

# node-eye マニュアル

Version 1.0.1  
2017/12/26

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 © 2016-2017 Atmark Techno, Inc.

Version 1.0.1  
2017/12/26

# 目次

1. はじめに .....	7
1.1. 表記について .....	7
1.1.1. フォント .....	7
1.1.2. コマンド入力例 .....	7
1.1.3. アイコン .....	7
2. node-eye とは .....	9
2.1. WebUI の概要 .....	9
3. node-eye を体験する .....	11
3.1. Armadillo の準備 .....	11
3.2. node-eye コントロールパネルを使う .....	14
3.2.1. node-eye コントロールパネルにログインする .....	14
3.2.2. 全体の稼働状況を見る .....	15
3.2.3. 各デバイスの稼働状況を見る .....	15
3.2.4. リソースモニタリングを行う .....	17
3.2.5. ファームウェアアップデートを行う .....	18
3.2.6. ログを見る .....	20
3.2.7. アカountの追加/変更/削除を行う .....	21
3.3. SACM を使う .....	26
3.3.1. SACM へログインする .....	26
3.3.2. リモートコンフィグを行う .....	27
3.3.3. フォルダ機能を使う .....	32
3.3.4. 接続状態が変化した時の通知先を設定する .....	37
3.3.5. その他 .....	46
4. Armadillo ソフトウェア仕様 .....	47
4.1. NOR フラッシュメモリのパーティション構成 .....	47
4.2. デフォルトでインストールされるアプリケーションの違い .....	48
4.3. リカバリイメージ .....	49
4.3.1. リカバリイメージの目的 .....	49
4.3.2. リカバリイメージの起動条件 .....	49
4.3.3. リカバリイメージの特徴 .....	49
4.3.3.1. インストールされているアプリケーション .....	49
4.3.3.2. リカバリイメージ起動中の WebUI 上の表示 .....	49
4.4. 監視方法 .....	49
4.5. リモートコンフィグ .....	50
4.5.1. node-eye におけるモジュールについて .....	50
4.6. ネットワーク設定 .....	51
4.6.1. ネットワークインターフェース設定シーケンス .....	51
4.7. ファームウェアアップデート .....	52
5. イメージファイルのビルド手順 .....	53
6. Howto .....	57
6.1. リカバリイメージを使って Armadillo を復旧する .....	57
6.2. リカバリイメージの起動をテストする .....	57
6.3. SACM からネットワークインターフェースの設定を行う .....	59
6.3.1. ネットワークモジュール仕様 .....	59
6.3.1.1. コンフィグの書式 .....	59
6.3.1.2. 使用可能な key 一覧 .....	60
6.3.2. 設定例 .....	61
6.3.2.1. 3G モバイル回線を使う .....	61
6.3.2.2. AWL13 を使う .....	61
6.3.2.3. 固定 IP アドレスを使う .....	61

---

6.3.2.4. 複数のインターフェースを同時に使う .....	62
6.3.3. SACM へ初めて接続する時のネットワーク設定を変更する .....	63
6.3.3.1. 使用可能な key 一覧 .....	67
6.3.3.2. 設定例 .....	67
6.4. Armadillo: モジュールの作成方法 .....	68
6.4.1. モジュールの配置 .....	68
6.4.2. モジュールのサンプル .....	69
6.4.3. モジュールのオペレーションの種類 .....	70
6.4.4. モジュールの設計の注意点 .....	71
6.5. テンプレートセットを活用した複数 Armadillo の一括設定 .....	71
6.6. Armadillo を node-eye 非対応の標準イメージに戻す .....	71
6.6.1. bootloader 領域の書き換え .....	72
6.6.2. 再起動 .....	72
6.6.3. kernel 領域の書き換え .....	72
6.6.4. userland 領域の書き換え .....	72
7. 機能制限 .....	73

## 図目次

3.1. ダッシュボード (node-eye コントロールパネル) .....	15
3.2. デバイス一覧 (node-eye コントロールパネル) .....	16
3.3. デバイス詳細 (node-eye コントロールパネル) .....	16
3.4. デバイス複数選択 (node-eye コントロールパネル) .....	17
3.5. デバイス複数選択 (node-eye コントロールパネル) .....	18
3.6. リソースモニタリング (node-eye コントロールパネル) .....	18
3.7. デバイス一覧 (node-eye コントロールパネル) .....	19
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) .....	19
3.9. タスク (node-eye コントロールパネル) .....	21
3.10. イベント (node-eye コントロールパネル) .....	21
3.11. コンフィグ反映 (SACM コントロールパネル) .....	30
3.12. 即時反映 (SACM コントロールパネル) .....	31
3.13. フォルダタブ (SACM コントロールパネル) .....	32
3.14. フォルダ名入力 (SACM コントロールパネル) .....	33
3.15. フォルダー一覧 (SACM コントロールパネル) .....	34
3.16. 所属 SA 変更 (SACM コントロールパネル) .....	35
3.17. フォルダに追加する SA を選択 (SACM コントロールパネル) .....	36
3.18. SA が追加されたフォルダ (SACM コントロールパネル) .....	37
3.19. 監視タブ (SACM コントロールパネル) .....	38
3.20. 監視グループ名入力 (SACM コントロールパネル) .....	39
3.21. 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) .....	40
3.22. 監視グループの詳細 (SACM コントロールパネル) .....	41
3.23. 通知先メールアドレス入力 (SACM コントロールパネル) .....	42
3.24. 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) .....	43
3.25. 監視グループ詳細 (SACM コントロールパネル) .....	44
3.26. 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) .....	45
3.27. SA が追加された監視グループ (SACM コントロールパネル) .....	46
6.1. ファームウェアアップデート (node-eye コントロールパネル) .....	58

## 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 .....	7
1.3. コマンド入力例での省略表記 .....	7
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. node-eye 標準イメージ一覧 .....	12
3.3. ファームウェアアップデートに利用するイメージファイルの一覧 .....	20
4.1. Armadillo-IoT ゲートウェイ スタンダードモデル G2 出荷時 (node-eye 非対応、Armadillo の型番が AG42*-ではじまる場合) .....	47
4.2. node-eye 対応(Armadillo の型番が AG42*-ではじまる場合) .....	47
4.3. Armadillo-IoT ゲートウェイ スタンダードモデル G2 出荷時 (node-eye 非対応、Armadillo の型番が AG43*-ではじまる場合) .....	48
4.4. node-eye 対応(Armadillo の型番が AG43*-ではじまる場合) .....	48
4.5. モジュールのタイプとその用途 .....	50
4.6. アットマークテクノが提供しているモジュール一覧 .....	51
5.1. ソースコードの一覧 .....	53
6.1. ネットワークモジュールの eth0 に関する key 一覧 .....	60
6.2. ネットワークモジュールの umts0 に関する key 一覧 .....	60
6.3. ネットワークモジュールの awlan0 に関する key 一覧 .....	60
6.4. ネットワークモジュールの resolv.conf に関する key 一覧 .....	61
6.5. ディレクトリの用途 .....	66
6.6. eth0 で 使用可能な key の説明 .....	67
6.7. umts0 で使用可能な key の説明 .....	67
6.8. awlan0 で使用可能な key の説明 .....	67
6.9. resolv で使用可能な key の説明 .....	67
6.10. 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 .....	70
6.11. 工場出荷状態に戻すため使用するイメージファイル一覧 .....	71

# 1. はじめに

## 1.1. 表記について

### 1.1.1. 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

### 1.1.2. 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
[armadillo ~/]#	Armadillo 上の root ユーザで実行
[armadillo ~/]\$	Armadillo 上の一般ユーザで実行
hermit>	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

### 1.1.3. 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 [http://node-eye.com] をご覧ください。



node-eye では NOR フラッシュメモリに書き込む 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>

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。

### 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のできる開発向けの 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 5.34 documentation [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 G2** (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。

以降の手順を行なうために、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
Armadillo-IoT ゲートウェイ G2 開発セット	
ATDE5	
ライセンスファイル(squashfs-license-ne- <i>[Distribution ID]</i> .img)	
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	

手順の中では Armadillo の NOR フラッシュメモリの書き換えとシリアルコンソールを使ったブートローダーの操作を行います。Armadillo-IoT ゲートウェイ G2 のスタートアップガイドと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 ドキュメント・ダウンロード

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



ライセンスファイルは node-eye 登録完了通知書と同時にメールでお送りしています。

### 3.1. Armadillo の準備

node-eye を利用するには、Armadillo の NOR フラッシュメモリに node-eye に対応したイメージファイルが書き込まれている必要があります。

Armadillo に node-eye 対応イメージを書き込む際に、パーティション構成の変更を伴うため、イメージの書き換えは順を追って行う必要があります。下記の順番でイメージを書き換えてください。

1. bootloader 領域の書き換えと再起動
2. recovery 領域の書き換え
3. kernel 領域と userland 領域の書き換え
4. license 領域の書き換え

ライセンスイメージファイル以外のイメージファイルは下記 URL からダウンロードすることができます。ライセンスイメージファイルは弊社営業担当よりメールにてお送りしています。

表 3.2 node-eye 標準イメージ一覧

リージョン	ファイル名
ブートローダー	<a href="http://download.atmark-techno.com/node-eye/image/loader-armadillo-iotg-std-ne-v3.10.0-ne2.bin">http://download.atmark-techno.com/node-eye/image/loader-armadillo-iotg-std-ne-v3.10.0-ne2.bin</a>
リカバリ	<a href="http://download.atmark-techno.com/node-eye/image/recovery-aiotg-std-ne2.img">http://download.atmark-techno.com/node-eye/image/recovery-aiotg-std-ne2.img</a>
カーネル	<a href="http://download.atmark-techno.com/node-eye/image/linux-aiotg-std-v2.09-ne2.bin.gz">http://download.atmark-techno.com/node-eye/image/linux-aiotg-std-v2.09-ne2.bin.gz</a>
ユーザーランド	<a href="http://download.atmark-techno.com/node-eye/image/romfs-aiotg-std-v2.06-ne2.img.gz">http://download.atmark-techno.com/node-eye/image/romfs-aiotg-std-v2.06-ne2.img.gz</a>



型番が AG43\*-ではじまる Armadillo をご利用の場合、以下バージョンのイメージファイルを使用する必要があります。

- ・ loader-armadillo-iotg-std-ne-v3.10.0-ne2.bin 以降
- ・ recovery-aiotg-std-ne2.img 以降
- ・ linux-aiotg-std-v2.09-ne2.bin.gz 以降
- ・ romfs-aiotg-std-v2.06-ne2.img.gz 以降

### 手順 3.1 Armadillo の準備

#### 1. イメージファイルの準備

Armadillo にイメージの書き込みを行なう前に ATDE5 の /var/lib/tftpboot/ ディレクトリに各イメージをコピーしておいてください。

```
[PC ~]$ cd /var/lib/tftpboot/
[PC /var/lib/tftpboot/]$ ls
linux-aiotg-std-v2.09-ne2.bin.gz      romfs-aiotg-std-v2.06-ne2.img.gz
loader-armadillo-iotg-std-ne-v3.10.0-ne2.bin  squashfs-license-ne-[Distribution ID].img
recovery-aiotg-std-ne2.img
```

#### 2. ブートローダーイメージの書き換え

次に、Armadillo を保守モードで起動し、下記コマンドを実行してください。

```
hermit> tftpdn [Armadillo の IP アドレス] [ATDE5 の IP アドレス] \
--bootloader=loader-armadillo-iotg-std-ne-v[BASE_VERSION]-ne[VERSION].bin
```

tftpdn コマンドの詳細は Armadillo-IoT ゲートウェイ G2 製品マニュアル「12.4. TFTP を使用してフラッシュメモリを書き換える [http://manual.atmark-techno.com/armadillo-iot/armadillo-iotg-std\_product\_manual\_ja-2.5.2/ch12.html#sct.update\_image-with-tftpdn]」を参照してください。

一度 Armadillo の電源を一度切断し、再度保守モードで起動してください。memmap コマンドを実行し、下記のようなパーティション構成となっていれば node-eye 対応のブートローダーが動作しています。

型番が AG42\*-ではじまる Armadillo をご利用の場合、次の結果になります。

```
hermit> memmap
0xa0000000:0xa1ffffff FLA all bf:8K bl:4x32K/l, 255x128K/l
```

```
0xa0000000:0xa001ffff FLA bootloader bf:8K bl:4x32K/l
0xa0020000:0xa081ffff FLA recovery bf:8K bl:64x128K/l
0xa0820000:0xa0c1ffff FLA kernel bf:8K bl:32x128K
0xa0c20000:0xa1edffff FLA userland bf:8K bl:150x128K
0xa1ee0000:0xa1efffff FLA license bf:8K bl:1x128K/l
0xa1f00000:0xa1ffffff FLA config bf:8K bl:8x128K
0x80000000:0x87ffffff RAM dram-1
```

型番が AG43\*-ではじまる Armadillo をご利用の場合、次の結果になります。

```
hermit> memmap
0xa0000000:0xa1ffffff FLA all bf:8K bl:256x128K/l
0xa0000000:0xa003ffff FLA bootloader bf:8K bl:2x128K/l
0xa0040000:0xa083ffff FLA recovery bf:8K bl:64x128K/l
0xa0840000:0xa0c3ffff FLA kernel bf:8K bl:32x128K
0xa0c40000:0xa1edffff FLA userland bf:8K bl:149x128K
0xa1ee0000:0xa1efffff FLA license bf:8K bl:1x128K/l
0xa1f00000:0xa1ffffff FLA config bf:8K bl:8x128K
0x80000000:0x87ffffff RAM dram-1
```

### 3. リカバリイメージの書込み

続いて、リカバリイメージを書き込みます。下記のコマンドを実行してください。

```
hermit> tftpd [Armadillo の IP アドレス] [ATDE5 の IP アドレス] --blksize=1024 \
--recovery=recovery-aiotg-std-ne[VERSION].img
```

### 4. プライマリイメージの書き込み

プライマリイメージを書き込みます。下記のコマンドを実行してください。

```
hermit> tftpd [Armadillo の IP アドレス] [ATDE5 の IP アドレス] --blksize=1024 \
--kernel=linux-iotg-std-v[BASE_VERSION]-ne[VERSION].bin.gz \
--userland=romfs-aiotg-std-v[BASE_VERSION]-ne[VERSION].img.gz
```

### 5. license 領域の書き込み

license 領域用イメージを書き込みます。下記のコマンドを実行してください。複数の Armadillo で node-eye を評価される場合、それぞれの個体に別々のライセンスファイルを書き込んでください。

```
hermit> tftpd [Armadillo の IP アドレス] [ATDE5 の IP アドレス] \
--license=squashfs-license-ne-[Distribution ID].img
```

license 領域用イメージのファイル名は、squashfs-license-ne-[Distribution ID].img となっています。Distribution ID は、WebUI から操作するデバイスを識別するために必要になるので、必ず Distribution ID と書き込んだ Armadillo の対応関係を控えておいてください。

### 6. Armadillo の config を初期化する

次のように"boot"コマンドを実行して Linux システムを起動してください。

```
hermit> boot
Uncompressing kernel.....
.....
done.
Uncompressing ramdisk.....
.....
.....
```

起動が完了するとログインプロンプトが表示されます。root ユーザーでログインしてください。ユーザー名: root パスワード: root

```
armadillo-iotg login: root
Password:
[root@armadillo-iotg (ttymxc1) ~]#
```

ログインできたら、"flatfsd"コマンドを使ってコンフィグ領域を初期化します。初期化時には、node-eye 対応イメージの"/etc/default"ディレクトリ以下のファイルがコンフィグ領域に保存され、且つ"/etc/config"ディレクトリにファイルが複製されます。

```
[root@armadillo-iotg (ttymxc1) ~]# flatfsd -w
flatfsd: Nonexistent or bad flatfs (-1), creating new one...
flatfsd: saving fs to partition 0, tstamp=263
flatfsd: Wrote 5941 bytes to flash in 2 seconds
flatfsd: Created 14 configuration files (3795 bytes)
```

Armadillo の電源を入れなおして再起動してください。

## 7. Armadillo を起動する

保守モードで"boot"コマンドを実行し、Linux システムを起動すると、Armadillo は起動後に自動的に SACM へ接続を行います。

SACM への接続が成功すると、起動から 10 分以内に node-eye コントロールパネル、SACM コントロールパネル でデバイスの接続状態が [切断中]から[接続中]に切り替わります。

これで Armadillo の準備は終了しました。実際に WebUI を操作してみましょう。

## 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。

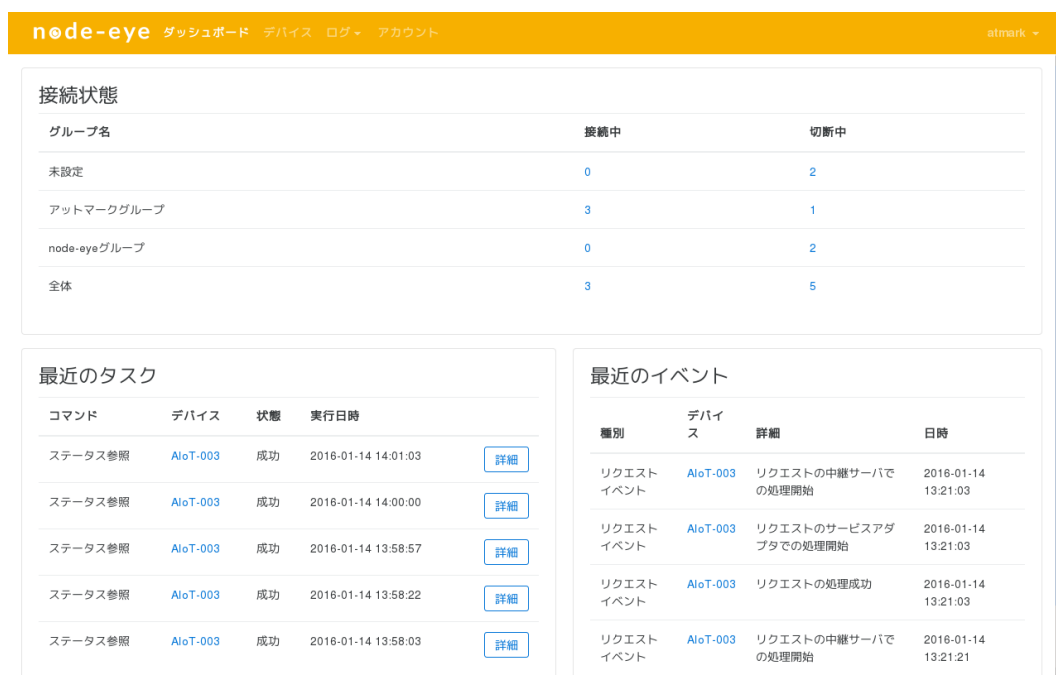


図 3.1 ダッシュボード(node-eye コントロールパネル)

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。

node-eye ダッシュボード デバイス ログ アカウント				atmark
グループ <input type="text" value="すべて"/> 接続状態 <input type="text" value="すべて"/> 設定状態 <input type="text" value="すべて"/> デバイス名 / SAコード <input type="text"/>	<input type="checkbox"/> デバイス			
	<input type="checkbox"/> AIoT-006 SAコード: tss18740226 / ディストリビューションID: 0001-0000-0000-0001-0000-0000-0000-0016	アットマークグループ	接続中	
	<input type="checkbox"/> AIoT-004 SAコード: tss18740233 / ディストリビューションID: 0001-0000-0000-0009-0000-0000-0000-0000		切断中	
	<input type="checkbox"/> AIoT-007 SAコード: tss18740264 / ディストリビューションID: 0001-0000-0000-0009-0000-0000-0000-0BFC	node-eyeグループ	切断中	
	<input type="checkbox"/> AIoT-005 SAコード: tss18740271 / ディストリビューションID: 0001-0000-0000-0001-0000-0000-0000-0019	node-eyeグループ	切断中	
	<input type="checkbox"/> AIoT-001 SAコード: tss21373220 / ディストリビューションID: 0001-0000-0000-0009-0000-0000-0000-1084	アットマークグループ	接続中	
	<input type="checkbox"/> AIoT-003 SAコード: tss21373251 / ディストリビューションID: 0001-0000-0000-0009-0000-0000-0000-1551	アットマークグループ	接続中	
	<input type="checkbox"/> AIoT-002 SAコード: tss21373268 / ディストリビューションID: 0001-0000-0000-0009-0000-0000-0000-1550	アットマークグループ	切断中	
	<input type="checkbox"/> AIoT-000 SAコード: tss21545443 / ディストリビューションID: 0001-0000-0000-0001-0000-0000-0000-001F		切断中	
0 / 8 選択		<input type="button" value="ファームウェアアップデート"/> <input type="button" value="再起動"/> <input type="button" value="コマンド実行"/> <input type="button" value="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。

node-eye

ダッシュボード

デバイス

ログ

アカウント

atmark

戻る

AIoT-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				atmark
グループ すべて	接続状態 すべて	設定状態 すべて	デバイス名 / SAコード	
グループ		デバイス	グループ	接続状態
<input checked="" type="checkbox"/>	AIoT-001	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
<input checked="" type="checkbox"/>	AIoT-002	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
<input checked="" type="checkbox"/>	AIoT-003	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
<input type="checkbox"/>	AIoT-006	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"/>	AIoT-007	SAコード: tss18740264 / ディストリビューションID: 0001-0000-0000-0009-0000-0000-0BFC	node-eyeグループ	切断中
<input type="checkbox"/>	AIoT-005	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"/>	AIoT-004	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"/>	AIoT-000	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。

node-eye ダッシュボード デバイス ログ アカウント				atmark
グループ すべて	接続状態 すべて	設定状態 すべて	デバイス名 / SAコード	
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

デバイス	グループ	接続状態
<input checked="" type="checkbox"/> AIoT-001 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
<input checked="" type="checkbox"/> AIoT-002 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
<input checked="" type="checkbox"/> AIoT-003 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
<input type="checkbox"/> AIoT-006 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"/> AIoT-007 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"/> AIoT-005 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"/> AIoT-004 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"/> AIoT-000 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

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。

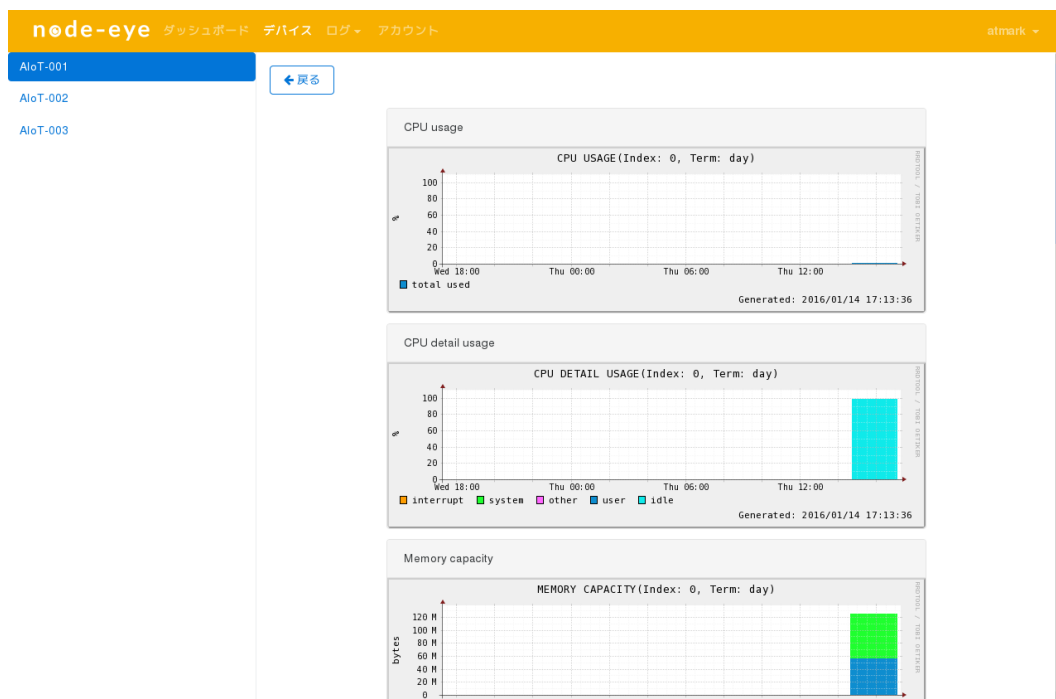


図 3.6 リソースモニタリング (node-eye コントロールパネル)

### 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。

アップデートで書き込むファイルは、Armadillo が接続されているネットワーク経由でダウンロードする事が可能な Web サーバーに公開されている必要があります。ここでは、例としてアットマークテクノが公開している node-eye 対応のイメージファイルを使います。

まず、メニューから[デバイス]を選択します

node-eye ダッシュボード デバイス ログ アカウント atmark			
グループ すべて	<input type="checkbox"/> デバイス	グループ	接続状態
接続状態 すべて	<input type="checkbox"/> AIoT-006 SAコード: tss18740226 / ディストリビューションID: 0001-0000-0000-0001-0000-0000-0000-0016	アットマークグループ	接続中
設定状態 すべて	<input type="checkbox"/> AIoT-004 SAコード: tss18740293 / ディストリビューションID: 0001-0000-0000-0009-0000-0000-0000-0000		切断中
デバイス名 / SAコード <input type="text"/>	<input type="checkbox"/> AIoT-007 SAコード: tss18740264 / ディストリビューションID: 0001-0000-0000-0009-0000-0000-0000-00FC	node-eyeグループ	切断中
	<input type="checkbox"/> AIoT-005 SAコード: tss21373220 / ディストリビューションID: 0001-0000-0000-0001-0000-0000-0000-0019	node-eyeグループ	切断中
	<input type="checkbox"/> AIoT-001 SAコード: tss21373220 / ディストリビューションID: 0001-0000-0000-0009-0000-0000-0000-1084	アットマークグループ	接続中
	<input type="checkbox"/> AIoT-003 SAコード: tss21373251 / ディストリビューションID: 0001-0000-0000-0009-0000-0000-0000-1551	アットマークグループ	接続中
	<input type="checkbox"/> AIoT-002 SAコード: tss21373268 / ディストリビューションID: 0001-0000-0000-0009-0000-0000-0000-1550	アットマークグループ	切断中
	<input type="checkbox"/> AIoT-000 SAコード: tss21546443 / ディストリビューションID: 0001-0000-0000-0001-0000-0000-0000-001F		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 3.7 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。

node-eye ダッシュボード デバイス ログ アカウント atmark																											
AIoT-001 AIoT-003 AIoT-002	<input type="button" value="戻る"/> Ping Traceroute 再起動 <input type="button" value="ファームウェアアップデート"/>	● プライマリ ○ リカバリ カーネル <input type="text" value="http://developers.node-eye.com/device-management/downloads/aiot_g2/images/linux-aiotg-std-v2.04-ne1.b"/> ユーザランド <input type="text" value="http://developers.node-eye.com/device-management/downloads/aiot_g2/images/romfs-aiotg-std-v2.06-ne1"/> <input type="checkbox"/> アップデート完了時に再起動する ● すぐに実行する ○ 実行時間を指定する <input type="text"/> <input type="button" value="実行"/>																									
		<table> <thead> <tr> <th>デバイス</th><th>状態</th><th>実行日時</th><th></th></tr> </thead> <tbody> <tr> <td>AIoT-003</td><td>成功</td><td>2016-01-14 18:43:55</td><td><input type="button" value="詳細"/></td></tr> <tr> <td>AIoT-003</td><td>成功</td><td>2016-01-14 17:28:20</td><td><input type="button" value="詳細"/></td></tr> <tr> <td>AIoT-003</td><td>成功</td><td>2016-01-14 13:57:33</td><td><input type="button" value="詳細"/></td></tr> <tr> <td>AIoT-003</td><td>失敗</td><td>2016-01-14 13:27:05</td><td><input type="button" value="詳細"/></td></tr> <tr> <td>AIoT-002</td><td>成功</td><td>2016-01-14 13:27:05</td><td><input type="button" value="詳細"/></td></tr> </tbody> </table>	デバイス	状態	実行日時		AIoT-003	成功	2016-01-14 18:43:55	<input type="button" value="詳細"/>	AIoT-003	成功	2016-01-14 17:28:20	<input type="button" value="詳細"/>	AIoT-003	成功	2016-01-14 13:57:33	<input type="button" value="詳細"/>	AIoT-003	失敗	2016-01-14 13:27:05	<input type="button" value="詳細"/>	AIoT-002	成功	2016-01-14 13:27:05	<input type="button" value="詳細"/>	
デバイス	状態	実行日時																									
AIoT-003	成功	2016-01-14 18:43:55	<input type="button" value="詳細"/>																								
AIoT-003	成功	2016-01-14 17:28:20	<input type="button" value="詳細"/>																								
AIoT-003	成功	2016-01-14 13:57:33	<input type="button" value="詳細"/>																								
AIoT-003	失敗	2016-01-14 13:27:05	<input type="button" value="詳細"/>																								
AIoT-002	成功	2016-01-14 13:27:05	<input type="button" value="詳細"/>																								

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)

フォームに下記のイメージファイルの URL を入力してください。

表 3.3 ファームウェアアップデートに利用するイメージファイルの一覧

リージョン	ファイル名
カーネル	<a href="http://download.atmark-techno.com/node-eye/image/linux-aiotg-std-v2.09-ne2.bin.gz">http://download.atmark-techno.com/node-eye/image/linux-aiotg-std-v2.09-ne2.bin.gz</a>
ユーザーランド	<a href="http://download.atmark-techno.com/node-eye/image/romfs-aiotg-std-v2.06-ne2.img.gz">http://download.atmark-techno.com/node-eye/image/romfs-aiotg-std-v2.06-ne2.img.gz</a>

「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>



ファームウェアのアップデートは 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。



「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。

<sup>[1]</sup> 「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。

node-eye ダッシュボード デバイス ログ アカウント					atmark
グループ <input type="text" value="すべて"/> 接続状態 <input type="text" value="すべて"/> 設定状態 <input type="text" value="すべて"/> デバイス名 / SAコード <input type="text"/>	コマンド	タスク イベント	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
	ステータス参照		AloT-003	成功	2016-01-14 13:57:52
	任意コマンド実行		AloT-003	成功	2016-01-14 13:57:45
	動作コンフィグ反映		AloT-003	成功	2016-01-14 13:57:41
	ステータス参照		AloT-003	成功	2016-01-14 13:36:50
	ステータス参照		AloT-003	成功	2016-01-14 13:35:47
	ステータス参照		AloT-003	成功	2016-01-14 13:34:44
	ステータス参照		AloT-003	成功	2016-01-14 13:33:41
	ステータス参照		AloT-003	成功	2016-01-14 13:32:39

図 3.9 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。

node-eye ダッシュボード デバイス ログ アカウント					atmark
グループ <input type="text" value="すべて"/> 接続状態 <input type="text" value="すべて"/> 設定状態 <input type="text" value="すべて"/> デバイス名 / SAコード <input type="text"/>	種別	タスク イベント	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

図 3.10 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 アカウントの作成

### 1. [新規アカウント]をクリック

node-eye ダッシュボード デバイス ログ 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

### 2. 項目を入力し[作成]をクリック

node-eye ダッシュボード デバイス ログ アカウント atmark

← 戻る

ユーザ名  
atmark-operator

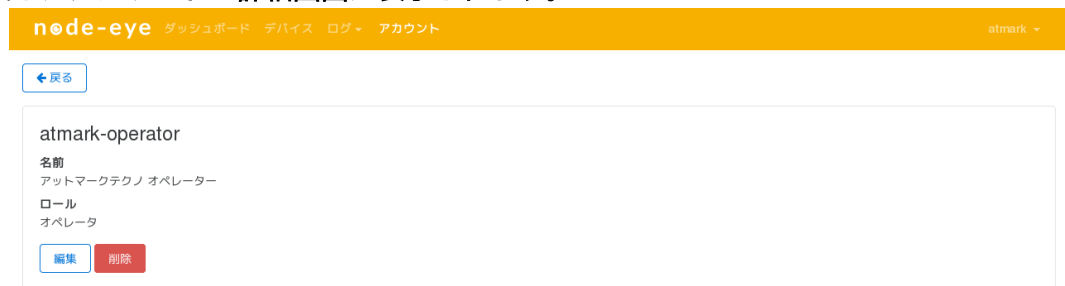
パスワード  
\*\*\*\*\*

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

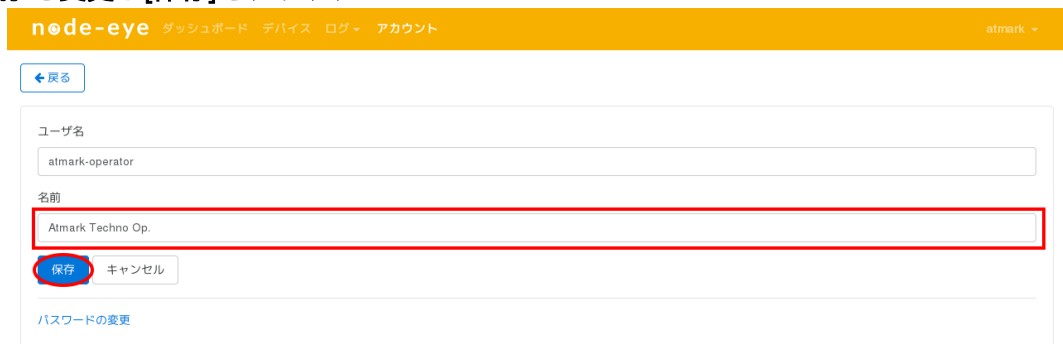
node-eye ダッシュボード デバイス ログ アカウント atmark ▾		
<a href="#">+ 新規アカウント</a>		
ユーザ名	名前	ロール
atmark	atmark	管理者
atmark-operator	アットマークテクノ オペレーター	オペレータ

### 2. [編集]をクリック

node-eye ダッシュボード デバイス ログ アカウント atmark ▾	
<a href="#">← 戻る</a>	
<div>atmark-operator</div> <div>名前 アットマークテクノ オペレーター</div> <div>ロール オペレータ</div> <div><a href="#">編集</a> <a href="#">削除</a></div>	



### 3. "名前"を変更し[保存]をクリック



The screenshot shows the 'node-eye' dashboard with the 'アカウント' (Account) tab selected. The '名前' (Name) field is highlighted with a red box, and the '保存' (Save) button is highlighted with a red circle. The 'ユーザー名' (Username) field contains 'atmark-operator' and the 'パスワードの変更' (Change Password) link is visible below the form.

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

ユーザー名  
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



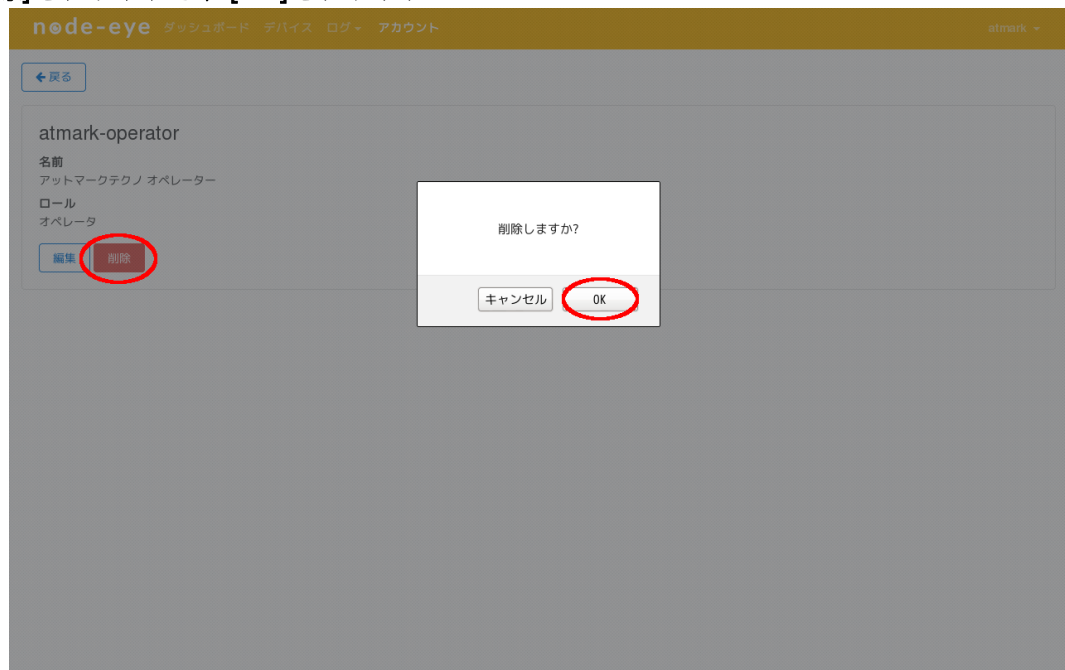
The screenshot shows the 'node-eye' dashboard with the 'アカウント' (Account) tab selected. A table lists the users, and the 'atmark-operator' user is highlighted with a red circle. The table has columns for 'ユーザー名' (Username), '名前' (Name), and 'ロール' (Role).

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

ユーザー名	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

## 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

The screenshot shows the 'node-eye' dashboard with a table of users. The table has three columns: 'ユーザ名' (Username), '名前' (Name), and 'ロール' (Role). The user 'atmark' is listed in the table.

ユーザ名	名前	ロール
atmark	atmark	管理者

## 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「6. Howto」の「6.3. SACM からネットワークインターフェースの設定を行う」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

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

SA

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

1 - 8 / 8 20 < 前へ 次へ >

SA		
AIoT-006	切断中	tss18740226 不明なデバイス
AIoT-004	切断中	tss18740223 Armadillo-IoT G2
AIoT-007	切断中	tss18740264 Armadillo-IoT G2
AIoT-005	切断中	tss18740271 不明なデバイス
AIoT-001	接続中	tss21373220 Armadillo-IoT G2
AIoT-003	切断中	tss21373251 Armadillo-IoT G2
AIoT-002	接続中	tss21373268 Armadillo-IoT G2
AIoT-000	切断中	tss21545443 不明なデバイス

1 - 8 / 8 20 < 前へ 次へ >

## 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

1 - 8 / 8 20

SA IoT-001

AIoT-001  
SAコード: tss21373220 白デバイス: 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

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

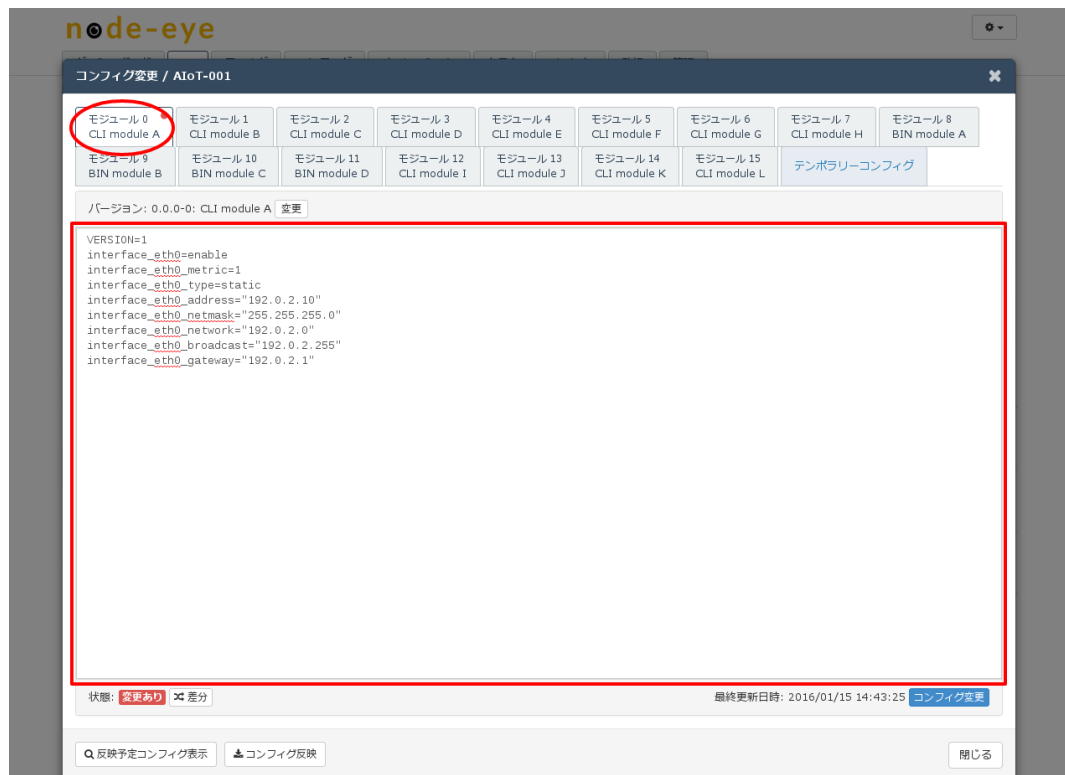
☐ 反映予定コンフィグ表示

コンフィグ反映

## 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=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"
```



モジュール 0 の設定項目は Armadillo の **/etc/network/interfaces** 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。

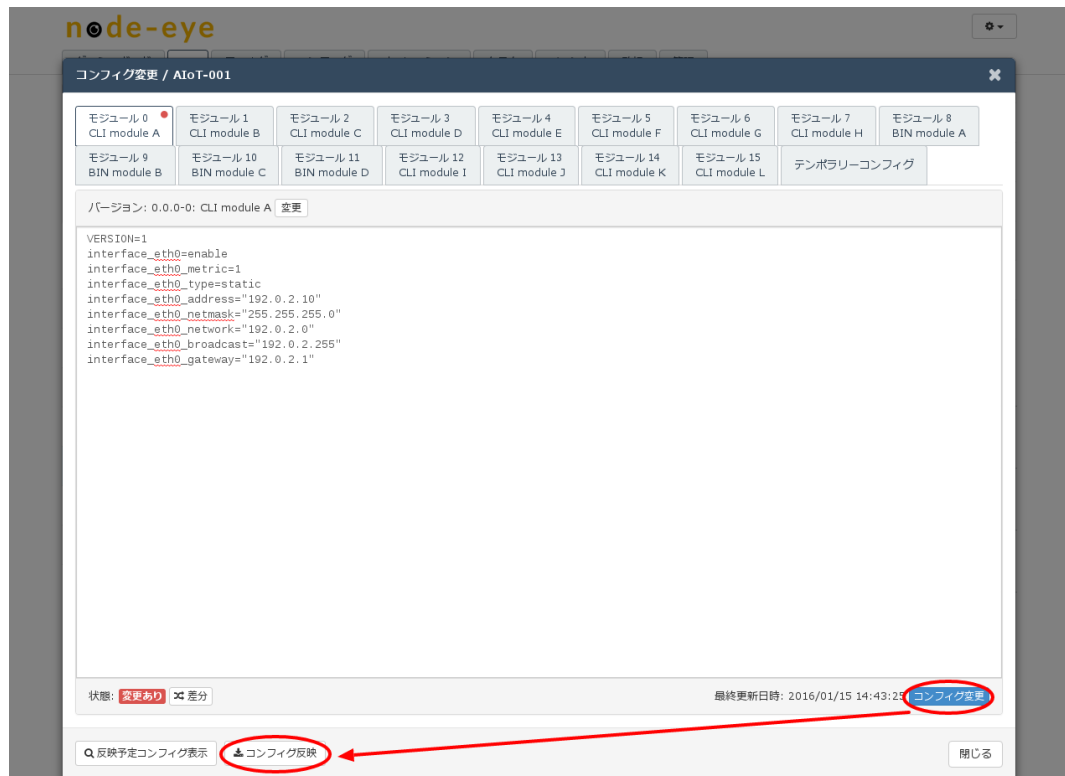


図 3.11 コンフィグ反映 (SACM コントロールパネル)

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 3.12 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 3.13 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

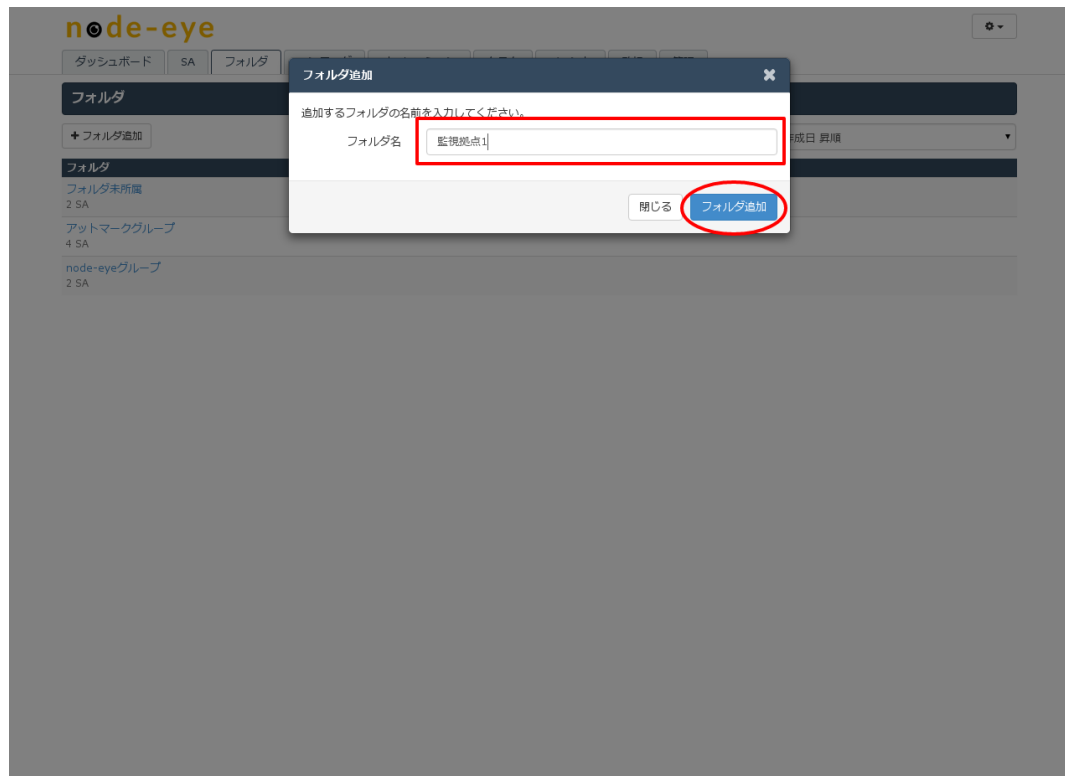


図 3.14 フォルダ名入力 (SACM コントロールパネル)

### 3. フォルダー一覧から、先ほど入力したフォルダ名をクリック



図 3.15 フォルダー一覧 (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 3.16 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

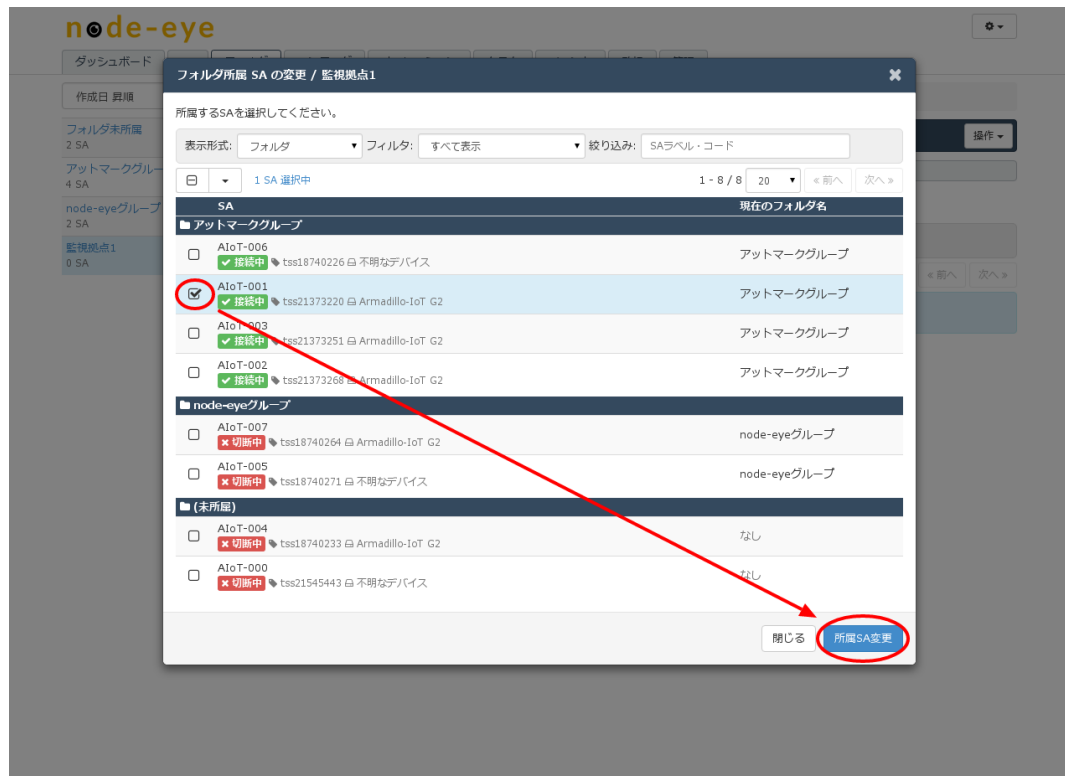


図 3.17 フォルダに追加する SA を選択 (SACM コントロールパネル)

## 6. チェックを入れた SA が表示されます。



図 3.18 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事ができます。

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 3.19 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

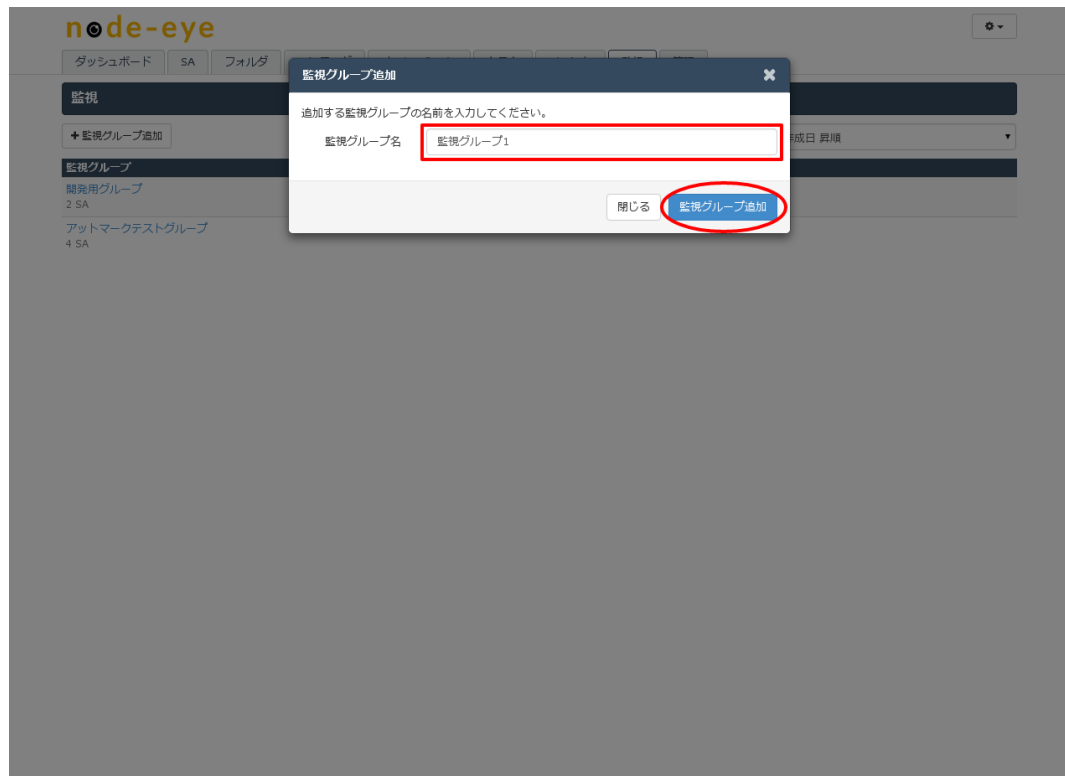


図 3.20 監視グループ名入力 (SACM コントロールパネル)

3. 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 3.21 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



図 3.22 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

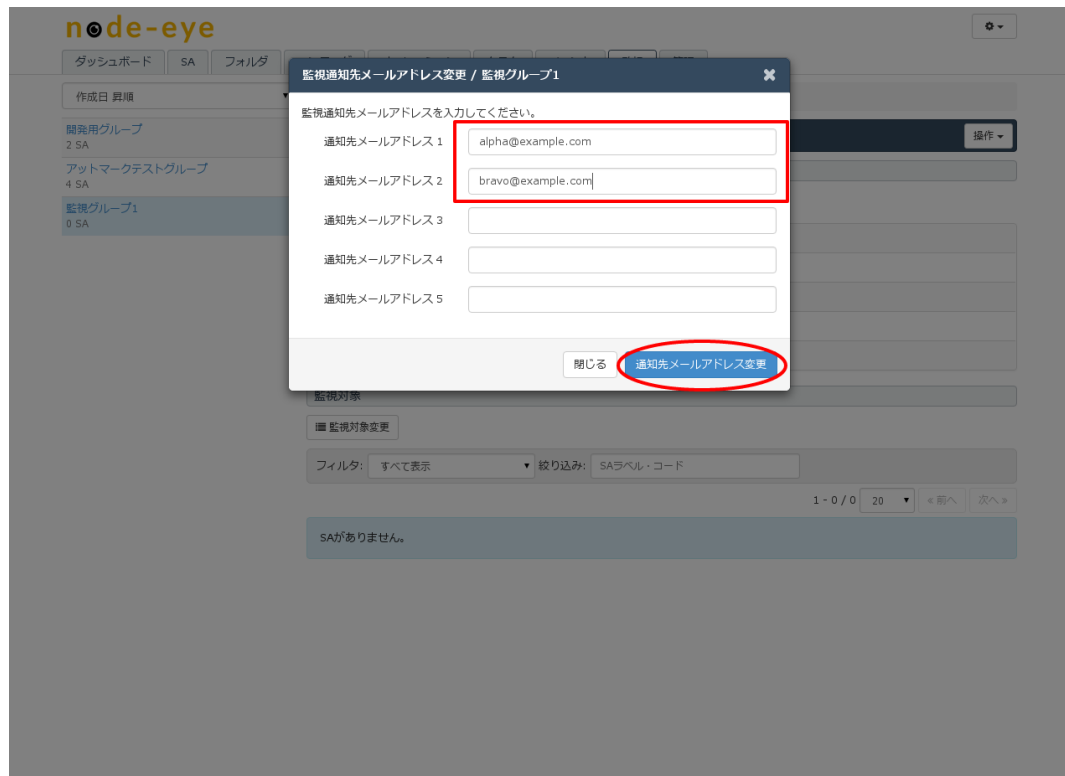


図 3.23 通知先メールアドレス入力 (SACM コントロールパネル)

## 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。

The screenshot displays the '監視' (Monitoring) section of the node-eye SACM Control Panel. A green notification bar at the top indicates that the notification email address has been successfully updated. The left sidebar shows a navigation menu with options like '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 main content area is titled '監視 / 監視グループ1'. It features a '監視グループ1' header with a '操作' (Action) button. Below this, there is a section for '通知先メールアドレス' (Notification Email Address) 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 a '監視対象' (Monitoring Target) section with a '監視対象変更' (Change Monitoring Target) button. A filter section shows 'フィルタ: すべて表示' (Filter: Show All) and '絞り込み: SAラベル・コード' (Narrowing: SA Label・Code). 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	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

図 3.24 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



図 3.25 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

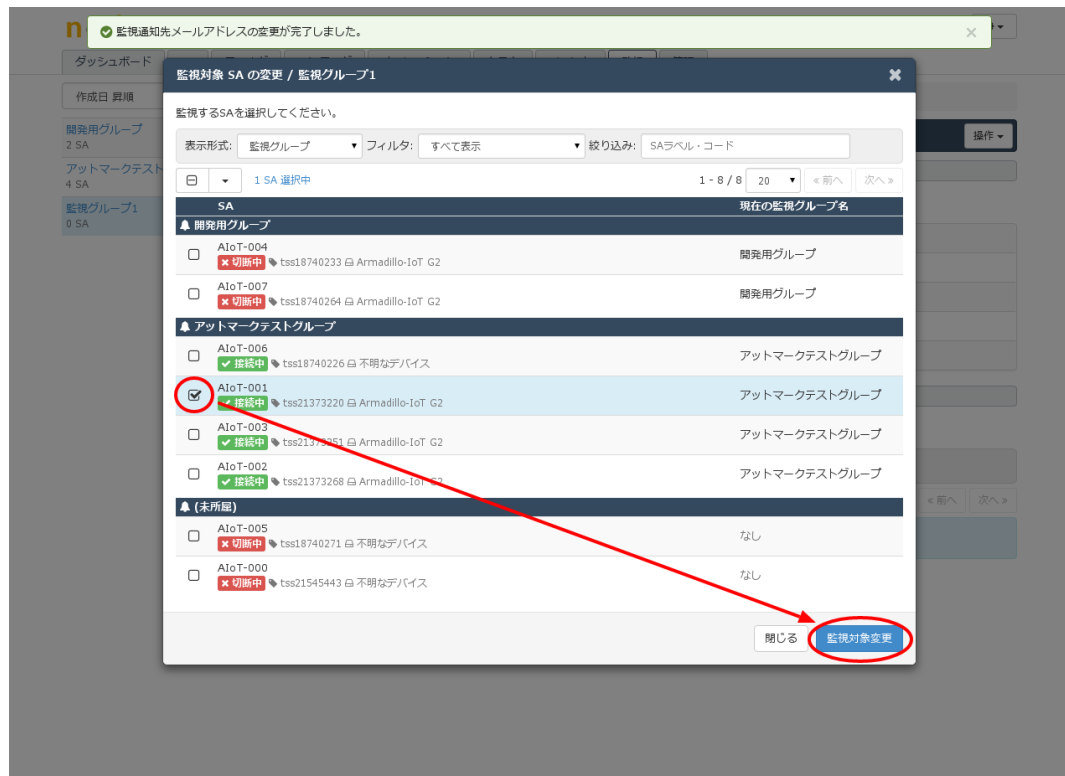


図 3.26 監視グループに追加する SA を選択 (SACM コントロールパネル)

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 3.27 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. NOR フラッシュメモリのパーティション構成

node-eye に対応した NOR フラッシュメモリのパーティション構成は、Armadillo-IoT ゲートウェイ G2 の出荷時の構成と異なります。また、Armadillo-IoT ゲートウェイ G2 の型番によっても、パーティション構成が異なります。

Armadillo-IoT ゲートウェイ G2 の node-eye 非対応構成(型番が AG42\*-ではじまる場合)、node-eye 対応構成(型番が AG42\*-ではじまる場合)、Armadillo-IoT ゲートウェイ G2 の node-eye 非対応構成(型番が AG43\*-ではじまる場合)、node-eye 対応構成(型番が AG43\*-ではじまる場合)、それぞれを以下に示します。

**表 4.1 Armadillo-IoT ゲートウェイ スタンダードモデル G2 出荷時 (node-eye 非対応、Armadillo の型番が AG42\*-ではじまる場合)**

物理アドレス	パーティション名	サイズ	ソフトウェア
0xA0000000   0xA001FFFF	bootloader	128kByte	ブートローダーイメージ
0xA0020000   0xA041FFFF	kernel	4MByte	Linux カーネルイメージ
0xA0420000   0xA1EFFFFFFF	userland	26.875Mbyte	ユーザーランドイメージ
0xA1F00000   0xA1FFFFFFF	config	1MByte	アプリケーションの設定情報など

**表 4.2 node-eye 対応(Armadillo の型番が AG42\*-ではじまる場合)**

物理アドレス	パーティション名	サイズ	ソフトウェア
0xA0000000   0xA001FFFF	bootloader	128kByte	node-eye 対応ブートローダーイメージ
0xA0020000   0xA081FFFF	recovery	8MByte	リカバリイメージ
0xA0820000   0xA0C1FFFF	kernel	4MByte	Linux カーネルイメージ(ユーザー作成)
0xA0C20000   0xA1EDFFFF	userland	18.75Mbyte	ユーザーランドイメージ(ユーザー作成)
0xA1EE0000   0xA1EFFFFFFF	license	128kByte	ハードウェア固有情報
0xA1F00000   0xA1FFFFFFF	config	1MByte	アプリケーションの設定情報など

表 4.3 Armadillo-IoT ゲートウェイ スタンダードモデル G2 出荷時 (node-eye 非対応、Armadillo の型番が AG43\*-ではじまる場合)

物理アドレス	パーティション名	サイズ	ソフトウェア
0xA0000000   0xA003FFFF	bootloader	256kByte	ブートローダーイメージ
0xA0040000   0xA043FFFF	kernel	4MByte	Linux カーネルイメージ
0xA0440000   0xA1EFFFFF	userland	26.75Mbyte	ユーザーランドイメージ
0xA1F00000   0xA1FFFFFF	config	1MByte	アプリケーションの設定情報など

表 4.4 node-eye 対応(Armadillo の型番が AG43\*-ではじまる場合)

物理アドレス	パーティション名	サイズ	ソフトウェア
0xA0000000   0xA003FFFF	bootloader	256kByte	node-eye 対応ブートローダーイメージ
0xA0040000   0xA083FFFF	recovery	8MByte	リカバリイメージ
0xA0840000   0xA0C3FFFF	kernel	4MByte	Linux カーネルイメージ(ユーザー作成)
0xA0C40000   0xA1EDFFFF	userland	18.625Mbyte	ユーザーランドイメージ(ユーザー作成)
0xA1EE0000   0xA1EFFFFF	license	128kByte	ハードウェア固有情報
0xA1F00000   0xA1FFFFFF	config	1MByte	アプリケーションの設定情報など

node-eye 対応のパーティション構成には、新たに recovery と license を追加しています。

recovery パーティションについて詳しくは本文「4.3. リカバリイメージ」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. デフォルトでインストールされるアプリケーションの違い

node-eye 対応 パーティション構成では、userland パーティションのサイズは 18.75MB です。

容量制限のため node-eye 対応のユーザーランドには、Oracle Java SE Embedded 8、LUA インタプリタ はデフォルトで組み込まれないようになっています。

また、node-eye のデバイス運用管理サービスにいくつか必要なアプリケーションが追加されています。



## 4.3. リカバリイメージ

recovery 領域には、Linux カーネルとユーザーランド(atmark-dist)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イメージといいます。

### 4.3.1. リカバリイメージの目的

リカバリイメージの目的は、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が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

### 4.3.2. リカバリイメージの起動条件

リカバリイメージを使って 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

### 4.3.3. リカバリイメージの特徴

#### 4.3.3.1. 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。

ユーザーランドにインストールされるアプリケーションの差分は以下のコマンドで確認することができます。

```
[PC] $ cd atmark-dist/vendors/AtmarkTechno
[PC] $ diff Armadillo-IoTG-Std-NE/config.vendor Armadillo-IoTG-Std-NE.Recover/
config.vendor
```



#### 4.3.3.2. リカバリイメージ起動中の WebUI 上の表示

リカバリイメージで 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。

## 4.4. 監視方法

Armadillo の接続/切断状態の監視は、Armadillo にインストールされた armsd が SACM に Heartbeat パケットを送信することによって実現しています。

- ・送信間隔

- ・ 5 分に 1 回 Heartbeat を Armadillo から送信
- ・ 状態変化
  - ・ [切断中]のとき、1 回 Heartbeat 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 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 を送信しません。従って、接続状態は [切断中]となりますが、Armadillo は SACM に接続可能なため復旧作業 (ファームウェアアップデート等の)は実行できます。

## 4.5. リモートコンフィグ

node-eye のリモートコンフィグ機能は、SACM のモジュールを使用しています。そのため、この章では "node-eye におけるモジュールについて" 説明します。

### 4.5.1. node-eye における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 種類あります。Armadillo-IoT ゲートウェイ スタンダードモデル G2 では、16 個のモジュールが用意されています。

表 4.5 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 4.6 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

モジュールに関する詳細な情報は、別途 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。

## 4.6. ネットワーク設定

node-eye 対応イメージでは armsd がネットワーク設定を管理します。

Armadillo 起動直後は、armsd が `/etc/network/interfaces` を生成してネットワーク接続を行います。SACM からコンフィグを取得できたら、一度、すべてのネットワークインターフェースをダウンした後、取得したコンフィグを元に `/etc/network/interfaces` を生成し、再度ネットワークインターフェースをアップします。コンフィグの管理はネットワークモジュールが行っています。

armsd のネットワーク設定の詳細は、本文「3.3.2. リモートコンフィグを行う」を参照してください。

### 4.6.1. ネットワークインターフェース設定シーケンス

Armadillo が起動してから、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説明します。下記のシーケンスは SACM 動作シーケンス Pull 動作 [[http://manual.sacm.jp/service\\_spec/sequence.html](http://manual.sacm.jp/service_spec/sequence.html)] に沿った動作です。

1. Armadillo が起動し、armsd が起動すると `/etc/armsd/scripts/line` が実行されます。 `/etc/armsd/scripts/line` は `/etc/config/line.conf` を元に `/etc/network/interfaces`(以下、`interfaces` と表記します)を生成します。生成される `interfaces` は USB メモリーを使うことによって変更することができます。
2. 次に、生成された `interfaces` 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/config/armsd.cache` として保存されます) 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 `interfaces` 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 `interfaces` は、Armadillo 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 `interfaces` 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。
  - ・ Service-Config により設定した `interfaces` 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。

9. 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。

## 4.7. ファームウェアアップデート

本章では node-eye コントロールパネル の ファームウェアアップデート機能 の仕様について説明します。ファームウェアアップデート機能の使い方は「3. node-eye を体験する」を参照してください。

ファームウェアアップデートにより書き込むことのできる領域は以下の 3 つです。

- ・ kernel 領域
- ・ userland 領域
- ・ recovery 領域

※ただし、同時に書き込めるのは kernel と userland の組み合わせのみ

また、ファームウェアアップデートの際には、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置き、その URL を指定する必要があります。

## 5. イメージファイルのビルド手順

本章では、ATDE5 を使って node-eye 対応のイメージファイルをビルドする方法を説明します。

以下の手順は「リカバリイメージを作成する場合」を例に書かれていますが、kernel/userland 領域に書き込む、標準 Linux カーネルイメージ/標準ユーザーランドイメージを作成する場合も [ステップ 5] を除いて同様の手順となります。



ATDE5 での操作は全て root ユーザーではなく、**一般ユーザー**で行ってください。

### 手順 5.1 イメージファイルのビルド

#### 1. ソースコードの準備

下記 URL からソースコードをダウンロードします。

表 5.1 ソースコードの一覧

ファイル名	説明
<a href="http://download.atmark-techno.com/node-eye/source/linux-3.14-at10-ne2.tar.gz">http://download.atmark-techno.com/node-eye/source/linux-3.14-at10-ne2.tar.gz</a>	Linux カーネル
<a href="http://download.atmark-techno.com/node-eye/source/atmark-dist-20151218-ne2.tar.gz">http://download.atmark-techno.com/node-eye/source/atmark-dist-20151218-ne2.tar.gz</a>	Atmark-dist(ユーザーランド)
<a href="http://armadillo.atmark-techno.com/files/downloads/armadillo-wlan/source/driver/AWL13/awl13-3.0.2-3.tar.gz">http://armadillo.atmark-techno.com/files/downloads/armadillo-wlan/source/driver/AWL13/awl13-3.0.2-3.tar.gz</a>	AWL13 デバイスドライバ



型番が AG43\*-ではじまる Armadillo をご利用の場合、以下バージョンのソースコードを使用する必要があります。

- ・ linux-3.14-at10-ne2.tar.gz 以降
- ・ atmark-dist-20151218-ne2.tar.gz 以降

#### 2. アーカイブの展開

各ソースコードアーカイブを展開します。

```
[ATDE ~]$ ls
atmark-dist-[version]-ne[version].tar.gz  awl13-[version].tar.gz  linux-3.14-
at[version]-ne[version].tar.gz
[ATDE ~]$ tar xzf atmark-dist-[version].tar.gz
[ATDE ~]$ ar xzf awl13-[version].tar.gz
[ATDE ~]$ tar xzf linux-3.14-at[version].tar.gz
[ATDE ~]$ ls
atmark-dist-[version]-ne[version]          awl13-[version]          linux-3.14-
at[version]-ne[version]
```



```
atmark-dist-[version]-ne[version].tar.gz    awl13-[version].tar.gz linux-3.14-
at[version]-ne[version].tar.gz
```



### 3. シンボリックリンクの作成

Atmark Dist に、AWL13、Linux カーネルのシンボリックリンクを作成します。

```
[ATDE ~]$ cd atmark-dist-[version]-ne[version]
[ATDE ~/atmark-dist-[version]-ne[version]]$ ln -s ../awl13-[version] awl13
[ATDE ~/atmark-dist-[version]-ne[version]]$ ln -s ../linux-3.14-at[version]-ne[version]
linux-3.x
```



以降のコマンド入力例では、各ファイルからバージョンを省略した表記を用います。

### 4. コンフィギュレーションの開始

Atmark Dist ディレクトリに入り、コンフィギュレーションを行います。ここでは、menuconfig を利用します。

```
[ATDE ~]$ cd atmark-dist
[ATDE ~/atmark-dist]$ make menuconfig
atmark-dist v1.45.0 Configuration

-----
----- Main Menu -----
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module < > module capable

-----
Vendor/Product Selection --->
Kernel/Library/Defaults Selection --->
---
Load an Alternate Configuration File
Save Configuration to an Alternate File

-----
<Select>    < Exit >    < Help >
-----
```

### 5. ベンダー/プロダクト名の選択

メニュー項目は、上下キーで移動することができます。下部の Select/Exit/Help は左右キーで移動することができます。選択するには Enter キーを押下します。 "Vendor/Product Selection --->" に移動して Enter キーを押下します。 Vendor には "AtmarkTechno" を選択し、AtmarkTechno Products には "Armadillo-IoTG-Std-NE.Recover" を選択します。

```
atmark-dist v1.45.0 Configuration
-----
```

```

----- Vendor/Product Selection -----
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module < > module capable

--- Select the Vendor you wish to target
      (AtmarkTechno) Vendor
--- Select the Product you wish to target
      (Armadillo-IoTG-Std-NE.Recover) AtmarkTechno Products

-----
-----
      <Select>    < Exit >    < Help >
-----

```

## 6. デフォルトコンフィギュレーションの適用

前のメニューに戻るには、"Exit"に移動してEnterキーを押下します。続いて、"Kernel/Library/Defaults Selection --->"に移動してEnterキーを押下します。"Default all settings (lose changes)"に移動して"Y"キーを押下します。押下すると"[\*]"のように選択状態となります。

```

atmark-dist v1.45.0 Configuration
-----
----- Kernel/Library/Defaults Selection -----
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module < > module capable

--- Kernel is linux-3.x
      (default) Cross-dev
      (None) Libc Version
      [*] Default all settings (lose changes)
      [ ] Customize Kernel Settings
      [ ] Customize Vendor/User Settings
      [ ] Update Default Vendor Settings

-----
-----
      <Select>    < Exit >    < Help >
-----

```

## 7. コンフィギュレーションの終了

前のメニューに戻るため、"Exit"に移動してEnterキーを押下します。コンフィギュレーションを抜けるためにもう一度"Exit"に移動してEnterキーを押下します。

## 8. コンフィギュレーションの確定

コンフィギュレーションを確定させるために"Yes"に移動して Enter キーを押下します。

```
atmark-dist v1.45.0 Configuration
```

```
-----  
Do you wish to save your new kernel configuration?  
-----
```

```
< Yes >    < No  >  
-----
```

## 9. ビルド

コンフィギュレーションが完了するので、続いてビルドを行います。ビルドは"make"コマンドを実行します。

```
[ATDE ~/atmark-dist]$ make
```

## 10. イメージファルの生成確認

ビルドが終了すると、atmark-dist/images/ディレクトリ以下にイメージファイルが作成されています。Armadillo-IoT では圧縮済みのイメージ(拡張子が".gz"のもの)を利用します。

```
[ATDE ~/atmark-dist]$ ls images/  
linux.bin  linux.bin.gz  romfs.img  romfs.img.gz
```



## 6. Howto

---

### 6.1. リカバリイメージを使って Armadillo を復旧する

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

#### 手順 6.1 リカバリイメージを使った Armadillo の復旧

##### 1. リカバリ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. リカバリイメージからプライマリ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] に切りわか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。

### 6.2. リカバリ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 6.2 リカバリ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 ATDE5 等の Linux マシンで行ってください。

```
[PC~]$ echo | gzip > dummy.gz && md5sum dummy.gz > dummy.gz.md5
[PC~]$ ls dummy*
dummy.gz  dummy.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 "dummy.gz" と "dummy.gz.md5" を対象となる 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.gz" への URL を [カーネル] 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。

[ユーザーランド] 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。

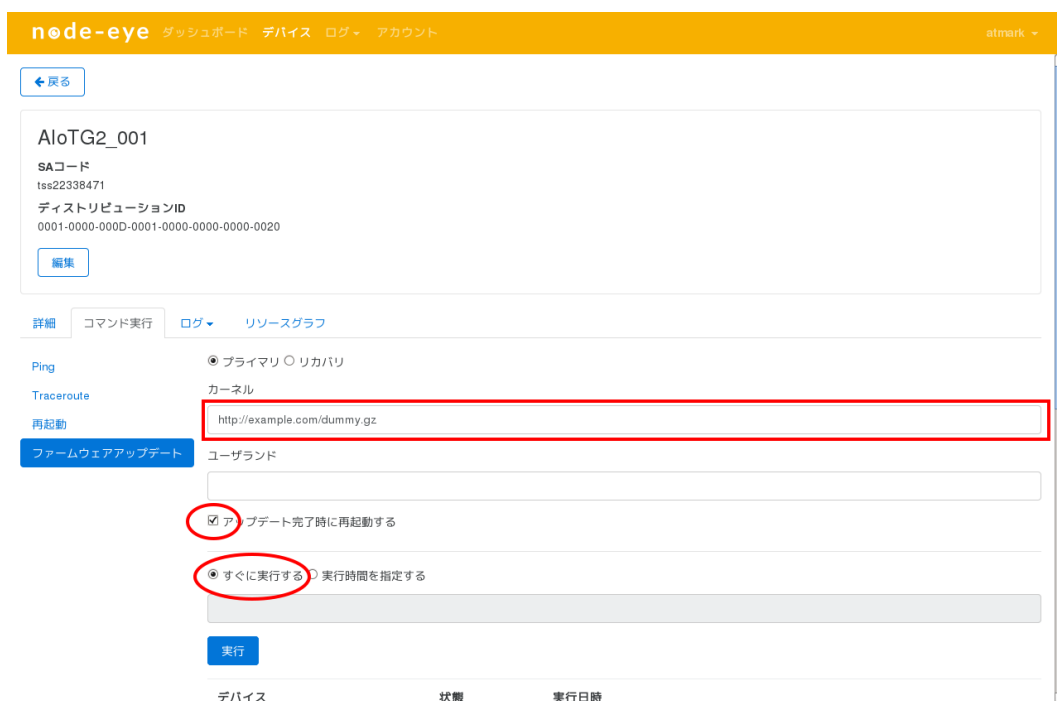


図 6.1 ファームウェアアップデート (node-eye コントロールパネル)

## 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、自動的にリカバリイメージで再起動します。

リカバリ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。

## 6.3. SACM からネットワークインターフェースの設定を行う

本章では、アットマークテクノが提供するネットワークモジュールのコンフィグの仕様と、各種ネットワークインターフェースを使うための設定例を紹介します。

ネットワークモジュールは モジュール 0 に割り当てています。



SACM の操作手順については、「3. node-eye を体験する」の「3.3.2. リモートコンフィグを行う」を参照してください。

### 6.3.1. ネットワークモジュール仕様

#### 6.3.1.1. 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 G2 の標準イメージ(node-eye 対応)では "1" 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

### 6.3.1.2. 使用可能な key 一覧

#### 6.3.1.2.1. eth0

ネットワークモジュールの eth0 に関する key 一覧です。 interface\_eth0\_ から始まる変数の値は /etc/network/interfaces に書き込まれます。

表 6.1 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_network	192.168.0.0	ネットワーク
interface_eth0_broadcast	192.168.255.255	ブロードキャストアドレス
interface_eth0_gateway	192.168.0.1	ゲートウェイ
interface_eth0_metric	1	メトリック値

#### 6.3.1.2.2. umts0

ネットワークモジュールの umts0 に関する key 一覧です。 interface\_umts0\_ から始まる変数の値は /etc/network/interfaces に書き込まれます。

表 6.2 ネットワークモジュールの umts0 に関する key 一覧

key	value (参考例)	説明
interface_umts0	enable	eth0 の有効/無効 (enable / disable)
interface_umts0_type	dhcp	IP アドレスの設定方法 (dhcp のみサポート)
interface_umts0_apn	APN	アクセスポイント名 (APN)
interface_umts0_id	"user name"	認証 ID(ユーザー名)
interface_umts0_auth_type	CHAP	認証方式
interface_umts0_pass	"password"	パスワード
interface_umts0_pdp_type	IP	PDP タイプ
interface_umts0_metric	20	メトリック値

#### 6.3.1.2.3. awlan0

ネットワークモジュールの awlan0 に関する key 一覧です。 interface\_awlan0\_ から始まる変数の値は /etc/network/interfaces に書き込まれます。

表 6.3 ネットワークモジュールの awlan0 に関する key 一覧

key	value (参考例)	説明
interface_awlan0	enable	enable または disable
interface_awlan0_type	dhcp	dhcp または static
interface_awlan0_address	192.168.10.2	IP アドレス
interface_awlan0_netmask	255.255.0.0	ネットマスク
interface_awlan0_network	192.168.0.0	ネットワーク
interface_awlan0_broadcast	192.168.255.255	ブロードキャストアドレス
interface_awlan0_gateway	192.168.0.1	ゲートウェイ
interface_awlan0_metric	40	メトリック値
interface_awlan0_passphrase	"password"	パスフレーズ
interface_awlan0_cryptmode	"WPA2-AES"	暗号化方式*1 [http://manual.atmark-techno.com/armadillo-wlan/armadillo-wlan_software_manual_ja-3.3.0/ch07.html#table-setting-cryptmode]

key	value (参考例)	説明
interface_awlan0_essid	"mywpa2aes"	ESSID
interface_awlan0_channel	1	チャンネル(interface_awlan0_wireless_mode が master の時のみ設定可能)
interface_awlan0_wireless_mode	managed	接続モード (ステーションモード(STA): managed / アクセスポイントモード(AP): master)

### 6.3.1.2.4. 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 6.4 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

## 6.3.2. 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=1, interface\_\*\*\*=enable が書かれていることに注目してください。

### 6.3.2.1. 3G モバイル回線を使う

```
VERSION=1

interface_umts0=enable
interface_umts0_type=dhcp
interface_umts0_id="myid"
interface_umts0_pass="mypass"
interface_umts0_apn="myapn"
interface_umts0_auth_type="PAP"
interface_umts0_pdp_type="IP"
interface_umts0_metric=10
```

### 6.3.2.2. AWL13 を使う

```
VERSION=1

interface_awlan0=enable
interface_awlan0_type=dhcp
interface_awlan0_essid="my-wireless-network"
interface_awlan0_passphrase="password"
interface_awlan0_wireless_mode="managed"
interface_awlan0_cryptmode="WPA2-AES"
interface_awlan0_metric=20
```

### 6.3.2.3. 固定 IP アドレスを使う

eth0 を使用する場合

```
VERSION=1
```

```
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_network="192.168.10.0"
interface_eth0_broadcast="192.168.10.255"
interface_eth0_gateway="192.168.10.1"

resolv_conf_search="local-network"
resolv_conf_nameserver="192.168.10.100"
```

awlan0 を使用する場合

```
VERSION=1

interface_awlan0=enable
interface_awlan0_type=dhcp
interface_awlan0_address="192.168.10.10"
interface_awlan0_netmask="255.255.255.0"
interface_awlan0_network="192.168.10.0"
interface_awlan0_broadcast="192.168.10.255"
interface_awlan0_gateway="192.168.10.1"
interface_awlan0_essid="my-wireless-network"
interface_awlan0_passphrase="password"
interface_awlan0_wireless_mode="managed"
interface_awlan0_cryptmode="WPA2-AES"
interface_awlan0_metric=20

resolv_conf_search="local-network"
resolv_conf_nameserver="192.168.10.100"
```

#### 6.3.2.4. 複数のインターフェースを同時に使う

```
VERSION=1

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_network="192.168.10.0"
interface_eth0_broadcast="192.168.10.255"
interface_eth0_gateway="192.168.10.1"

interface_umts0=enable
interface_umts0_type=dhcp
interface_umts0_id="myid"
interface_umts0_pass="mypass"
interface_umts0_apn="myapn"
interface_umts0_auth_type="PAP"
interface_umts0_pdp_type="IP"
interface_umts0_metric=10

interface_awlan0=enable
```

```
interface_awlan0_type=dhcp
interface_awlan0_essid="my-wireless-network"
interface_awlan0_passphrase="password"
interface_awlan0_wireless_mode="managed"
interface_awlan0_cryptmode="WPA2-AES"
interface_awlan0_metric=20

resolv_conf_search="local-network"
resolv_conf_nameserver="192.168.10.100"
```

### 6.3.3. SACM へ初めて接続する時のネットワーク設定を変更する

Armadillo のリモートコンフィグと遠隔監視を行うためには、Armadillo と SACM サーバー が互いに通信できる環境を用意する必要があります。デフォルトの設定では、Armadillo の Ethernet ポート (eth0) を使って、DHCP による設定を使って、通信を試みます。それ以外の環境、例えば、3G モバイル回線だけで Armadillo の遠隔監視を行う場合は、この章で説明する手順で設定をする必要があります。

本題に入る前に、SACM から Armadillo に対してリモートコンフィグを行うまでの流れを本文の「4.6.1. ネットワークインターフェース設定シーケンス」で確認してください。

シーケンス内で示すリモートコンフィグ前の `/etc/network/interfaces` の変更は `/etc/armsd/scripts/line` の設定ファイル(`/etc/config/line.conf`)を変更することで出来ることがわかります。

`/etc/config/line.conf` の変更は、以下のいずれかの手順で行うことができます。

- ・「手順 6.3. 設定ファイルを変更し、コンフィグ領域に保存する」

これは Armadillo と PC を接続し Armadillo の端末を操作して設定する方法です。これは試作、開発時に向いています。"イメージの書き換えが不要"というメリットはありますが、PC と Armadillo を何らかの方法で接続する必要があるため、複数台を同時に設定する場合などに向きません。また、`flatfsd -w` でコンフィグ領域を初期化した場合に再設定が必要です。

- ・「手順 6.4. 設定ファイルを変更し、イメージファイルを書き込む」

これは、`atmark-dist` のソースコード内にある、`/etc/armsd/scripts/line` の設定ファイルを編集し、ビルドしたイメージ書き込むことで、設定ファイルを変更する方法です。「手順 6.3. 設定ファイルを変更し、コンフィグ領域に保存する」の方法と比較して、作業は多くなってしまいますが、`flatfsd -w` を行っても、`atmark-dist` ビルド時の `/etc/armsd/scripts/line` の設定が保持されるという特徴があります。

- ・「手順 6.5. 設定ファイルを USB メモリーに入れ、設定を取り込む」

これは、`node-eye` 標準イメージの機能を使った方法です。Armadillo に USB メモリーを接続した状態で起動することによって、USB メモリー内の設定ファイルを Armadillo に取り込むことが可能です。PC を使わずに設定することができるため、多数の Armadillo を連続してセットアップする場合に向いています。

ただし、この方法も `flatfsd -w` でコンフィグ領域を初期化した場合は再設定が必要です。



`/etc/armsd/scripts/line` が実行されるタイミングや、`interfaces` が生成されてからの処理については、前述の「4.6.1. 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。



ご使用の環境が、以下の条件をすべて満たしている場合、これらの設定を行う必要はありません。

- ・ Armadillo のネットワークインターフェースに 有線 LAN を使用している
- ・ Armadillo が DHCP で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

## 手順 6.3 設定ファイルを変更し、コンフィグ領域に保存する

### 1. line.conf の編集

vi エディタなどで、Armadillo 上の /etc/armsd/scripts/line の設定ファイルを変更します。

```
[Armadillo ~]# vi /etc/config/line.conf
```

以下は、3G モバイル回線と DHCP による有線 LAN 接続を試行する設定例です。先頭に# が付いた行はコメントなので、評価されません。他の設定例については、本文「6.3.3. SACM へ初めて接続する時のネットワーク設定を変更する」に記載の設定例を参照してください。

```
MOBILE_ID="example@example.jp"
MOBILE_PASS="example_mobile_password"
MOBILE_APN="example.jp"
MOBILE_AUTH_TYPE="CHAP"
MOBILE_PDP_TYPE="IP"

#RESOLV_SEARCH="local-network"
#RESOLV_NAMESERVER="192.168.10.1"
#STATIC_ADDRESS="192.0.2.10"
#STATIC_NETMASK="255.255.255.0"
#STATIC_NETWORK="192.0.2.0"
#STATIC_BROADCAST="192.0.2.255"
#STATIC_GATEWAY="192.0.2.1"
```

### 2. line.conf の保存と再起動

まず、キャッシュファイルを削除し、編集した設定ファイルをフラッシュメモリに保存してから、再起動します。

```
[Armadillo ~]# rm -f /etc/config/armsd.cache
[Armadillo ~]# flatfsd -s
[Armadillo ~]# reboot
```

再起動後、先ほど書き込んだ設定ファイルを基に /etc/network/interfaces が生成され、SACM への接続がおこなわれます。



## 手順 6.4 設定ファイルを変更し、イメージファイルを書き込む

### 1. ソースコードの用意

作業用 PC 上に node-eye 対応の atmark-dist のソースコードを用意します。

### 2. line.conf の編集

/etc/armstd/scripts/line の設定ファイルは、プライマリ、リカバリそれぞれにあります。そのため、どちらも同じ設定にしたい場合は両方編集する必要があります。

- ・ プライマリ
  - ・ atmark-dist/vendors/AtmarkTechno/Armadillo-IoTG-Std-NE/etc/default/line.conf
- ・ リカバリ
  - ・ atmark-dist/vendors/AtmarkTechno/Armadillo-IoTG-Std-NE.Recover/etc/default/line.conf

編集内容は「6.3.3.2. 設定例」で紹介しているものを参考にしてください。

### 3. イメージの書き込み

編集後、ユーザーランドをビルドし、Armadillo へ書き込みを行ってください。ユーザーランドのビルド方法は Armadillo-IoT ゲートウェイ スタンダードモデル製品マニュアルの 11.1. Linux カーネル/ユーザーランドをビルドする [http://manual.atmark-techno.com/armadillo-iot/armadillo-iotg-std\_product\_manual\_ja-2.5.2/ch11.html#sct.build-dist] を参照してください。Armadillo へのイメージの書き込み方法は、同マニュアルの 第 12 章 フラッシュメモリの書き換え方法 [http://manual.atmark-techno.com/armadillo-iot/armadillo-iotg-std\_product\_manual\_ja-2.5.2/ch12.html] を参照してください。

### 4. コンフィグ領域の初期化

書き込みが完了し、Armadillo が再起動した後はログイン後に "flatfsd" コマンドを使ってコンフィグ領域を初期化してください。初期化時には、新たに書き込んだ node-eye 対応イメージの "/etc/default" ディレクトリ以下のファイルがコンフィグ領域に保存され、且つ "/etc/config" ディレクトリにファイルが複製されます。

```
[Armadillo ~]# flatfsd -w
```

## 手順 6.5 設定ファイルを USB メモリーに入れ、設定を取り込む

Armadillo に USB メモリーに入れた設定ファイルを取り込ませる方法を紹介します。設定ファイルは「特定のディレクトリ」に「特定の名前」で、USB メモリーに保存しておく必要があります。

### 1. USB メモリーをフォーマットする

USB メモリーは **FAT32** でフォーマットしてください。

### 2. 設定ファイルを作成する

設定ファイルは、USB メモリーに以下のように配置します。

```

USB_ROOT
|-- config
|   |-- attached
|   |   |-- 0001-0000-000D-0001-0000-0000-XXXX-0001.conf
|   |   |-- 0001-0000-000D-0001-0000-0000-XXXX-0002.conf
|   |-- pool
|   |   |-- 1.conf
|   |   |-- 2.conf
|   |   |-- 3.conf
|   |-- fixed
|   |   |-- default.conf

```

3つのディレクトリ(attached, pool, fixed)の用途を下記の表に示します。目的に合ったディレクトリに設定ファイルを置いてください。

```
/mnt/config/[directory_name]
```

表 6.5 ディレクトリの用途

directory_name	説明
attached	Distribution-ID とネットワーク設定を 1 対 1 で対応させたい場合に使うファイル名は [Distribution-ID].conf のフォーマットが有効
pool	設定する台数分のファイルを用意して、Distribution-ID とは無関係に順次設定を適用する場合に使うファイル名の昇順で USB メモリーの接続毎に設定が適用され、適用済みの設定は attached に [Distribution-ID].conf として移動される任意のファイル名が有効
fixed	Distribution-ID に関係なく 1 つのネットワーク設定を使いまわしたい場合に使うファイル名は default.conf のみ有効

※ ディレクトリごとに設定ファイル名の書式が異なります。注意して下さい。

設定ファイルは、以下の優先順位で Armadillo に読み込まれます。

```
[first] attached -> pool -> fixed [last]
```

設定ファイルは attached/ のものが最優先で読み込まれます。Armadillo は attached/ 以下に、自らの Distribution-ID に一致する設定ファイル([Distribution-ID].conf)が見つかった場合、これを取り込みます。設定ファイルは、1 つ見つかり、その時点で、pool/、fixed/ のコンフィグは使われません。

pool に入っているコンフィグは Armadillo にコンフィグが書かれた後、attached/ に {Distribution-ID}.conf の名前で移動されます。

fixed に入っているコンフィグは Armadillo にコンフィグが書かれた後、attached/ に {Distribution-ID}.conf の名前でコピーされます。

### 6.3.3.1. 使用可能な key 一覧

#### 6.3.3.1.1. eth0 固定 IP

表 6.6 eth0 で 使用可能な key の説明

key	説明
STATIC_ADDRESS	IP アドレス
STATIC_NETMASK	ネットマスク
STATIC_NETWORK	ネットワークアドレス
STATIC_BROADCAST	ブロードキャストアドレス
STATIC_GATEWAY	ゲートウェイ

#### 6.3.3.1.2. umts0

表 6.7 umts0 で使用可能な key の説明

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

#### 6.3.3.1.3. awlan0

表 6.8 awlan0 で使用可能な key の説明

key	説明
WIRELESS_LAN_ENABLE	無線 LAN の有効/無効 (true/false)
WIRELESS_LAN_ESSID	ESSID
WIRELESS_LAN_PASSPHRASE	パスフレーズ
WIRELESS_LAN_CRYPTMODE	暗号化方式
WIRELESS_LAN_WIRELESS_MODE	接続モード



LS Pull における eth0 と awlan0 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。

#### 6.3.3.1.4. resolv

表 6.9 resolv で使用可能な key の説明

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

### 6.3.3.2. 設定例

3G モバイル回線を使う

```
MOBILE_ID="User name"
MOBILE_PASS="Password"
```

```
MOBILE_APN="APN"  
MOBILE_AUTH_TYPE="NON, PAP or CHAP"  
MOBILE_PDP_TYPE="IP or PPP"
```

固定 IP を使う

```
STATIC_ADDRESS="192.0.2.10"  
STATIC_NETMASK="255.255.255.0"  
STATIC_NETWORK="192.0.2.0"  
STATIC_BROADCAST="192.0.2.255"  
STATIC_GATEWAY="192.0.2.1"  
  
RESOLV_SEARCH="local-network"  
RESOLV_NAMESERVER="192.168.10.1"
```

無線 LAN(aw113)を使う

```
WIRELESS_LAN_ENABLE=true  
WIRELESS_LAN_ESSID="myssid"  
WIRELESS_LAN_PASSPHRASE="mypresharedkey"  
WIRELESS_LAN_CRYPTMODE="WPA2-AES"  
WIRELESS_LAN_WIRELESS_MODE="managed"
```

無線 LAN で固定 IP を使う

```
WIRELESS_LAN_ENABLE=true  
WIRELESS_LAN_ESSID="myssid"  
WIRELESS_LAN_PASSPHRASE="mypresharedkey"  
WIRELESS_LAN_CRYPTMODE="WPA2-AES"  
WIRELESS_LAN_WIRELESS_MODE="managed"  
  
STATIC_ADDRESS="192.0.2.10"  
STATIC_NETMASK="255.255.255.0"  
STATIC_NETWORK="192.0.2.0"  
STATIC_BROADCAST="192.0.2.255"  
STATIC_GATEWAY="192.0.2.1"  
  
RESOLV_SEARCH="local-network"  
RESOLV_NAMESERVER="192.168.10.1"
```

## 6.4. Armadillo: 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み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み込むためには、本文「5. イメージファイルのビルド手順」でビルドする Atmark Dist のソースコードに変更を加えて、イメージをビルドしなおす必要があります。

### 6.4.1. モジュールの配置

モジュールの実体となるファイルは展開した Atmark Dist のソースコードの下記のパスに 配置します。

- ・ プライマリ用: atmark-dist/vendors/AtmarkTechno/Armadillo-IoTG-Std-NE/etc/armsd/user\_modules/
- ・ リカバリ用: atmark-dist/vendors/AtmarkTechno/Armadillo-IoTG-Std-NE.Recover/etc/armsd/user\_modules/

モジュールは基本的にシェルスクリプトとして、ファイル名がモジュール番号(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。

## 6.4.2. モジュールのサンプル

モジュールのサンプルは node-eye 対応 の Atmark Dist のソースコード内の下記のパスに配置されています。

- ・ atmark-dist/vendors/AtmarkTechno/Armadillo-IoTG-Std-NE/etc/armsd/user\_modules/4\_command

下記にその内容を記載します。

```
#!/bin/sh

# This module 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 を参考になさってください。

- ・ atmark-dist/vendors/AtmarkTechno/Armadillo-IoTG-Std-NE/etc/armsd/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

```

作成したモジュールはサンプルコードと同じディレクトリに配置して、「5. イメージファイルのビルド手順」に沿ってユーザーランドイメージをビルドしてください。ビルドが完了したイメージファイルは node-eye コントロールパネル からファームウェア更新機能を使って 書き込み、再起動が完了すると SACM コントロールパネル から操作する事ができるようになります。

### 6.4.3. 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 6.10 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。

#### 6.4.4. 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。

### 6.5. 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。

### 6.6. Armadillo を node-eye 非対応の標準イメージに戻す

本章では node-eye 対応のイメージが書かれた Armadillo を、Armadillo-IoT ゲートウェイ G2 の工場出荷状態にする方法を紹介します。再び Armadillo に node-eye 対応のイメージを書き込む方法については「3. node-eye を体験する」の「3.1. Armadillo の準備」を参照してください。

Armadillo-IoT ゲートウェイ G2 の工場出荷状態に戻すため使用するイメージファイルを下記に示します。

表 6.11 工場出荷状態に戻すため使用するイメージファイル一覧

パーティション	イメージファイル名
bootloader	loader-armadillo-iotg-std- <i>[version]</i> .bin
kernel	linux-aiotg-std- <i>[version]</i> .bin.gz
userland	romfs-aiotg-std- <i>[version]</i> .img.gz



最新版のブートローダー、Linux カーネルイメージファイルは Armadillo サイトから、ユーザーランドイメージファイルはユーザーズサイトからダウンロード可能です。

#### Armadillo-IoT ドキュメント・ダウンロード

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

#### Armadillo-IoT ユーザーランド イメージファイル

[https://users.atmark-techno.com/armadillo-iot/userland-image/license\\_agreement](https://users.atmark-techno.com/armadillo-iot/userland-image/license_agreement)

node-eye 非対応の標準イメージに戻すにはパーティション構成の変更を伴うため、イメージの書き換えは順を追って行う必要があります。下記の順番でイメージを書き換えてください。また、書き換えは Hermit-At の保守モードで tftpdلを使用してください。

1. bootloader 領域の書き換え (loader-armadillo-iotg-std-[version].bin)
2. 再起動
3. kernel 領域の書き換え (linux-aiotg-std-[version].bin.gz)
4. userland 領域の書き換え (romfs-aiotg-std-[version].img.gz)

### 6.6.1. bootloader 領域の書き換え

最初に、node-eye 対応ブートローダーイメージを書き込みます。tftp を使用してイメージを書き換えるため、事前に、Armadillo からアクセスすることのできる有線 LAN に接続できるよう ATDE5 を起動して、ATDE5 の /var/lib/tftpboot/ ディレクトリにイメージファイルをコピーしておいてください。Armadillo を保守モードで起動し、下記コマンドを実行してください。

```
hermit> tftpdل [Armadillo の IP アドレス] [ATDE5 の IP アドレス] --boot loader=loader-armadillo-iotg-std-[version].bin
```



### 6.6.2. 再起動

Armadillo の電源を入れなおして、再起動してください。

### 6.6.3. kernel 領域の書き換え

続いて、カーネルの書き換えを行います。下記のコマンドを実行してください。

```
hermit> tftpdل [Armadillo の IP アドレス] [ATDE5 の IP アドレス] --blksize=1024 --kernel=linux-aiotg-std-[version].bin.gz
```



### 6.6.4. userland 領域の書き換え

ユーザーランドの書き換えを行います。下記のコマンドを実行してください。

```
hermit> tftpdل [Armadillo の IP アドレス] [ATDE5 の IP アドレス] --blksize=1024 --userland=romfs-aiotg-std-[version].img.gz
```





## 7. 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	2016/06/10	・ 初版発行
1.0.1	2017/12/22	・ 新フラッシュメモリ適用品(型番 AG43*-ではじまる Armadillo-IoT G2)に対応 ・ 他、軽微な誤記修正

