv4/IPv6*/UDP/TCP
Connection protocol	IPCP/PPP/PPPoE (IPv4/IPv6*)
Dynamic IP address	DHCP server/client (IPv4/IPv6*) A DHCP server can be set for each interface.
Domain name resolution	DNS relay/client
Routing	Static routing
Address/port conversion	NAT and NAPT (destination)
VPN	IPsec/remote.it VPN
Clock synchronization	NTP/GPS
Security	Packet filters
Operational management	
Setting method	amsh Dedicated command line interface (CLI) for amnimo Gateway
Firmware update	apt (incremental update) / amfirm (full update)
Log management	syslog
Device management	amnimo DMS
Development tool chain	<ul style="list-style-type: none"> <li>● gcc-7 7.5.0-3ubuntu1 to 18.04</li> <li>● binutils 2.30-21ubuntu1 to 18.04.2</li> </ul>

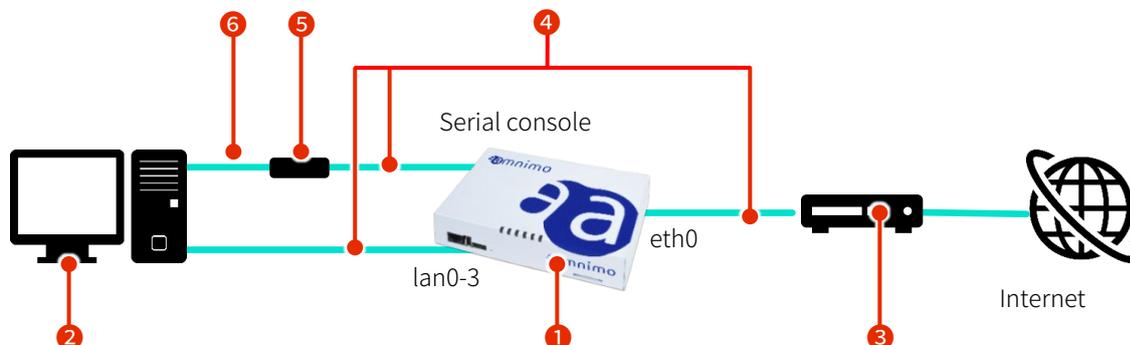
\* IPv6 is scheduled to be supported in Ver. 2 or later.

## 1.2 Setup Preparation

This section explains what needs to be prepared before you build your Edge Gateway development environment.

### 1.2.1 Hardware configuration

The hardware configuration for building an Edge Gateway development environment is shown as follows.



A description of the required hardware

No.	Hardware	Description
①	Edge Gateway	Refers to an environment in which your application runs. For a native development environment, it refers to an environment in which you build an application. Apply the settings as described in "1.2.2".
②	PC	For a cross development environment, it refers to the environment in which you build an application. Initially, SSH is disabled, and the RS232C interface is required to make setting changes. If a USB port is present, a USB-RS232C conversion adapter or other adapter may be used as an alternative.
③	Wired router	Prepared to connect the gateway to the Internet. Set the DHCP server function.
④	LAN cable	Used to connect to the serial console eth0/lan0-3.
⑤	RJ45-RS232C conversion connector	Used to convert the RJ45 (LAN cable) to the RS232C (RS232C cable).
⑥	RS232C cable	Connects the RS232C connector on the PC and the RJ45-RS232C conversion connector.



The developing user must prepare the items except for Edge Gateway. Refer to the "User's Manual" for details about the hardware.



Do not connect the hardware to the corporate LAN, and configure it as an independent network.

## 1.2.2 Setting up Edge Gateway

This subsection explains a setting example regarding how to make the hardware configuration described in "1.2.1" operational as a development environment. A setting example of Edge Gateway used in this chapter is shown as follows.



Setting example of Edge Gateway

Setting item	Setting details
Interface	eth0 : dhcp* br0 : 192.168.0.254*
SSH	Enabled
Host name	amnimo
Time zone	Asia/Tokyo

\* This is the factory default setting.

Display a setting list on the amsh configuration mode to check the setting details.

### Config Mode

```

gateway
amnimo(cfg)# show config ↵
# ---- transition to configure mode ----
configure
# ---- hostname configure ----
hostname amnimo
# ---- timezone configure ----
timezone Asia Tokyo
# ---- interface eth0 configure ----
interface eth0
enable
pmtu auto
dhcp4
dhcp4 dns 30
dhcp4 ntp
dhcp4 mtu
dhcp4 route 30
mtu 1500
mode 100baseT-Auto
proxy-arp
no optional
exit
# ---- interface lan0 configure ----
interface lan0
enable
pmtu auto
mtu 1500
mode 100baseT-Auto
proxy-arp
no optional

```

```
exit
# ---- interface lan1 configure ----
interface lan1
enable
pmtu auto
mtu 1500
mode 100baseT-Auto
proxy-arp
no optional
exit
# ---- interface lan2 configure ----
interface lan2
enable
pmtu auto
mtu 1500
mode 100baseT-Auto
proxy-arp
no optional
exit
# ---- interface lan3 configure ----
interface lan3
enable
pmtu auto
mtu 1500
mode 100baseT-Auto
proxy-arp
no optional
exit
# ---- interface br0 configure ----
interface br0
enable
bridge lan0
bridge lan1
bridge lan2
bridge lan3
mac lan0
pmtu auto
address 192.168.0.254/24
mtu 1500
proxy-arp
no optional
exit
# ---- filter input configure ----
filter input default-policy accept
# ---- filter output configure ----
filter output default-policy accept
# ---- filter forward configure ----
filter forward default-policy accept
# ---- rule 100 ----
filter forward 100
enable
policy drop
match protocol udp dst-port 137:138
exit
# ---- rule 110 ----
filter forward 110
enable
policy drop
match protocol udp src-port 137:138
```

```
exit
# ---- rule 120 ----
filter forward 120
enable
policy drop
match protocol tcp dst-port 137
exit
# ---- rule 130 ----
filter forward 130
enable
policy drop
match protocol tcp src-port 137
exit
# ---- rule 140 ----
filter forward 140
enable
policy drop
match protocol tcp dst-port 139
exit
# ---- rule 150 ----
filter forward 150
enable
policy drop
match protocol tcp src-port 139
exit
# ---- rule 160 ----
filter forward 160
enable
policy drop
match protocol tcp dst-port 445
exit
# ---- rule 170 ----
filter forward 170
enable
policy drop
match protocol tcp src-port 445
exit
# ---- dns configure ----
dns
no enable
exit
# ---- ipsec log-level configure ----
ipsec log-level
asn control
cfg control
chd control
dmn control
enc control
esp control
ike control
imc control
imv control
job control
knl control
lib control
mgr control
net control
pts control
tls control
```

```

tnc control
exit
# ---- ntp configure ----
ntp
no enable
exit
# ---- ssh configure ----
ssh
no enable
exit
# ---- syslog local configure ----
syslog local
enable
rotate-size 10240
rotate-count 8
level informational
exit
# ---- syslog remote configure ----
syslog remote
no enable
server-port 514
level informational
exit
# ---- storage sda1 configure ----
storage mount sda1 /media/ssd type ext4 options defaults
storage fsck sda1 preen
storage monitor sda1 retry 3 interval 10m reboot 3
storage failsafe sda1 retry 3 interval 10 reboot 3
# ---- poe lan0 configure ----
poe lan0
enable
limit-current auto
ondelay 0
exit
# ---- poe lan1 configure ----
poe lan1
enable
limit-current auto
ondelay 0
exit
# ---- poe lan2 configure ----
poe lan2
enable
limit-current auto
ondelay 0
exit
# ---- poe lan3 configure ----
poe lan3
enable
limit-current auto
ondelay 0
exit
# ---- cpufreq configure ----
cpufreq ondemand
# ---- thermal polling configure ----
thermal polling 1000
# ---- thermal cpufreq high configure ----
thermal cpufreq high
enable

```

```

mode high
temperature 100.0
hysteresis 10.0
log detection warnings
log restoration notifications
state 200MHZ
exit
# ---- thermal cpufreq low configure ----
thermal cpufreq low
enable
mode low
temperature -10.0
hysteresis 5.0
log detection warnings
log restoration notifications
state 1000MHZ
exit
# ---- thermal mobile high configure ----
thermal mobile high
enable
mode high
temperature 100.0
hysteresis 10.0
log detection warnings
log restoration notifications
state disable
exit
# ---- thermal interface high configure ----
thermal interface high
enable
mode high
temperature 100.0
hysteresis 10.0
log detection warnings
log restoration notifications
state 100baseT-Auto
exit
# ---- dms configure ----
dms
no enable
exit
# ---- nxwitness configure ----
nxwitness
enable
port 7001
database /mnt/share/nxwitness/database/file.db
password secret 1sxWjNj/NBbdEfGFmP6vrw==
exit
# ---- remoteit configure ----
remoteit
no enable
exit
# ---- gui configure ----
gui
enable
protocol http
port 80
exit

```

## 1.2.3 Preparing your PC environment

For your PC's OS, install the same Ubuntu 18.04 LTS version as Edge Gateway.

Download URL: <https://releases.ubuntu.com/18.04/>



If you are developing in a Windows environment, it is recommended that you build a virtual environment such as by using Virtual Box or VMware virtualization software before developing (explanation in this manual shall be omitted).

### How to check the version of Ubuntu

host

```
$ lsb_release -a ↵
No LSB modules are available.
Distributor ID: Ubuntu
Description:   Ubuntu 18.04.4 LTS
Release:      18.04
Codename:     bionic
```



The version of Ubuntu may be updated in association with an update of the user environment.

### Setting/checking IP addresses

In the following example, the PC interface (ens36) is connected to the same network (192.168.0.0/24) as Edge Gateway.

host

```
$ sudo ip addr add 192.168.0.1 dev ens36 ↵
$ ip addr show dev ens36 ↵
3: ens36: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:6d:95:a9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.1/24 brd 192.168.0.255 scope global ens36
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe6d:95a9/64 scope link
        valid_lft forever preferred_lft forever
```

## 1.3 Creating a Cross Development Environment

This section explains how to develop applications for Edge Gateway in a cross development environment.

### 1.3.1 Preparing your PC environment

#### ■ Installing the apt package

Update and upgrade the apt package list on the PC side's Ubuntu environment to install the following apt packages.

```
host
$ sudo apt update ↵
$ sudo apt upgrade ↵
$ sudo apt install build-essential crossbuild-essential-arm64 device-tree-compiler libs
sl-dev gcc-arm-linux-gnueabi file tree ↵
```

#### ■ Checking the tool chain version

Check the tool chain version.

```
host
$ aarch64-linux-gnu-gcc --version ↵
aarch64-linux-gnu-gcc (Ubuntu/Linaro 7.5.0-3ubuntu1~18.04) 7.5.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

#### ■ List of tool chain commands

apt package	Command
binutils-aarch64-linux-gnu	aarch64-linux-gnu-addr2line
	aarch64-linux-gnu-ar
	aarch64-linux-gnu-as
	aarch64-linux-gnu-c++filt
	aarch64-linux-gnu-dwp
	aarch64-linux-gnu-elfedit
	aarch64-linux-gnu-gprof
	aarch64-linux-gnu-ld
	aarch64-linux-gnu-ld.bfd
	aarch64-linux-gnu-ld.gold
	aarch64-linux-gnu-nm
	aarch64-linux-gnu-objcopy
	aarch64-linux-gnu-objdump
	aarch64-linux-gnu-ranlib
	aarch64-linux-gnu-readelf
aarch64-linux-gnu-size	
aarch64-linux-gnu-strings	
aarch64-linux-gnu-strip	
cpp-aarch64-linux-gnu	aarch64-linux-gnu-cpp
g++-aarch64-linux-gnu	aarch64-linux-gnu-g++
g++-aarch64-linux-gnu	aarch64-linux-gnu-gcc
	aarch64-linux-gnu-gcc-ar

apt package	Command
	aarch64-linux-gnu-gcc-nm
	aarch64-linux-gnu-gcc-ranlib
	aarch64-linux-gnu-gcov
	aarch64-linux-gnu-gcov-dump
	aarch64-linux-gnu-gcov-tool

## 1.3.2 Executing a cross build

This subsection explains the procedure to execute a cross build using the sample code.

### ■ Creating sample code (main.c, Makefile)

The following shows the source code for a program that continuously outputs a string every 10 seconds.

#### main.c

```
#include <stdio.h>
#include <unistd.h>

int main(void)
{
    while(1){
        printf("Hello World!¥n");
        sleep(10);
    }
}
```

host

#### Makefile

```
CC = aarch64-linux-gnu-gcc
CFLAGS = -O3 -Wall
DESTDIR = /usr/bin
LIBS =
OBSJ = main.o
PROG = amnimo-sample-app

all: $(PROG)

$(PROG): $(OBSJ)
    $(CC) $(OBSJ) $(LIBS) -o $(PROG)

clean:
    rm -f *.o $(PROG)

install: $(PROG)
    sudo install -s $(PROG) $(DESTDIR)
```

The directory structure and file placement are shown as follows.

```
~/sample$ tree ↵
.
├─ main.c
└─ Makefile
```

### ■ Build

host

```
~/sample$ make all ↵
```

## Checking the executable file type for the build result

host

```
~/sample$ file amnimo-sample-app ↵
amnimo-sample-app: ELF 64-bit LSB shared object, ARM aarch64, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-aarch64.so.1, for GNU/Linux 3.7.0, BuildID[sha1]=1e9a9503efe79366a8c4f51bba5d20cc65f9a7fe, not stripped
```

## Transferring the executable file

Send the generated executable file amnimo-sample-app to Edge Gateway via scp.

host

```
~/sample$ scp amnimo-sample-app admin@192.168.0.254:/tmp/ ↵
The authenticity of host '192.168.0.254 (192.168.0.254)' can't be established.
ECDSA key fingerprint is SHA256:AJ0j48/CzTC8mETZnRjwHyGegbpq00vQOg6/8sB9npg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.254' (ECDSA) to the list of known hosts.
admin@192.168.0.254's password:
amnimo-sample-app                               100% 9080      3.0MB/s   00:00
```



The blue part (fingerprint confirmation) is output in the following circumstances.

- At the first ssh login
- When data is sent/received via scp

## Executing the executable file

Log in to Edge Gateway and execute the executable file.

gateway

```
admin@amnimo:~$ /tmp/amnimo-sample-app ↵
Hello World!
Hello World!
...
```

## 1.4 Native Development Environment

This section explains how to develop applications for Edge Gateway in a native development environment.

### 1.4.1 Preparing Edge Gateway

#### ■ Installed tool chains

The factory default Edge Gateway is provided with the following tool chains.

apt package	Command
binutils-aarch64-linux-gnu	aarch64-linux-gnu-addr2line
	aarch64-linux-gnu-ar
	aarch64-linux-gnu-as
	aarch64-linux-gnu-c++filt
	aarch64-linux-gnu-dwp
	aarch64-linux-gnu-elfedit
	aarch64-linux-gnu-gprof
	aarch64-linux-gnu-ld
	aarch64-linux-gnu-ld.bfd
	aarch64-linux-gnu-ld.gold
	aarch64-linux-gnu-nm
	aarch64-linux-gnu-objcopy
	aarch64-linux-gnu-objdump
	aarch64-linux-gnu-ranlib
	aarch64-linux-gnu-readelf
	aarch64-linux-gnu-size
	aarch64-linux-gnu-strings
aarch64-linux-gnu-strip	

#### ■ Installing the apt package required for a build

Log in to Edge Gateway, update and upgrade the apt package list in the Ubuntu environment of the Edge Gateway to install the following apt packages.

Before proceeding, check "4.5 Update of Edge Gateway apt Package Repository" in the "Edge Gateway amnimo G series User's Manual".

```
gateway
$ sudo apt update ↵
$ sudo apt upgrade ↵
$ sudo apt install gcc file tree ↵
```



If you want to use C++ (g++-aarch64-linux-gnu) or cmake, you must prepare them in advance.

## ■ Checking the tool chain version

Check the tool chain version.

```
admin@amnimo:~$ gcc --version ↵
gcc (Ubuntu/Linaro 7.5.0-3ubuntu1~18.04) 7.5.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOS
```

## 1.4.2 Executing a native build

This subsection explains the procedure to execute a native build using the sample code.

### ■ Creating sample code (main.c, Makefile)

Use the sample code in "1.3.2".

main.c

```
#include <stdio.h>
#include <unistd.h>

int main(void)
{
    while(1){
        printf("Hello World!¥n");
        sleep(10);
    }
}
```

Makefile

```
CC = gcc
CFLAGS = -O3 -Wall

DESTDIR := /
LIBS =
OBJS = main.o
PROG = amnimo-sample-app

all: $(PROG)

$(PROG): $(OBJS)
    $(CC) $(OBJS) $(LIBS) -o $(PROG)

clean:
    rm -f *.o $(PROG)

install: $(PROG)
    install -D -s $(PROG) $(DESTDIR)/usr/sbin/$(PROG)
```

The directory structure and file placement are shown as follows.

```
admin@amnimo:~/sample$ tree ↵
.
├─ main.c
└─ Makefile
```

## ■ Executing a native build

gateway

```
admin@amnimo:~$ make all ↵
```

## ■ Checking the executable file type for the build result

gateway

```
$ file amnimo-sample-app ↵  
amnimo-sample-app: ELF 64-bit LSB shared object, ARM aarch64, version 1 (SYSV), dynamic  
ally linked, interpreter /lib/ld-linux-aarch64.so.1, for GNU/Linux 3.7.0, BuildID[sha1]  
=1436db4cc909f72d8565e5f8d14a2b4be47c0515, not stripped
```

## ■ Executing the executable file

Log in to Edge Gateway and execute the executable file.

gateway

```
admin@amnimo:~$ /tmp/amnimo-sample-app ↵  
Hello World!  
Hello World!  
...
```

## 1.5 Building an environment for a distributed build

In a native development environment, the CPU processing capacity of Edge Gateway underperforms that of a general-purpose PC. Therefore, it takes longer than the build time of a cross development environment. This difference becomes prominent as the program scale increases.

The following subsection explains how to speed up the build process of a native development environment by building a distributed build environment using distcc.

### 1.5.1 Preparing your PC environment

Use the host side as the distribution point for the build process.

#### ■ Installing the distcc/ccache package

Update and upgrade the apt package list on the PC side's Ubuntu environment to install the following apt packages.

```
$ sudo apt install distcc ccache ↵
```

host

#### ■ Modifying configuration files of distcc

Edit the part after `/etc/default/distcc`.

```
# Defaults for distcc initscript
# sourced by /etc/init.d/distcc

#
# should distcc be started on boot?
#
STARTDISTCC="true" # Make STARTDISTCC="false" to be "true"

#
# Which networks/hosts should be allowed to connect to the daemon?
# You can list multiple hosts/networks separated by spaces.
# Networks have to be in CIDR notation, f.e. 192.168.0.0/24
# Hosts are represented by a single IP Address
#
ALLOWEDNETS="192.168.0.0/24" # Specify the subnet including the IP address of the requestor (E
dge Gateway) in ALLOWEDNETS=

#
# Which interface should distccd listen on?
# You can specify a single interface, identified by it's IP address, here.
#
LISTENER="192.168.0.1" # Specify the PC's IP address (the IP address of the interface accessible
from Edge Gateway) in LISTENER=

#
# You can specify a (positive) nice level for the distcc process here
#
NICE="10" # Specify NICE (high priority - 20 to 19 low priority). Set it according to your develop
ment environment.

#
# You can specify a maximum number of jobs, the server will accept concurrently
#
JOBS="8" # Specify the number of jobs to do a distributed build. Set it according to your develo
pment environment.
```

host

```
#
# Enable Zeroconf support?
# If enabled, distccd will register via mDNS/DNS-SD.
# It can then automatically be found by zeroconf enabled distcc clients
# without the need of a manually configured host list.
#
# ZEROCONF="true"

ZEROCONF="false"
```

## Restarting the distcc daemon

```
host
$ sudo systemctl daemon-reload ↵
$ sudo systemctl enable distcc ↵
$ sudo systemctl restart distcc ↵
$ netstat -a | grep distcc
tcp        0      0 localhost:distcc    0.0.0.0:*          LISTEN     ↵Check the
output
```

## 1.5.2 Preparing Edge Gateway

As on the host side, install distcc/ccache, edit the configuration file, and restart the daemon. However, only STARTDISTCC is to be modified in the configuration file.

### Installing the distcc/ccache package

```
gateway
admin@amnimo:~$ sudo apt install distcc ccache ↵
```

### Modifying configuration files of distcc

Edit the part after /etc/default/distcc.

```
gateway
# Defaults for distcc initscript
# sourced by /etc/init.d/distcc

#
# should distcc be started on boot?
#
STARTDISTCC="true" # make STARTDISTCC="false" to be "true"

# After here is by default

...
```

### Restarting the distcc daemon

```
gateway
admin@amnimo:~$ sudo systemctl daemon-reload ↵
admin@amnimo:~$ sudo systemctl enable distcc ↵
admin@amnimo:~$ sudo systemctl restart distcc ↵
admin@amnimo:~$ netstat -a | grep distcc ↵
tcp        0      0 localhost:distcc    0.0.0.0:*          LISTEN     ↵Check th
e output
```

## Setting environment variables

Set the environment variables for a distributed build.

```
export CCACHE_PREFIX=distcc
export CC="ccache /usr/bin/aarch64-linux-gnu-gcc"
export CXX="ccache /usr/bin/aarch64-linux-gnu-g++"
export AS="/usr/bin/aarch64-linux-gnu-as"
export AR="/usr/bin/aarch64-linux-gnu-ar"
export CPP="/usr/bin/aarch64-linux-gnu-cpp"
export LD="/usr/bin/aarch64-linux-gnu-ld"
export PKG_CONFIG="/usr/bin/aarch64-linux-gnu-pkg-config"
export DISTCC_HOSTS="192.168.0.1/8 127.0.0.1"
```



### DISTCC\_HOSTS

- Multiple IP addresses can be added. Arrange them in order of performance, with the highest server first and the lowest server last, from left to right.
- The processing will be sorted in the order set in the DISTCC\_HOSTS. If a connection cannot be made with the distcc of the target IP address, the processing will be sorted to the IP address specified next.
- 8 in 192.168.0.1/8 indicates the number of threads executed at the same time. If the number is omitted, 4 will be set by default. Specify the same value as the jobs in the configuration file in the distribution point's distcc.

## 1.5.3 Executing a distributed build

### Specifying parallel processing when using make

This subsection provides an explanation assuming a build environment in which the Makefile is available. You can specify the number of parallel processing by specifying values after -j.

```
admin@amnimo:~$ make -j8 ↵
```

### Checking the status of the distributed build

You can check the operation status of distcc by using the following command.

```
admin@amnimo:~$ watch distccmon-text ↵
```

#### Normal operation

```
8999 Compile    yyy.c          127.0.0.1[1]
9100 Compile    zzz.c          192.168.0.1[0]
9094 Compile    aaa.c          192.168.0.1[1]
(Press Ctrl + C to terminate)
```

## 2 Firmware for Edge Gateway

This chapter explains how to provide the unique Edge Gateway applications that users created as firmware or packages.

### 2.1 File System Configuration for Edge Gateway

This section explains the correspondence between Edge Gateway memory devices and the file system.

Memory device	Device file	Mount point	Capacity	File system	Redundant area <sup>*1</sup>	Use
eMMC	/dev/mmcblk0boot0	/mnt/config/area0	2.9MB	ext4	AREA0	Setting storage area
	/dev/mmcblk1boot0	/mnt/config/area1	2.9MB	ext4	AREA1	Setting storage area
	/dev/mmcblk0p1	/	8.3GB	ext4	AREA0	rootfs area
	/dev/mmcblk0p2	/	8.3GB	ext4	AREA1	rootfs area
	/dev/mmcblk0p3	/opt/local	4.1GB	ext4	AREA0	userfs area
	/dev/mmcblk0p4	/opt/local	4.1GB	ext4	AREA1	userfs area
	/dev/mmcblk0p5	/mnt/share <sup>*2</sup>	3.5GB	ext4	-	shared area
SSD <sup>*3</sup>	/dev/sda1	/var/log	64GB to 2TB	ext4	-	For NxWitness Data storage area

\*1 Edge gateway has redundant areas. For information on how to manage redundant areas, refer to "4.4 Operating the Firmware" in the "Edge Gateway amnimo G series User's Manual".

\*2 /mnt/share/log is mounted as /var/log and /mnt/share/common is mounted as /opt/common.

\*3 SSD is optional. In addition, users can select the mount point or the file system. The description above is an example. For information on how to enable SSDs, refer to "Chapter 6 Storage Operations" in the "Edge Gateway amnimo G series User's Manual".

## 2.2 Creating a Firmware Image for Edge Gateway

First, install the user-created applications and required packages on the gateway. Then create a firmware image in the environment.

### 2.2.1 Installing applications and packages

Save the application you have created in the userfs area (under /opt/local).

Use the following command to install the packages required for running the application.

```
gateway
admin@amnimo:~$ sudo apt update
admin@amnimo:~$ sudo apt install <package name>
```

### 2.2.2 Creating a firmware image

The following command creates a firmware image from the specified redundant area. When you execute the command, it will initiate a shutdown and create a firmware image that includes the rootfs and userfs areas during the shutdown.

#### Syntax

```
amfwgen snapshot --target=<target> --partition=<partition> <amf>
```

#### Setting item

Item	Details						
target	Specify the redundant area where the firmware image is created.						
	<table border="1"> <thead> <tr> <th>Display</th> <th>Details</th> </tr> </thead> <tbody> <tr> <td>fore</td> <td>Specify the redundant area that is currently running. Used as the default value in cases where the target is omitted.</td> </tr> <tr> <td>back</td> <td>Specify the redundant area that is not currently running.</td> </tr> </tbody> </table>	Display	Details	fore	Specify the redundant area that is currently running. Used as the default value in cases where the target is omitted.	back	Specify the redundant area that is not currently running.
	Display	Details					
fore	Specify the redundant area that is currently running. Used as the default value in cases where the target is omitted.						
back	Specify the redundant area that is not currently running.						
partition	Specify the redundant area where the firmware image is created.						
	<table border="1"> <thead> <tr> <th>Display</th> <th>Details</th> </tr> </thead> <tbody> <tr> <td>rootfs</td> <td>Specify the rootfs.</td> </tr> <tr> <td>userfs</td> <td>Specify the userfs.</td> </tr> </tbody> </table>	Display	Details	rootfs	Specify the rootfs.	userfs	Specify the userfs.
	Display	Details					
rootfs	Specify the rootfs.						
userfs	Specify the userfs.						
amf	<p>Specify the output destination for the firmware image. Upon restart, the firmware image is generated in the path specified here.</p> <p> Specify the path on the rootfs as the output destination. You cannot specify the mount destination as the userfs/sharefs or an SD card.</p>						



- When you execute this command, it will initiate a shutdown.
- The amount of time it takes to restart after executing this command depends on the used size of the redundant area. It typically takes from an hour to a few hours.
- It is recommended that you reduce the size of the redundant area before executing the command.

#### Example

The following is an example to generate a firmware image that includes the rootfs and userfs in the currently running area.

```
gateway
admin@amnimo:~$ sudo amfwgen snapshot --target=fore --partition=rootfs,userfs /home/admin/firmware.amf
```

# 3 Edge Gateway Utilization

This chapter introduces how to utilize Edge Gateway when it is incorporated with applications.

## 3.1 Utilizing Hook Processing

Edge gateway has a mechanism to hook users' own processing in certain cases. The script files located under the following directory will be executed.

### Hook processing list

Function	Directory	Execution condition
Ethernet interface	/etc/amnimo/if-up.d	Interface link up
	/etc/amnimo/if-down.d	Interface link down
PoE	/etc/poe/hook/temp-alert-occur.d	Occurrence of a thermal error within the PoE controller
	/etc/poe/hook/temp-alert-clear.d	Recovery from a thermal error within the PoE controller
PPP	/etc/ppp/ip-up.d	Allocation of an IP address
	/etc/ppp/ip-down.d	Release of an IP address
Communication module	/etc/ecm/if-up.d	Line connection
	/etc/ecm/if-down.d	Line disconnection
Low voltage monitoring function	/etc/amnimo/uvol-deteccion.d	Low voltage detection
	/etc/amnimo/uvol-recovery.d	Recovery from low voltage
Temperature monitoring	/etc/amnimo/thermal-detection.d	Occurrence of a thermal error
	/etc/amnimo/thermal-restoration.d	Recovery from a thermal error



The file names must consist of ASCII upper and lower case characters, ASCII numbers, ASCII underscores (\_), and ASCII hyphens (-).

## 3.2 Service Management Using systemd

Ubuntu uses systemd as the service management function. Therefore, startup of user-provided applications can also be managed by preparing service type unit files of systemd.

This section explains how to create a unit file of systemd when you use the sample program `amnimo-sample-app` in "1.3.2" as a service.

### 3.2.1 Creating a unit file

Create a unit file (`amnimo-sample-app.service`) for the `amnimo-sample-app` under `/lib/systemd/system`.

#### `amnimo-sample-app.service`

```
[Unit]
Description= amnimo sample application
After=syslog.target

[Service]
Type=simple
ExecStart=/usr/sbin/amnimo-sample-app

[Install]
WantedBy=multi-user.target
```

#### Unit section

Provide items that are independent from the unit type, such as the unit description, dependency, and ordered relation.

Parameter	Details	Setting example for the sample
Description	Provide a description related to the program.	amnimo sample application
After	This unit will be executed after the specified unit list starts.	syslog.target (Start this sample program after syslog to save logs in syslog)

#### Service section

Provide the settings that are specific to the service unit.

Parameter	Details	Setting example for the sample
Type	Specify how the service unit is started.	simple
ExecStart	Specify the command that starts the service.	/usr/sbin/amnimo-sample-app (This sample command)

#### Install section

Provide the dependency so that the unit will start when automatic startup is enabled for the specified target.

Parameter	Details	Setting example for the sample
WantedBy	When you enable the service unit, a link to the unit file will be created in the <code>.wants</code> directory of the set unit.	multi-user.target (A target for an environment with multiple users. Equivalent to runlevels 2 to 4 for <code>systinit</code> )



When you customize the unit files, refer to the following websites.

- Unit files <https://www.freedesktop.org/software/systemd/man/systemd.unit.html>
- Service units <https://www.freedesktop.org/software/systemd/man/systemd.service.html>

## 3.2.2 Control and management of the service state

Start the service unit that you have created and check its operation.

### ■ Operation after modifying the unit file

When you modify the unit file, you must execute the following command to reflect the setting details.

```
admin@amnimo:~$ sudo systemctl daemon-reload ↵
```

### ■ Service unit startup and status check

When you start the service unit, execute the command as follows.

```
admin@amnimo:~$ sudo systemctl start amnimo-sample-app ↵
```

You can check if the service unit has started successfully in the following way.

```
admin@amnimo:~$ sudo systemctl status amnimo-sample-app ↵
● amnimo-sample-app.service - amnimo sample application
   Loaded: loaded (/lib/systemd/system/amnimo-sample-app.service; disabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-08-19 18:22:05 JST; 10s ago
     Main PID: 30823 (amnimo-sample-a)
    CGroup: /system.slice/amnimo-sample-app.service
            └─30823 /usr/sbin/amnimo-sample-app
...
gateway
```

### ■ Enabling/disabling the units

If you want to start the unit automatically at the system startup, enable the unit.

```
admin@amnimo:~$ sudo systemctl enable amnimo-sample-app ↵
```

To disable the unit, execute the command as follows.

```
admin@amnimo:~$ sudo systemctl disable amnimo-sample-app ↵
```

## 3.3 Creating a Package

You can use a package management system in Edge Gateway. You can also manage dependencies between packages by managing user-created applications as a package.

This section explains how to create a package in a native development environment.



Refer to the Ubuntu website for details of the package management system.  
<https://ubuntu.com/server/docs/package-management>

### 3.3.1 Preparation

Install the required applications.

#### ■ Installing related packages

```
gateway
admin@amnimo:~$ sudo apt update ↵
admin@amnimo:~$ sudo apt upgrade ↵
admin@amnimo:~$ sudo apt install devscripts cdbshelper debhelper dh-make ↵
```

#### ■ Source code for the sample application

For the sample program, use the sample code in "1.3.2". For the unit file for the service, use the unit file created in "3.2.1".

#### main.c

```
#include <stdio.h>
#include <unistd.h>

int main(void)
{
    while(1){
        printf("Hello World!\n");
        sleep(10);
    }
    return 0;
}
```

## Makefile

Add the service unit installation processing to the Makefile in "1.3.2".

```
CC = gcc
CFLAGS = -O3 -Wall

DESTDIR :=
LIBS =
OBJS = main.o
PROG = amnimo-sample-app
UNITFILE = amnimo-sample-app.service

all: $(PROG)
$(PROG): $(OBJS)
    $(CC) $(OBJS) $(LIBS) -o $(PROG)

clean:
    rm -f *.o $(PROG)

install: $(PROG)
    install -D -s $(PROG) $(DESTDIR)/usr/sbin/$(PROG)
    install -D -m 644 $(UNITFILE) $(DESTDIR)/lib/systemd/system/$(UNITFILE)
```

## amnimo-sample-app.service

```
[Unit]
Description= amnimo sample application
After=syslog.target

[Service]
Type=simple
ExecStart=/usr/sbin/amnimo-sample-app

[Install]
WantedBy=multi-user.target
```

The directory structure and file placement are shown as follows.

```
admin@amnimo:~/sample$ tree ↵
.
├── amnimo-sample-app.service
├── main.c
└── Makefile
```

gateway

## 3.3.2 Creating a package

### ■ Creating a package template

Use `dh_make` to create a package template in the sample application directory.

```

gateway
admin@amnimo:~$ cd sample ↵
admin@amnimo:~/sample$ dh_make --native --single --email support@amnimo.com --packagename
amnimo-sample-app_1.0.0 ↵
Maintainer Name: unknown
Email-Address   : support@amnimo.com
Date            : Wed, 19 Aug 2020 10:33:40 +0000
Package Name    : amnimo-sample-app
Version         : 1.0.0
License         : gpl3
Package Type    : single
Are the details correct? [Y/n/q]          ← Press the [ y ] key followed by the [ Enter ] key
to exit
admin@amnimo:~/sample$

```



Options for `dh_make` used in the example above are shown as follows.

Option	Setting details
<code>1.0--native</code>	Creating an Ubuntu-compliant package
<code>--single</code>	Creating an application
<code>--email</code>	Settings for email addresses on each file
<code>--packagename</code>	A package name for each package file

In addition, the parameters set in the example above are shown below.

Parameter	Setting details
Package name	amnimo-sample-app
Version	1.0.0  Specified according to Semantic Versioning 2.0.0. <a href="https://semver.org/lang/ja/">https://semver.org/lang/ja/</a>
Email address	<a href="mailto:support@amnimo.com">support@amnimo.com</a>

When you create a package template, the following files in blue are output.

gateway

```
admin@amnimo:~/sample$ tree ↵
.
├── amnimo-sample-app.service
├── debian
│   ├── amnimo-sample-app.cron.d.ex
│   ├── amnimo-sample-app.doc-base.EX
│   ├── amnimo-sample-app-docs.docs
│   ├── changelog
│   ├── compat
│   ├── control
│   ├── copyright
│   ├── manpage.1.ex
│   ├── manpage.sgml.ex
│   ├── manpage.xml.ex
│   ├── menu.ex
│   ├── postinst.ex
│   ├── postrm.ex
│   ├── preinst.ex
│   ├── prerm.ex
│   ├── README
│   ├── README.Debian
│   ├── README.source
│   ├── rules
│   └── source
│       └── format
├── main.c
└── Makefile
```

### Main generated files

File name	Details
control	Information used by the package management tool.
copyright	Information related to package copyright and license. By default, GPLv3 is set.  This file is not required.
changelog	Used to identify the version number, revision, distribution, and urgency of the package. It must be provided in the following format. <a href="http://www.debian.org/doc/debian-policy/ch-source.html#s-dpkgchangelog">http://www.debian.org/doc/debian-policy/ch-source.html#s-dpkgchangelog</a>  The debchange command can be used to get the format template.
rules	Contains rules for creating packages. For basic Makefile descriptions (all, clean, install, etc.), you do not need to make any changes.



When you customize package information, check the Debian site for more information.  
<https://www.debian.org/doc/manuals/maint-guide/dreq.ja.html>

## ■ Creating a package

By executing `dh_make`, you have created the files necessary to create the package. Next, create the package with `debuild`.

gateway

```
admin@amnimo:~/sample$ debuild -us -uc -ui ↵
```

When you execute this, the following files in blue are output. The files related to the package are output one level higher in the execution directory.

gateway

```
admin@amnimo:~/sample$ tree ../ ↵
../
├── amnimo-sample-app_1.0.0_arm64.build
├── amnimo-sample-app_1.0.0_arm64.buildinfo
├── amnimo-sample-app_1.0.0_arm64.changes
├── amnimo-sample-app_1.0.0_arm64.deb
├── amnimo-sample-app_1.0.0.dsc
├── amnimo-sample-app_1.0.0.tar.xz
├── sample
│   ├── amnimo-sample-app
│   └── debian
│       ├── amnimo-sample-app
│       │   ├── DEBIAN
│       │   │   ├── control
│       │   │   └── md5sums
│       │   ├── lib
│       │   │   ├── systemd
│       │   │   └── system
│       │   └── amnimo-sample-app.service
│       ├── usr
│       │   ├── sbin
│       │   │   └── amnimo-sample-app
│       │   ├── share
│       │   │   └── doc
│       │   │       └── amnimo-sample-app
│       │   │           ├── changelog.gz
│       │   │           ├── copyright
│       │   │           └── README.Debian
│       └── amnimo-sample-app.cron.d.ex
│           ├── amnimo-sample-app.doc-base.EX
│           ├── amnimo-sample-app-docs.docs
│           ├── amnimo-sample-app.substvars
│           ├── changelog
│           ├── compat
│           ├── control
│           ├── copyright
│           ├── debhelper-build-stamp
│           ├── files
│           └── manpage.1.ex
```

(Omitted)

```
├── source
│   └── format
├── main.c
├── main.o
└── Makefile
```

## Main generated files

File name	Details
amnimo-sample-app_1.0.0_arm64.deb	A binary package. It can be installed/uninstalled using the dpkg command.
amnimo-sample-app_1.0.0.dsc	A summary of the source code details.
amnimo-sample-app_1.0.0_arm64.changes	This file contains all the changes in the revision package.
amnimo-sample-app_1.0.0_arm64.build	Logs when the package was generated.
amnimo-sample-app_1.0.0_arm64.buildinfo	They contain package dependency information and build times.
amnimo-sample-app_1.0.0.tar.xz	A source code tar archive. It contains the contents of the Debian directory in the sample directory.

## ■ Checking the contents in the created package

Check the contents in the created deb package archive.

```

gateway
admin@amnimo:~$ dpkg -c amnimo-sample-app_1.0.0_arm64.deb ↵
drwxr-xr-x root/root      0 2020-01-02 00:53 ./
drwxr-xr-x root/root      0 2020-01-02 00:53 ./lib/
drwxr-xr-x root/root      0 2020-01-02 00:53 ./lib/systemd/
drwxr-xr-x root/root      0 2020-01-02 00:53 ./lib/systemd/system/
-rwxr-xr-x root/root    163 2020-01-02 00:53 ./lib/systemd/system/amnimo-sample-app.
service
drwxr-xr-x root/root      0 2020-01-02 00:53 ./
drwxr-xr-x root/root      0 2020-01-02 00:53 ./usr/
drwxr-xr-x root/root      0 2020-01-02 00:53 ./usr/sbin/
-rwxr-xr-x root/root   5864 2020-01-02 00:53 ./usr/sbin/amnimo-sample-app
drwxr-xr-x root/root      0 2020-01-02 00:53 ./usr/share/
drwxr-xr-x root/root      0 2020-01-02 00:53 ./usr/share/doc/
drwxr-xr-x root/root      0 2020-01-02 00:53 ./usr/share/doc/amnimo-sample-app/
-rw-r--r-- root/root    193 2020-01-02 00:53 ./usr/share/doc/amnimo-sample-app/README
E.Debian
-rw-r--r-- root/root    145 2020-01-02 00:53 ./usr/share/doc/amnimo-sample-app/chang
elog.gz
-rw-r--r-- root/root   1412 2020-01-02 00:53 ./usr/share/doc/amnimo-sample-app/copyr
ight

```

### 3.3.3 Installing/uninstalling packages

You can install/uninstall the generated packages using the dpkg command.

#### ■ Installing packages

Specify the package file name.

The following is an example of installing the package under /home/admin.

```
admin@amnimo:~$ sudo apt install /home/admin/amnimo-sample-app_1.0.0_arm64.deb ↵ gateway
```

#### ■ Checking the installed packages

Specify the package name.

```
admin@amnimo:~$ apt list amnimo-sample-app ↵ gateway
Listing... Done
amnimo-sample-app/now 1.0.0 arm64 [installed,local]
```

#### ■ Uninstalling packages

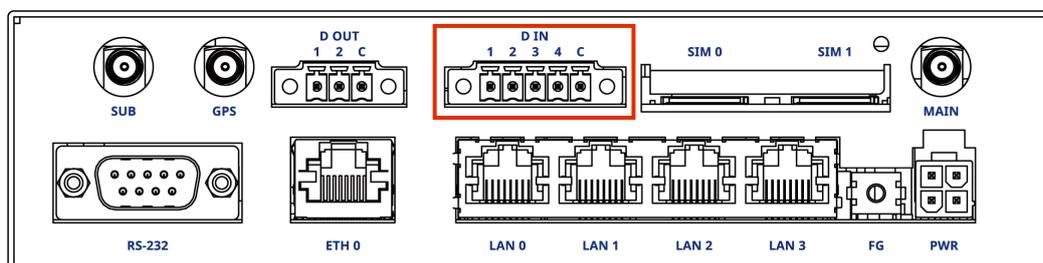
Specify the package name.

```
admin@amnimo:~$ sudo apt remove amnimo-sample-app ↵ gateway
```

### 3.4 Linking the Digital Input (D IN) with Nx Witness

Edge Gateway has been preinstalled with amnimo-dimoni, an application that monitors changes in the digital input (D IN) terminals on the rear of the device and notifies Nx Witness of them as events.

By linking the digital input (D IN) with Nx Witness, you will be able to know events in real time and quickly search videos taken when an event occurred from recorded data.



The firmware version V1.3.0 or later of Edge Gateway has been preinstalled on this function.

➔ For details of the digital input (D IN), refer to "1.2.6 D IN/D OUT ports" in the "Edge Gateway User's Manual".

#### 3.4.4 Setting of amnimo-dimoni

Configure the settings to notify Nx Witness of changes of the digital input (D IN).

##### ■ Changing the amnimo-dimoni configuration file

Open the `/etc/amnimo/dimoni.conf` file, and set the admin password of Nx Witness for `NX_PASSWORD`.



- The admin password of Nx Witness is the password set when connecting to Edge Gateway from the Nx Witness Client application of your PC for the first time.
- If you specify a port number other than 7001 for Nx Witness, change the `NX_PORT` settings (7001 by default) as well.

➔ For details, refer to "Step 6 Setting VMS" in the "Edge Gateway Startup Guide".

gateway

```
admin@amnimo:~$ sudo cat /etc/amnimo/dimoni.conf ↵
#
# Nx Witness Generic Events
#
NX_PASSWORD=PASSWORD           ←Set the password used for Nx Witness.
NX_PORT=7001                     ←Set the port number (7001 by default) used for Nx Witness.
NX_DI_1_UP=EVENT_DI-1_UP        ↓ ↓ ↓ ↓ You do not need to change the following. ↓ ↓ ↓ ↓
NX_DI_1_DN=EVENT_DI-1_DN
NX_DI_2_UP=EVENT_DI-2_UP
NX_DI_2_DN=EVENT_DI-2_DN
NX_DI_3_UP=EVENT_DI-3_UP
NX_DI_3_DN=EVENT_DI-3_DN
NX_DI_4_UP=EVENT_DI-4_UP
NX_DI_4_DN=EVENT_DI-4_DN
:
```

## Restarting the service

If the settings have been changed, you will need to restart the amnimo-dimoni service.

```
admin@amnimo:~$ sudo systemctl restart amnimo-dimoni ↵
```

## Disabling the linkage settings

amnimo-dimoni is automatically started when starting the system. If you disable Nx Witness, you will need to delete the password set for NX\_PASSWOARD and restart the amnimo-dimoni service as shown below.

```
admin@amnimo:~$ sudo vi /etc/amnimo/dimoni.conf ↵

#
# Nx Witness Generic Events
#
NX_PASSWORD=                ←Delete the password settings.
NX_PORT=7001
NX_DI_1_UP=EVENT_DI-1_UP
NX_DI_1_DN=EVENT_DI-1_DN
NX_DI_2_UP=EVENT_DI-2_UP
NX_DI_2_DN=EVENT_DI-2_DN
NX_DI_3_UP=EVENT_DI-3_UP
NX_DI_3_DN=EVENT_DI-3_DN
NX_DI_4_UP=EVENT_DI-4_UP
NX_DI_4_DN=EVENT_DI-4_DN
:
admin@amnimo:~$ sudo systemctl restart amnimo-dimoni ↵ ←Restart the service.
```

## Saving the configuration file

With amsh, write the contents of the amnimo-dimoni configuration file (/etc/amnimo/dimoni.conf) to the one of Edge Gateway. By performing this operation, even if the whole firmware is updated with the firmware area update command or if the system is started in a redundant area that is not running currently, the settings will be retained.

Admin Mode Config Mode

```
amnimo# config file save startup-config ↵
```

### 3.4.5 Setting Events of Nx Witness

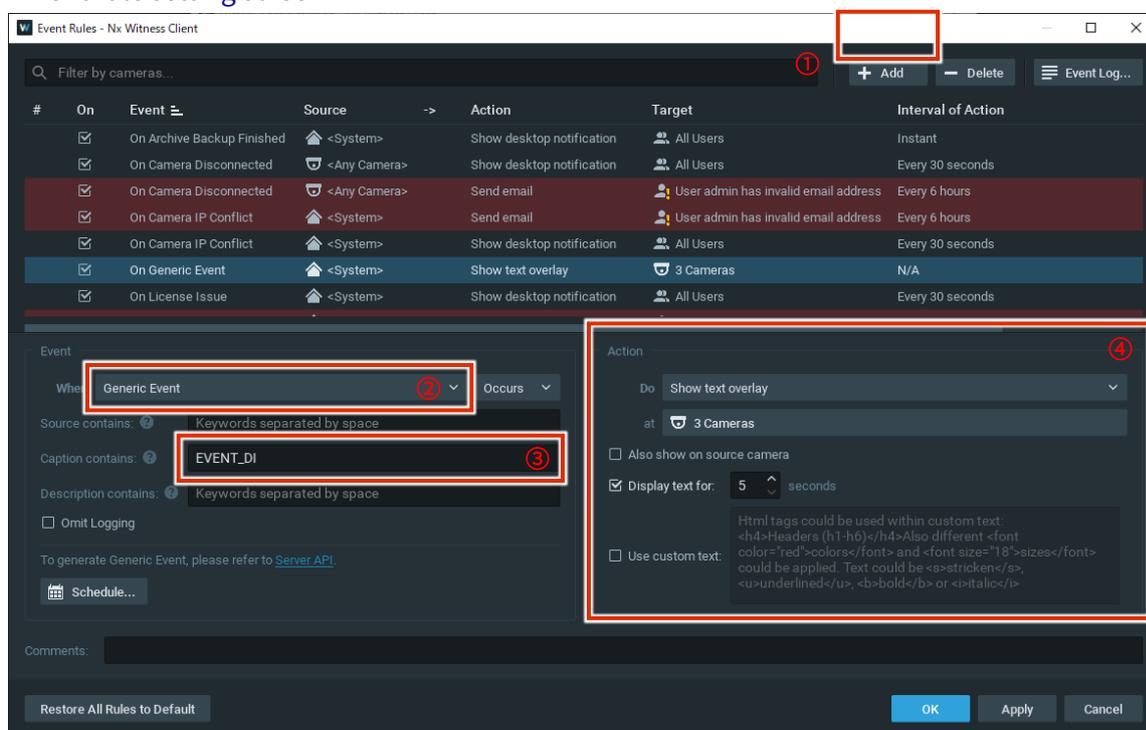
You can specify actions such as displaying texts on the LIVE video screen, assigning bookmarks to recorded videos, and sending messages when changing the digital input (D IN).

#### Operating the Nx Witness Client application

Start the Nx Witness Client application from your PC, and set the event. After connecting Edge Gateway with Nx Witness, click  shown at the top left of the screen, and click [System Administration] -> [Event Rules] to display the event rule setting screen. Configure the following settings on this screen.

- ① Click the [ Add ] button to create a new event.
- ② Set [Generic Event] for [ When ].
- ③ Set the character string (listed in the table below) for [ Caption contains ], and specify the event trigger.
- ④ For [ Action ], set the action you want to execute when the event occurs.

#### Event rule setting screen



#### List of character strings that can be set for [ Caption contains ] and event occurrence timings

Character string that can be set for [ Caption contains ]	Event occurrence timing
EVENT_DI-1_UP	DI-1 changes from Low to High
EVENT_DI-1_DN	DI-1 changes from High to Low
EVENT_DI-2_UP	DI-2 changes from Low to High
EVENT_DI-2_DN	DI-2 changes from High to Low
EVENT_DI-3_UP	DI-3 changes from Low to High
EVENT_DI-3_DN	DI-3 changes from High to Low
EVENT_DI-4_UP	DI-4 changes from Low to High
EVENT_DI-4_DN	DI-4 changes from High to Low
EVENT_DI-1	Changes in DI-1 (both High and Low)
EVENT_DI-2	Changes in DI-2 (both High and Low)
EVENT_DI-3	Changes in DI-3 (both High and Low)
EVENT_DI-4	Changes in DI-4 (both High and Low)

Character string that can be set for [ Caption contains ]	Event occurrence timing
EVENT_DI	Changes in one of the DIs (both High and Low)

## ■ Saving the Nx Witness settings

Save the Nx Witness settings including the event settings mentioned earlier for the amsh configuration mode. By performing this operation, even if the whole firmware is updated with the firmware area update command or if the system is started in a redundant area that is not running currently, the settings will be retained.



- Set the admin password of Nx Witness for password.
- If you specify a port number other than 7001 for Nx Witness, change the port settings as well.

### Config Mode

gateway

```
amnimo(cfg)# nxwitness ↵
amnimo(cfg-nxwitness)# password ↵
Enter new password:
Retype new password:
passwd: password updated successfully.
amnimo(cfg-nxwitness)# port 7001 ↵
amnimo(cfg-nxwitness)# exit ↵
amnimo(cfg)# config nxwitness save ↵
amnimo(cfg)# config file save startup-config ↵
```

## 4 Troubleshooting

This chapter explains troubleshooting for cases where abnormal behaviors occur in the application development for Edge Gateway.

### 4.1 Utilizing Edge Gateway Setting Functions

Setting functions for Edge Gateway can be utilized in the following cases.

- You want to revert to the original behavior
- You want to use the same Edge Gateway settings
- You want to create multiple Edge Gateway settings which are partly different, in order to check the operation



You can set Edge Gateway by using CLI (amsh) or by using the regular Linux console. This manual uses the method using CLI (amsh) for explanation. For setting methods, refer to the following manuals.

- ➔ Using CLI (amsh)  
"Chapter 5, Operation of Configuration Files" in the "Edge Gateway amnimo G series User's Manual"
- ➔ Using the Linux console  
"13.1 Controlling the Configuration File" in the "Edge Gateway amnimo G series User's Manual"

#### 4.1.1 Default configuration file

The default configuration file for Edge Gateway is "startup-config". This file is loaded as a configuration file when you start Edge Gateway.

The configuration file (extension: dat) is saved with a file (extension: sha256) that contains the SHA-2 hash value as follows. They are saved in two places for redundant management.

```
admin@amnimo:~$ ls -l /mnt/config/area0/ ↵
total 22
drwx----- 2 root root 12288 Feb 15 1974 lost+found
-rw-r--r-- 1 root root 3588 Aug 7 10:57 startup-config.dat
-rw-r--r-- 1 root root 89 Aug 7 10:57 startup-config.sha256
admin@amnimo:~$ ls -l /mnt/config/area1/ ↵
total 22
drwx----- 2 root root 12288 Feb 15 1974 lost+found
-rw-r--r-- 1 root root 3588 Aug 7 10:57 startup-config.dat
-rw-r--r-- 1 root root 89 Aug 7 10:57 startup-config.sha256
```

#### 4.1.2 Backing up the configuration file

Save the backup file using the function that renames the configuration file.

- ➔ Refer to "5.6 Changing the Name of a Configuration File" in the "Edge Gateway amnimo G series User's Manual" for details.

##### Example

In administrator mode, save the startup-config with the name backup-config.

**Admin Mode**

```
amnimo# config file copy startup-config backup-config ↵
```

Check the backed up configuration file.

```

admin@amnimo:~$ ls -l /mnt/config/area0/ ↵
total 22
-rw-rw-rw- 1 root root  3588 Aug  7 11:44 backup-config.dat
-rw-rw-rw- 1 root root    89 Aug  7 11:44 backup-config.sha256
drwx----- 2 root root 12288 Feb 15  1974 lost+found
-rw-r--r-- 1 root root  3588 Aug  7 10:57 startup-config.dat
-rw-r--r-- 1 root root    89 Aug  7 10:57 startup-config.sha256
admin@amnimo:~$ ls -l /mnt/config/area1/
total 22
-rw-rw-rw- 1 root root  3588 Aug  7 11:44 backup-config.dat
-rw-rw-rw- 1 root root    89 Aug  7 11:44 backup-config.sha256
drwx----- 2 root root 12288 Feb 15  1974 lost+found
-rw-r--r-- 1 root root  3588 Aug  7 10:57 startup-config.dat
-rw-r--r-- 1 root root    89 Aug  7 10:57 startup-config.sha256

```

gateway

### 4.1.3 Managing generations of configuration files

Users can manage generations of the configuration files by saving them under any file name.

#### Example

Change the settings in administrator mode, then save each setting with the names "pattern-A-config", "pattern-B-config", and "pattern-C-config". Finally, display a list of the configuration files.

#### Admin Mode

```

After completing the various settings, execute the following
amnimo# config file save pattern-A-config ↵

```

```

After completing some settings, execute the following
amnimo# config file save pattern-B-config ↵

```

```

After completing some settings, execute the following
amnimo# config file save pattern-C-config ↵
amnimo# show config file ↵
startup-config 2020-01-02T00:00:00+09:00
pattern-A-config 2020-01-01T00:00:05+09:00
pattern-B-config 2020-01-01T00:00:10+09:00
pattern-C-config 2020-01-01T00:00:12+09:00

```

gateway

## 4.1.4 Reflecting the configuration file

You can load the saved file and reflect it to Edge Gateway.

### ■ When multiple configuration files exist on the same device

#### Example

Load one configuration file (pattern-A-config) saved in "4.1.3" in administrator mode and reflect the settings.

#### Admin Mode

```
gateway
amnimo# config file load pattern-A-config ↵ ← Load the settings
amnimo# config file save ↵ ← Save the loaded settings as startup-c
onfig and reflect it as the settings for the next startup
startup-config file already exists. Do you want to overwrite? (y/N): y ↵
amnimo# reboot type soft ↵ ← Execute a software reboot
```

### ■ When you use another Edge Gateway's configuration file

#### Example

Load the configuration file (/media/sdcard/pattern-D-config) that was transferred from another Edge Gateway using the external command amcfg instead of amsh.

```
gateway
admin@amnimo:~$ sudo amcfg load /media/sdcard/pattern-D-config ↵ ← Load the settings
admin@amnimo:~$ sudo amcfg save ↵ ← Save the loaded settings as startup-config and reflect it
as the settings for the next startup
startup-config file already exists. Do you want to overwrite? (y/N): y
admin@amnimo:~$ sudo amctrl reboot -t soft ↵ ← Execute a software reboot
```

## 4.2 Troubleshooting for Cases of Startup Failure

You may need user-installed applications to develop an application for Edge Gateway. Under some circumstances, you could experience a situation where Linux does not start.

Since Edge Gateway has redundant areas, you can start it by switching the startup area by the boot loader in the event of a startup failure.

### 4.2.1 About the sample case

In the sample cases assumed in this section, the problem is addressed in the following sequence.

- Area 0 and area 1 were initially operated in the same configuration.
- The area 0 side had become unable to start properly while the application development on area 0 was in progress.
- Restore area 0 by reverting it to the same configuration as area 1.

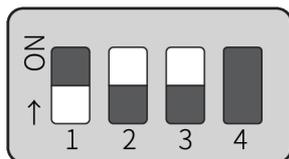
### 4.2.2 Step 1. Switching the startup areas on the boot loader

You can switch the startup areas by the boot loader. Set it on the serial console.

#### Starting up in U-Boot command mode

Before connecting the power, set the DIP switch to "U-Boot command mode". Then connect the power.

DIP switch settings for U-Boot command mode



When powered on, the following password input screen (input time: 10 seconds) is displayed. Enter the password and press the [ Enter ] key to log in.

```

gateway
TIM-1.0
WTMI-devel-18.12.1-118f0bd
WTMI: system early-init
SVC REV: 5, CPU VDD voltage: 1.108V
(Omitted)
STATUS:SN=[300002],MAC0=[E8:1B:4B:00:30:02],BS=[a:0 b:385 h:0 s:0],DIPBM=[ubootcommand]
am_show_board_status: CNTFRQ_EL0=12500000 Hz

Please enter password - autoboot in 10 sec... ← Enter the password and press the [
Enter ] key
Return to boot status(0x55) for login
Amnimo>> run stopwdt ← It is reset by Watchdog IC after a few minutes. If the o
peration takes time, execute the stop process.
Amnimo>>

```



- For the initial password, please contact our Customer Support.
- Password input failure is allowed up to three times. If the failure is repeated more than three times, U-Boot will start up in Linux startup mode.

## Switching the startup areas by U-Boot

You can switch the startup areas using the `ambootsw` command. Restart with the reset command after switching the startup areas.

### Syntax

```
ambootsw < get | set <0|1> >
```

### Setting item

Item	Details	
get	Check the startup area.	
	<b>Output format</b>	
	Boot Area: <b>AREA_NO</b> Boot Count: <b>BOOT_COUNT</b>	
	<b>Item</b>	<b>Display</b>
	AREA_NO	0 (1st Area)
		Area 0 Setting storage area : /dev/mmcblk0boot0 rootfs :           /dev/mmcblk0p1 userfs :           /dev/mmcblk0p3
		1 (2nd Area)
		Area 0 Setting storage area : /dev/mmcblk0boot0 rootfs :           /dev/mmcblk0p1 userfs :           /dev/mmcblk0p3
	BOOT_COUNT	Numerical value
		Number of startups
set AREA_NO	The startup area is switched.	
	<b>Item</b>	<b>Details</b>
	AREA_NO	Specify the number for the startup area.

### Example

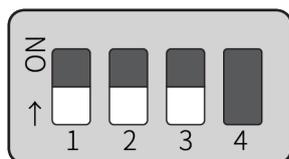
```
Amnimo>> ambootsw get ↵          ← Check the current startup area
Boot Area : 0 (1st Area)         ← Area 0
Boot Count : 385
Amnimo>> ambootsw set 1 ↵       ← Change the startup area to area 1
Amnimo>> ambootsw get ↵          ← Check the startup area again
Boot Area : 1 (2nd Area)        ← Area 1
Boot Count : 385
```

gateway

## Starting up in Linux startup mode

Before restarting, set to the "Linux boot command mode" by using the DIP switch and carry out the restart process.

DIP switch settings for Linux startup mode



```
Amnimo>> reset ↵          ↵ Restart
```

gateway

### 4.2.3 Step 2. Check the startup area after starting Linux

You can check the startup area after you start Linux. After performing a restart, make sure that it is the startup area that you have set on the U-Boot side.

➔ Refer to "4.4.6 Configuring a redundant area to boot" in the "Edge Gateway amnimo G series User's Manual" or "13.2.5 Controlling the startup area" in the "Edge Gateway amnimo G series User's Manual" for details.

#### Example

For the amctrl command, execute as follows.

```
admin@amnimo:~$ sudo amctrl boot ↵
AREA: 1
```

gateway

For amsh, execute as follows. The following is an example of executing a command in general user mode.

User Mode Admin Mode

```
amnimo$ show device boot ↵
1
```

gateway

## 4.2.4 Step 3. Synchronizing the firmware

Now that you have confirmed successful startup on the Area 1 side of the startup area, synchronize the firmware contents of the Area 1 side to the Area 0 side so that the Area 0 side starts successfully.

➔ Refer to "4.4.5 Synchronizing redundant areas of firmware" in the "Edge Gateway amnimo G series User's Manual" for details.

### Example

Admin Mode

```
gateway
amnimo# firmware area sync ↵
reboot to sync? (y/N):          ↵ ←Press the [ y ] key followed by the [ Enter ] key
amnimo# device boot 0 ↵       ↵ ← Revert the startup area to 0
amnimo# reboot type soft↵     ↵ ← Execute a software reboot
```



To cancel the execution of the firmware area sync command, press the [ Enter ] key, or press the [ n ] key followed by the [ Enter ] key.

## Revision history

1 <sup>st</sup> edition	Issued in November 2020
2 <sup>nd</sup> edition	Issued in December 2020
3 <sup>rd</sup> edition	Issued in March 2021
4 <sup>th</sup> edition	Issued in August 2021
5 <sup>th</sup> edition	Issued in January 2022
6 <sup>th</sup> edition	Issued in May 2022



Edge Gateway amnimo G series Developer's Manual  
6<sup>th</sup> edition, May 6, 2022

IM AMD03A01-51EN

---

All Rights Reserved. Copyright © 2020, amnimo Inc