

# node-eye マニュアル

Armadillo-IoT ゲートウェイ G3L

Version 1.1.1  
2018/11/28

株式会社アットマークテクノ [<http://www.atmark-techno.com>]

Armadillo サイト [<http://armadillo.atmark-techno.com>]

---

# node-eye マニュアル

株式会社アットマークテクノ

製作著作 © 2016-2018 Atmark Techno, Inc.

Version 1.1.1  
2018/11/28

# 目次

1. はじめに	7
1.1. 本書の対応製品について	7
1.2. 表記について	7
1.2.1. フォント	7
1.2.2. コマンド入力例	7
1.2.3. アイコン	8
2. node-eye とは	9
2.1. WebUI の概要	9
3. node-eye を体験する	11
3.1. 準備	11
3.2. node-eye コントロールパネルを使う	16
3.2.1. node-eye コントロールパネルにログインする	16
3.2.2. 全体の稼働状況を見る	16
3.2.3. 各デバイスの稼働状況を見る	16
3.2.4. リソースモニタリングを行う	18
3.2.5. ファームウェアアップデートを行う	19
3.2.6. ログを見る	23
3.2.7. アカウントの追加/変更/削除を行う	24
3.3. SACM を使う	29
3.3.1. SACM へログインする	29
3.3.2. リモートコンフィグを行う	30
3.3.3. フォルダ機能を使う	35
3.3.4. 接続状態が変化した時の通知先を設定する	40
3.3.5. その他	49
4. Armadillo ソフトウェア仕様	50
4.1. メモリマップ	50
4.2. node-eye パッケージ	50
4.2.1. node-eye パッケージのインストール・アンインストール	51
4.2.2. node-eye.service の有効化・無効化	51
4.3. armsd	51
4.3.1. 死活監視機能	51
4.3.2. リモートコンフィグ機能	52
4.3.2.1. モジュールについて	52
4.3.3. その他・詳細仕様	53
4.4. アットマークテクノが提供するモジュールの仕様	53
4.4.1. ネットワークモジュール	53
4.4.1.1. ネットワークインターフェース設定シーケンス	53
4.4.1.2. コンフィグの書式	54
4.4.1.3. ネットワークモジュールコンフィグ一覧	54
4.4.2. ファームウェアモジュール	56
4.5. リカバリーイメージ	56
4.5.1. リカバリーイメージの目的	56
4.5.2. リカバリーイメージの起動条件	57
4.5.3. リカバリーイメージの特徴	57
4.5.3.1. インストールされているアプリケーション	57
4.5.3.2. リカバリーイメージ起動中の WebUI 上の表示	57
4.5.3.3. RAM disk に展開されるユーザーランド	57
4.5.3.4. リカバリーイメージが使用する line.conf	57
4.5.4. リカバリーイメージのビルド方法	57
4.5.5. リカバリーイメージの書き込み方法	59

---

5. Howto .....	61
5.1. ネットワーク設定方法 .....	61
5.1.1. line.conf .....	61
5.1.1.1. line.conf の設定方法 .....	61
5.1.1.2. line.conf コンフィグ一覧 .....	62
5.1.1.3. 設定例 .....	63
5.1.2. ネットワークモジュールコンフィグ .....	64
5.1.2.1. 使用可能な key 一覧 .....	64
5.1.2.2. 設定例 .....	64
5.2. リカバリーイメージを使って Armadillo を復旧する .....	66
5.3. リカバリーイメージの起動をテストする .....	66
5.4. node-eye インストール済のユーザーランド ビルド方法 .....	68
5.5. モジュールの作成方法 .....	71
5.5.1. モジュールの配置 .....	71
5.5.2. モジュールのサンプル .....	71
5.5.3. モジュールのオペレーションの種類 .....	73
5.5.4. モジュールの設計の注意点 .....	73
5.6. テンプレートセットを活用した複数 Armadillo の一括設定 .....	73
6. 機能制限 .....	74

## 目次

3.1. ダッシュボード (node-eye コントロールパネル) .....	16
3.2. デバイス一覧 (node-eye コントロールパネル) .....	17
3.3. デバイス詳細 (node-eye コントロールパネル) .....	17
3.4. デバイス複数選択 (node-eye コントロールパネル) .....	18
3.5. デバイス複数選択 (node-eye コントロールパネル) .....	19
3.6. リソースモニタリング (node-eye コントロールパネル) .....	19
3.7. デバイス一覧 (node-eye コントロールパネル) .....	21
3.8. ファームウェアアップデート (node-eye コントロールパネル) .....	21
3.9. ファームウェアアップデート実行 (node-eye コントロールパネル) .....	22
3.10. タスク (node-eye コントロールパネル) .....	23
3.11. イベント (node-eye コントロールパネル) .....	24
3.12. コンフィグ反映 (SACM コントロールパネル) .....	33
3.13. 即時反映 (SACM コントロールパネル) .....	34
3.14. フォルダタブ (SACM コントロールパネル) .....	35
3.15. フォルダ名入力 (SACM コントロールパネル) .....	36
3.16. フォルダ一覧 (SACM コントロールパネル) .....	37
3.17. 所属 SA 変更 (SACM コントロールパネル) .....	38
3.18. フォルダに追加する SA を選択 (SACM コントロールパネル) .....	39
3.19. SA が追加されたフォルダ (SACM コントロールパネル) .....	40
3.20. 監視タブ (SACM コントロールパネル) .....	41
3.21. 監視グループ名入力 (SACM コントロールパネル) .....	42
3.22. 監視グループが追加された監視タブ (SACM コントロールパネル) .....	43
3.23. 監視グループの詳細 (SACM コントロールパネル) .....	44
3.24. 通知先メールアドレス入力 (SACM コントロールパネル) .....	45
3.25. 通知先メールアドレスが追加された監視グループ詳細 (SACM コントロールパネル) .....	46
3.26. 監視グループ詳細 (SACM コントロールパネル) .....	47
3.27. 監視グループに追加する SA を選択 (SACM コントロールパネル) .....	48
3.28. SA が追加された監視グループ (SACM コントロールパネル) .....	49
4.1. リカバリーイメージアップデート実行 (node-eye コントロールパネル) .....	60
5.1. ファームウェアアップデート (node-eye コントロールパネル) .....	67

## 表目次

1.1. 使用しているフォント .....	7
1.2. 表示プロンプトと実行環境の関係 .....	8
1.3. コマンド入力例での省略表記 .....	8
2.1. WebUI の機能比較 .....	9
3.1. node-eye を体験するためのチェックリスト .....	11
3.2. Linux と ATDE のバージョン関係 .....	11
3.3. node-eye の要求バージョン(Linux 3.14) .....	14
3.4. node-eye の要求バージョン(Linux 4.9) .....	14
3.5. ファームウェアアップデートのためにダウンロードするファイル一覧(Linux 3.14) .....	20
3.6. ファームウェアアップデートのためにダウンロードするファイル一覧(linux4.9) .....	20
3.7. ファームウェアアップデート .....	22
4.1. QSPI フラッシュメモリ メモリマップ .....	50
4.2. eMMC メモリマップ .....	50
4.3. モジュールのタイプとその用途 .....	52
4.4. アットマークテクノが提供しているモジュール一覧 .....	53
4.5. ネットワークモジュールの eth0 に関する key 一覧 .....	54
4.6. ネットワークモジュールの wwan に関する key 一覧 .....	55
4.7. ネットワークモジュールの wlan0 に関する key 一覧 .....	55
4.8. ネットワークモジュールの resolv.conf に関する key 一覧 .....	56
4.9. ファームウェアアップデートに必要なファイル .....	56
4.10. リカバリーイメージアップデート .....	59
5.1. ネットワーク設定ファイル .....	61
5.2. eth0 で使用可能な key 一覧 .....	62
5.3. wwan で使用可能な key 一覧 .....	62
5.4. wlan0 で使用可能な key 一覧 .....	63
5.5. resolv で使用可能な key 一覧 .....	63
5.6. 各モジュールの case 文に書く事のできるオペレーションの種類と実行条件 .....	73

# 1. はじめに

---

## 1.1. 本書の対応製品について

本書は以下の Armadillo に対応しています。その他 Armadillo では本書をご利用いただけません。

- ・ Armadillo-IoT ゲートウェイ G3L



## 1.2. 表記について

### 1.2.1. フォント

本書では以下のような意味でフォントを使いわけています。

表 1.1 使用しているフォント

フォント例	説明
本文中のフォント	本文
[PC ~]\$ ls	プロンプトとユーザ入力文字列
text	編集する文字列や出力される文字列。またはコメント

### 1.2.2. コマンド入力例

本書に記載されているコマンドの入力例は、表示されているプロンプトによって、それぞれに対応した実行環境を想定して書かれています。「/」の部分はカレントディレクトリによって異なります。各ユーザのホームディレクトリは「~」で表わします。

表 1.2 表示プロンプトと実行環境の関係

プロンプト	コマンドの実行環境
[PC /]#	作業用 PC 上の root ユーザで実行
[PC /]\$	作業用 PC 上の一般ユーザで実行
[ATDE/ ^]#	ATDE 上の root ユーザで実行
[ATDE/ ^]\$	ATDE 上の一般ユーザで実行
[armadillo /]#	Armadillo 上の root ユーザで実行
[armadillo /]\$	Armadillo 上の一般ユーザで実行
=>	Armadillo 上の保守モードで実行

コマンド中で、変更の可能性のあるものや、環境により異なるものに関しては以下のように表記します。適時読み替えて入力してください。

表 1.3 コマンド入力例での省略表記

表記	説明
[version]	ファイルのバージョン番号

### 1.2.3. アイコン

本書では以下のようにアイコンを使用しています。



注意事項を記載します。



役に立つ情報を記載します。



## 2. node-eye とは

node-eye(ノード・アイ)は、IoT ゲートウェイの稼働状況を確認するためのリソースの監視・管理機能、ファームウェアの機能追加やバグ修正に対応するリモートアップデート機能など、IoT ゲートウェイの基本的な運用・管理機能を一括して提供するサービスです。node-eye を使ったシステムのユースケース、サービスのオプションについては node-eye サイトをご覧ください。

node-eye はデバイス運用管理機能 ユーザーインターフェースとして、開発向けの SACM コントロールパネルと運用/保守向けの node-eye コントロールパネルという 2 種類の WebUI を用意しています。WebUI に表示されるデバイスの状態表示や、ファームウェアの更新等の操作は、Armadillo にインストールされた armsd <sup>[1]</sup> が SACM と相互に通信を行う事で実現しています。



node-eye では eMMC に書き込む kernel、userland、recovery イメージファイルをファームウェアと総称しています。

node-eye サイト

<http://node-eye.com>

### 2.1. WebUI の概要

node-eye のデバイス運用管理サービスでは node-eye コントロールパネル と SACM コントロールパネル の 2 つの WebUI を用意しています。node-eye コントロールパネルは、デバイス状況の把握、ファームウェアアップデートなどが行える運用/保守向けのシンプルな UI となっています。SACM コントロールパネル は、モジュールのリモートコンフィグ、監視対象のデバイスの表示名の変更、電子メールによるネットワークからの接続/切断状態の通知を設定することができる開発向けの UI となっています。

表 2.1 WebUI の機能比較

対象	node-eye コントロールパネル	SACM コントロールパネル
	運用/保守向け	開発向け
Ping/Traceroute/再起動	✓	✓
ログ(イベント/タスク)の表示	✓	✓
リソースグラフ表示	✓	✓
ファームウェアアップデート	✓	
電子メールによる接続状態通知		✓
デバイスのラベル(表示名)の変更		✓
モジュールのリモートコンフィグ		✓

<sup>[1]</sup>armsd は、SACM との間で通信を行うクライアントプログラムです。armsd の詳しい情報は クライアント実装 - libarms documentation [[https://dev.smf.jp/libarms/armsd/armsd\\_top.html](https://dev.smf.jp/libarms/armsd/armsd_top.html)] を参照してください。



node-eye では Armadillo で動作する監視対象のアプリケーションをモジュールという単位で管理します。

## 3. node-eye を体験する

本章ではトライアルをご契約されたお客様を対象に、Armadillo-IoT ゲートウェイ G3L (以降、Armadillo と表記)を使って node-eye の基本的な機能を使う手順を説明します。

以降の手順を行うために、次のものを用意してください。

表 3.1 node-eye を体験するためのチェックリスト

用意するもの	チェック
ATDE6 または ATDE7	
Armadillo-IoT ゲートウェイ G3L 開発セット	
node-eye 登録完了通知書に記載のアカウント情報	
DHCP サーバーが稼動している有線 LAN ネットワーク	

導入する ATDE のバージョンは、お使いの Armadillo にインストールされている[Linux][ユーザーランド]のバージョンに合わせて決定します。以下の「表 3.2. Linux と ATDE のバージョン関係」を確認し、適切なバージョンの ATDE を用意してください。

表 3.2 Linux と ATDE のバージョン関係

Linux/ユーザーランド バージョン	ATDE バージョン
Linux3.14/Jessie	ATDE6
Linux4.9/Stretch	ATDE7

本手順を行う前に、Armadillo-IoT ゲートウェイ G3L の製品マニュアルに記載された操作を一通り確認する事をお勧めします。

Armadillo-IoT ゲートウェイ G3L ドキュメント・ダウンロード

<http://armadillo.atmark-techno.com/armadillo-iot-g3l/downloads>

### 3.1. 準備

以下の手順で、node-eye を利用するための準備を行います。

1. SACM コントロールパネルで Armadillo のネットワーク設定を行う
2. ブートローダーおよびカーネルバージョンの確認
3. node-eye パッケージのインストール
4. armsd 起動

#### 手順 3.1 準備

1. SACM コントロールパネルで Armadillo のネットワーク設定を行う

SACM コントロールパネルにログインし、対象の Armadillo のネットワーク設定を行います。ログインするためのアカウントは登録完了通知書に記載されています。

## SACM コントロールパネル

<https://atmark-techno.sacm.jp/>

### 1. メニューから[SA]をクリックし、いずれかの Armadillo をクリック

The screenshot shows the node-eye SACM control panel. At the top, there is a navigation menu with buttons for 'ダッシュボード', 'SA', 'フォルダ', 'コンフィグ', 'オペレーション', 'タスク', 'イベント', '監視', and '管理'. The 'SA' button is highlighted. Below the navigation menu, there is a header for 'SA' and a search bar with 'SAラベル・コード' entered. The main content area displays a list of Armadillo devices. The device 'AloT-001' is circled in red. The list includes columns for device ID, name, and status. The status for 'AloT-001' is '接続中' (Connected).

Device ID	Device Name	Status	Group
AloT-006	tss18740226 不明なデバイス	切断中	アットマークグループ 未所属 開発用グループ
AloT-004	tss18740233 Armadillo-IoT G2	切断中	未所属 未所属 開発用グループ
AloT-007	tss18740264 Armadillo-IoT G2	切断中	node-eyeグループ 未所属 開発用グループ
AloT-005	tss18740271 不明なデバイス	切断中	node-eyeグループ 未所属 未所属
AloT-001	tss21373220 Armadillo-IoT G2	接続中	アットマークグループ 未所属 アットマークテストグループ
AloT-003	tss21373251 Armadillo-IoT G2	切断中	アットマークグループ 未所属 アットマークテストグループ
AloT-002	tss21373268 Armadillo-IoT G2	接続中	アットマークグループ 未所属 アットマークテストグループ
AloT-000	tss21545443 不明なデバイス	切断中	未所属 未所属 未所属

- SA のメニューから[コンフィグ]をクリックして、ページ下部の[コンフィグ変更]をクリック

The screenshot shows the node-eye interface for SA AIoT-001. The 'Config' tab is selected and circled in red. Below the table, the 'Config Change' button is also circled in red.

モジュールID	作業コンフィグ	次回起動時コンフィグ	動作コンフィグ
モジュール 0	バージョン: 0.0.0-0 CLI module A	バージョン: 0.0.0-0 CLI module A	バージョン: 0.0.0-0 CLI module A
モジュール 1	バージョン: 0.0.0-0 CLI module B	バージョン: 0.0.0-0 CLI module B	バージョン: 0.0.0-0 CLI module B
モジュール 2	バージョン: 0.0.0-0 CLI module C	バージョン: 0.0.0-0 CLI module C	バージョン: 0.0.0-0 CLI module C
モジュール 3	バージョン: 0.0.0-0 CLI module D	バージョン: 0.0.0-0 CLI module D	バージョン: 0.0.0-0 CLI module D
モジュール 4	バージョン: 0.0.0-0 CLI module E	バージョン: 0.0.0-0 CLI module E	バージョン: 0.0.0-0 CLI module E
モジュール 5	バージョン: 0.0.0-0 CLI module F	バージョン: 0.0.0-0 CLI module F	バージョン: 0.0.0-0 CLI module F
モジュール 6	バージョン: 0.0.0-0 CLI module G	バージョン: 0.0.0-0 CLI module G	バージョン: 0.0.0-0 CLI module G
モジュール 7	バージョン: 0.0.0-0 CLI module H	バージョン: 0.0.0-0 CLI module H	バージョン: 0.0.0-0 CLI module H
モジュール 8	バージョン: 0.0.0-0 BIN module A	バージョン: 0.0.0-0 BIN module A	バージョン: 0.0.0-0 BIN module A
モジュール 9	バージョン: 0.0.0-0 BIN module B	バージョン: 0.0.0-0 BIN module B	バージョン: 0.0.0-0 BIN module B
モジュール 10	バージョン: 0.0.0-0 BIN module C	バージョン: 0.0.0-0 BIN module C	バージョン: 0.0.0-0 BIN module C
モジュール 11	バージョン: 0.0.0-0 BIN module D	バージョン: 0.0.0-0 BIN module D	バージョン: 0.0.0-0 BIN module D
モジュール 12	バージョン: 0.0.0-0 CLI module I	バージョン: 0.0.0-0 CLI module I	バージョン: 0.0.0-0 CLI module I
モジュール 13	バージョン: 0.0.0-0 CLI module J	バージョン: 0.0.0-0 CLI module J	バージョン: 0.0.0-0 CLI module J
モジュール 14	バージョン: 0.0.0-0 CLI module K	バージョン: 0.0.0-0 CLI module K	バージョン: 0.0.0-0 CLI module K
モジュール 15	バージョン: 0.0.0-0 CLI module L	バージョン: 0.0.0-0 CLI module L	バージョン: 0.0.0-0 CLI module L

コンフィグ変更  
 コンフィグ変更  
 コンフィグ表示  
 コンフィグ表示  
 反映予定コンフィグ表示  
 反映予定コンフィグ表示  
 コンフィグ反映  
 コンフィグ反映

- モジュール 0 のコンフィグを以下のように書き換えます。

```
VERSION=2
interface_eth0=enable
interface_eth0_type=dhcp
```

- [コンフィグ変更]をクリックし、[コンフィグ反映]をクリックします。

5. 反映スケジュールを[即時反映]にチェックを入れて、[コンフィグ反映]をクリックします。



## 2. ブートローダーおよびカーネルバージョンの確認

次に、Armadillo に書き込まれているブートローダーおよびカーネルのバージョン確認を行います。

node-eye を利用するためには下記に示すバージョンより新しい必要があります。お使いのLinux カーネルバージョンに合わせて要求バージョンを確認してください。

表 3.3 node-eye の要求バージョン(Linux 3.14)

イメージファイル	ファイル名	要求バージョン
ブートローダー	u-boot-x1-atX.bin	at3
カーネル	ulmage-x1-vX.XX	v3.00

表 3.4 node-eye の要求バージョン(Linux 4.9)

イメージファイル	ファイル名	要求バージョン
ブートローダー	u-boot-x1-atXX.bin	at17
カーネル	ulmage-x1-v4.9-atX	at3

Armadillo と 作業用 PC(ATDE) を USB シリアルケーブルで接続し、minicom などのシリアル通信ソフトウェアを起動してください。

ブートローダーのバージョンを確認するには、保守モードで下記のコマンドを実行してください。

```
=> version
```

```
U-Boot 2016.07-at3 (Sep 20 2016 - 10:13:10 +0900)
arm-linux-gnueabi-hf-gcc ( 4.9.2-10) 4.9.2
GNU ld (GNU Binutils for Debian) 2.25
```

カーネルのバージョンを確認するには、Debian ユーザーランド起動後、下記のコマンドを実行してください。

```
[armadillo ~]# uname -a
Linux armadillo 3.14.76-at3 #1080 SMP PREEMPT Fri Sep 16 17:15:59 JST 2016 armv7l GNU/
Linux
```

表示されたバージョンが「表 3.3. node-eye の要求バージョン(Linux 3.14)」よりも低い場合は、最新のインストールディスクなどでイメージの書き換えを行ってから次の手順に進んでください。

### 3. node-eye パッケージのインストール

次に、SACM と通信を行うためのクライアントプログラム(armsd)が含まれている node-eye パッケージを Armadillo にインストールします。node-eye パッケージには armsd の他、リモートコンフィグ、ファームウェアアップデートに必要なスクリプトなどがまとめられています。

node-eye パッケージおよび含まれるソフトウェアに関して、詳しくは「4.2. node-eye パッケージ」を参照してください。

node-eye パッケージのインストールは apt-get コマンドを利用して行います。下記コマンドを実行して、node-eye パッケージをインストールしてください。

```
[armadillo ~]# apt-get install node-eye
```

インストールが完了するとパッケージリストに node-eye が増えていることが確認できます。

```
[armadillo ~]# dpkg -l | grep node-eye
```

### 4. armsd 起動

最後に、下記コマンドを実行して armsd を起動します。armsd はデーモンとして起動するためプロンプトがすぐに戻ってきます。

```
[armadillo ~]# armsd
```

ps コマンドで armsd が起動していることを確認できます。

```
[armadillo ~]# ps ax
```

armsd が SACM と通信を行いネットワーク設定が完了すると eth0 が DHCP 接続になります。

## 3.2. node-eye コントロールパネルを使う

### 3.2.1. node-eye コントロールパネルにログインする

http://apps.node-eye.com/device-management/ にアクセスし、**node-eye 登録完了通知書**に記載されている node-eye コントロールパネルアカウントを使用してログインしてください。



**node-eye 登録完了通知書**に記載しているアカウントは管理者(admin)権限です。第3者に漏れることの無いよう厳重に保管してください。アカウントの追加/削除権限を持たないオペレーターアカウントは、本文「3.2.7. アカウントの追加/変更/削除を行う」の手順で追加することができます。

### 3.2.2. 全体の稼働状況を見る

node-eye コントロールパネルにアクセスするとダッシュボードが表示されます。

The screenshot shows the node-eye dashboard with a yellow header containing navigation links: node-eye ダッシュボード デバイス ログ アカウント and a user profile 'atmark'. The main content is divided into three sections:

- 接続状態 (Connection Status):** A table showing connection counts for different groups.
 

グループ名	接続中	切断中
未設定	0	2
アットマークグループ	3	1
node-eyeグループ	0	2
全体	3	5
- 最近のタスク (Recent Tasks):** A table listing recent tasks with columns for Command, Device, Status, and Execution Time.
 

コマンド	デバイス	状態	実行日時
ステータス参照	AloT-003	成功	2016-01-14 14:01:03
ステータス参照	AloT-003	成功	2016-01-14 14:00:00
ステータス参照	AloT-003	成功	2016-01-14 13:58:57
ステータス参照	AloT-003	成功	2016-01-14 13:58:22
ステータス参照	AloT-003	成功	2016-01-14 13:58:03
- 最近のイベント (Recent Events):** A table listing recent events with columns for Category, Device, Details, and Date/Time.
 

種別	デバイス	詳細	日時
リクエストイベント	AloT-003	リクエストの中継サーバでの処理開始	2016-01-14 13:21:03
リクエストイベント	AloT-003	リクエストのサービスアダプタでの処理開始	2016-01-14 13:21:03
リクエストイベント	AloT-003	リクエストの処理成功	2016-01-14 13:21:03
リクエストイベント	AloT-003	リクエストの中継サーバでの処理開始	2016-01-14 13:21:21

図 3.1 ダッシュボード(node-eye コントロールパネル)

この画面では node-eye に登録されているデバイスの接続状況の一覧、直近 5 件のタスクおよびイベントが表示されます。

接続中/切断中の列に表示された台数をクリックすることで、それぞれの状態にあるデバイスの一覧を表示することができます。

### 3.2.3. 各デバイスの稼働状況を見る

メニューから[デバイス]をクリックするとデバイスの一覧を見ることができます。



グループ	デバイス	グループ	接続状態
すべて	<input type="checkbox"/> AloT-006 SAコード: tss18740226 / ディストリビューションID: 0001-0000-000D-0001-0000-0000-0000-0016	アットマークグループ	接続中
すべて	<input type="checkbox"/> AloT-004 SAコード: tss18740233 / ディストリビューションID: 0001-0000-000D-0009-0000-0000-0000-0000		切断中
すべて	<input type="checkbox"/> AloT-007 SAコード: tss18740264 / ディストリビューションID: 0001-0000-000D-0009-0000-0000-0000-06FC	node-eyeグループ	切断中
	<input type="checkbox"/> AloT-005 SAコード: tss18740271 / ディストリビューションID: 0001-0000-000D-0001-0000-0000-0000-0019	node-eyeグループ	切断中
	<input type="checkbox"/> AloT-001 SAコード: tss21373220 / ディストリビューションID: 0001-0000-000D-0009-0000-0000-0000-1084	アットマークグループ	接続中
	<input type="checkbox"/> AloT-003 SAコード: tss21373251 / ディストリビューションID: 0001-0000-000D-0009-0000-0000-0000-1551	アットマークグループ	接続中
	<input type="checkbox"/> AloT-002 SAコード: tss21373268 / ディストリビューションID: 0001-0000-000D-0009-0000-0000-0000-1550	アットマークグループ	切断中
	<input type="checkbox"/> AloT-000 SAコード: tss21545443 / ディストリビューションID: 0001-0000-000D-0001-0000-0000-0000-001F		切断中

0 / 8 選択

ファームウェアアップデート 再起動 コマンド実行 リソース表示

図 3.2 デバイス一覧 (node-eye コントロールパネル)

この画面では、「グループ」「接続状態」「設定状態」「デバイス名/SAコード」で目的のデバイスを絞り込み、探すことができます。目的のデバイスをクリックすることで、デバイスの詳細な情報を確認できます。

← 戻る

AloT-001  
SAコード  
tss21373220  
ディストリビューションID  
0001-0000-000D-0009-0000-0000-0000-1084

編集

詳細 コマンド実行 ログ▼ リソースグラフ

グループ	アットマークグループ
テンプレート	(未設定)
監視	アットマークテストグループ
接続状態	接続中 (更新日: 2016-01-14 14:40:03)
	IPアドレス 121.101.71.133
	ポート 63271
	接続方式 接続持続型 (優先:)
設定状態	操作・コンフィグ有効
	初回 2015-12-16 19:20:32

図 3.3 デバイス詳細 (node-eye コントロールパネル)

また、複数のデバイスにチェックを入れ、一括で「ファームウェアアップデート」「再起動」「コマンド実行」「リソース表示」を行うことも可能です。

node-eye ダッシュボード デバイス ログ アカウント <span style="float: right;">atmark</span>			
グループ	デバイス	グループ	接続状態
すべて	<input checked="" type="checkbox"/> <b>AloT-001</b> SAコード: tss21373220 / ディストリビューションID: 0001-0000-0000-0009-0000-0000-1084	アットマークグループ	接続中
接続状態 すべて	<input checked="" type="checkbox"/> <b>AloT-002</b> SAコード: tss21373268 / ディストリビューションID: 0001-0000-0000-0009-0000-0000-1550	アットマークグループ	切断中
設定状態 すべて	<input checked="" type="checkbox"/> <b>AloT-003</b> SAコード: tss21373251 / ディストリビューションID: 0001-0000-0000-0009-0000-0000-1551	アットマークグループ	接続中
デバイス名 / SAコード	<input type="text"/>		
	<input type="checkbox"/> <b>AloT-006</b> SAコード: tss18740226 / ディストリビューションID: 0001-0000-0000-0001-0000-0000-0016	アットマークグループ	切断中
	<input type="checkbox"/> <b>AloT-007</b> SAコード: tss18740264 / ディストリビューションID: 0001-0000-0000-0009-0000-0000-06FC	node-eyeグループ	切断中
	<input type="checkbox"/> <b>AloT-005</b> SAコード: tss18740271 / ディストリビューションID: 0001-0000-0000-0001-0000-0000-0019	node-eyeグループ	切断中
	<input type="checkbox"/> <b>AloT-004</b> SAコード: tss18740233 / ディストリビューションID: 0001-0000-0000-0009-0000-0000-0000		切断中
	<input type="checkbox"/> <b>AloT-000</b> SAコード: tss21545443 / ディストリビューションID: 0001-0000-0000-0001-0000-0000-001F		切断中

3 / 8 選択

ファームウェアアップデート 再起動 コマンド実行 リソース表示

図 3.4 デバイス複数選択 (node-eye コントロールパネル)



ファームウェアアップデート実行中でも「再起動」を行うことができます。ただし、これを行うとプライマリイメージが起動しなくなることがあります。

### 3.2.4. リソースモニタリングを行う

メニューから[デバイス]を選択します リソースを確認したいデバイスのチェックボックスを選択します。

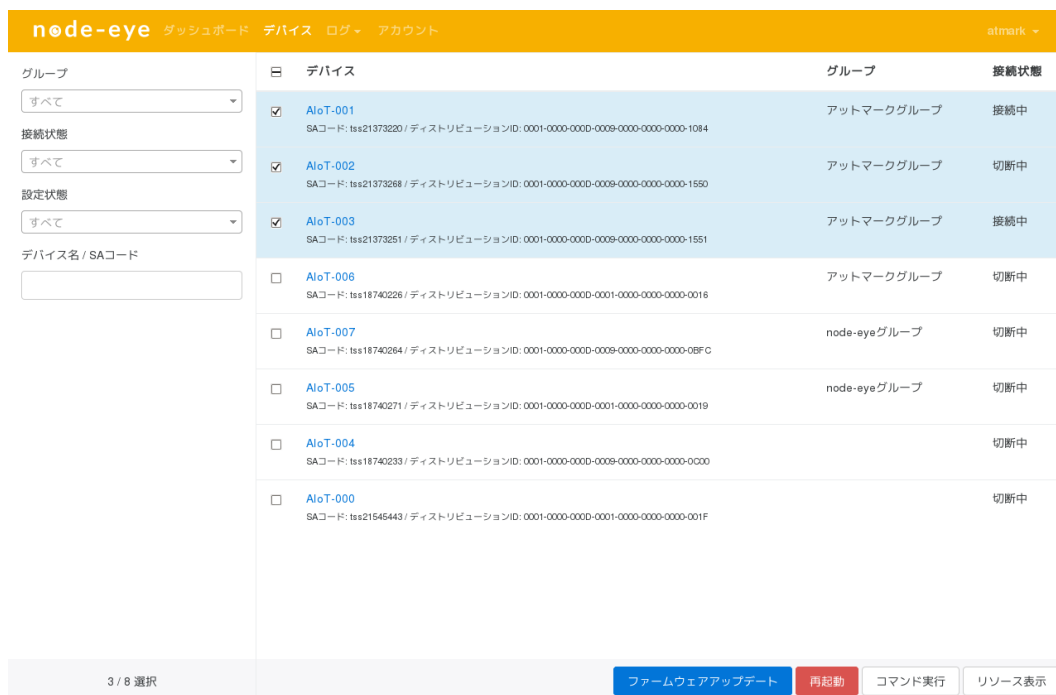


図 3.5 デバイス複数選択 (node-eye コントロールパネル)

画面下の[リソース表示]をクリックすると、CPU 使用率、転送量等のデバイスのリソースが表示されます。

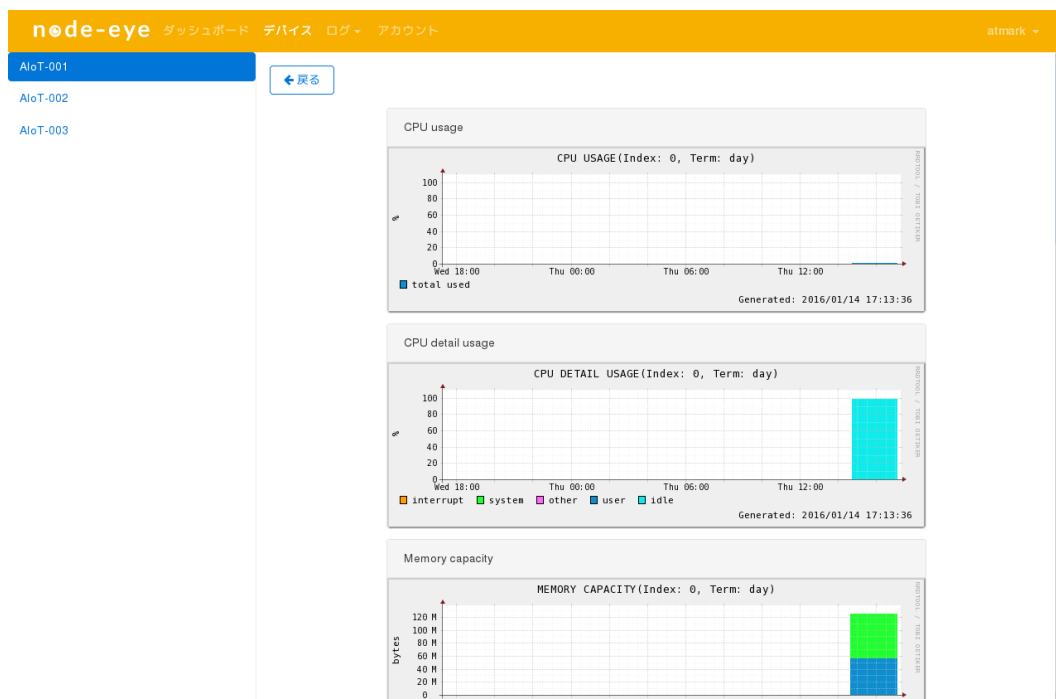


図 3.6 リソースモニタリング (node-eye コントロールパネル)

### 3.2.5. ファームウェアアップデートを行う

本章では、node-eye コントロールパネルを使って Armadillo のファームウェアアップデートを実行する方法を説明します。

ファームウェアアップデートで書き込むファイルは、Armadillo がアクセスすることのできる Web サーバーに公開されている必要があります。ここでは Web サーバーを ATDE として、標準 Linux カーネルイメージと DTB,ユーザーランドにアップデートを行います。

はじめに Web サーバー側の準備を行います。ATDE に必要なファイルを用意し、Linux カーネル,ユーザーランドのファイル名変更と ZIP 圧縮を行ってください。

導入したい Linux カーネルのバージョンによって、用意するファイル名が変わります。以下を確認し、必要なファイルを用意してください。

表 3.5 ファームウェアアップデートのためにダウンロードするファイル一覧(Linux 3.14)

リージョン	ファイル名
Linux 3.14 カーネル	ulmage-x1-vXX.00
Linux 3.14 Device Tree Blob	armadillo_iotg_g3l-vXX.00.dtb
Linux 3.14 ユーザーランド	debian-jessie-armhf_aiotg3l_YYYYMMDD.tar.gz

表 3.6 ファームウェアアップデートのためにダウンロードするファイル一覧(linux4.9)

リージョン	ファイル名
Linux 4.9 カーネル	ulmage-x1-v4.9-atX
Linux 4.9 Device Tree Blob	armadillo_iotg_g3l-v4.9-atX.dtb
Linux 4.9 ユーザーランド	debian-stretch-armhf_aiotg3l_YYYYMMDD.tar.gz

```
[ATDE ~]$ ls
armadillo_iotg_g3l-vXX.00.dtb  uImage-x1-vXX.00  debian-jessie-armhf_aiotg3l_YYYYMMDD.tar.gz
[ATDE ~]$ mv uImage-x1-vXX.00 uImage
[ATDE ~]$ mv armadillo_iotg_g3l-vXX.00.dtb armadillo_iotg_g3l.dtb
[ATDE ~]$ zip uImage.dtb.zip armadillo_iotg_g3l.dtb
[ATDE ~]$ mv debian-jessie-armhf_aiotg3l_YYYYMMDD.tar.gz debian-armhf.tar.gz
[ATDE ~]$ ls
armadillo_iotg_g3l.dtb  uImage  uImage.dtb.zip  debian-armhf.tar.gz
[ATDE ~]$ md5sum uImage > uImage.md5
[ATDE ~]$ md5sum uImage.dtb.zip > uImage.dtb.zip.md5
[ATDE ~]$ md5sum debian-armhf.tar.gz > debian-armhf.tar.gz.md5
[ATDE ~]$ ls
armadillo_iotg_g3l.dtb  uImage  uImage.dtb.zip  uImage.dtb.zip.md5  uImage.md5  debian-armhf.tar.gz
debian-armhf.tar.gz.md5
```

次に、用意したファイルを Armadillo がダウンロードできるように /var/www/html/ にコピーします。

```
[ATDE ~]$ sudo cp uImage* /var/www/html
[ATDE ~]$ sudo cp debian-armhf* /var/www/html
```



ファームウェアアップデートの仕様について詳しくは「4.4.2. ファームウェアモジュール」を参照してください。

次に、ブラウザで node-eye コントロールパネルを開き、メニューから[デバイス]を選択します

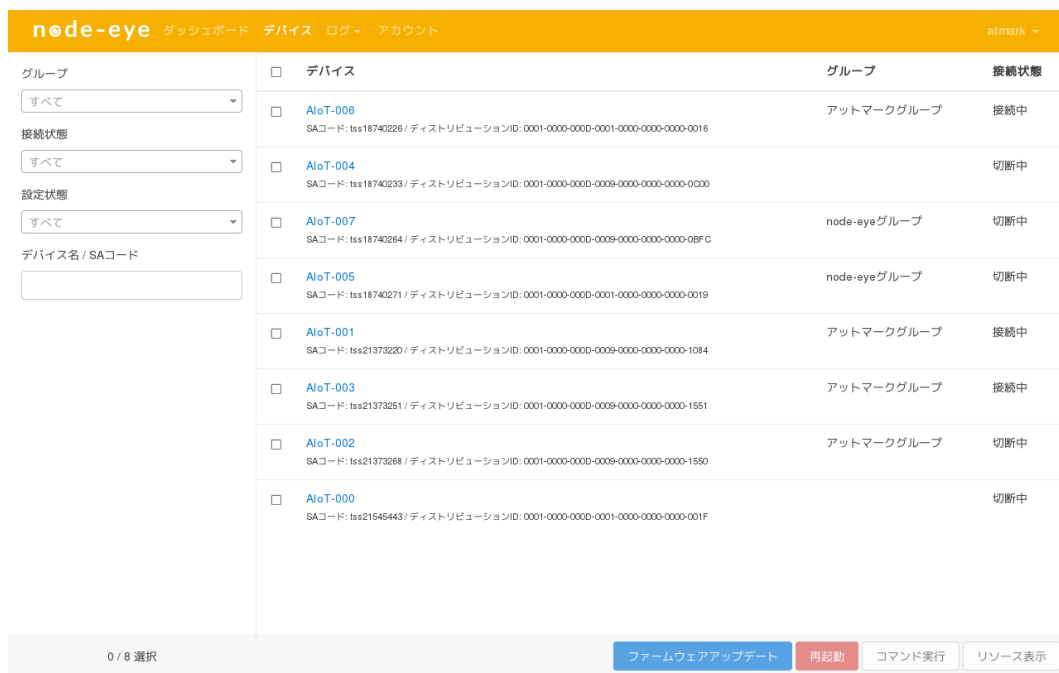


図 3.7 デバイス一覧 (node-eye コントロールパネル)

デバイスの中から[接続状態]が **接続中** になっているデバイスのチェックボックスを選択し、画面下の [ファームウェアアップデート] をクリックします。



図 3.8 ファームウェアアップデート (node-eye コントロールパネル)

下記のようにフォームへ入力してください([ATDE の IP アドレス]はご利用の環境にあわせて変更してください)。

表 3.7 ファームウェアアップデート

フォーム	入力内容
プライマリ/リカバリ	プライマリ
カーネル	http://[ATDE の IP アドレス]/ulmage
ユーザーランド	http://[ATDE の IP アドレス]/debian-armhf.tar.gz
アップデート完了時に再起動する	✓

「アップデート完了時に再起動する」 にチェックを入れて [実行] をクリックしファームウェアアップデートを開始します。<sup>[1]</sup>

図 3.9 ファームウェアアップデート実行 (node-eye コントロールパネル)



ファームウェアアップデートは node-eye コントロールパネルから与えた URL を使って、Armadillo がダウンロードを行います。上記以外の新たにビルドしたイメージファイルを使う場合は、Armadillo からアクセスすることのできる HTTP サーバーにファイルを配置してください。



ファームウェアアップデートは[カーネル][ユーザーランド]のどちらかを入力し、もう片方を空欄にすることで片方だけアップデートすることも可能です。



ファイルを HTTP サーバーに配置する際には、ファイルの MD5 チェックサムを "~.md5" という拡張子を付けて同階層からダウンロードできるようにしてください。MD5 チェックサムはファームウェアアップデートの際にファイルの破損を検知するために同時にダウンロードされます。

[1] 「実行時間を指定する」で過去の時間を指定した場合には、「すぐに実行する」と同じ動作になります。



「アップデート完了時に再起動する」にチェックを入れない場合は、Armadillo が再起動するまで古いファームウェアで稼働し続けます。

### 3.2.6. ログを見る

ログにはタスク、イベントの2つがあり、それぞれ、表示される内容が異なります。

[タスク]ではデバイスに対して再起動やコマンド実行を行った結果を参照することができます。[イベント]では「デバイスの状態が接続中から切断中に変化した」など、デバイスに関するイベントを参照することができます。

[タスク]の一覧を見るには、メニューから[ログ]->[タスク]をクリックしてください。

グループ	コマンド	タスク	デバイス	状態	実行日時	
すべて	コマンド	タスク	AioT-003	成功	2016-01-14 14:01:03	詳細
接続状態	コマンド	イベント	AioT-003	成功	2016-01-14 14:00:00	詳細
設定状態	ステータス参照		AioT-003	成功	2016-01-14 13:58:57	詳細
デバイス名 / SAコード	ステータス参照		AioT-003	成功	2016-01-14 13:58:22	詳細
	ステータス参照		AioT-003	成功	2016-01-14 13:58:03	詳細
	ステータス参照		AioT-003	成功	2016-01-14 13:57:52	詳細
	任意コマンド実行		AioT-003	成功	2016-01-14 13:57:45	詳細
	動作コンフィグ反映		AioT-003	成功	2016-01-14 13:57:41	詳細
	ステータス参照		AioT-003	成功	2016-01-14 13:36:50	詳細
	ステータス参照		AioT-003	成功	2016-01-14 13:35:47	詳細
	ステータス参照		AioT-003	成功	2016-01-14 13:34:44	詳細
	ステータス参照		AioT-003	成功	2016-01-14 13:33:41	詳細
	ステータス参照		AioT-003	成功	2016-01-14 13:32:39	詳細

図 3.10 タスク (node-eye コントロールパネル)

[イベント]の一覧を見るには、メニューから[ログ]->[イベント]をクリックしてください。

種別	タスク	デバイス	詳細	日時
リクエストイベント		AloT-003	リクエストの中継サーバでの処理開始	2016-01-14 13:21:03
リクエストイベント		AloT-003	リクエストのサービスアダプタでの処理開始	2016-01-14 13:21:03
リクエストイベント		AloT-003	リクエストの処理成功	2016-01-14 13:21:03
リクエストイベント		AloT-003	リクエストの中継サーバでの処理開始	2016-01-14 13:21:21
リクエストイベント		AloT-003	リクエストのサービスアダプタでの処理開始	2016-01-14 13:21:22
リクエストイベント		AloT-003	リクエストの処理成功	2016-01-14 13:21:22
リクエストイベント		AloT-003	リクエストの中継サーバでの処理開始	2016-01-14 13:21:56
リクエストイベント		AloT-003	リクエストのサービスアダプタでの処理開始	2016-01-14 13:21:57
リクエストイベント		AloT-003	リクエストの処理成功	2016-01-14 13:21:57
リクエストイベント		AloT-003	リクエストの中継サーバでの処理開始	2016-01-14 13:22:59
リクエストイベント		AloT-003	リクエストのサービスアダプタでの処理開始	2016-01-14 13:23:00
リクエストイベント		AloT-003	リクエストの処理成功	2016-01-14 13:23:00
リクエストイベント		AloT-003	リクエストの中継サーバでの処理開始	2016-01-14 13:24:02
リクエストイベント		AloT-003	リクエストのサービスアダプタでの処理開始	2016-01-14 13:24:03
リクエストイベント		AloT-003	リクエストの処理成功	2016-01-14 13:24:03

図 3.11 イベント (node-eye コントロールパネル)

### 3.2.7. アカウントの追加/変更/削除を行う

メニューから[アカウント]をクリックします。この画面から、オペレーター向けのアカウント作成を行います。

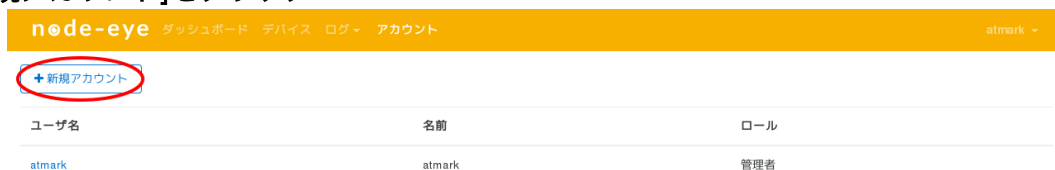


管理者権限のアカウントのみ、この操作が行えます オペレーター権限ではメニューに[アカウント]が表示されません



## 手順 3.2 アカウムの作成

### 1. [新規アカウント]をクリック



### 2. 項目を入力し[作成]をクリック

The screenshot shows the account creation form in the node-eye dashboard. The form has a yellow header bar and a blue '戻る' button. The form fields are: 'ユーザ名' (atmark-operator), 'パスワード' (masked with dots), 'パスワード (確認)' (masked with dots), and '名前' (アットマークテクノ オペレーター). A red circle highlights the '作成' button at the bottom left of the form.

ユーザ名  
atmark-operator

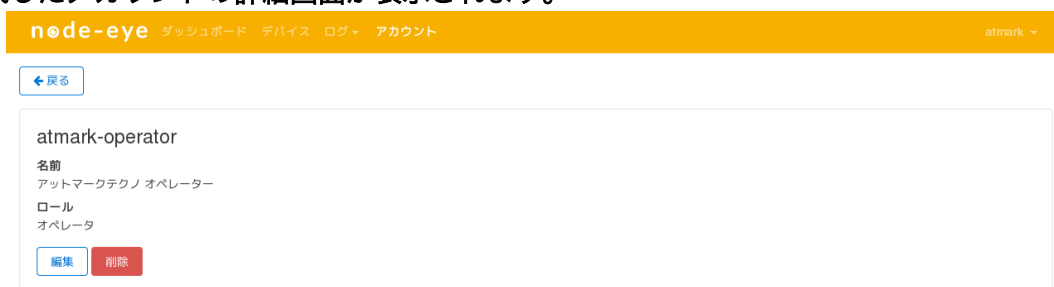
パスワード  
.....

パスワード (確認)  
.....

名前  
アットマークテクノ オペレーター

作成 キャンセル

### 3. 作成したアカウントの詳細画面が表示されます。



### 4. メニューの[アカウント]をクリックしてユーザー一覧を表示すると、作成したユーザーが増えていることがわかります。



## 手順 3.3 アカウントの編集

### 1. ユーザーをクリック



node-eye ダッシュボード デバイス ログアウト アカウント atmark

+ 新規アカウント

ユーザ名	名前	ロール
atmark	atmark	管理者
atmark-operator	アットマークテクノ オペレーター	オペレーター

### 2. [編集]をクリック



node-eye ダッシュボード デバイス ログアウト アカウント atmark

< 戻る

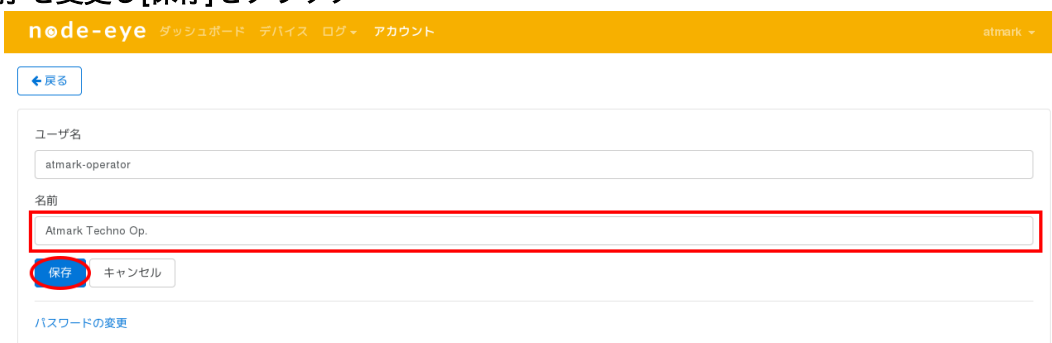
atmark-operator

名前  
アットマークテクノ オペレーター

ロール  
オペレーター

編集 削除

### 3. "名前"を変更し[保存]をクリック



node-eye ダッシュボード デバイス ログアウト アカウント atmark

← 戻る

ユーザ名  
atmark-operator

名前  
Atmark Techno Op.

保存 キャンセル

パスワードの変更

## 手順 3.4 アカウントの削除

### 1. ユーザーをクリック

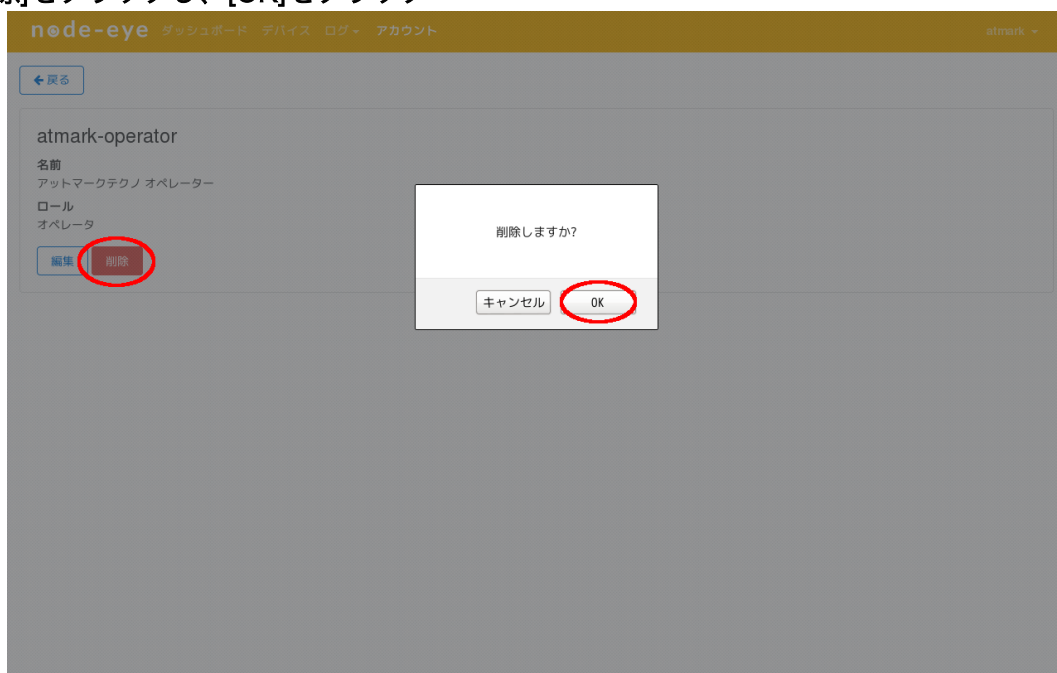


node-eye ダッシュボード デバイス ログアウト アカウント atmark

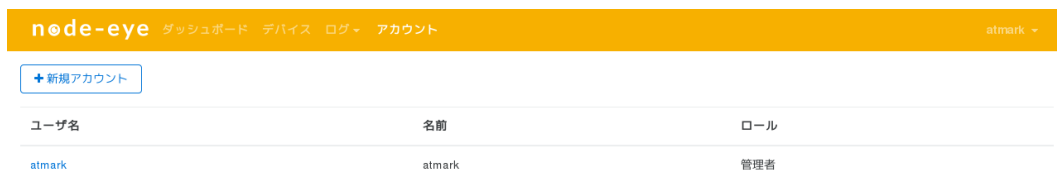
+ 新規アカウント

ユーザ名	名前	ロール
atmark	atmark	管理者
atmark-operator	アットマークテクノ オペレーター	オペレーター

## 2. [削除]をクリックし、[OK]をクリック



## 3. [OK]をクリックした後に、ユーザー一覧から選択したユーザーが消えていることが確認できます



## 3.3. SACM を使う

### 3.3.1. SACM へログインする

<https://atmark-techno.sacm.jp/user/> にアクセスし、**node-eye 登録完了通知書**に記載されている **SACM コントロールパネルアカウント**を使用してログインしてください。

## 3.3.2. リモートコンフィグを行う

本章では、DHCP で稼働してる Armadillo を固定 IP にする方法を例に、リモートコンフィグ機能の使い方を紹介します。その他の設定方法については本文「5. Howto」の「5.1. ネットワーク設定方法」を参照してください。

### 手順 3.5 リモートコンフィグを行う

1. メニューから[SA]をクリックし、[接続中]になっている Armadillo をクリック

The screenshot shows the node-eye web interface. At the top, there's a navigation bar with 'node-eye' logo and a user profile icon. Below it, a menu bar contains 'ダッシュボード', 'SA', 'フォルダ', 'コンフィグ', 'オペレーション', 'タスク', 'イベント', '監視', and '管理'. The 'SA' menu is active. Below the menu, there's a search and filter area with '表示形式: フラット', 'フィルタ: すべて表示', and '絞り込み: SAラベル・コード'. A pagination bar shows '1 - 8 / 8' and '20' items per page. The main content area lists several IoT devices (AIoT-006 to AIoT-000). The device 'AIoT-001' is circled in red and has a green checkmark and the text '接続中' (Connected) next to it. Other devices are marked as '切断中' (Disconnected) with a red 'x'.

SA	接続状態	デバイス名	グループ
AIoT-006	切断中	tss18740226 不明なデバイス	アットマークグループ 未所属 開発用グループ
AIoT-004	切断中	tss18740233 不明なデバイス	未所属 未所属 開発用グループ
AIoT-007	切断中	tss18740264 Armadillo-IoT G2	node-eyeグループ 未所属 開発用グループ
AIoT-005	切断中	tss18740271 不明なデバイス	node-eyeグループ 未所属 未所属
AIoT-001	接続中	tss21373220 Armadillo-IoT G2	アットマークグループ 未所属 アットマークテストグループ
AIoT-003	切断中	tss21373251 Armadillo-IoT G2	アットマークグループ 未所属 アットマークテストグループ
AIoT-002	接続中	tss21373268 Armadillo-IoT G2	アットマークグループ 未所属 アットマークテストグループ
AIoT-000	切断中	tss21545443 不明なデバイス	未所属 未所属 未所属

## 2. SA のメニューから[コンフィグ]をクリックして、ページ下部の[コンフィグ変更]をクリック

node-eye

ダッシュボード SA フォルダ コンフィグ オペレーション タスク イベント 監視 管理

表示形式: フラット  
フィルタ: すべて表示  
絞り込み: SAラベル・コード

SA / AIoT-001

AIoT-001  
SAコード: tss21373220 自デバイス: Armadillo-IoT G2  
Distribution ID: 0001-0000-0000-0009-0000-0000-0000-1084  
接続中 操作: コンフィグ有効 ★ アクティブバージョン済み

詳細情報 **コンフィグ** オペレーション タスク イベント グラフ

モジュールID	作業コンフィグ	次回起動時コンフィグ	動作コンフィグ
モジュール 0	バージョン: 0.0.0-0 CLI module A	バージョン: 0.0.0-0 CLI module A	バージョン: 0.0.0-0 CLI module A
モジュール 1	バージョン: 0.0.0-0 CLI module B	バージョン: 0.0.0-0 CLI module B	バージョン: 0.0.0-0 CLI module B
モジュール 2	バージョン: 0.0.0-0 CLI module C	バージョン: 0.0.0-0 CLI module C	バージョン: 0.0.0-0 CLI module C
モジュール 3	バージョン: 0.0.0-0 CLI module D	バージョン: 0.0.0-0 CLI module D	バージョン: 0.0.0-0 CLI module D
モジュール 4	バージョン: 0.0.0-0 CLI module E	バージョン: 0.0.0-0 CLI module E	バージョン: 0.0.0-0 CLI module E
モジュール 5	バージョン: 0.0.0-0 CLI module F	バージョン: 0.0.0-0 CLI module F	バージョン: 0.0.0-0 CLI module F
モジュール 6	バージョン: 0.0.0-0 CLI module G	バージョン: 0.0.0-0 CLI module G	バージョン: 0.0.0-0 CLI module G
モジュール 7	バージョン: 0.0.0-0 CLI module H	バージョン: 0.0.0-0 CLI module H	バージョン: 0.0.0-0 CLI module H
モジュール 8	バージョン: 0.0.0-0 BIN module A	バージョン: 0.0.0-0 BIN module A	バージョン: 0.0.0-0 BIN module A
モジュール 9	バージョン: 0.0.0-0 BIN module B	バージョン: 0.0.0-0 BIN module B	バージョン: 0.0.0-0 BIN module B
モジュール 10	バージョン: 0.0.0-0 BIN module C	バージョン: 0.0.0-0 BIN module C	バージョン: 0.0.0-0 BIN module C
モジュール 11	バージョン: 0.0.0-0 BIN module D	バージョン: 0.0.0-0 BIN module D	バージョン: 0.0.0-0 BIN module D
モジュール 12	バージョン: 0.0.0-0 CLI module I	バージョン: 0.0.0-0 CLI module I	バージョン: 0.0.0-0 CLI module I
モジュール 13	バージョン: 0.0.0-0 CLI module J	バージョン: 0.0.0-0 CLI module J	バージョン: 0.0.0-0 CLI module J
モジュール 14	バージョン: 0.0.0-0 CLI module K	バージョン: 0.0.0-0 CLI module K	バージョン: 0.0.0-0 CLI module K
モジュール 15	バージョン: 0.0.0-0 CLI module L	バージョン: 0.0.0-0 CLI module L	バージョン: 0.0.0-0 CLI module L

コンフィグ変更  
 コンフィグ変更  
コンフィグ表示  
 コンフィグ表示  
反映予定コンフィグ表示  
 反映予定コンフィグ表示  
コンフィグ反映  
 コンフィグ反映

## 3. モジュール 0 のコンフィグを以下のように書き換えます。

```
VERSION=2
interface_eth0=enable
interface_eth0_metric=1
interface_eth0_type=static
interface_eth0_address="192.0.2.10"
interface_eth0_netmask="255.255.255.0"
interface_eth0_gateway="192.0.2.1"
```

node-eye

コンフィグ変更 / IoT-001

モジュール 0 CLI module A	モジュール 1 CLI module B	モジュール 2 CLI module C	モジュール 3 CLI module D	モジュール 4 CLI module E	モジュール 5 CLI module F	モジュール 6 CLI module G	モジュール 7 CLI module H	モジュール 8 BIN module A
モジュール 9 BIN module B	モジュール 10 BIN module C	モジュール 11 BIN module D	モジュール 12 CLI module I	モジュール 13 CLI module J	モジュール 14 CLI module K	モジュール 15 CLI module L	テンポラリーコンフィグ	

バージョン: 0.0.0-0: CLI module A 変更

```
VERSION=1
interface_eth0=enable
interface_eth0_metric=1
interface_eth0_type=static
interface_eth0_address="192.0.2.10"
interface_eth0_netmask="255.255.255.0"
interface_eth0_network="192.0.2.0"
interface_eth0_broadcast="192.0.2.255"
interface_eth0_gateway="192.0.2.1"
```

状態: 変更あり 差分 最終更新日時: 2016/01/15 14:43:25 コンフィグ変更

反映予定コンフィグ表示 コンフィグ反映 閉じる



モジュール 0 の設定項目は Armadillo の `/etc/NetworkManager/system-connections/` に保存される接続ファイルに反映されます。実際のネットワークの構成にあわせて値を決めてください。



## 4. [コンフィグ変更]をクリックし、[コンフィグ反映]をクリックします。

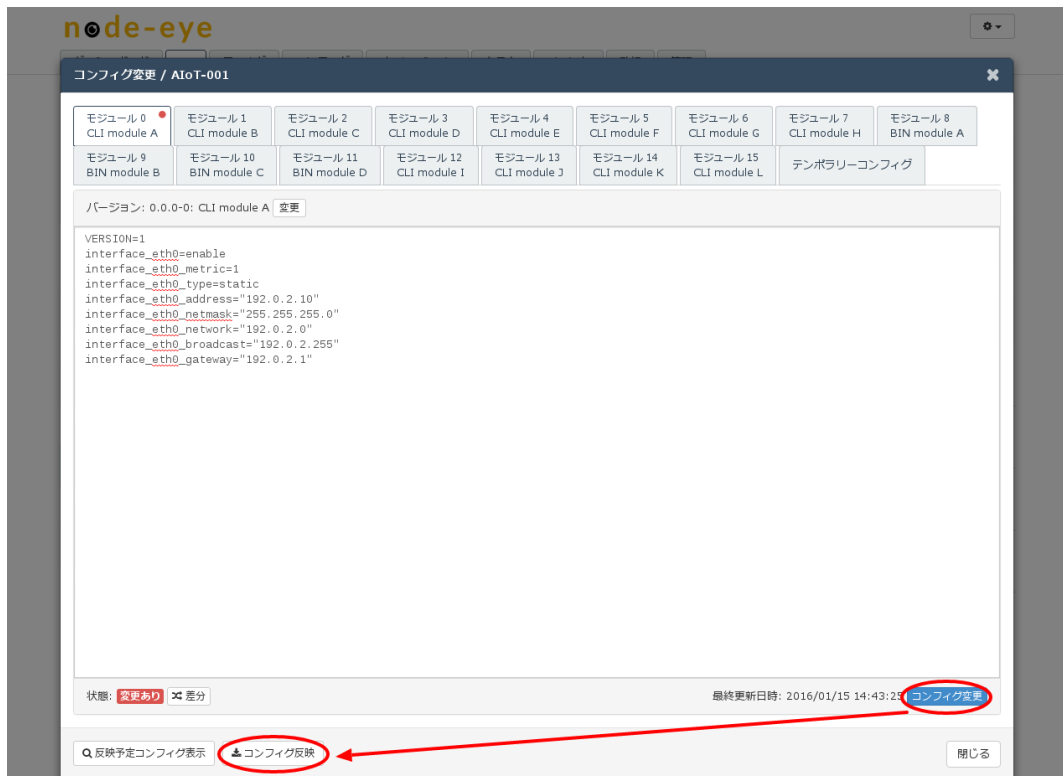


図 3.12 コンフィグ反映 (SACM コントロールパネル)

5. 反映スケジュールを[即時反映]にチェックを入れて、[コンフィグ反映]をクリックします。

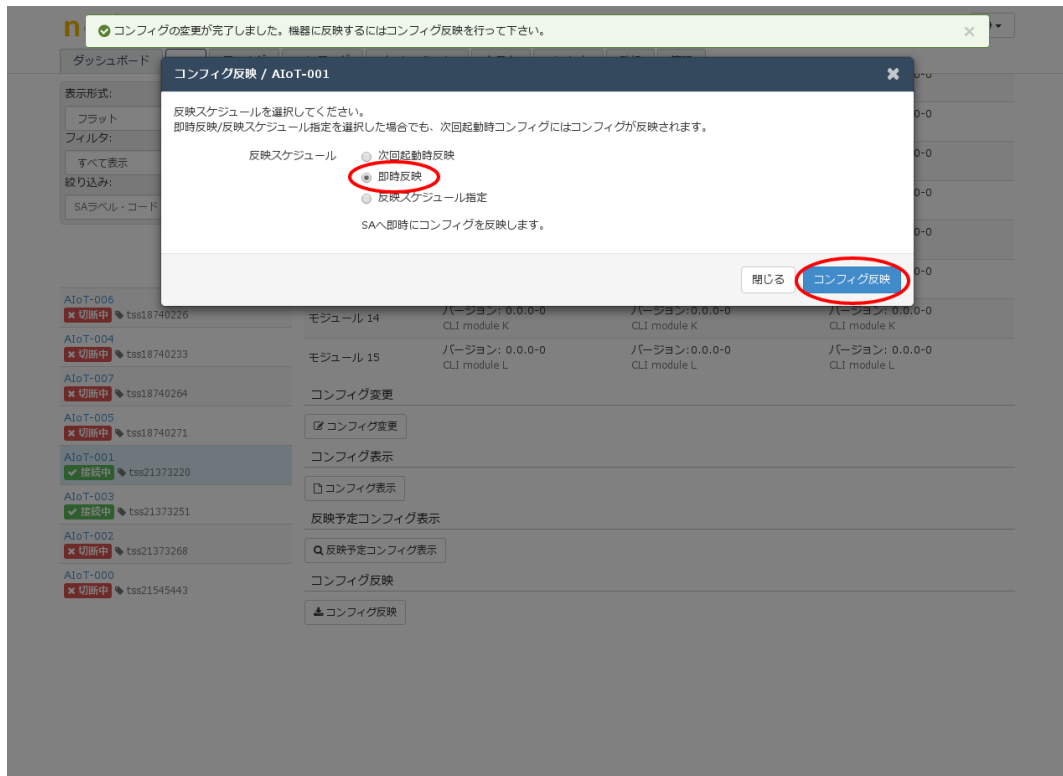


図 3.13 即時反映 (SACM コントロールパネル)



執筆時点で以下の不具合が判明しているため、モジュール 0 のコンフィグと、他のモジュールのコンフィグを同時に反映しないでください。

モジュールコンフィグのロールバックが発生した時、Armadillo 上で動作しているコンフィグと、SACM コントロールパネルに表示されるコンフィグが一致しなくなる場合があります。

6. 反映が完了した後に、Armadillo に接続したシリアルコンソールで `ifconfig` を実行すると、ネットワークインターフェースにモジュール 0 のコンフィグが反映されている事を確認することができます。

```
[Armadillo ~]# ifconfig
eth0    Link encap:Ethernet  HWaddr 00:11:0C:18:10:83
        inet addr:192.0.2.10  Bcast:192.0.2.255  Mask:255.255.255.0
        inet6 addr: fe80::211:cff:fe18:1083/64  Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:17608  errors:0  dropped:1  overruns:0  frame:0
        TX packets:1488  errors:3  dropped:0  overruns:0  carrier:0
        collisions:0  txqueuelen:1000
        RX bytes:2584712 (2.4 MiB)  TX bytes:194540 (189.9 KiB)

lo      Link encap:Local Loopback
```

```
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:60 errors:0 dropped:0 overruns:0 frame:0
TX packets:60 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:5280 (5.1 KiB) TX bytes:5280 (5.1 KiB)
```



リモートコンフィグ機能によるネットワーク設定の内容に問題があり、SACM と接続できなくなった場合、 Armadillo で動作している armsd が約 5 分後にコンフィグ反映直前の設定に戻します。

### 3.3.3. フォルダ機能を使う

Armadillo の数が多い場合や拠点ごとに管理を行いたい場合に便利なフォルダ機能を紹介します。フォルダ分けすることにより、フォルダ単位でコマンドが実行できるようになったり、コンフィグの一斉反映がより簡単に行なえるようになります。

#### 手順 3.6 SA をフォルダに追加する

1. メニューから[フォルダ]をクリックし、[フォルダ追加]をクリック



図 3.14 フォルダタブ (SACM コントロールパネル)

## 2. 任意のフォルダ名を入力し[フォルダ追加]をクリック

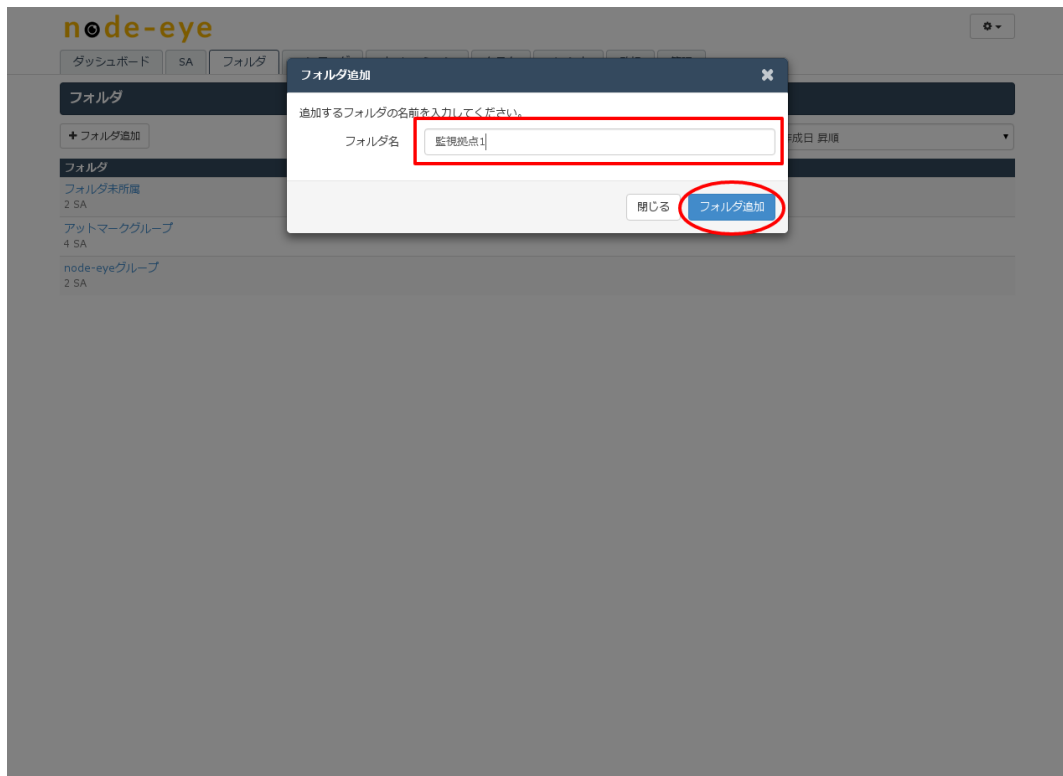


図 3.15 フォルダ名入力 (SACM コントロールパネル)

### 3. フォルダ一覧から、先ほど入力したフォルダ名をクリック



図 3.16 フォルダ一覧 (SACM コントロールパネル)

#### 4. フォルダに Armadillo を登録するため、[所属 SA 変更]をクリック



図 3.17 所属 SA 変更 (SACM コントロールパネル)

## 5. 任意の SA にチェックを入れ、[所属 SA 変更]をクリック

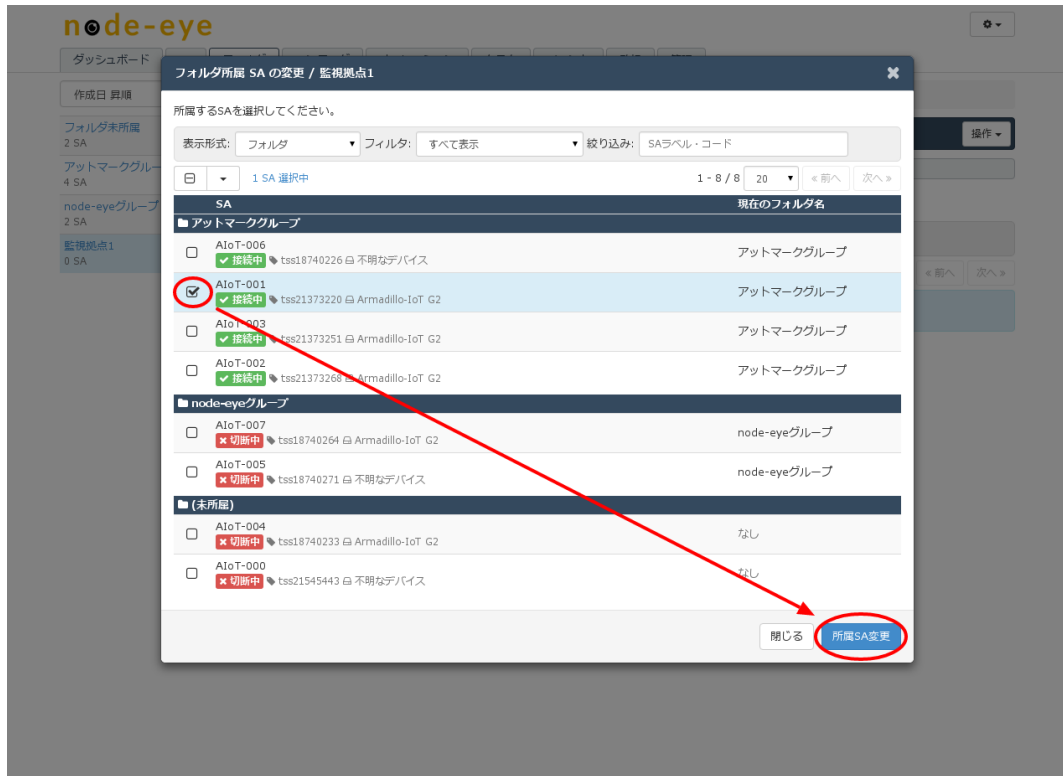


図 3.18 フォルダに追加する SA を選択 (SACM コントロールパネル)

## 6. チェックを入れた SA が表示されます。



図 3.19 SA が追加されたフォルダ (SACM コントロールパネル)

### 3.3.4. 接続状態が変化した時の通知先を設定する

SACM コントロールパネルでは、Armadillo と SACM の接続状態が変化した時に、電子メールを送信する機能を設定することができます。

接続状態が変化するのとは以下のようなことが考えられます。

- ・ [接続中] → [切断中]: Armadillo がリカバリーイメージで起動した
- ・ [接続中] → [切断中]: Armadillo と SACM 間のネットワークに障害が発生し通信できなくなった
- ・ [切断中] → [接続中]: Armadillo が SACM 間のネットワークに障害が解消され通信できるようになった



[接続中] → [切断中] の接続状態の変化には実際に通信が途絶してから、最大で 15 分程度かかります





[切断中] → [接続中] の接続状態の変化には実際に通信が確立してから、最大で 10 分程度かかります

通知先の設定には SACM コントロールパネルの監視グループを使います。監視グループには複数の Armadillo と、通知先のメールアドレスを登録することができます。

### 手順 3.7 SA の接続状態が変化した時の通知先を設定する

1. まず、監視グループの作成を行います。メニューから[監視]をクリックし、[監視グループ追加]をクリック



図 3.20 監視タブ (SACM コントロールパネル)

## 2. 任意の監視グループ名を入力し、[監視グループ追加]をクリック

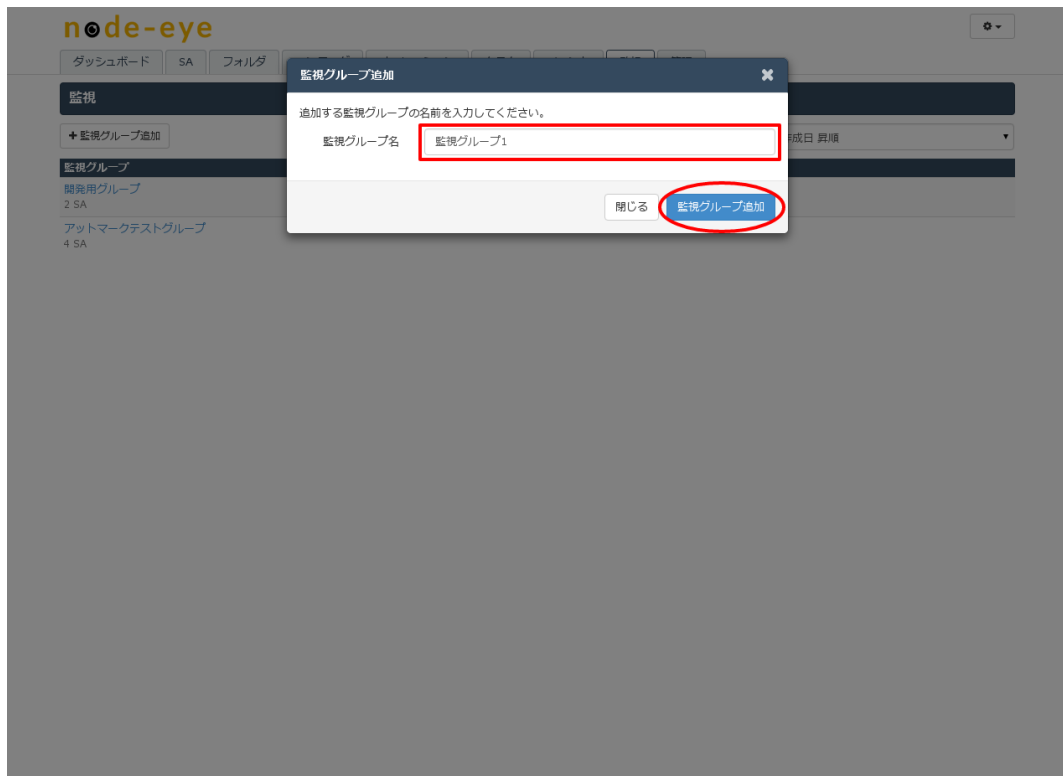


図 3.21 監視グループ名入力 (SACM コントロールパネル)

- これで監視グループの作成は完了です。次に監視グループに電子メールの送信先を設定します。監視グループ一覧から、先ほど入力した監視グループ名をクリック



図 3.22 監視グループが追加された監視タブ (SACM コントロールパネル)

#### 4. [通知先メールアドレス変更]をクリック

The screenshot shows the 'node-eye' dashboard with the '監視' (Monitoring) tab selected. The main content area displays '監視グループ1' (Monitoring Group 1) with a '通知先メールアドレス' (Notification email address) section. A button labeled '通知先メールアドレス変更' (Change notification email address) is circled in red. Below this, there is a table with 5 rows, each labeled '通知先メールアドレス' followed by a number (1-5), with the value 'なし' (None) in the adjacent column. At the bottom, a blue message box states 'SAがありません。' (No SA).

通知先メールアドレス	値
通知先メールアドレス 1	なし
通知先メールアドレス 2	なし
通知先メールアドレス 3	なし
通知先メールアドレス 4	なし
通知先メールアドレス 5	なし

図 3.23 監視グループの詳細 (SACM コントロールパネル)

## 5. 任意の通知先メールアドレスを入力し、[通知先メールアドレス変更]をクリック

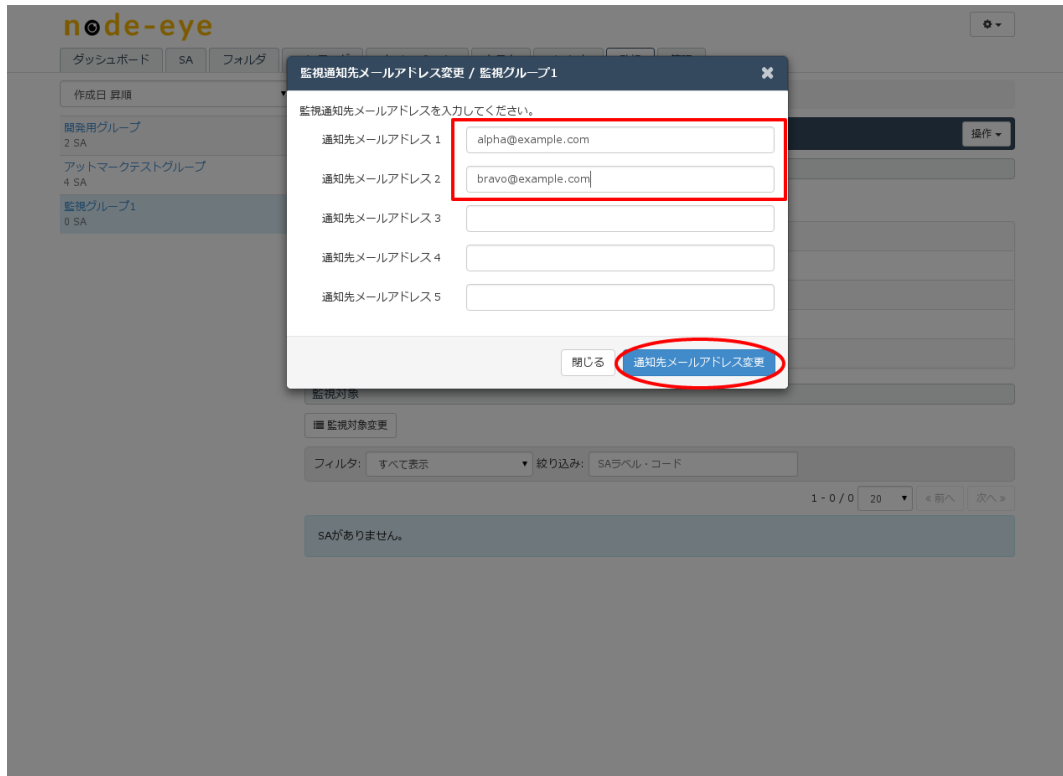


図 3.24 通知先メールアドレス入力 (SACM コントロールパネル)

## 6. これで Armadillo の動作状況が変化した時の通知先を設定することができました。

監視通知先メールアドレスの変更が完了しました。

ダッシュボード SA フォルダ コンフィグ オペレーション タスク イベント 監視 管理

作成日 昇順

開発用グループ  
2 SA

アットマークテストグループ  
4 SA

監視グループ1  
0 SA

監視 / 監視グループ1

監視グループ1 操作

通知先メールアドレス

通知先メールアドレス変更

通知先メールアドレス 1	alpha@example.com
通知先メールアドレス 2	bravo@example.com
通知先メールアドレス 3	なし
通知先メールアドレス 4	なし
通知先メールアドレス 5	なし

監視対象

監視対象変更

フィルタ: すべて表示 絞り込み: SAラベル・コード

1 - 0 / 0 20 < 前へ 次へ >

SAがありません。

図 3.25 通知先メールアドレスが追加された監視グループ詳細 (SACM コントロールパネル)

## 7. 最後に、監視グループに Armadillo を登録します [監視対象変更]をクリック

The screenshot shows the '監視' (Monitoring) section of the node-eye interface. A notification at the top states '監視通知先メールアドレスの変更が完了しました。' (Monitoring notification email address change is complete). The main content area is titled '監視グループ1' (Monitoring Group 1) and includes a '操作' (Action) dropdown menu. Below this, there is a section for '通知先メールアドレス' (Notification email addresses) with a '通知先メールアドレス変更' (Change notification email address) button. A table lists five notification email addresses: 'alpha@example.com', 'bravo@example.com', and three 'なし' (None) entries. Below the table is the '監視対象' (Monitoring target) section, where the '監視対象変更' (Change monitoring target) button is circled in red. At the bottom, there is a filter section with 'すべて表示' (Show all) and '絞り込み: Sラベル・コード' (Filter: S label code), and a status bar indicating '1 - 0 / 0' items and 'SAがありません。' (No SAs).

図 3.26 監視グループ詳細 (SACM コントロールパネル)

## 8. 監視グループに追加したい Armadillo にチェックを入れ[監視対象変更]をクリック

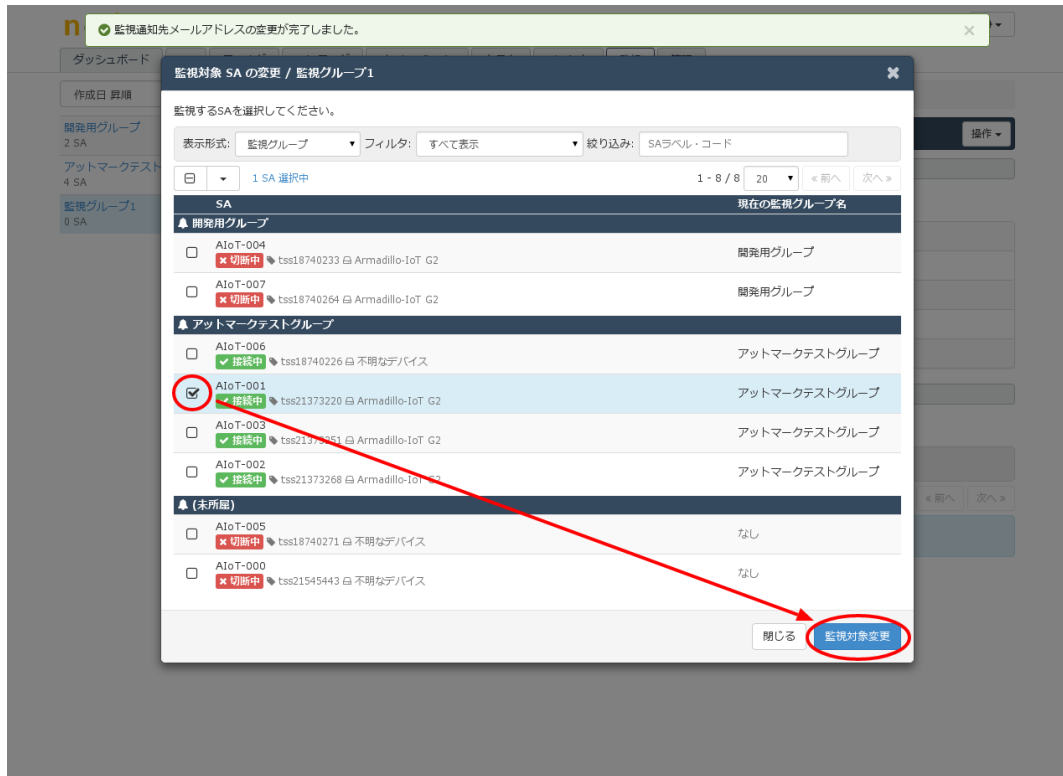


図 3.27 監視グループに追加する SA を選択 (SACM コントロールパネル)



9. これで監視グループに登録した Armadillo の動作状況が変わった時に、電子メールが送信されるようになりました。



図 3.28 SA が追加された監視グループ (SACM コントロールパネル)

電子メールが送信されることを確認するには、登録した Armadillo の Ethernet ケーブルを外すか、その時点で[切断中]となっている Armadillo を起動し、SACM と接続する必要があります。

SACM は一時的なネットワーク障害と区別するため、[接続中]から[切断中]に状態の表示を変更するまでに 15 分の猶予期間を設けています。そのため、Ethernet ケーブルを外す等の方法で SACM との接続を遮断した場合は 15 分後に、電子メールの送信と node-eye コントロールパネル、SACM コントロールパネルの表示が変化します。

### 3.3.5. その他

この他の使い方や、詳しい情報は <http://manual.sacm.jp/> を参照してください

## 4. Armadillo ソフトウェア仕様

### 4.1. メモリマップ

Armadillo-IoT の QSPI フラッシュメモリのメモリマップを「表 4.1. QSPI フラッシュメモリ メモリマップ」に示します。

表 4.1 QSPI フラッシュメモリ メモリマップ

物理アドレス	サイズ	説明
0x00000000   0x001003FF	約 1 MByte	U-Boot ブートローダーイメージ
0x00100400   0x001403FF	256 KBytes	ライセンス情報
0x00140400   0x007FFFFFFF	約 6.7 MBytes	予約領域

Armadillo-IoT の eMMC のメモリマップを「表 4.2. eMMC メモリマップ」に示します。

表 4.2 eMMC メモリマップ

パーティション	サイズ	説明
1	32 MBytes	Linux カーネルイメージ/Device Tree Blob
2	約 3.4 GBytes	Debian GNU/Linux
3	128 MBytes	リカバリーイメージ

recovery パーティションについては本文「4.5. リカバリーイメージ」を参照してください。license パーティションには、Distribution ID に対応した SACM との通信に使う認証キー (LS-SA key) が格納されます。

kernel、userland パーティションは、通常の Armadillo 利用時と同様、ユーザーが自由に変更可能です。

### 4.2. node-eye パッケージ

本章では node-eye パッケージについてを説明します。

node-eye パッケージはアットマークテクノが提供する Debian パッケージです。node-eye を利用するためのクライアントソフトやファームウェアアップデートのためのスクリプトなどがまとめられています。node-eye を利用するためには node-eye パッケージを Armadillo にインストールする必要があります。

node-eye パッケージをインストール可能な製品を以下に示します。この他の製品に node-eye パッケージはインストール出来ません。

- ・ Armadillo-IoT ゲートウェイ G3
- ・ Armadillo-IoT ゲートウェイ G3L
- ・ Armadillo-X1

### 4.2.1. node-eye パッケージのインストール・アンインストール

node-eye パッケージをインストールするには、下記のコマンドを実行します。

```
[armadillo ~]# apt-get update
[armadillo ~]# apt-get install node-eye
```

node-eye パッケージをアンインストールするには、下記のコマンドを実行します。

```
[armadillo ~]# apt-get remove node-eye
```

### 4.2.2. node-eye.service の有効化・無効化

node-eye.service を有効化することで、Armadillo の起動後自動的に armsd を立ち上げることができます。

node-eye.service を有効化するには、下記のコマンドを実行します。

```
[armadillo ~]# systemctl enable node-eye.service
```

node-eye.service を無効化するには、下記のコマンドを実行します。

```
[armadillo ~]# systemctl disable node-eye.service
```

node-eye.service が有効化されているかどうかを確認するには、下記のコマンドを実行します。

```
[armadillo ~]# systemctl is-enabled node-eye.service
```

## 4.3. armsd

本章では node-eye パッケージに含まれている ARMS プロトコルのクライアントソフト "armsd" について説明します。

node-eye の死活監視、リモートコンフィグ、ファームウェアアップデートなどの機能は、armsd と SACM が互いに通信することで実現しています。以下では armsd の標準機能である死活監視機能、リモートコンフィグ機能について説明します。

### 4.3.1. 死活監視機能

armsd は SACM から死活監視を行うために、定期的に SACM へ Heartbeat パケットを送信しています。SACM が Heartbeat パケットを受信すると、SACM は Heartbeat パケットを送ってきた Armadillo と接続中であると判断します。

- ・ armsd の Heartbeat パケット送信間隔
- ・ 5 分に 1 回 Heartbeat を Armadillo から送信

- ・ SACM の接続状態の表示
  - ・ [切断中]のとき、1 回 Heartbeat パケットが SACM に到達すると、接続状態を[接続中]に変化させる
  - ・ [接続中]のとき、3 回連続で Heartbeat パケットが SACM に到達しなかった場合、接続状態を[切断中]に変化させる



リカバリーイメージは、Heartbeat パケットを送信しません。従って、接続状態の表示は[切断中]となりますが、Armadillo と SACM は互いに通信できるため復旧作業(ファームウェアアップデートなど)は可能です。

## 4.3.2. リモートコンフィグ機能

SACM から Armadillo 上の設定ファイルの修正を行ったり、なんらかのアプリケーションを実行するためには、armsd のモジュールという仕組みを使います。

以下では、リモートコンフィグ機能に欠かせないモジュールについて説明します。

### 4.3.2.1. モジュールについて

モジュールは Armadillo のユーザーランドに配置された start/stop/reconfig/command といったオペレーションを受け付けるスクリプトとして実装されています。

モジュールにはバイナリモジュール(BIN module)、コマンドラインインターフェースモジュール(CLI module)の 2 種類あり、16 個のモジュールが用意されています。

表 4.3 モジュールのタイプとその用途

No.	Module type	用途
0 - 3	CLI module	アットマークテクノが提供する機能に使用
4 - 7	CLI module	お客様が自由に使用
8 - 11	BIN module	お客様が自由に使用
12 - 15	CLI module	お客様が自由に使用

CLI module は、SACM 上で直接、モジュールのコンフィグを編集、参照することができます。バイナリモジュールは、SACM 上でコンフィグの内容を直接参照することはできず、ダウンロードとアップロードのみが可能です。

また、コマンドラインインターフェースモジュールのコンフィグで扱えるのは、US-ASCII のテキストのみに限られます。日本語などのマルチバイト文字が含まれる場合、バイナリモジュールで取り扱う必要があります。

モジュールのコンフィグについては SACM コンフィグの概念 [[http://manual.sacm.jp/service\\_spec/config.html](http://manual.sacm.jp/service_spec/config.html)] を参照してください。

モジュール 0 ~ 3 はアットマークテクノが提供する機能(ネットワーク設定、ファームウェア管理など)に使用しています。モジュール 0 ~ 3 に変更を加えることは可能ですが、変更を加えた場合にはアットマークテクノから提供する機能(サービス等)を受けられなくなる場合があります。

現在、アットマークテクノから提供しているモジュールを表に示します。

表 4.4 アットマークテクノが提供しているモジュール一覧

No.	Module type	Module name	用途
0	CLI module	ネットワークモジュール	ネットワークインターフェースの設定に使用
1	CLI module	ファームウェアモジュール	ファームウェアアップデートに使用

モジュール 4 ~ 15 はお客様が自由に使用することができます。モジュールを作成・追加を行うには「5.5. モジュールの作成方法」を参照してください。

その他、モジュールに関する情報は SACM モジュールの概念 [[http://manual.sacm.jp/service\\_spec/module.html](http://manual.sacm.jp/service_spec/module.html)] を参照してください。

### 4.3.3. その他・詳細仕様

armsd についてより詳しい情報は libarms のドキュメントを参照してください

クライアント実装 - libarms documentation

[https://dev.smf.jp/libarms/armsd/armsd\\_top.html](https://dev.smf.jp/libarms/armsd/armsd_top.html)

## 4.4. アットマークテクノが提供するモジュールの仕様

本章ではアットマークテクノが提供する armsd のモジュールについて説明します。

### 4.4.1. ネットワークモジュール

ネットワークモジュールは Armadillo のネットワーク設定を管理します。ネットワークモジュールを使用することで、SACM コントロールパネルから Armadillo のネットワーク設定を変更することができます。ネットワークモジュールを使ったネットワークの設定方法は「5.1. ネットワーク設定方法」を参照してください。

#### 4.4.1.1. ネットワークインターフェース設定シーケンス

以下に SACM 上に設定したネットワークインターフェース設定が反映されるまでのシーケンスを示します。

1. armsd が起動すると/etc/armsd/scripts/line が実行されます。/etc/armsd/scripts/line は /etc/armsd/line.conf を元 に /etc/NetworkManager/system-connections/armsd\_line\_connection (以下、コネクションと表記します)を生成します。
2. 次に、生成されたコネクションを使用して LS(Location Server)に接続します。
3. 接続が確立すると、LS から RS(Resource Server)へ接続するために必要な情報(Location-Config)を取得します。これを LS Pull といいます。Armadillo は LS Pull に成功すると、Location-Config をキャッシュします。(etc/armsd/armsd.cache として保存します) Location-Config がある場合、LS Pull をスキップします。
4. 次に、再び/etc/armsd/scripts/line でコネクションを生成し、これを用いて RS へ接続します。ここで生成されるコネクションは、armsd が起動した時と同じものが生成されます。
5. 接続が確立すると、RS から SACM のモジュール N に設定されたコンフィグ(Service-Config)を取得します。これを RS Pull といいます。
6. 取得した Service-Config を用いて、post-pull script がコネクションを生成します。

7. こうして、Service-Config の設定でネットワークインターフェースがアップします。
8. 最後に接続性の確認を行います。
9. - Service-Config により設定したコネクシオンで SACM と接続できるかを確認します。接続は 5 回試行しますが、5 回とも接続できなかった場合、新たに設定する前の Service-Config にロールバックします。
10. SACM との接続が確認されると、ネットワーク設定のシーケンスは終了となります。

#### 4.4.1.2. コンフィグの書式

ネットワークモジュール(モジュール 0)のコンフィグの書式を以下に示します。

```
version=[version]

[key]=[value]
[key]=[value]
:
```

- ・ version
  - ・ モジュール内の処理とコンフィグの内容を対応付けるために使用するバージョン番号です
  - ・ Armadillo-IoT G3, Armadillo-IoT G3L, Armadillo-X1 では "2" 固定となっています
- ・ key
  - ・ コンフィグの変数名です
  - ・ 同一の key を同じコンフィグ内に書くと、後に書かれたものが使用されます
  - ・ key の前に#をつけることで、コメントアウトできます
- ・ value
  - ・ key に対応する値です

#### 4.4.1.3. ネットワークモジュールコンフィグ一覧

##### 4.4.1.3.1. eth0

ネットワークモジュールの eth0 に関する key 一覧です。

表 4.5 ネットワークモジュールの eth0 に関する key 一覧

key	value (参考例)	説明
interface_eth0	enable	eth0 の有効/無効 (enable / disable)
interface_eth0_type	static	IP アドレスの設定方法 (dhcp / static)
interface_eth0_address	192.168.10.1	IP アドレス
interface_eth0_netmask	255.255.0.0	ネットマスク
interface_eth0_gateway	192.168.0.1	ゲートウェイ
interface_eth0_metric	1	メトリック値
interface_eth0_never-default	true	デフォルトゲートウェイとして使用しない (true / false) 未設定の場合 false



`interface_eth0_never-default` は node-eye パッケージ v0.2.0-1 以降で使用することができます。

#### 4.4.1.3.2. wwan

ネットワークモジュールの wwan に関する key 一覧です。

表 4.6 ネットワークモジュールの wwan に関する key 一覧

key	value (参考例)	説明
<code>interface_wwan</code>	enable	wwan の有効/無効 (enable / disable)
<code>interface_wwan_type</code>	dhcp	IP アドレスの設定方法 (dhcp のみサポート)
<code>interface_wwan_apn</code>	APN	アクセスポイント名 (APN)
<code>interface_wwan_id</code>	"user name"	認証 ID(ユーザー名)
<code>interface_wwan_pass</code>	"password"	パスワード
<code>interface_wwan_metric</code>	20	メトリック値

#### 4.4.1.3.3. wlan0

ネットワークモジュールの wlan0 に関する key 一覧です。

表 4.7 ネットワークモジュールの wlan0 に関する key 一覧

key	value (参考例)	説明
<code>interface_wlan0</code>	enable	enable または disable
<code>interface_wlan0_type</code>	dhcp	dhcp または static
<code>interface_wlan0_address</code>	192.168.10.2	IP アドレス
<code>interface_wlan0_netmask</code>	255.255.0.0	ネットマスク
<code>interface_wlan0_gateway</code>	192.168.0.1	ゲートウェイ
<code>interface_wlan0_metric</code>	40	メトリック値
<code>interface_wlan0_never-default</code>	true	デフォルトゲートウェイとして使用しない (true / false) 未設定の場合 false
<code>interface_wlan0_passphrase</code>	"passphrase"	パスフレーズ
<code>interface_wlan0_cryptmode</code>	"WPA2-AES"	暗号化方式*1 [ <a href="http://manual.atmark-techno.com/armadillo-wlan/armadillo-wlan_software_manual_ja-3.3.0/ch07.html#table-setting-cryptmode">http://manual.atmark-techno.com/armadillo-wlan/armadillo-wlan_software_manual_ja-3.3.0/ch07.html#table-setting-cryptmode</a> ]
<code>interface_wlan0_essid</code>	"mywpa2aes"	ESSID



`interface_wlan0_never-default` は node-eye パッケージ v0.2.0-1 以降で使用することができます。

#### 4.4.1.3.4. resolv.conf

ネットワークモジュールの resolv.conf に関する key 一覧です。 `resolv_conf_` から始まる変数の値は `/etc/resolv.conf` に書き込まれます。

表 4.8 ネットワークモジュールの resolv.conf に関する key 一覧

key	説明
resolv_conf_search	resolv.conf の search
resolv_conf_nameserver	resolv.conf の nameserver

## 4.4.2. ファームウェアモジュール

ファームウェアモジュールは node-eye コントロールパネルからファームウェアアップデートを行う際に使用されます。ファームウェアモジュールを使ったファームウェアアップデートの方法は「3.2.5. ファームウェアアップデートを行う」を参照してください。

ファームウェアモジュールで書き換えることのできるファイルは以下の3つです。kernel, dtb は必ず同時にアップデートを行います。個別にアップデートすることはできません。dtb のみ、kernel のみ更新したい場合は、Armadillo に既書き込まれているイメージを指定する必要があります。

- ・ kernel ・ dtb
- ・ userland
- ・ recovery

ファームウェアアップデートを行う際に、用意する必要があるファイルは以下のとおりです。

表 4.9 ファームウェアアップデートに必要なファイル

アップデート対象	種別	ファイル名
kernel ・ dtb	Linux カーネル	ulmage
	Linux カーネルチェックサム	ulmage.md5
	Linux カーネル DTB アーカイブ	ulmage.dtb.zip
	Linux カーネル DTB アーカイブ チェックサム	ulmage.dtb.zip.md5
userland	Debian GNU/Linux ユーザーランドアーカイブ	*.tar.gz
	Debian GNU/Linux ユーザーランドアーカイブチェックサム	*.tar.gz.md5
recovery	リカバリー Linux カーネル	ulmage.recovery
	リカバリー Linux カーネルチェックサム	ulmage.recovery.md5
	リカバリー Linux カーネル DTB アーカイブ	ulmage.recovery.dtb.zip
	リカバリー Linux カーネル DTB アーカイブ チェックサム	ulmage.recovery.dtb.zip.md5



ファームウェアアップデートを行うためには、Armadillo が HTTP プロトコルでアクセス可能な Web サーバーにイメージファイルを事前に置いておく必要があります。

## 4.5. リカバリーイメージ

recovery 領域には、Linux カーネルとユーザーランドを一つにまとめたイメージを書き込みます。このイメージをリカバリーイメージといいます。

### 4.5.1. リカバリーイメージの目的

リカバリーイメージの目的は、次のような人為的ミスによって正常に起動できなくなった Armadillo を node-eye コントロールパネルからの復旧をできるようにすることです。



- ・ ファームウェアアップデート機能を使って Linux カーネルまたはユーザーランドに誤ったイメージを書き込んでしまった
- ・ ファームウェアアップデート機能を使ったイメージ書き込みの最中に、電源が遮断されて書き込みが完了せず再起動に失敗した

## 4.5.2. リカバリーイメージの起動条件

リカバリーイメージを使って Armadillo の起動する条件を次に示します。

- ・ ファームウェアアップデートが正常に完了していない状態で、Armadillo が再起動した
- ・ ファームウェアアップデートは完了したが、更新したイメージで起動できなかった

## 4.5.3. リカバリーイメージの特徴

### 4.5.3.1. インストールされているアプリケーション

リカバリーイメージは、Linux カーネルとユーザーランドを復旧するために SACM に接続することのできる最小の構成となっています。そのため、インストールされているアプリケーションは標準イメージと比べ少なくなっています。

### 4.5.3.2. リカバリーイメージ起動中の WebUI 上の表示

リカバリーイメージで Armadillo が起動すると、SACM コントロールパネル、node-eye コントロールパネル上では [切断中] と表示されます。ただし、Armadillo がリカバリーイメージで起動し、SACM と通信できる場合は、SACM コントロールパネル、node-eye コントロールパネルから Ping と Traceroute を実行することができます。

### 4.5.3.3. RAM disk に展開されるユーザーランド

リカバリーイメージが起動すると、ユーザーランドは RAM disk 上に展開されます。そのため、ルートファイルシステムへ書き込みを行っても変更は保存されません。また、eMMC 第 2 パーティションに保存されているユーザーランドが RAM 上に展開されるのではなく、リカバリーイメージに含まれるユーザーランドが RAM 上に展開されます。

### 4.5.3.4. リカバリーイメージが使用する line.conf

リカバリーイメージは eMMC 第 3 パーティションに保存された `/etc/armsd/line.conf` を使用します。eMMC 第 2 パーティションの `/etc/armsd/line.conf` は使用されません。

## 4.5.4. リカバリーイメージのビルド方法

リカバリーイメージをビルドする方法を示します。リカバリーイメージのビルドは ATDE 上で行います。



eMMC 第 2 パーティションに書き込む GNU/Linux Debian アーカイブの構築方法は、製品マニュアルを参照してください。

## 手順 4.1 リカバリーイメージのビルド

### 1. x1-node-eye-recovery-builder の準備

はじめに、x1-node-eye-recovery-builder を使用するための準備を行います。下記のコマンドを実行する前に、最新の x1-node-eye-recovery-builder を Armadillo サイトからダウンロードしておいてください。

```
[PC ~]$ sudo apt-get update && sudo apt-get install qemu-user-static
[PC ~]$ tar xf x1-node-eye-recovery-builder-[version].tar.gz
```

### 2. リカバリーイメージ用の initramfs アーカイブ作成

x1-node-eye-recovery-builder を使ってリカバリーイメージ用の initramfs アーカイブを作成します。build-recovery-initramfs.sh を実行してから initramfs アーカイブが完成するまでに 40 分程度かかります。これは x1-node-eye-recovery-builder がパッケージをインターネット上から取得するためです。回線速度によって完成までにかかる時間は前後します。

```
[PC ~]$ cd x1-node-eye-recovery-builder-[version]
[PC ~/x1-node-eye-recovery-builder-[version]]$ sudo ./build-recovery-initramfs.sh
```

### 3. Linux カーネルのソースコードの準備

Armadillo サイトから Linux カーネルのソースコードアーカイブをダウンロードした後、アーカイブの展開をします。

```
[PC ~]$ ls
linux-3.14.38-at[version].tar.gz  x1-node-eye-recovery-builder-[version]
[PC ~]$ tar xf linux-3.14.38-at[version].tar.gz
[PC ~]$ ls
linux-3.14.38-at[version]          x1-node-eye-recovery-builder-[version]
linux-3.14.38-at[version].tar.gz
```

### 4. initramfs アーカイブへのシンボリックリンクを作成

Linux カーネルディレクトリに移動して、initramfs アーカイブへのシンボリックリンクを作成します。

```
[PC ~]$ cd linux-3.14.38-at[version]
[PC ~/linux-3.14.38-at[version]]$ ln -s ../x1-node-eye-recovery-builder-[version]/
initramfs_x1_node-eye_recovery_[YYYYMMDD].cpio.gz initramfs_x1.cpio.gz
```

### 5. ビルド

ビルドするには、次のようにコマンドを実行します。

```
[PC ~/linux-3.14.38-at[version]]$ make ARCH=arm x1_defconfig
[PC ~/linux-3.14.38-at[version]]$ make CROSS_COMPILE=arm-linux-gnueabihf- ARCH=arm
[PC ~/linux-3.14.38-at[version]]$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf-
```

```
LOADADDR=0x80008000 uImage
```

## 6. イメージファイルの生成確認

ビルドが終了すると、arch/arm/boot/ディレクトリと、arch/arm/boot/dts/以下にイメージファイル(Linux カーネルと DTB)が作成されています。

```
[PC ~/linux-3.14.38-at[version]]$ ls arch/arm/boot/uImage
uImage
[PC ~/linux-3.14.38-at[version]]$ ls arch/arm/boot/dts/armadillo_iotg_g3l.dtb
armadillo_iotg_g3l.dtb
```

### 4.5.5. リカバリーイメージの書き込み方法

「4.5.4. リカバリーイメージのビルド方法」で作成したイメージファイルを、Armadillo に書き込む方法について説明します。「3.2.5. ファームウェアアップデートを行う」で説明した方法と同様に、ATDE で必要なファイルの名前変更と ZIP 圧縮を行った後に所定の位置へ設置します。

```
[PC ~]$ ls
armadillo_iotg_g3l.dtb uImage
[PC ~]$ mv uimage uImage.recovery
[PC ~]$ mv armadillo_iotg_g3l.dtb armadillo_iotg_g3l.dtb.recovery
[PC ~]$ zip uImage.recovery.dtb.zip armadillo_iotg_g3l.dtb.recovery
[PC ~]$ ls
armadillo_iotg_g3l.dtb.recovery uImage.recovery uImage.recovery.dtb.zip
[PC ~]$ md5sum uImage.recovery > uImage.recovery.md5
[PC ~]$ md5sum uImage.recovery.dtb.zip > uImage.recovery.dtb.zip.md5
[PC ~]$ ls
armadillo_iotg_g3l.dtb.recovery uImage.recovery uImage.recovery.dtb.zip
uImage.recovery.dtb.zip.md5 uImage.recovery.md5
[PC ~]$ sudo cp uImage.recovery* /var/www/html
```

次に、「3.2.5. ファームウェアアップデートを行う」で説明した方法と同様に、接続中デバイスから[ファームウェアアップデート]項目に移動します。その後、下記のようにフォームへ入力してください([ATDE の IP アドレス]はご利用の環境にあわせて変更してください)。

表 4.10 リカバリーイメージアップデート

フォーム	入力内容
プライマリ/リカバリ	リカバリ
リカバリ	http://[ATDE の IP アドレス]/uimage.recovery
アップデート完了時に再起動する	

[実行]をクリックし、リカバリーイメージのアップデートを行ってください。

The screenshot shows the 'node-eye' control panel with the following elements:

- Header: 'node-eye' logo, navigation links for 'ダッシュボード', 'デバイス', 'ログ', and 'アカウント', and a user profile 'atmark'.
- Navigation: '詳細', 'コマンド実行', 'ログ', and 'リソースグラフ' tabs.
- Configuration Section:
  - 'Ping' section: Radio buttons for 'プライマリ' and 'リカバリ' (selected).
  - 'Traceroute' section: A text input field containing 'http://192.1680.100/uImage.recovery'.
  - '再起動' section: A checkbox for 'アップデート完了時に再起動する'.
  - 'ファームウェアアップデート' section: Radio buttons for 'すぐに実行する' (selected) and '実行時間を指定する'. Below this is a date input field showing '年 / 月 / 日 --:--'.
- Action: A blue '実行' button.

図 4.1 リカバリーイメージアップデート実行 (node-eye コントロールパネル)

# 5. Howto

## 5.1. ネットワーク設定方法

本章では、ネットワークモジュール用いて各種ネットワークインターフェースを設定するための方法を紹介します。

node-eye においてネットワークの設定ファイルは 2 つあり、それぞれ用途と設定ファイルが使われるタイミングが異なります。タイミングについては「4.4.1.1. ネットワークインターフェース設定シーケンス」を参照してください。

表 5.1 ネットワーク設定ファイル

設定ファイル	用途
line.conf	SACM に接続するためのネットワーク設定を管理します。この設定ファイルは Armadillo 上に設定・保存します。
ネットワークモジュールコンフィグ	SACM と接続した後のネットワーク設定を管理します。この設定ファイルは SACM 上に設定・保存します。

### 5.1.1. line.conf

リモートコンフィグや死活監視などを行うためには、Armadillo と SACM が互いに通信できる環境を用意する必要があります。line.conf は Armadillo と SACM が互いに通信できる環境を作るための設定ファイルです。

line.conf にはデフォルト設定が存在します。デフォルト設定ではネットワークインターフェースに eth0 を使い、DHCP による IP アドレス取得を行って SACM と通信します。

そのため、下記の 3 つの条件が揃っている場合、line.conf を変更する必要ありません。

- ・ Armadillo のネットワークインターフェースに有線 LAN(eth0)を使用している
- ・ Armadillo が DHCP によるアドレスを取得できる環境である
- ・ サービスアダプタの通信要件 [[http://manual.sacm.jp/service\\_spec/condition.html#id2](http://manual.sacm.jp/service_spec/condition.html#id2)]を満たしている

#### 5.1.1.1. line.conf の設定方法

line.conf の設定にはテキストエディタを使用します。node-eye パッケージをインストールすると、line.conf は /etc/armsd/ に配置されます。

vi で line.conf を編集するには、下記のコマンドを実行してください。

```
[armadillo /]# vi /etc/armsd/line.conf
```



line.conf の編集は「5.1.1.2. line.conf コンフィグ一覧」と「5.1.1.3. 設定例」を参考に行ってください。

line.conf のデフォルト設定を使用するには line.conf を空にするか、書き込んでいるコンフィグをすべてコメントアウトしてください。

line.conf を編集した後は、recovery パーティションに line.conf コピーしてください。

```
[armadillo /]# mount /dev/mmcblk2p3 /mnt
[armadillo /]# mkdir -p /mnt/etc/arm_sd/
[armadillo /]# cp /etc/arm_sd/line.conf /mnt/etc/arm_sd/line.conf
[armadillo /]# umount /mnt
```



recovery パーティションにコピーした line.conf はリカバリーイメージが使用します。ファームウェアアップデートの失敗などによりリカバリーイメージが起動した場合、SACM と通信を行うために line.conf が必要となります。

## 5.1.1.2. line.conf コンフィグ一覧

### 5.1.1.2.1. eth0 固定 IP

表 5.2 eth0 で使用可能な key 一覧

key	説明
ETHERNET_ENABLE	無線 LAN の有効/無効 (true/false)
STATIC_ADDRESS	IP アドレス
STATIC_NETMASK	ネットマスク
STATIC_GATEWAY	ゲートウェイ



ETHERNET\_ENABLE は node-eye パッケージ v0.2.0-1 以降で使用することができます。

### 5.1.1.2.2. wwan

表 5.3 wwan で使用可能な key 一覧

key	説明
MOBILE_ID	認証 ID(ユーザー名)
MOBILE_PASS	パスワード
MOBILE_APN	アクセスポイント名(APN)
MOBILE_AUTH_TYPE	認証方式
MOBILE_PDP_TYPE	PDP タイプ

### 5.1.1.2.3. wlan0

表 5.4 wlan0 で使用可能な key 一覧

key	説明
WIRELESS_LAN_ENABLE	無線 LAN の有効/無効 (true/false)
WIRELESS_LAN_ESSID	ESSID
WIRELESS_LAN_PASSPHRASE	パスワード



LS Pull における eth0 と wlan0 は排他になっているため、WIRELESS\_LAN\_ENABLE=true に設定した場合は eth0 による LS Pull はできません。



WIRELESS\_LAN\_ENABLE=false に設定した場合は ETHERNET\_ENABLE=true の設定を行わなくても eth0 が有効になります。

### 5.1.1.2.4. resolv

表 5.5 resolv で使用可能な key 一覧

key	説明
RESOLV_SEARCH	resolv.conf の search
RESOLV_NAMESERVER	resolv.conf の nameserver

### 5.1.1.3. 設定例

#### 5.1.1.3.1. 3G モバイル回線を使う

```
MOBILE_ID="User name"
MOBILE_PASS="Password"
MOBILE_APN="APN"
MOBILE_AUTH_TYPE="PAP"
MOBILE_PDP_TYPE="IP"
```

#### 5.1.1.3.2. 固定 IP を使う

```
ETHERNET_ENABLE=true
STATIC_ADDRESS="192.0.2.10"
STATIC_NETMASK="255.255.255.0"
STATIC_GATEWAY="192.0.2.1"

RESOLV_SEARCH="local-network"
RESOLV_NAMESERVER="192.168.10.1"
```

### 5.1.1.3.3. 無線 LAN(wlan)を使う

```
WIRELESS_LAN_ENABLE=true
WIRELESS_LAN_ESSID="myssid"
WIRELESS_LAN_PASSPHRASE="mypresharedkey"
```

### 5.1.1.3.4. 無線 LAN で固定 IP を使う

```
WIRELESS_LAN_ENABLE=true
WIRELESS_LAN_ESSID="myssid"
WIRELESS_LAN_PASSPHRASE="mypresharedkey"

STATIC_ADDRESS="192.0.2.10"
STATIC_NETMASK="255.255.255.0"
STATIC_GATEWAY="192.0.2.1"

RESOLV_SEARCH="local-network"
RESOLV_NAMESERVER="192.168.10.1"
```

## 5.1.2. ネットワークモジュールコンフィグ

ネットワークモジュールコンフィグを設定することで、リモートから Armadillo のネットワーク設定を管理できます。ネットワークモジュールコンフィグにはデフォルト設定が無いため、**必ず**設定する必要があります。

### 5.1.2.1. 使用可能な key 一覧

ネットワークモジュールコンフィグに設定可能な key は、「4.4.1.3. ネットワークモジュールコンフィグ一覧」を参照してください。

### 5.1.2.2. 設定例

以下はネットワークモジュールの設定例です。必ず、VERSION=2, interface\_\*\*\*=enable が書かれていることに注目してください。

#### 5.1.2.2.1. 3G モバイル回線を使う

```
VERSION=2

interface_wwan=enable
interface_wwan_type=dhcp
interface_wwan_id=user
interface_wwan_pass=password
interface_wwan_apn=apn
interface_wwan_auth_type=PAP
interface_wwan_pdp_type=IP
interface_wwan_metric=10
```

#### 5.1.2.2.2. wlan を使う

```
VERSION=2
```



```
interface_wlan0=enable
interface_wlan0_type=dhcp
interface_wlan0_essid=my-wireless-network
interface_wlan0_passphrase=preshared-key
interface_wlan0_metric=20
```

### 5.1.2.2.3. 固定 IP アドレスを使う

eth0 を使用する場合

```
VERSION=2

interface_eth0=enable
interface_eth0_metric=1
interface_eth0_type=static
interface_eth0_address=192.168.10.10
interface_eth0_netmask=255.255.255.0
interface_eth0_gateway=192.168.10.1

resolv_conf_search=local-network
resolv_conf_nameserver=192.168.10.100
```

wlan0 を使用する場合

```
VERSION=2

interface_wlan0=enable
interface_wlan0_type=static
interface_wlan0_address=192.168.10.10
interface_wlan0_netmask=255.255.255.0
interface_wlan0_gateway=192.168.10.1
interface_wlan0_essid=my-wireless-network
interface_wlan0_passphrase=preshared-key
interface_wlan0_metric=20

resolv_conf_search=local-network
resolv_conf_nameserver=192.168.10.100
```

### 5.1.2.2.4. 複数のインターフェースを同時に使う

```
VERSION=2

interface_eth0=enable
interface_eth0_metric=1
interface_eth0_type=static
interface_eth0_address=192.168.10.10
interface_eth0_netmask=255.255.255.0
interface_eth0_gateway=192.168.10.1

interface_wwan=enable
interface_wwan_type=dhcp
interface_wwan_id=user
interface_wwan_pass=password
```

```
interface_wwan_apn=apn
interface_wwan_auth_type=PAP
interface_wwan_pdp_type=IP
interface_wwan_metric=10

interface_wlan0=enable
interface_wlan0_type=dhcp
interface_wlan0_essid=my-wireless-network
interface_wlan0_passphrase=preshared-key
interface_wlan0_metric=20

resolv_conf_search=local-network
resolv_conf_nameserver=192.168.10.100
```

## 5.2. リカバリーイメージを使って Armadillo を復旧する

本章では、リカバリーイメージを使って Armadillo を復旧する手順を紹介します。

### 手順 5.1 リカバリーイメージを使った Armadillo の復旧

#### 1. リカバリーイメージで起動していることを確認する

はじめに、node-eye コントロールパネルにログインします。

リカバリーイメージで起動した Armadillo は、接続状態が [切断中] で表示されるため、[切断中] の Armadillo に対し ping を実行します。

"状態" が [成功] となった場合は**復旧作業ができます**。Armadillo はリカバリーイメージで起動しているので、次の手順に進んでください。

"状態" が [失敗] となった場合は**復旧作業ができません**。何らかの原因で Armadillo と SACM が接続できていないため、Armadillo を直接確認する必要があります。

#### 2. リカバリーイメージからプライマリーイメージを書き換える

ping が成功した Armadillo に対しファームウェアアップデートを行います。入力するカーネルとユーザーランドのダウンロード URL は、起動できるものを入力してください。ファームウェアアップデートには時間がかかる場合があるため、[アップデート完了時に再起動する] にチェックを入れてから、実行することをおすすめします。

#### 3. 書き込みを行ったイメージの確認

ファームウェアアップデートが成功し、再起動すると、書き込んだイメージが起動します。また、SACM と通信ができると接続状態が [接続中] になります。これでファームウェアアップデートは完了です。

再起動後、接続状態が [切断中] → [接続中] に切り替わるまでに時間がかかる場合があります。10 分以上切り替わらない場合は、リカバリーイメージで起動していることが考えられます。この場合は、再び復旧作業を行ってください。

## 5.3. リカバリーイメージの起動をテストする

本章では、Armadillo に書き込んだリカバリーイメージが正常に起動するか確認する方法を説明します。

手順の中で、既に Armadillo に書き込まれているプライマリイメージを上書きします。リカバリーイメージの起動を確認した後に、元に戻すための正常なプライマリイメージファイルを用意しておいてください。

## 手順 5.2 リカバリーイメージの起動テスト

### 1. 起動しないプライマリイメージファイルを作る

正常に起動することができないイメージを書き込んだ状況を再現するため、サイズが 1 バイトのファイルを gzip 圧縮して不正なプライマリイメージファイルとして用意します。次の手順を ATDE 等の Linux マシンで行ってください。

```
[PC~]$ echo | gzip > dummy.tar.gz
[PC~]$ md5sum dummy.tar.gz > dummy.tar.gz.md5
[PC~]$ ls dummy*
dummy.tar.gz  dummy.tar.gz.md5
```

### 2. 起動しないプライマリイメージを Armadillo から接続可能な Web サーバーに配置する

作成した 2 つのファイルを Armadillo がアクセスすることのできる Web サーバーに配置してください。

### 3. node-eye コントロールパネルから起動しないプライマリイメージを使ってファームウェアアップデートを実行する

node-eye コントロールパネル で対象となる Armadillo を選択して、不正なプライマリイメージファイル "dummy.tar.gz" への URL を [ユーザーランド] に入力し、[アップデート完了後に再起動する] にチェックを入れた上で、ファームウェアアップデートを実行してください。

[カーネル] のテキストボックスには何も入力する必要はありません。

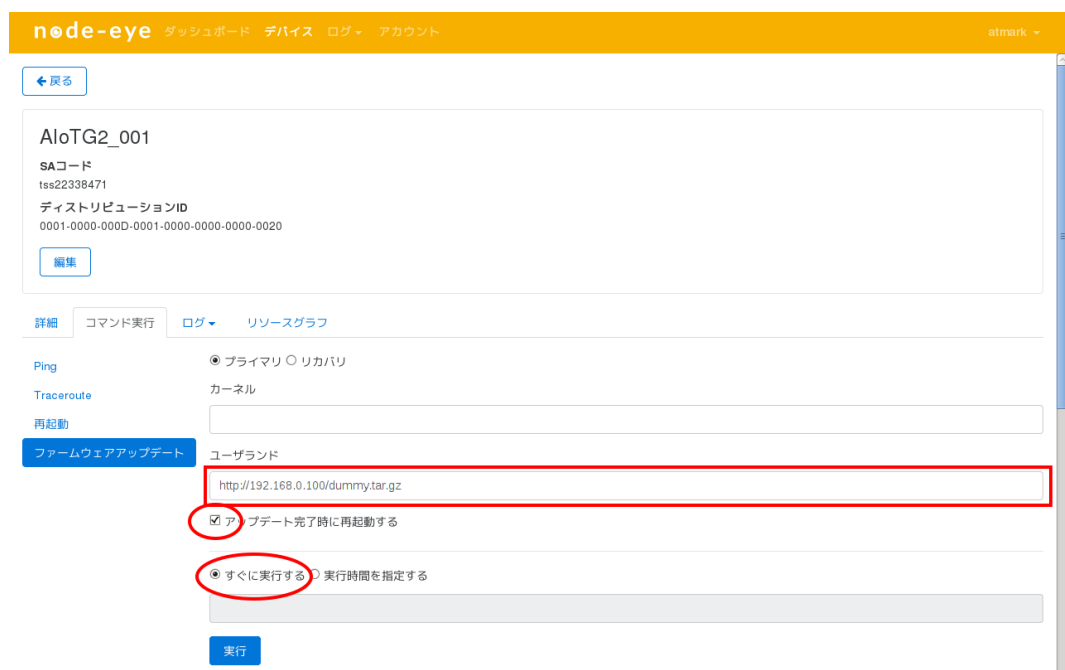


図 5.1 ファームウェアアップデート (node-eye コントロールパネル)

#### 4. リカバリーイメージによって Armadillo が起動した事を確認する

起動しないプライマリイメージファイルによるファームウェアアップデートに成功して Armadillo は再起動を実行します。しかし、起動しないプライマリイメージによって起動が失敗するため、自動的にリカバリーイメージで再起動します。

リカバリーイメージで再起動した Armadillo は node-eye コントロールパネル または SACM コントロールパネルからは [切断中] として認識されますが、Ping、Traceroute、再起動、ファームウェアアップデートは実行することができます。

起動しないプライマリイメージのアップデートによる再起動を実行してから 15 分以上経過した後に、対象の Armadillo が node-eye コントロールパネルから [切断中] として表示される事と、Ping が実行できる事を確認してください。

確認が終わったら正常なプライマリイメージを使ってファームウェアアップデートを実行してください。

## 5.4. node-eye インストール済のユーザーランド ビルド方法

アットマークテクノが提供しているユーザーランドアーカイブには、標準で node-eye がインストールされていません。そのため、そのままファームウェアアップデートしてしまうと SACM に接続できなくなってしまいます。

本章では、node-eye 環境をインストールしたユーザーランドの作成方法について説明します。

### 手順 5.3 node-eye 環境をインストールしたユーザーランドの作成方法

#### 1. x1-debian-builder の準備

カスタマイズしたユーザーランドの作成には、専用のツールが必要です。アットマークテクノのダウンロードサイトからお使いの環境に合わせた機種、バージョンを選択し、「Debian GNU/Linux ルートファイルシステム 構築ツール」である x1-debian-builder を ATDE 上にダウンロードしてください。

ダウンロード後は以下のように解凍し、該当するディレクトリ内に入ります。

```
[PC ~]$ ls
x1-debian-builder-[version].tar.gz
[PC ~]$ tar zxf x1-debian-builder-[version].tar.gz
[PC ~]$ cd x1-debian-builder-[version]
[PC ~]/x1-debian-builder-[version]$ ls
aiotg3_resources  ax1_resources  aiotg3l_resources  build.sh
```

#### 2. パッケージインストールリストに node-eye 追加

x1-debian-builder のビルド時にインストールされる、パッケージのリストへ node-eye を追加します。以下のように packages ファイルを開き、最下部に[node-eye]を追記してください。

```
[PC ~]/x1-debian-builder-[version]$ vi aiotg3il_resources/resources/packages
# List of packaes will install in the debootstrap chroot environment.
# A line start with # is comment. Separate packages name with new-line or space.
```

```
#Utils
sudo
locales
sed
less
expect
minicom
cu
vim-tiny
dosfstools
util-linux
rsyslog
bash-completion
man
dialog
cpufrequtils
pds6j-utils
atmark-x1-base

#Device
usbutils
i2c-tools
evtest
bluez

#Network
iptables
network-manager
modemmanager
net-tools
avahi-daemon
avahi-autoipd
ntpdate
lighttpd
curl
wget
traceroute
iputils-ping
iputils-arping
dnsutils
mosquitto-clients
openssh-client

#Dev
ruby2.1
rubygems
bundler
lua5.2
node-eye
```

### 3. line.conf の作成

line.conf を x1-debian-builder 下に作成します。

line.conf の説明,設定例は「5.1.1. line.conf」に記載しています。そちらを参考に、お客様のネットワーク環境に合わせた line.conf を作成してください。

```
[PC ~]/x1-debian-builder-[version]$ vi line.conf
```

#### 4. line.conf の配置

作成した line.conf をユーザーランドに配置します。

```
[PC ~]/x1-debian-builder-[version]$ cp line.conf aiotg3il_resources/resources/
```

その後 aiotg3il\_resources/resources/fixup に cp コマンドを追記します。これを行うことで、node-eye インストール時に任意の line.conf へ上書きできます。

```
[PC ~]/x1-debian-builder-[version]$ vi aiotg3il_resources/resources/fixup
#!/bin/sh

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

PKGS=$(cat /resources/packages | grep -v ^# | xargs)

cp /resources/sources.list /etc/apt/sources.list
cp /resources/atmark-techno-pin-1001 /etc/apt/preferences.d/

apt-key adv --keyserver pgp.nic.ad.jp --recv-keys 3874DA771B351757
apt-get -y update
apt-get -y install $PKGS

#Setup locales
locale-gen
sed -i 's/# ja_JP.UTF-8 UTF-8/ja_JP.UTF-8 UTF-8/' /etc/locale.gen
locale-gen

#Setup timezone
echo "Asia/Tokyo" > /etc/timezone
dpkg-reconfigure -f noninteractive tzdata

#user/password/hostname
useradd atmark
echo "atmark:atmark" | chpasswd
usermod -s /bin/bash -Gcdrom, floppy, audio, dip, video, plugdev atmark
mkhomedir_helper atmark

echo "root:root" | chpasswd
usermod -Guucp,dialout root
echo "armadillo" > /etc/hostname

#fix fstab
mkdir /opt/license
if [ $(grep -c proc /etc/fstab) -eq 0 ]; then
    echo "proc                /proc                proc                defaults                0 0"
>> /etc/fstab
    echo "/dev/mtdblock1          /opt/license          squashfs defaults,nofail      0 0"
>> /etc/fstab
fi
```

```
#fix rc.local
cp /resources/rc.local /etc/rc.local
cp /resources/hosts /etc/hosts
cp /resources/line.conf /etc/armsd/line.conf
```

## 5. x1-debian-builder のビルド

設定が完了したら、x1-debian-builder をビルドして node-eye インストール済のユーザーランドを生成します。

```
[PC ~]/x1-debian-builder-[version]$ sudo ./build.sh
```

## 6. ユーザーランドの生成確認

ビルド完了後、/x1-debian-builder-[version]下にユーザーランドが生成されます。

```
[PC ~]/x1-debian-builder-[version]$ ls
aiotg3_resources  ax1_resources  aiotg3l_resources  build.sh  debian-jessie-
armhf_aiotg3l_YYYYMMDD.tar.gz
```



## 5.5. モジュールの作成方法

本章では、Armadillo に独自のモジュールを組込む手順について説明します。

モジュールを作成して Armadillo のユーザーランドに組込むためには、node-eye パッケージをインストールしておく必要があります。

### 5.5.1. モジュールの配置

モジュールはユーザーランドの下記のパスに配置します。

```
/etc/armsd/user_modules/
```

モジュール名はモジュール番号(ユーザー向けは 4 から 15) + "\_"で始まるファイル名で作成してください。

例えば、5\_mymodule.sh というファイル名でモジュールを作成した場合、モジュール 5 として実行されます。

ファイル名のモジュール番号は重複しないようにしてください。同じ番号が複数ある場合、ファイル名を ascii でソートして若い方の 1 つだけがモジュールの実装として機能します。

### 5.5.2. モジュールのサンプル

モジュールのサンプルは下記のパスに配置されています。

- ・ /etc/armsd/user\_modules/4\_command

下記にその内容を記載します。

```

#!/bin/sh

# This modele is a sample.

PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
OPS=${1}
shift

case ${OPS} in
  command)
    sh ${2} > ${3} 2>&1
    ;;
  *)
    exit 1
    ;;
esac

exit 0

```

このモジュールは SACM コントロールパネル から モジュール 4 として操作することができ、**任意コマンド実行** で与えるパラメーターを **sh** の引数として、Armadillo 上で実行することができます。

文中の case 文で分岐する条件の OPS にはモジュールの **オペレーション** が文字列として代入されます。実装することのできる **オペレーション** は下記のサンプルコード N\_sample を参考にしてください。

- ・ /etc/armisd/user\_modules/N\_sample

```

#!/bin/sh

PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
OPS=${1}
shift

syslog_debug() {
  logger -t armsd.module.$(basename ${0}) ${1}
}

case ${OPS} in
  start)
    syslog_debug "START: <id=${1}> <version=${2}> <info=${3}> <configure=${4}>"
    ;;
  stop)
    syslog_debug "STOP: <id=${1}>"
    ;;
  reconfig)
    syslog_debug "RECONFIG: <id=${1}> <version=${2}> <info=${3}> <configure=${4}>"
    ;;
  status)
    syslog_debug "STATUS: <id=${1}> <requestfile=${2}> <resultfile=${3}>"
    syslog_debug "request: $(cat ${2})"
    echo "STATUS RESULT" > ${3}
    ;;
  command)
    syslog_debug "COMMAND: <id=${1}> <requestfile=${2}> <resultfile=${3}>"
    syslog_debug "request: $(cat ${2})"
    echo "COMMAND RESULT" > ${3}

```



```

;;
*)
  exit 1
;;
esac

exit 0

```

### 5.5.3. モジュールのオペレーションの種類

各モジュールの case 文に書く事のできるオペレーションの種類と、実行される条件は下記の通りです。

表 5.6 各モジュールの case 文に書く事のできるオペレーションの種類と実行条件

オペレーション	実行タイミング
start	armsd が SACM に接続を確立した直後
stop	SACM または node-eye コントロールパネルで <b>再起動</b> を実行
reconfig	SACM コントロールパネルで <b>コンフィグ反映</b> を実行
status	SACM コントロールパネルで <b>ステータス取得</b> を実行
command	SACM コントロールパネルで <b>任意コマンド</b> を実行

モジュールを作る場合は、必ずしも全ての **オペレーション** に対応する case 文を書く必要はありません。SACM コントロールパネルからの各オペレーションの実行方法については **SACM マニュアル** の「オペレーション」タブ [[http://manual.sacm.jp/management\\_spec/cp\\_screen/operation.html](http://manual.sacm.jp/management_spec/cp_screen/operation.html)] を参照してください。

### 5.5.4. モジュールの設計の注意点

個々のモジュールのタイムアウト時間は 180 秒です。start/stop/command 等の **オペレーション** を呼び出してから、180 秒以内に実行が終了しない場合はモジュールの実行が中断されます。

モジュールの返り値はモジュールへのオペレーションの成功/失敗の判定に影響することはありません。スクリプトのテスト等の用途で自由に設定することができます。

## 5.6. テンプレートセットを活用した複数 Armadillo の一括設定

SACM コントロールパネルではテンプレートセットという仕組みを使って 複数の Armadillo に対して、同時にモジュールのコンフィグを実行することができます。使い方は SACM service manual のテンプレートセットについて [[http://manual.sacm.jp/management\\_spec/template.html](http://manual.sacm.jp/management_spec/template.html)]を参照してください。

## 6. 機能制限

---

開発中の node-eye の実装には次の制限があります。

- ・ 複数のネットワークインターフェースを DHCP で設定した場合に、`resolve.conf` の上書きにより SACM に接続できない場合がある
- ・ 複数のネットワークインターフェースを DHCP で設定した場合に、インターネットにルーティングしないネットワークセグメントに接続したネットワークインターフェースが含まれる場合、SACM に接続できない場合がある

## 改訂履歴

バージョン	年月日	改訂内容
1.0.0	2016/12/16	・ 初版発行
1.0.1	2017/05/29	・ 「表 4.5. ネットワークモジュールの eth0 に関する key 一覧」に interface_eth0_never-default の説明を追加 ・ 「表 4.7. ネットワークモジュールの wlan0 に関する key 一覧」に interface_wlan0_never-default の説明を追加 ・ 「表 5.2. eth0 で使用可能な key 一覧」に ETHERNET_ENABLE の説明を追加
1.1.0	2018/11/20	・ Linux 4.9 対応に伴い、ファイルバージョンや説明の追加 ・ 「3.2.5. ファームウェアアップデートを行う」にユーザーランドのファームウェアアップデート方法を追加 ・ 「4.5.5. リカバリーイメージの書き込み方法」を追加 ・ 「5.4. node-eye インストール済のユーザーランドビルド方法」を追加
1.1.1	2018/11/28	・ 「4.5.5. リカバリーイメージの書き込み方法」の名前変更、zip 圧縮例にて dtb の名前変更が抜けていた不具合を修正

