

# 目次

- IJ SACM の利用** ..... 3
  - 準備** ..... 3
    - 動作検証用の無料SACMアカウント取得 ..... 3
  - セットアップ** ..... 3
    - パッケージの追加 ..... 3
    - init script の作成 ..... 4
    - 設定ファイルの編集 ..... 5
    - ダミーコンフィグの設定 ..... 6
  - armsd の起動** ..... 6
  - 要件ごとのカスタマイズ** ..... 8
  - ダウンロード** ..... 9



# IJJ SACM の利用



SACM (Service Adapter Control Manager) は、IJJ が提供する機器の自動接続、一元管理を可能にするマネージメントシステムサービスです。

MA-E3xx シリーズにパッケージの追加などを行うことで MA-E3xx シリーズの機器を SACM から管理することが可能になります。

## 準備

### 動作検証用の無料SACMアカウント取得

libarms を組み込んだ機器との連携動作を確認できる libarms 動作検証用 SACM が無償で提供されていますので、下記 URL よりアカウントを取得します。

<https://dev.smf.jp/sacm/>

SMF Developer

[Home](#) [Contact](#)

## SACM

libarms動作検証用の無料SACMアカウントを発行しております。下記のリンクよりお申し込みください。

### libarms動作検証用SACM利用申請

libarmsを組み込んだ機器とSMFシステムとの連携動作を確認できる、libarms動作検証用SACMを無償で提供しています。設定情報の自動取得 (Pull) をはじめ、コンフィグ反映、ステータス取得などのPushオペレーション、Heartbeat通信による監視など、SMFで提供される自動接続・完全管理の一連の機能をお試しいただけます。

[申し込み >>](#)

© 2012 Internet Initiative Japan Inc.

## セットアップ

### パッケージの追加

libarms および armsd のパッケージをインストールします。

ファームウェアに含める場合は [PC上で root filesystem を変更する\(QEMU使用\)](#) を参考にして root

filesystem にインストールします。  
簡単に試すには、実機上でインストールします。

File	Stat	SHA1SUM	Note
<a href="#">libarms_5.41_armhf.deb</a>	2017/01/12 10:12 100.5 KB	6d2a4893cf57d08924afc94af04105995c515ee9	
<a href="#">libarms-dev_5.41_armhf.deb</a>	2017/01/12 10:12 7.6 KB	2539ca0e61c9c1141f53f22912b23b0d63aefa6f	開発用 ヘッダ ファイル <sup>1)</sup>
<a href="#">armsd_1.0.7_armhf.deb</a>	2017/01/12 10:12 22.5 KB	2afcd226879b5a1876ec5d1a9ee7b4424f4e61b6	

```
user1@plum:~$ sudo dpkg -i /tmp/libarms_5.41_armhf.deb
/tmp/armsd_1.0.7_armhf.deb
[sudo] password for user1:
Selecting previously unselected package libarms.
(Reading database ... 20950 files and directories currently installed.)
Preparing to unpack /tmp/libarms_5.41_armhf.deb ...
Unpacking libarms (5.41) ...
Selecting previously unselected package armsd.
Preparing to unpack /tmp/armsd_1.0.7_armhf.deb ...
Unpacking armsd (1.0.7) ...
Setting up libarms (5.41) ...
Setting up armsd (1.0.7) ...
Processing triggers for libc-bin (2.19-0ubuntu6.9) ...
user1@plum:~$
```

## init script の作成

armsd パッケージには init script が含まれていませんので作成しておきます。  
Upstart/SysV init どちらでもかまいません。

### **/etc/init/armsd.conf**

#### [armsd.conf](#)

```
# SACM armsd
#

description "SACM armsd"

start on runlevel [2345]
stop on runlevel [!2345]
```

```
respawn

pre-start script
  [ -f /etc/default/armsd ] || { stop; exit 0; }

  . /etc/default/armsd

  [ "x$START" = "xyes" ] || { stop; exit 0; }
end script

script
  . /etc/default/armsd

  [ "x$START" = "xyes" ] || { stop; exit 0; }

  exec /usr/sbin/armsd -D -f /etc/armsd/armsd.conf
end script
```

## **/etc/default/armsd**

### armsd

```
START=yes
```

## 設定ファイルの編集

動作検証用の無料SACMアカウント取得で送られてきたアカウント情報を、設定ファイルに反映します。

## **/etc/armsd/armsd.conf**

### armsd.conf

```
distribution-id: 0001-0000-0101-0000-0000-0000-DEAD-BEEF <-----
ls-sa-key:      deadbeef <-----
#sa-model-name: arms client daemon
#sa-version:    0.00

#path-iconfig:  /etc/armsd/initial-config
#path-state-cache: /var/cache/armsd/state
#https-proxy-url: http://192.168.0.1:8080/

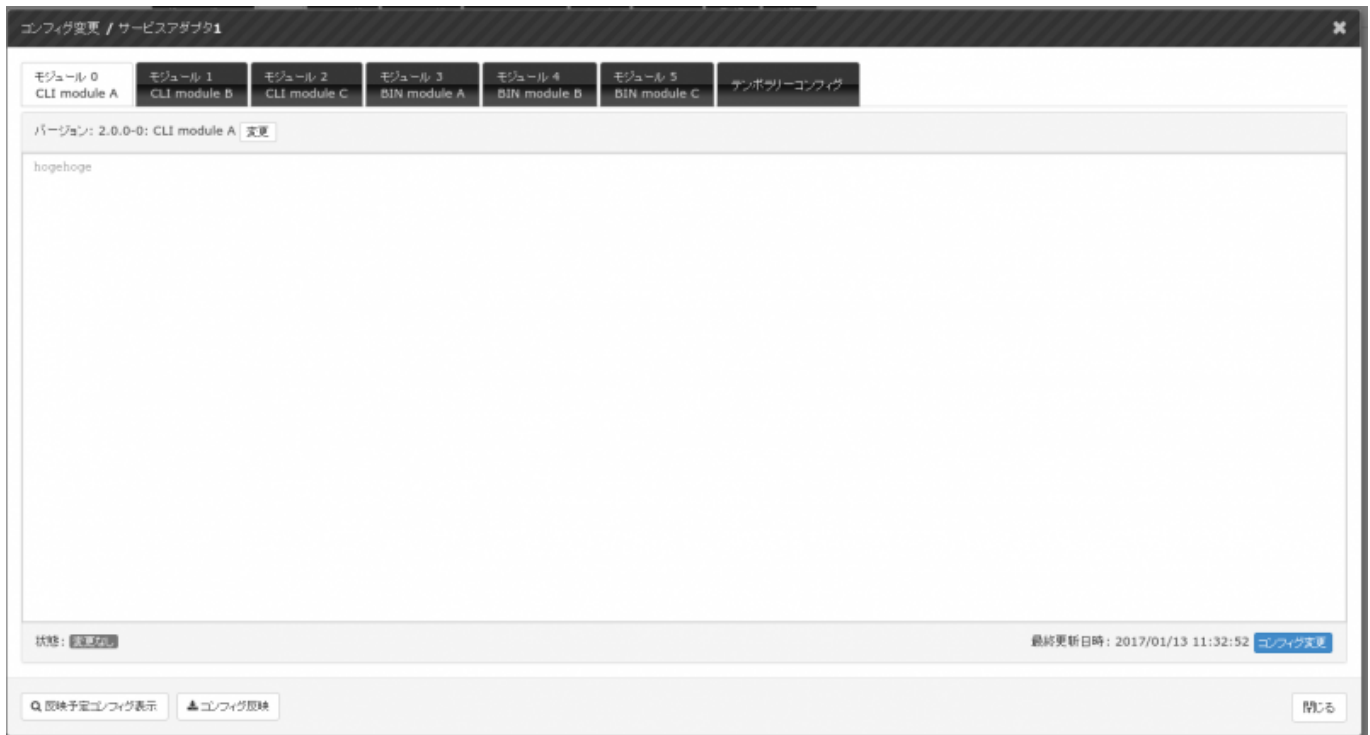
#hb-disk-usage0: /
```

```
#hb-traffic-if0:      eth0

### see sample scripts in /usr/share/armsd-X.X/examples
#script-app-event:   /etc/armsd/scripts/app-event
#script-clear:       /etc/armsd/scripts/clear
#script-command:     /etc/armsd/scripts/command
#script-post-pull:   /etc/armsd/scripts/post-pull
#script-reconfig:    /etc/armsd/scripts/reconfig
#script-start:       /etc/armsd/scripts/start
#script-status:      /etc/armsd/scripts/status
#script-stop:        /etc/armsd/scripts/stop
#script-reboot:      /etc/armsd/scripts/reboot
#script-line-ctrl:   /etc/armsd/scripts/line
#script-state-changed: /etc/armsd/scripts/state-changed
```

## ダミーコンフィグの設定

SACM にコンフィグが登録されていないとエラーとなり接続されないため、適当にダミーのコンフィグを入れておきます。



## armsd の起動

armsd を起動させてみます。

```

user1@plum:~$ sudo service armsd start
[sudo] password for user1:
armsd start/running, process 5176
user1@plum:~$

```

```

user1@plum:~$ ps ax|grep armsd
5176 ?        Ss        0:00 /usr/sbin/armsd -D -f /etc/armsd/armsd.conf
5200 pts/1    S+        0:00 grep --color=auto armsd

```

起動しました。

SACM の Web 画面で確認すると、状態が『接続中』になることが確認できます。

SA / サービスアダプタ1

サービスアダプタ1 操作 ▾

SAコード: [redacted] 白デバイス: libarms Test Client

Distribution ID: 0001-0000-[redacted]

✓ 接続中 操作・コンフィグ有効 ★ アクティベーション済み

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

メモ

---

接続状態

- ✓ 接続中
  - 通知日時: 2017/01/13 11:35:02

接続元情報

- IPアドレス: [redacted]
- ポート: 54866
- 接続モード: 接続持続型
  - 優先接続モード: 接続持続型

---

コンフィグ反映状態

- 操作・コンフィグ有効
  - 操作有効通知日時: 2017/01/13 12:07:08

コンフィグ取得情報

- 初回取得日時: 2017/01/13 10:31:35
- 最終取得日時: 2017/01/13 12:07:08
  - 取得理由: 電源投入

---

フォルダ情報

- なし

テンプレートセット情報

- なし

監視グループ情報

- なし

CPU使用率などのグラフを見ることもできます。

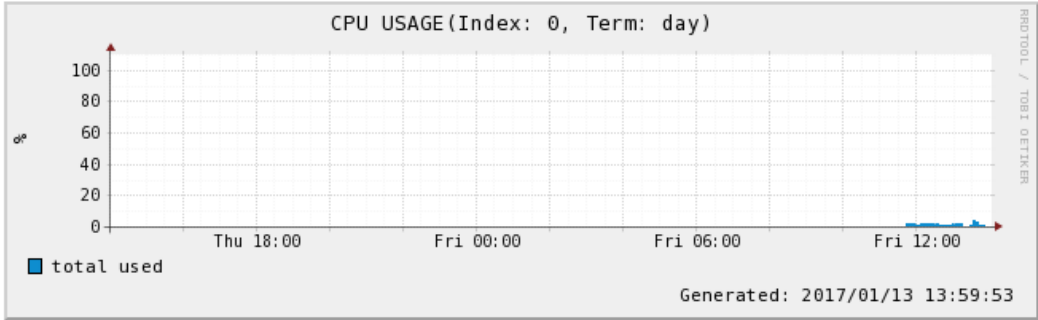
サービスアダプタ1  
SAコード: [redacted] 白デバイス: libarms Test Client  
Distribution ID: 0001-0000-[redacted]  
接続中 操作: コンフィグ有効  
★ アクティベーション済み

- 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

他拠点のグラフと比較 表示設定

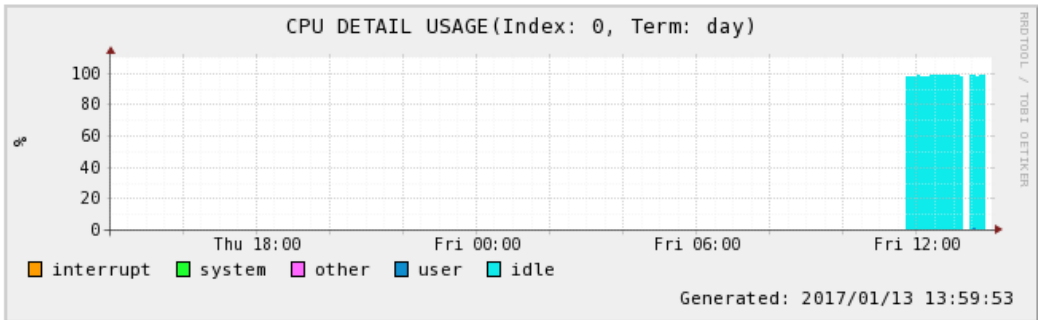
### CPU使用率

インデックス 0



### CPU詳細使用率

インデックス 0



## 要件ごとのカスタマイズ

あとは、アプリケーションの要件ごとにコールバックスクリプトを適宜作成し、

- コンフィグの更新
- 任意コマンドの実行 (ファームウェア更新など)

などを実装することができます。

armsd に関するドキュメントは [こちら](#) を参照して下さい。



## ダウンロード

libarms, armsd および init script を組み込んだサンプルファームウェアです。

File	Stat	SHA1SUM	Note
<a href="#">mae3xx_trusty_sacm_v2_6_12.img</a>	2017/01/13 14:57 67.2 MB	91492c0cd048c4ecd2e79aca8e58cc830ff4cf2d	v2.6.12 ベース
<a href="#">mae320_rootfs_trusty_sacm_v2_6_12.tar.xz</a>	2017/01/13 14:57 52.0 MB	5b8b98cf125ae82636b440398a28a1e18a23b9c1	

<sup>1)</sup>

armsd を利用する場合、インストール不要

From:

<https://centurysys.net/> - MA-X/MA-S/MA-E/IP-K Developers' WiKi

Permanent link:

[https://centurysys.net/doku.php?id=mae3xx\\_tips:iij\\_sacm:start](https://centurysys.net/doku.php?id=mae3xx_tips:iij_sacm:start)

Last update: **2017/01/13 15:55**

