

Armadillo-800 EVA 製品マニュアル

A8000-D00Z

Version 1.0.0
2012/01/05

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>]

Armadillo-800 EVA 製品マニュアル

株式会社アットマークテクノ

060-0035 札幌市中央区北 5 条東 2 丁目 AFT ビル
TEL 011-207-6550 FAX 011-207-6570

製作著作 © 2012 Atmark Techno, Inc.

Version 1.0.0
2012/01/05

目次

1. はじめに	10
1.1. 本書および関連ファイルについて	10
1.2. 本書の構成	11
1.3. 表記について	11
1.3.1. フォント	11
1.3.2. コマンド入力例	11
1.3.3. アイコン	11
1.4. 謝辞	12
2. 注意事項	13
2.1. 評価ボードについてのご注意	13
2.2. 安全に関する注意事項	13
2.3. 無線 LAN 機能搭載製品の使用上のご注意	14
2.4. 保証について	15
2.5. 輸出について	15
2.6. 商標について	15
3. 概要	16
3.1. ボード概要	16
3.2. ブロック図	18
4. インターフェースレイアウト	20
4.1. Armadillo-800 EVA のインターフェース配置	20
4.2. ディップスイッチ	21
5. 電源を入れる前に	23
5.1. 準備するもの	23
5.2. 接続方法	23
5.3. ディップスイッチの設定	25
5.4. シリアル通信ソフトウェアの設定	25
5.5. インストール済みのソフトウェア	25
6. Android を起動	27
6.1. 準備	27
6.2. 起動	27
6.3. 終了方法	28
6.4. 基本操作	28
6.4.1. ロック画面	29
6.4.2. ホーム画面	29
6.4.3. アプリケーション一覧	30
7. Debian GNU/Linux を起動	32
7.1. 準備	32
7.2. 起動	32
7.3. ログイン	33
7.4. 終了方法	34
7.5. 使用方法	34
7.5.1. 有線 LAN の使用方法	34
7.5.2. 時刻の設定	36
7.5.3. パッケージの管理方法	36
7.5.4. カメラを動作させる	38
7.5.5. オーディオを動作させる	39
8. 無線 LAN の使用方法	41
8.1. 準備	41
8.1.1. Debian の起動	41
8.1.2. カーネルモジュール/ファームウェアロードの自動化	41

8.2. 無線設定	42
8.2.1. 無線設定パラメータ	42
8.2.2. インフラストラクチャモード設定例	44
8.2.3. アドホックモード設定例	45
9. SD ブートの活用	47
9.1. ブートディスクを作成	47
9.1.1. ブートディスクの作成に必要なファイルの取得	48
9.1.2. パーティションの作成	48
9.1.3. ファイルシステムの構築	50
9.1.4. システムイメージの展開	51
9.1.5. 設定ファイルの編集	51
9.2. ブートディスクから起動する	52
10. リカバリ手順	53
10.1. パーティション構成	53
10.2. eMMC 全体をリカバリする	54
10.2.1. リカバリディスクの作成	54
10.2.2. リカバリを実行する	58
10.3. eMMC の特定ルートファイルシステムをリカバリする	58
10.3.1. Debian GNU/Linux をリカバリする	59
10.3.2. Android をリカバリする	60
10.4. eMMC のブートローダーをリカバリする	61
10.4.1. ブートローダーのリカバリに必要なファイルの取得	61
10.4.2. ブートローダーのリカバリを実行する	61
11. 開発環境の準備	63
11.1. ATDE の使用方法	63
11.1.1. ATDE の取得	63
11.1.2. ATDE の起動	64
11.2. クロス開発ツールのインストール方法	64
11.2.1. クロス開発ツール Debian パッケージの取得	64
11.2.2. クロス開発ツール Debian パッケージのインストール	64
12. カーネルのビルド	66
12.1. ソースアーカイブの取得	66
12.2. ソースアーカイブの展開	66
12.3. ビルド	66
13. SGX 用カーネルモジュールのビルド	67
13.1. ソースアーカイブ取得	67
13.2. カーネルの準備	67
13.3. ビルド	67
13.4. インストール	68
14. 無線 LAN(AWL13) 用 Linux デバイスドライバーのビルド	70
14.1. ソースアーカイブ取得	70
14.2. カーネルの準備	70
14.3. ビルド	70
15. ブートローダーのビルド	72
15.1. ソースアーカイブの取得	72
15.2. ソースアーカイブの展開	72
15.3. ビルド	72
16. JTAG ICE を利用する	74
16.1. 準備	74
16.1.1. JTAG ケーブルの接続	74
16.1.2. ディップスイッチの設定	74
16.2. 接続確認	74
16.3. 各種デバッグへの対応について	74

17. Linux ドライバー 一覧	75
18. インターフェース仕様	78
18.1. CON1(カメラモジュールインターフェース)	78
18.2. CON2(拡張バスインターフェース)	78
18.3. CON3(HDMI 出力インターフェース)	80
18.4. CON4(コンジットビデオ出力インターフェース)	80
18.5. CON5(H-UDI JTAG インターフェース)	80
18.6. CON6(ARM JTAG インターフェース)	81
18.7. CON7(SD インターフェース 1)	82
18.8. CON8(SD インターフェース 2)	82
18.9. CON9(RTC 外部バックアップインターフェース)	83
18.10. CON10~CON13(オーディオインターフェース)	83
18.10.1. CON10(モノラルマイク入力インターフェース)	83
18.10.2. CON11(ステレオヘッドホン出力インターフェース)	83
18.10.3. CON12(オーディオ ライン出力(L)インターフェース)	84
18.10.4. CON13(オーディオ ライン出力(R)インターフェース)	84
18.11. CON14(AWL13 モジュールインターフェース)	84
18.12. CON15(拡張インターフェース)	85
18.13. CON16(LCD インターフェース 1)	86
18.14. CON17(LCD インターフェース 2)	88
18.15. CON19(電源入力インターフェース)	89
18.16. CON20(USB インターフェース 1)	89
18.17. CON21(USB インターフェース 2)	90
18.18. CON22(シリアルインターフェース)	90
18.19. CON23(LAN インターフェース)	90
18.20. CON24(USB インターフェース 3)	91
18.21. LED1(カメラ LED)	91
18.22. LED2(電源 LED)	91
18.23. LED3~LED6(ユーザー LED)	91
18.24. LED7、LED8(LAN LED)	92
18.25. SW1(機能選択スイッチ)	92
18.26. SW2(リセットスイッチ)	92
18.27. SW3~SW6(ユーザースイッチ)	93
19. 基板形状図	94
A. Hermit-At ブートローダー	96
A.1. version	96
A.1.1. version 使用例	97
A.2. info	97
A.2.1. info 使用例	97
A.3. mac	97
A.3.1. mac 使用例	98
A.4. setenv と clearenv	98
A.4.1. setenv/clearenv 使用例	98
A.4.2. Linux カーネルパラメーター	98
A.5. setbootdevice	99
A.5.1. setbootdevice の使用例	99
A.6. frob	99
A.7. boot	99
A.7.1. boot 使用例	100

目次

3.1. Armadillo-800 EVA のブロック図	19
4.1. Armadillo-800 EVA のインターフェース配置図	20
4.2. ディップスイッチ(SW1)の設定(出荷時)	22
5.1. Armadillo-800 EVA 接続例	24
5.2. ディップスイッチの設定	25
6.1. 起動 OS 設定の変更手順(Android)	27
6.2. 起動ログ(Android)	27
6.3. ロック画面	29
6.4. ホーム画面	29
6.5. ホーム画面: アプリケーション一覧ボタン	30
6.6. アプリケーション一覧画面	30
7.1. 起動設定の変更手順(Debian)	32
7.2. 起動ログ(Linux)	32
7.3. 終了方法	34
7.4. ifup コマンドによる有線 LAN の有効化	34
7.5. ifdown コマンドによる有線 LAN の無効化	35
7.6. システムクロックの設定	36
7.7. X サーバーと GStreamer のインストール	38
7.8. X サーバーの起動	38
7.9. X サーバー起動画面	38
7.10. カメラの映像を LCD に表示	39
7.11. 音声ファイルの再生	39
7.12. マイク入力をスピーカに出力	39
7.13. gnome-audio パッケージの音声ファイルを利用	40
8.1. インフラストラクチャモード: WPA2-PSK(AES)設定手順	45
8.2. インフラストラクチャモード: WEP 設定手順	45
8.3. アドホックモード: WEP 設定手順	46
9.1. パーティション作成手順	48
9.2. パーティション確認手順	49
9.3. ファイルシステム作成手順	50
9.4. システムイメージ展開手順	51
9.5. fstab の編集	51
9.6. ブートディスクからの起動	52
10.1. パーティション作成手順	54
10.2. パーティション確認手順	56
10.3. ファイルシステム作成手順	56
10.4. ファイル展開手順	57
10.5. Debian GNU/Linux のリカバリ手順	59
10.6. Android のリカバリ手順	60
10.7. ブートローダーのリカバリ手順	61
11.1. 64-bit PC 用クロス開発ツール Debian パッケージインストールコマンド	65
11.2. 32-bit PC 用クロス開発ツール Debian パッケージインストールコマンド	65
12.1. ソースアーカイブの展開	66
12.2. カーネルのビルド	66
13.1. SGX 用カーネルモジュールのビルド	68
13.2. SGX 用カーネルモジュールのインストール	69
14.1. AWL13 ドライバーのビルド	71
15.1. ソースアーカイブの展開	72
15.2. ブートローダーのビルド	73
16.1. ディップスイッチの JTAG 設定(ARM)	74

18.1. AC アダプターの極性マーク	89
18.2. リセットブロック図	93
19.1. 基板形状および固定穴寸法	94
19.2. コネクタ中心寸法	95
A.1. version 構文	97
A.2. version の使用例	97
A.3. info 構文	97
A.4. info の使用例	97
A.5. mac 構文	97
A.6. mac の使用例	98
A.7. setenv/clearenv 構文	98
A.8. setenv と clearenv の使用例	98
A.9. setbootdevice 構文	99
A.10. ブートデバイスに内蔵ストレージのパーティション 4 を指定する	99
A.11. ブートデバイスに SD カードを指定する	99
A.12. boot 構文	100
A.13. boot の使用例	100

表目次

1.1. 使用しているフォント	11
1.2. 表示プロンプトと実行環境の関係	11
1.3. コマンド入力例での省略表記	11
3.1. Armadillo-800 EVA の仕様	16
4.1. Armadillo-800 EVA のインターフェース内容	21
4.2. ディップスイッチの設定(出荷時)	21
4.3. ディップスイッチ(SW1)のスイッチの機能	22
5.1. ディップスイッチの設定	25
5.2. シリアル通信設定	25
5.3. インストールされている OS	25
6.1. ユーザースイッチの名称と機能	28
7.1. シリアルコンソールログイン時のユーザ名とパスワード	34
8.1. インフラストラクチャモード: WPA-PSK/WPA2-PSK パラメータ例	45
8.2. インフラストラクチャモード: WEP パラメータ例	45
8.3. アドホックモード: WEP パラメータ例	46
9.1. ブートディスクの構成	47
9.2. ブートディスクの作成に必要なファイル	48
10.1. 内蔵ストレージのパーティション構成	53
10.2. eMMC のブートパーティション	53
10.3. 内蔵ストレージ各領域の用途	53
10.4. リカバリディスクの作成に必要なファイル	54
10.5. リカバリ進捗と LED の対応	58
10.6. リカバリ対象のルートファイルシステム	58
10.7. Debian GNU/Linux のリカバリに必要なファイル	59
10.8. Android のリカバリに必要なファイル	60
10.9. ブートローダーのリカバリに必要なファイル	61
11.1. ATDE4 の種類	63
11.2. ユーザ名とパスワード	64
18.1. CON1 信号配列	78
18.2. CON2 信号配列	78
18.3. CON3 信号配列	80
18.4. CON4 信号配列	80
18.5. CON5 信号配列	81
18.6. CON6 信号配列	81
18.7. CON7 信号配列	82
18.8. CON8 信号配列	82
18.9. CON9 信号配列	83
18.10. CON10 信号配列	83
18.11. CON11 信号配列	84
18.12. CON12 信号配列	84
18.13. CON13 信号配列	84
18.14. CON14 信号配列	84
18.15. CON15 信号配列	85
18.16. CON16 信号配列	87
18.17. CON17 信号配列	88
18.18. CON19 信号配列	89
18.19. CON20 信号配列	90
18.20. CON21 信号配列	90
18.21. CON22 信号配列	90
18.22. CON23 信号配列	90

18.23. CON24 信号配列	91
18.24. LED1 の挙動	91
18.25. LED の挙動	91
18.26. LED3～LED6 の挙動	92
18.27. LED7、LED8 の挙動	92
18.28. SW1 信号配列	92
18.29. SW2 の機能	93
18.30. SW3～SW6 の機能	93
A.1. よく使用される Linux カーネルパラメーター	98
A.2. frob コマンド	99

1. はじめに

このたびは Armadillo-800 EVA をお求めいただき、誠にありがとうございます。



製品本体を開封する前に必ずお読みください

本製品には、「ソフトウェア使用許諾契約書」に同意いただく必要のあるソフトウェアが含まれています。はじめに「ソフトウェア使用許諾契約書」(同梱の資料「はじめにお読みください」に記載)をご確認いただき、同意の上で開封してください。

本製品の本体基板のパッケージを開封した時点、または本製品に付属するDVDのパッケージを開封した時点で、「ソフトウェア使用許諾契約書」に同意したものとさせていただきます。

1.1. 本書および関連ファイルについて

本書を含めた関連マニュアル、ソースファイルやイメージファイルなどの関連ファイルは最新版を使用することをおすすめいたします。本書を読み進める前に、Armadillo サイト [<http://armadillo.atmark-techno.com>]から最新版の情報をご確認ください。

また、Armadillo-800 EVA をご購入の上でアットマークテクノ ユーザーズサイト [<https://users.atmark-techno.com>]から購入製品登録を行った方に限定して、下記のユーザー限定コンテンツを公開します。

- ・ ご購入ユーザー限定のソフトウェア
- ・ Armadillo-800 EVA 回路図



アットマークテクノ ユーザーズサイトでは、ユーザー登録されたお客様に向けて、アップデートの情報や、サポート情報、新製品のご案内などの情報を提供いたします。



購入製品登録を行うには、ユーザーズサイトの「ユーザー限定コンテンツ」メニューにアクセスしてください。

※購入製品登録には、製品本体から「正規認証ファイル」を取得する必要があります。登録を行う際は、お手元に購入製品をご用意ください。(正規認証ファイルの取得方法については、ユーザーズサイトの「ユーザー限定コンテンツ」メニューページに記載されています。)

1.2. 本書の構成

本書には、Armadillo-800 EVA の下記の項目について記載します。

- ・ ハードウェア仕様
- ・ インストール済みソフトウェアの説明
- ・ 起動方法
- ・ リカバリ方法
- ・ 開発環境の構築方法
- ・ ソフトウェアのビルド方法

1.3. 表記について

1.3.1. 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

1.3.2. 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

1.3.3. 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。

1.4. 謝辞

Armadillo で使用しているソフトウェアは Free Software / Open Source Software で構成されています。Free Software / Open Source Software は世界中の多くの開発者の成果によってなっています。この場を借りて感謝の意を表します。

2. 注意事項

2.1. 評価ボードについてのご注意

「開発セット」「評価セット」として販売している製品(以下「評価ボード」といいます)は、評価目的、技術開発またはデモンストレーション用途向けです。以下の事項をご理解・ご了承いただいた上で、ご使用いただきますようお願いいたします。



- ・ 評価ボードは、電子工学に関する技術知識と実務経験を有する技術者によって、良識ある技術的・実務的基準に従って取り扱われることを想定しています。
- ・ 評価ボードは、一般消費者が利用する最終製品において通常要求されるような設計上、販売上、または製造上の保護的措置については未完成品です。
- ・ 弊社は評価ボードについて、弊社の製品保証規定に従いご購入後 1 年間の交換保証のみを行うものとします。
- ・ 弊社は評価ボードのご購入者に対し、上記の交換保証を除き、評価ボードが特定目的に合致することの保証を含む明示的・黙示的な保証、その他ありとあらゆる保証に関する一切の責任を負わないものとします。
- ・ 評価ボードまたはその構成部品に不具合が発生した場合であっても、弊社はその原因の解析を行いません。

2.2. 安全に関する注意事項

本製品を安全にご使用いただくために、特に以下の点にご注意ください。



- ・ ご使用の前に必ず製品マニュアルおよび関連資料をお読みにになり、使用上の注意を守って正しく安全にお使いください。
- ・ マニュアルに記載されていない操作・拡張などを行う場合は、弊社 Web サイトに掲載されている資料やその他技術情報を十分に理解した上で、お客様自身の責任で安全にお使いください。
- ・ 水・湿気・ほこり・油煙等の多い場所に設置しないでください。火災、故障、感電などの原因になる場合があります。
- ・ 本製品に搭載されている部品の一部は、発熱により高温になる場合があります。周囲温度や取扱いによってはやけどの原因となる恐れがあります。本体の電源が入っている間、または電源切断後本体の温度が下がるまでの間は、基板上の電子部品、及びその周辺部分には触れないでください。

- ・ 本製品を使用して、お客様の仕様による機器・システムを開発される場合は、製品マニュアルおよび関連資料、弊社 Web サイトで提供している技術情報のほか、関連するデバイスのデータシート等を熟読し、十分に理解した上で設計・開発を行ってください。また、信頼性および安全性を確保・維持するため、事前に十分な試験を実施してください。
- ・ 本製品は、機能・精度において極めて高い信頼性・安全性が必要とされる用途(医療機器、交通関連機器、燃焼制御、安全装置等)での使用を意図しておりません。これらの設備や機器またはシステム等に使用された場合において、人身事故、火災、損害等が発生した場合、当社はいかなる責任も負いかねます。
- ・ 本製品には、一般電子機器用(OA 機器・通信機器・計測機器・工作機械等)に製造された半導体部品を使用しています。外来ノイズやサージ等により誤作動や故障が発生する可能性があります。万一誤作動または故障などが発生した場合に備え、生命・身体・財産等が侵害されることのないよう、装置としての安全設計(リミットスイッチやヒューズ・ブレーカー等の保護回路の設置、装置の多重化等)に万全を期し、信頼性および安全性維持のための十分な措置を講じた上でお使いください。
- ・ 無線 LAN 機能を搭載した製品は、心臓ペースメーカーや補聴器などの医療機器、火災報知器や自動ドアなどの自動制御器、電子レンジ、高度な電子機器やテレビ・ラジオに近接する場所、移動体識別用の構内無線局および特定小電力無線局の近くで使用しないでください。製品が発生する電波によりこれらの機器の誤作動を招く恐れがあります。

2.3. 無線 LAN 機能搭載製品の使用上のご注意

本製品は、2.4GHz 帯域の電波を使用します。稼動時に電波を使用しますので、電磁妨害や電波干渉が発生する恐れがあります。



- ・ 心臓ペースメーカーや補聴器などの医療機器、火災報知器や自動ドアなどの自動制御器、電子レンジ、高度な電子機器やテレビ・ラジオに近接する場所で使用しないでください。
- ・ 移動体識別用の構内無線局および特定小電力無線局の近くで使用しないでください。
- ・ 万一、本製品と同種無線局や他の機器との電波干渉が発生した場合は、すみやかに使用場所を変えるか、製品の稼動を停止（電波の使用を停止）してください。

2.4. 保証について

本製品の本体基板は、製品に添付もしくは弊社 Web サイトに記載している「製品保証規定」に従い、ご購入から 1 年間の交換保証を行っています。添付品およびソフトウェアは保証対象外となりますのでご注意ください。

製品保証規定 <http://www.atmark-techno.com/support/warranty-policy>

2.5. 輸出について

本製品の開発・製造は、原則として日本国内での使用を想定して実施しています。本製品を輸出する際は、輸出者の責任において、輸出関連法令等を遵守し、必要な手続きを行ってください。海外の法令および規則への適合については当社はなんらの保証を行うものではありません。本製品および関連技術は、大量破壊兵器の開発目的、軍事利用その他軍事用途の目的、その他国内外の法令および規則により製造・使用・販売・調達が禁止されている機器には使用することができません。

2.6. 商標について

- ・ Armadillo は株式会社アットマークテクノの登録商標です。その他の記載の商品名および会社名は、各社・各団体の商標または登録商標です。™、®マークは省略しています。
- ・ SD、SDHC、microSD、microSDHC、SDIO ロゴは SD-3C、LLC の商標です。



3. 概要

3.1. ボード概要

Armadillo-800 EVA はルネサス エレクトロニクス製のプロセッサ「R-Mobile A1」(ARM Cortex-A9、CPU クロック 最大800MHz)と、512MB DRAM(DDR3-800)、8GB フラッシュメモリ(eMMC)を搭載した評価ボードです。5 インチ LCD(WVGA)静電容量方式タッチパネル、HDMI 出力対応、3.1Mピクセル CMOS カメラモジュールなど、マルチメディア機能を多数搭載しています。また、IEEE802.11b/g/n 準拠の産業用組み込み機器向け無線 LAN モジュール「Armadillo-WLAN(AWL13)」を標準搭載しています。入手しづらいモジュールを一式取り揃え、多数の機能を評価することができます。

Armadillo-800 EVA の主な仕様は次の通りです。

表 3.1 Armadillo-800 EVA の仕様

プロセッサ	ルネサス エレクトロニクス R-Mobile A1 (R8A77404DBA)	
CPU コア	ARM Cortex-A9 シングルコア 命令/データキャッシュ 32kByte/32kByte L2 キャッシュ 256kByte メディアプロセッシングエンジン(NEON)搭載 浮動小数点コプロセッサ(VFPv3)搭載	
システムクロック	CPU コアクロック: 792MHz (最大 800MHz ^[1]) DDR クロック: 396MHz (最大 400MHz ^[1]) 内部 BUS クロック: 198MHz (最大 200MHz ^[1]) 拡張 BUS クロック: 99MHz (最大 100MHz ^[1]) 源発振クロック: 24MHz	
RAM	DDR3 SDRAM : 512MByte (32bit 幅) (DDR3-800)	
フラッシュメモリ	eMMC NAND フラッシュメモリ : 8GByte	
LAN(Ethernet)	10BASE-T/100BASE-TX AUTO-MDIX 対応	
無線 LAN	SDHI1 ^[2]	Armadillo-WLAN モジュール(AWL13) IEEE 802.11b/g/n 対応 (最大 72.2Mbps ^[3])
シリアル(UART)	最大 7 チャンネル	
	SCIFA0 ^[4]	+3.3V CMOS レベル フロー制御ピン有り(CTS、RTS) 拡張 I/O コネクタ
	SCIFA1	RS-232C レベル フロー制御ピンなし D-Sub9 ピンコネクタ
	SCIFA2 ^[5]	+3.3V CMOS レベル フロー制御ピン有り(CTS、RTS) 同期クロックピン有り(SCK) 拡張 I/O コネクタ
	SCIFA4 ^[6]	+3.3V CMOS レベル フロー制御ピンなし 拡張 I/O コネクタ
	SCIFA6 ^[4]	+3.3V CMOS レベル フロー制御ピンなし 同期クロックピン有り(SCK) 拡張 I/O コネクタ
	SCIFA7 ^[7]	+3.3V CMOS レベル フロー制御ピンなし LCD 拡張 I/F コネクタ
	SCIFB ^[8]	+3.3V CMOS レベル フロー制御ピンなし 同期クロックピン有り(SCK) 拡張 I/O コネクタまたは LCD 拡張 I/F コネクタ

USB	2チャンネル	
	USB0 ^[9]	USB HOST(High Speed 対応) TYPE A コネクタ USB DEVICE(High Speed 対応) TYPE B コネクタ
	USB1	USB HOST(High Speed 対応) TYPE A コネクタ
SD/MMC	最大2チャンネル ^[10]	
	SDHIO	SD スロット
	SDHI1 ^[11]	SD スロット
ビデオ	LCDC0 ^[12]	RGB/YCrCb インターフェース 最大 1440×900 ドット/24bit カラー 液晶パネル ^[13] - MPIRE 社製 AM-800480L1TMQW-T00H - 5 インチ WVGA(800×480 ドット/24bit カラー) - LED バックライト - 静電容量式マルチタッチパネル付) LCD 拡張 I/F コネクタ ^[13]
	LCDC1	HDMI 出力 ^{[14][15]} Type-A コネクタ ^[16] コンポジットビデオ出力 ^[15] RCA ピンジャック
	VOU ^[12]	デジタルビデオ出力、LCD 拡張 I/F コネクタ
地デジチューナー I/F	SSP ^[4]	8bit パラレルバスまたは 1bit シリアルバス 拡張 I/O コネクタ
スマートカード I/F	SIM	拡張 I/O コネクタ
カメラ	最大2チャンネル ^[17]	
	CEU0	最大 16bit データバス ^[18] カメラモジュール - CRESYN 社製 DCB-NSB55QFMRB - 3.1M ピクセル QXGA:最大 15fps, VGA:最大 30fps 8bit データバス) 拡張 I/O コネクタ
	CEU1	8bit データバス 拡張 I/O コネクタ
オーディオ	FSIA	ステレオヘッドホン出力ミニジャック ステレオライン出力 RCA ピンジャック モノラルマイク入力ミニジャック
	FSIB	HDMI 出力 ^[14] Type-A コネクタ ^[16]
I2C	3チャンネル	
	I2C0	ボード内部デバイス用 LCD 拡張 I/F コネクタ
	I2C1	外部拡張利用専用 拡張 I/O コネクタ
	I2C2(GPIO)	RTC 専用
SPI I2S Microwire	最大2チャンネル	
	MSIOF1 ^[19]	拡張バスコネクタ
	MSIOF2 ^[20]	LCD 拡張 I/F コネクタ
GPIO	13bit(最大 188bit ^[21])	
拡張バス	アドレスバス: 25bit データバス: 最大 16bit ^[22] チップセレクト: 最大 4bit ^[22]	
カレンダー時計	リアルタイムクロック	
RTC バックアップ電源	コイン電池対応 ^[23]	
スイッチ	タクトスイッチ x 4	
LED	LED(黄色、面実装) x 4	

JTAG	ARM 標準 20 ピンコネクタ ^[24] H-UDI 14 ピンコネクタ(コネクタ非搭載) ^[25]
電源電圧	5V±5%
使用周囲温度	10~40°C(ただし結露なきこと)
基板サイズ	183×135mm(突起部含まず)

- [1]R-Mobile A1 の最大クロック周波数
- [2]SD スロットと排他で利用可能
- [3]規格上の理論値
- [4]CEU1 の一部の信号と排他で利用可能
- [5]SSP および SIM の一部の信号と排他で利用可能
- [6]SSP の一部の信号と排他で利用可能
- [7]LCDC0 の一部の信号と排他で利用可能
- [8]SSP または LCDC0 の一部の信号と排他で利用可能
- [9]USB HOST と USB DEVICE は排他で利用可能
- [10]SDHI1 を SD スロットに選択した場合のチャンネル数
- [11]無線 LAN と排他で利用可能
- [12]LCDC0 と VOU は排他で利用可能
- [13]液晶パネルと LCD 拡張 I/F コネクタは排他で利用可能
- [14]HDMI 認証申請中
- [15]HDMI とコンポジットビデオは排他で利用可能
- [16]ビデオとオーディオの HDMI 出力 Type-A コネクタは同一
- [17]CEU0 を 8bit データバスで使用した場合のチャンネル数
- [18]CEU1 を使用しない場合に可能な最大 bit 数
- [19]拡張バスの一部の信号と排他で利用可能
- [20]LCDC0 信号の一部と排他で利用可能
- [21]R-Mobile A1 のピンマルチプレクス機能で、GPIO として利用可能なピンの合計
- [22]eMMC を使用しない場合に可能な最大 bit 数
- [23]電池は付属しません
- [24]ARM コネクタと H-UDI コネクタは信号線を共有しているため排他で利用可能
- [25]コネクタは非搭載

3.2. ブロック図

Armadillo-800 EVA のブロック図は次の通りです。

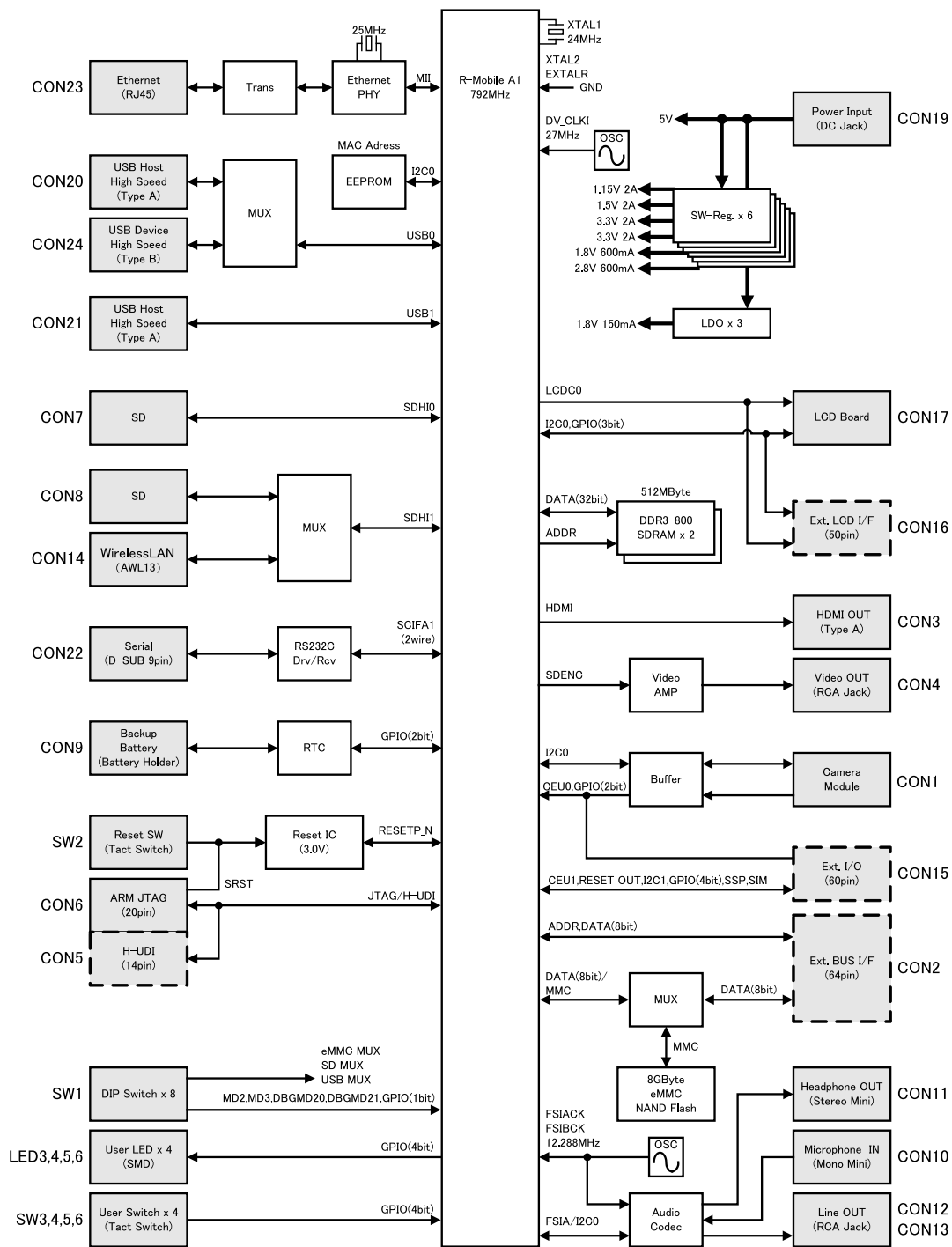


図 3.1 Armadillo-800 EVA のブロック図

4. インターフェースレイアウト

Armadillo-800 EVA のインターフェースレイアウトです。各インターフェースの配置場所等を確認してください。

4.1. Armadillo-800 EVA のインターフェース配置

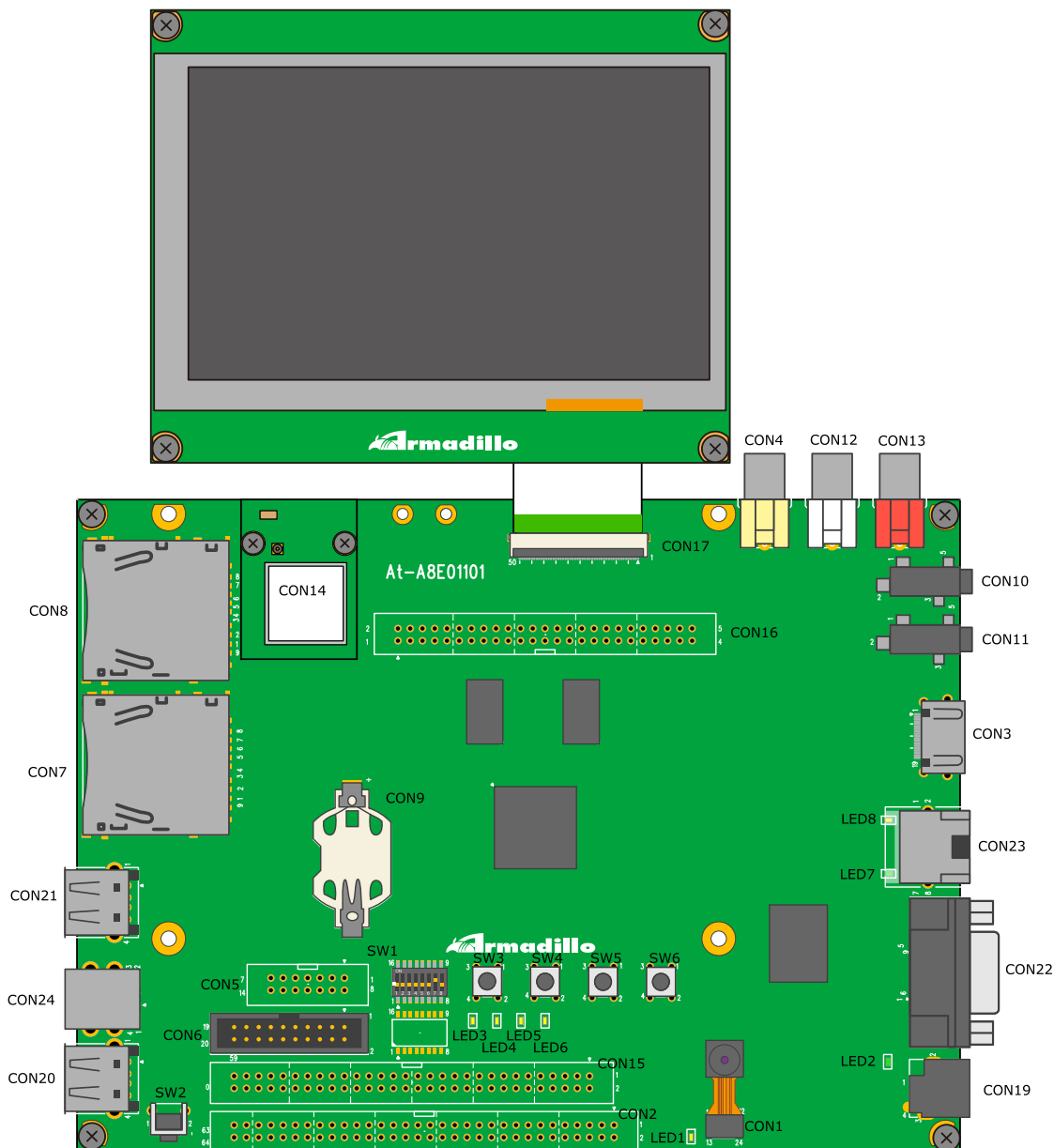


図 4.1 Armadillo-800 EVA のインターフェース配置図

表 4.1 Armadillo-800 EVA のインターフェース内容

部品番号	インターフェース	形状	備考
CON1	カメラモジュール	FPC コネクタ(24P) (0.4mm ピッチ)	挿抜寿命:50 回
CON2	拡張バス	ピンヘッダ(64P) (2.54mm ピッチ)	コネクタ非搭載
CON3	HDMI 出力	HDMI Type A コネクタ	
CON4	コンポジットビデオ出力	RCA ジャック黄色	
CON5	H-UDI JTAG	ピンヘッダ(14P) (2.54mm ピッチ)	コネクタ非実装
CON6	ARM JTAG	ピンヘッダ(20P) (2.54mm ピッチ)	
CON7	SD1	SD スロット	
CON8	SD2	SD スロット	
CON9	RTC 外部バックアップ	電池ボックス	対応電池: CR2032
CON10	モノラルマイク入力	ミニジャック (φ3.5mm)	
CON11	ステレオヘッドホン出力	ミニジャック (φ3.5mm)	
CON12	オーディオ ライン出力(L)	RCA ジャック白色	
CON13	オーディオ ライン出力(R)	RCA ジャック赤色	
CON14	AWL13 モジュール	狭ピッチコネクタ(34P) (0.5mm ピッチ)	挿抜寿命: 50 回
CON15	拡張	ピンヘッダ(60P) (2.54mm ピッチ)	コネクタ非実装
CON16	LCD1	ピンヘッダ(50P) (2.54 ピッチ)	コネクタ非実装
CON17	LCD2	FFC コネクタ(50P) (0.5mm ピッチ)	挿抜寿命: 20 回
CON19	電源入力	DC ジャック	対応プラグ: EIAJ #2
CON20	USB1	USB Type A コネクタ	
CON21	USB2	USB Type A コネクタ	
CON22	シリアル	D-Sub9 ピン(オス)	
CON23	LAN	RJ-45 コネクタ	
CON24	USB3	USB Type B コネクタ	
LED1	カメラ LED(黄色)	LED(面実装)	
LED2	電源 LED(緑色)	LED(面実装)	
LED3~LED6	ユーザー LED(黄色)	LED(面実装)	
LED7	LAN リンク LED(緑色)	LED(面実装)	
LED8	LAN アクティビティ LED(黄色)	LED(面実装)	
SW1	機能選択スイッチ	ディップスイッチ(8 接点)	
SW2	リセットスイッチ	タクトスイッチ(l=3.85mm) ライトアングル	
SW3~SW6	ユーザースイッチ	タクトスイッチ(h=5mm) ストレート	

4.2. ディップスイッチ

Armadillo-800 EVA に電源を入れる前にディップスイッチ(SW1)を設定する必要があります。出荷時は「表 4.2. ディップスイッチの設定(出荷時)」のように設定されています。

表 4.2 ディップスイッチの設定(出荷時)

SW1.1	SW1.2	SW1.3	SW1.4	SW1.5	SW1.6	SW1.7	SW1.8
OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF

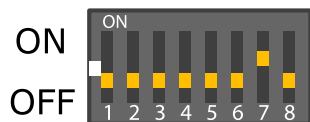


図 4.2 ディップスイッチ(SW1)の設定(出荷時)

各スイッチの機能は次の通りです。

表 4.3 ディップスイッチ(SW1)のスイッチの機能

機能	スイッチ		動作
起動モード設定	SW1.1		
	OFF		OS 自動起動モード
	ON		保守モード
起動デバイス設定	SW1.2	SW1.3	
	OFF	OFF	eMMC
	ON	OFF	SDHI0(CON7)
	OFF	ON	(未設定)
	ON	ON	拡張バス(CS0)
拡張バス設定	SW1.4		
	OFF		データバス上位 8bit(D8~D15)無効/eMMC 有効
	ON		データバス上位 8bit(D8~D15)有効/eMMC 無効
SDHI1 設定	SW1.5		
	OFF		SD インターフェース 2(CON8)無効/AWL13 モジュールインターフェース (CON14)有効
	ON		SD インターフェース 2(CON8)有効/AWL13 モジュールインターフェース (CON14)無効
USB0 設定	SW1.6		
	OFF		USB インターフェース 1(CON20)有効/USB インターフェース 3(CON24)無効
	ON		USB インターフェース 1(CON20)無効/USB インターフェース 3(CON24)有効
JTAG 設定	SW1.7	SW1.8	
	OFF	OFF	SH-X2
	ON	OFF	ARM
	OFF	ON	(未設定)
	ON	ON	バウンダリスキャン

5. 電源を入れる前に

5.1. 準備するもの

Armadillo-800 EVA を評価するには、以下の機材を準備する必要があります。

作業用 PC	Debian GNU/Linux もしくは Windows が動作し、1 ポート以上のシリアルインターフェースを持つ PC です。
シリアルクロスケーブル	Armadillo-800 EVA と作業用 PC を接続するための、D-Sub9 ピン（メス - メス）のクロス接続用ケーブルがです。
シリアル通信ソフトウェア	Linux では「minicom」、Windows では「Tera Term Pro」などです。Armadillo を制御するために使用します。作業用 PC にインストールしてください。

また、以下の機材を利用する場合があります。必要に応じて準備してください。

SD カード	容量が 1GByte 以上のものを準備してください。Armadillo-800 EVA を工場出荷状態に戻すためなどに利用します。
USB メモリ	Armadillo-800 EVA と作業用 PC 間のデータ転送等に利用します。
LAN ケーブル	Armadillo-800 EVA と LAN を経由した通信を行う場合に必要となります。 ^[1]
HDMI ケーブル	CON3 から映像を出力させる場合に必要となります。
RCA ケーブル	CON12, CON13 及び CON4 から映像または音声を出力させる場合に必要となります。
スピーカー又はヘッドフォン	CON11 から音声を出力させる場合に必要となります。スピーカーを利用する場合は、別途ステレオミニプラグケーブルが必要となる場合があります。
マイク	CON10 から音声を入力させる場合に必要となります。
ディスプレイ	HDMI 端子または RCA 入力端子をもつものです。映像または音声を出力させる場合に必要となります。

5.2. 接続方法

Armadillo-EVA の接続例です。

^[1]Armadillo-800 EVA は Auto MDIX に対応しているため、作業用 PC と LAN ケーブルで直接接続することができます。

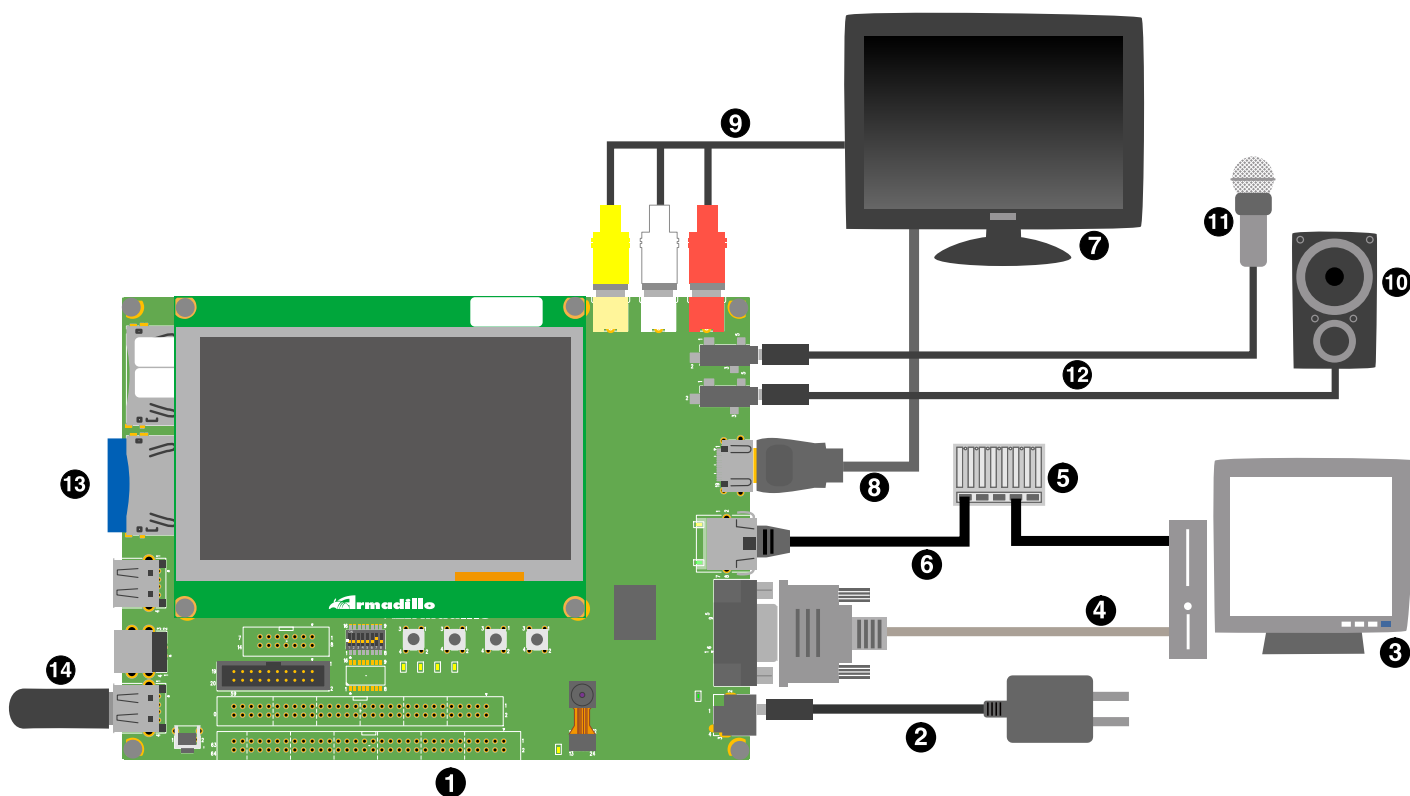



図 5.1 Armadillo-800 EVA 接続例

- ① Armadillo-800 EVA
- ② AC アダプター(付属品)
- ③ 作業用 PC
- ④ シリアルクロスケーブル(付属品)
- ⑤ LAN HUB
- ⑥ LAN ケーブル
- ⑦ ディスプレイ
- ⑧ HDMI ケーブル
- ⑨ RCA ケーブル
- ⑩ スピーカ
- ⑪ マイク
- ⑫ ステレオミニプラグケーブル
- ⑬ SD カード
- ⑭ USB メモリ



製品に付属の LCD モジュールおよびカメラモジュールには出荷時、保護フィルムが貼られています。保護フィルムをはがした状態でご使用ください。

5.3. ディップスイッチの設定

SW1 を「表 5.1. ディップスイッチの設定」の設定にしてください。SW1 の機能詳細については「表 4.3. ディップスイッチ(SW1)のスイッチの機能」で確認してください。

表 5.1 ディップスイッチの設定

1	2	3	4	5	6	7	8
OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF

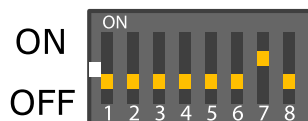



図 5.2 ディップスイッチの設定

5.4. シリアル通信ソフトウェアの設定

シリアル通信ソフトウェアを起動し、シリアルの通信設定を「表 5.2. シリアル通信設定」のように設定してください。

表 5.2 シリアル通信設定

項目	設定
転送レート	115,200bps
データ長	8bit
ストップビット	1bit
パリティ	なし
フロー制御	なし



シリアル通信ソフトウェアの横幅を 80 文字以上にしてください。横幅が 80 文字より小さい場合、コマンド入力中に表示が乱れることがあります。

5.5. インストール済みのソフトウェア

工場出荷状態で内蔵ストレージにインストールされている OS については、次のとおりとなります。

表 5.3 インストールされている OS

OS	バージョン
Debian GNU/Linux	6.0
Android	2.3.7



工場出荷状態の Armadillo-800 EVA は、電源を投入すると Android が起動するよう設定されています。電源を投入後に起動する OS は、ブートローダーの設定(起動 OS 設定)により決定されます。起動する OS を変更するには、次に示す項を参照してください。

- ・「6. Android を起動」
- ・「7. Debian GNU/Linux を起動」

6. Android を起動

6.1. 準備

Android を起動させる場合の起動 OS 設定の変更について説明します。

ブートローダーを保守モードで起動し、起動 OS 設定の変更をおこなうことで起動する OS を変更することができます。Armadillo-800 EVA に電源を投入する前にディップスイッチの起動モード設定(SW1.1)を ON に設定し、Armadillo-800 EVA に電源を投入するとブートローダーが保守モードで起動します。

ブートローダーを保守モードで起動すると、ブートローダーのコマンドプロンプトが表示されます。

```
Hermit-At v3.0.0 (Armadillo-800 EVA) compiled at 22:22:10, Dec 21 2011
hermit>
```

Android を起動するには「図 6.1. 起動 OS 設定の変更手順(Android)」のように起動 OS 設定を変更します。

```
hermit> setbootdevice mmcblk0p4 ❶
hermit> setenv console=ttySC1,115200 noinitrd rootwait root=/dev/mmcblk0p4 init=/init ❷
```

- ❶ Android 用の Linux カーネルを格納しているブートデバイスを設定
- ❷ Linux カーネルパラメータを設定

図 6.1 起動 OS 設定の変更手順(Android)

以上で起動 OS 設定の変更は完了です。boot コマンドで Android を起動することができます。

```
hermit> boot
```



Armadillo-800 EVA の電源を切断し、ディップスイッチの起動モード設定(SW1.1)を OFF に設定すると、電源投入後自動的に Android が起動するようになります。

6.2. 起動

Android を起動します。次のように起動ログがシリアル通信ソフトウェアに表示されます。

```
mmcsd: SD card at address 0x00000001
mmcsd: M8G2FA 1048576KiB
```


```

gendisk: /dev/mmcblk0p4: start=0x000f4280, size=0x001dcdc0
gendisk: Image.bin is found. (4390496 Bytes)
Copying      kernel...done.
Doing console=ttySC1,115200
Doing noinitrd
Doing rootwait
Doing root=/dev/mmcblk0p4
Doing init=/init
Linux version 2.6.35.7 (atmark@atde4) (gcc version 4.4.5 (Debian 4.4.5-8) ) #1 P
REEMPT Wed Dec 21 22:37:47 JST 2011
CPU: ARMv7 Processor [412fc093] revision 3 (ARMv7), cr=10c53c7f
CPU: VIPT nonaliasing data cache, VIPT nonaliasing instruction cache
Machine: Armadillo-800EVA
:
: (割愛)
:
VFS: Mounted root (ext3 filesystem) on device 179:4.
Freeing init memory: 124K
init: cannot open '/initlogo.rle'
init: cannot find '/system/etc/install-recovery.sh', disabling 'flash_recovery'
sh: can't access tty; job control turned off
$ net eth0: attached phy 0 to driver Generic PHY
WM8978 0-001a: Imprecise sampling rate: 48000Hz, consider using PLL
PHY: 0:00 - Link is Up - 100/Full
warning: `zygote' uses 32-bit capabilities (legacy support in use)
request_suspend_state: wakeup (3->0) at 11694431586 (2000-01-01 00:12:25.6519288
34 UTC)
    
```

図 6.2 起動ログ(Android)

6.3. 終了方法

Android では、終了方法が確立されていません。電源を切断して終了させてください。



リムーバブルディスクにデータを書き込んでいる途中で電源を切断した場合、ファイルシステム、及びデータが破損する恐れがあります。

6.4. 基本操作

Android の基本的な操作方法について説明します。

Android の操作の多くは、タッチパネルとユーザースイッチからおこないます。ユーザースイッチの名称と主な機能については、「表 6.1. ユーザースイッチの名称と機能」を参照してください。

表 6.1 ユーザースイッチの名称と機能

ユーザースイッチ	名称	主な機能
SW3	決定ボタン	フォーカスのあるキーを実行します
SW4	戻るボタン	1 つ前の画面に戻ります
SW5	メニューボタン	メニューを開きます
SW6	ホームボタン	ホーム画面に遷移します

6.4.1. ロック画面

Android の起動が完了すると、「図 6.3. ロック画面」が表示されます。

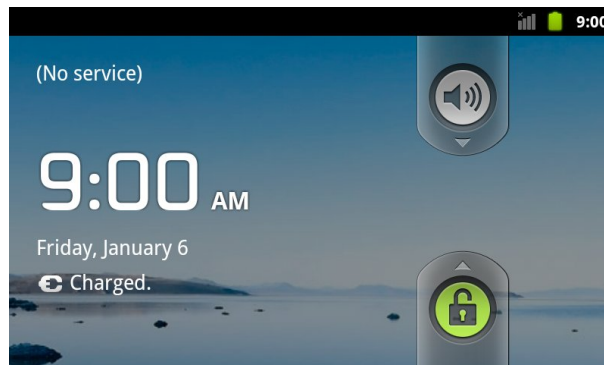


図 6.3 ロック画面

ロックは「メニューボタン」を 2 回押下すると解除できます。Android が起動直後の場合は、「6.4.2. ホーム画面」に示すホーム画面が表示されます。

ロック画面以外の画面が表示されている状態で、何も操作をせずに一定時間が経過すると、ロック画面が表示されます。

6.4.2. ホーム画面

ロック画面以外の任意の画面で「ホームボタン」を押下すると、「図 6.4. ホーム画面」が表示されます。



図 6.4 ホーム画面

「図 6.5. ホーム画面: アプリケーション一覧ボタン」で「アプリケーション一覧ボタン」(図中、赤枠のアイコン)を押下すると、「図 6.6. アプリケーション一覧画面」に示す画面が表示されます。

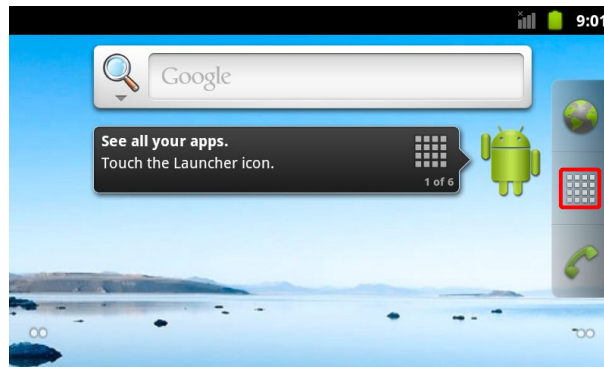


図 6.5 ホーム画面: アプリケーション一覧ボタン

6.4.3. アプリケーション一覧

ホーム画面で「アプリケーション一覧ボタン」を押下すると、「図 6.6. アプリケーション一覧画面」が表示されます。この画面には、インストールされているアプリケーションの一覧が表示されます。

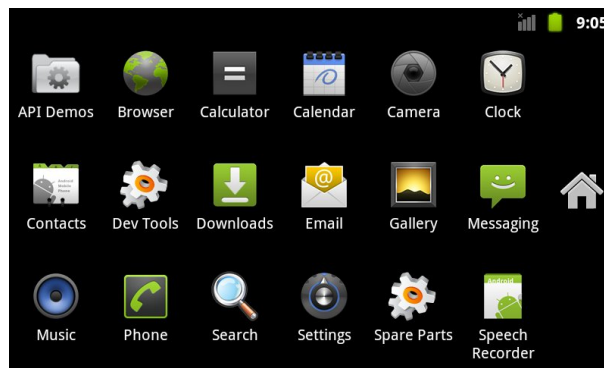


図 6.6 アプリケーション一覧画面

各アプリケーションのアイコンを押下すると、アプリケーションを起動することができます。「API Demos」のアイコンを押下すると、画面描画や音楽再生のデモを行うことができます。



工場出荷状態の Android では、以下のデバイスには対応していません。

- ・ カメラ
- ・ 無線 LAN
- ・ SD カード



Android のソースコード^[1]は、Armadillo サイト (<http://armadillo.atmark-techno.com>)から取得可能です。

^[1]利用制限のあるソフトウェアは含まれていません。

7. Debian GNU/Linux を起動

7.1. 準備

Debian GNU/Linux(以降、単に Debian と表記します)を起動させる場合の起動 OS 設定の変更について説明します。

ブートローダーを保守モードで起動し、起動 OS 設定の変更をおこなうことで起動する OS を変更することができます。Armadillo-800 EVA に電源を投入する前にディップスイッチの起動モード設定(SW1.1)を ON に設定し、Armadillo-800 EVA に電源を投入するとブートローダーが保守モードで起動します。

ブートローダーを保守モードで起動すると、ブートローダーのコマンドプロンプトが表示されます。

```
Hermit-At v3.0.0 (Armadillo-800 EVA) compiled at 22:22:10, Dec 21 2011
hermit>
```

Debian を起動するには「図 7.1. 起動設定の変更手順(Debian)」のように起動設定を変更します。

```
hermit> setbootdevice mmcblk0p2 ❶
hermit> setenv console=ttySC1,115200 noinitrd rootwait root=/dev/mmcblk0p2 ❷
```

- ❶ Debian 用の Linux カーネルを格納しているブートデバイスを設定
- ❷ Linux カーネルパラメータを設定

図 7.1 起動設定の変更手順(Debian)

以上で起動設定の変更は完了です。boot コマンドで Debian を起動することができます。

```
hermit> boot
```



Armadillo-800 EVA の電源を切断し、ディップスイッチの起動モード設定(SW1.1)を OFF に設定すると、電源投入後自動的に Debian が起動するようになります。

7.2. 起動

Debian を起動します。次のように起動ログがシリアル通信ソフトウェアに表示されます。

```
mmcscd: SD card at address 0x00000001
mmcscd: M8G2FA 1048576KiB
```



```
gendisk: /dev/mmcblk0p2: start=0x004ade00, size=0x009ba200
gendisk: Image.bin is found. (4390496 Bytes)
Copying      kernel...done.
Doing console=ttySC1,115200
Doing noinitrd
Doing rootwait
Doing root=/dev/mmcblk0p2
Linux version 2.6.35.7 (atmark@atde4) (gcc version 4.4.5 (Debian 4.4.5-8) ) #1 P
REEMPT Wed Dec 21 22:37:47 JST 2011
CPU: ARMv7 Processor [412fc093] revision 3 (ARMv7), cr=10c53c7f
CPU: VIPT nonaliasing data cache, VIPT nonaliasing instruction cache
Machine: Armadillo-800EVA
:
: (割愛)
:
VFS: Mounted root (ext3 filesystem) on device 179:2.
Freeing init memory: 124K
INIT: version 2.88 booting
Using makefile-style concurrent boot in runlevel S.
Starting the hotplug events dispatcher: udevd.
Synthesizing the initial hotplug events...done.
Waiting for /dev to be fully populated...done.
Activating swap...done.
Checking root file system...fsck from util-linux-ng 2.17.2
/dev/mmcblk0p2: Backing up journal inode block information.

/dev/mmcblk0p2: clean, 11589/637728 files, 88453/1274944 blocks
done.
EXT3-fs (mmcblk0p2): using internal journal
Cleaning up ifupdown...
Setting up networking...
Loading kernel modules...done.
Activating lvm and md swap...done.
Checking file systems...fsck from util-linux-ng 2.17.2
done.
Mounting local filesystems...done.
Activating swapfile swap...done.
Cleaning up temporary files...
Configuring network interfaces...done.
Cleaning up temporary files...
Setting kernel variables ...done.
INIT: Entering runlevel: 2
Using makefile-style concurrent boot in runlevel 2.
Starting enhanced syslogd: rsyslogd.
Starting periodic command scheduler: cron.

Debian GNU/Linux 6.0 debian ttySC1

debian login:
```

図 7.2 起動ログ(Linux)

7.3. ログイン

起動が完了するとログインプロンプトが表示されます。「表 7.1. シリアルコンソールログイン時のユーザ名とパスワード」に示すユーザーでログインすることができます。

表 7.1 シリアルコンソールログイン時のユーザ名とパスワード

ユーザ名	パスワード	権限
root	root	root ユーザ

7.4. 終了方法

安全に終了させる場合は、次のように halt コマンドを実行し、「System halted.」と表示されたのを確認してから電源を切断します。

```
[Armadillo ~]# halt

Broadcast message from root@debian (ttySC1) (Sat Jan 1 04:25:31 2000):

The system is going down for system halt NOW!
INIT: Switching to runlevel: 0
INIT: Sending processes the TERM signal
Using makefile-style concurrent boot in runlevel 0.
Asking all remaining processes to terminate...done.
All processes ended within 1 seconds...done.
Stopping enhanced syslogd: rsyslogd.
Deconfiguring network interfaces...done.
Cleaning up ifupdown...
Saving the system clock.
Deactivating swap...done.
Will now halt.
System halted.
```

図 7.3 終了方法

7.5. 使用方法

7.5.1. 有線 LAN の使用方法

Debian を有効に利用するには、パッケージ管理システム(後述)を利用します。パッケージ管理システムは、インターネット上の Debian サイトにアクセスし、パッケージ情報を集約し、パッケージの検索やインストールなどの一連の制御を提供してくれます。このシステムを利用する為には、インターネットへアクセスできる環境が必要となります。ここでは、インターネットを利用するための 1 つの手段「有線 LAN」を利用可能にします。

工場出荷状態の Debian は有線 LAN が無効になっています。ifup コマンドを使用して有線 LAN を有効化する手順を、「図 7.4. ifup コマンドによる有線 LAN の有効化」に示します。

```
[Armadillo ~]# ifup eth0 ❶
Internet Systems Consortium DHCP Client 4.1.1-P1
Copyright 2004-2010 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

net eth0: attached phy 0 to driver Generic PHY
Listening on LPF/eth0/36:f9:5d:6b:54:ac
Sending on   LPF/eth0/36:f9:5d:6b:54:ac
Sending on   Socket/fallback
```

```

DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 3
PHY: 0:00 - Link is Up - 100/Full
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 7
DHCPOFFER from 172.16.0.1
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 172.16.0.1
bound to 172.16.2.237 -- renewal in 33138 seconds.
[Armadillo ~]# ifconfig eth0 ②
eth0      Link encap:Ethernet  HWaddr 36:f9:5d:6b:54:ac
          inet addr:172.16.2.237  Bcast:172.16.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:876 errors:0 dropped:0 overruns:0 frame:0
          TX packets:109 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:316114 (308.7 KiB)  TX bytes:13236 (12.9 KiB)
          Interrupt:142 DMA chan:ff

```

- ① 有線 LAN インターフェースを有効化します。工場出荷状態の Debian では、DHCP を利用してネットワーク設定が行われるようになっていきます。
- ② 有線 LAN の状態を表示します。

図 7.4 ifup コマンドによる有線 LAN の有効化

ifup コマンドを使用して有効化された有線 LAN を再度無効化するには、ifdown コマンドを使用します。手順を「図 7.5. ifdown コマンドによる有線 LAN の無効化」に示します。

```

[Armadillo ~]# ifdown eth0 ①
Internet Systems Consortium DHCP Client 4.1.1-P1
Copyright 2004-2010 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth0/36:f9:5d:6b:54:ac
Sending on   LPF/eth0/36:f9:5d:6b:54:ac
Sending on   Socket/fallback
DHCPRELEASE on eth0 to 172.16.0.1 port 67
[Armadillo ~]# ifconfig eth0 ②
eth0      Link encap:Ethernet  HWaddr 36:f9:5d:6b:54:ac
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:963 errors:0 dropped:0 overruns:0 frame:0
          TX packets:111 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:332666 (324.8 KiB)  TX bytes:13638 (13.3 KiB)
          Interrupt:142 DMA chan:ff

```

- ① 有線 LAN インターフェースを無効化します。
- ② 有線 LAN の状態を表示します。

図 7.5 ifdown コマンドによる有線 LAN の無効化



ifup および ifdown コマンドは、/etc/network/interfaces に記述された動作をおこないます。工場出荷状態では、DHCP サーバから IP アドレス

を取得するよう設定されています。interfaces を変更することで、静的 IP アドレスの設定や、起動時に自動的に有効化することができます。詳しくは interfaces のマニュアルページを参照してください。

```
[Armadillo ~]# man 5 interfaces
```

7.5.2. 時刻の設定

システムクロックが大幅にずれた状態で、パッケージ管理システムを利用すると警告メッセージが出力される場合があります。事前に本項を参照してシステムクロックを合わせてください。

```
[Armadillo ~]# date 010609002012 ❶  
[Armadillo ~]# date ❷  
Fri Jan 6 09:00:05 UTC 2012  
[Armadillo ~]# hwclock -wu ❸  
[Armadillo ~]# hwclock ❹  
Fri Jan 6 09:00:14 2012 -0.319894 seconds
```

- ❶ システムクロックを 2011 年 1 月 6 日 9 時 00 分^[1]に設定します。
- ❷ システムクロックを確認します。
- ❸ リアルタイムクロックにシステムクロックを設定します。
- ❹ リアルタイムクロックを確認します。

図 7.6 システムクロックの設定

リアルタイムクロックを設定しておくことで、電源を切断/接続した場合でも起動時にリアルタイムクロックの値がシステムクロックに設定されるため、システムクロックを再設定する必要がなくなります。電源を長時間切断する場合は、RTC 外部バックアップインターフェース(CON9)に電池を接続する必要があります。

7.5.3. パッケージの管理方法

工場出荷状態の Debian には動作に必要な最低限のパッケージしかインストールされていません。しかし、パッケージ管理システム APT(Advanced Packaging Tool)を使用することで、簡単にパッケージを追加することができます。以下に APT の基本的な使用方法を示します。



APT は、/etc/apt/sources.list に記述されている場所から利用可能なパッケージのインデックスを取得します。デフォルトの設定では、HTTP サーバが指定されているため、APT を使用するには Armadillo-800 EVA がネットワークに接続されている必要があります。「7.5.1. 有線 LAN の使用方法」または、「8. 無線 LAN の使用方法」を参照してネットワークを有効にしてください。

[1]フォーマットは、MMDDhhmm[[CC]YY][.ss]。詳しくは、ヘルプや man ページをご覧ください。

apt-get update

パッケージインデックスファイルを最新の状態にアップデートします。

引数 無し

使用例

```
[Armadillo ~]# apt-get update
```

apt-get upgrade

現在インストールされている全てのパッケージを最新バージョンにアップグレードします。

引数 無し

使用例

```
[Armadillo ~]# apt-get upgrade
```

apt-get install [パッケージ名]

引数に指定したパッケージをインストールします。すでにインストール済みの場合はアップグレードします。

引数 パッケージ名(複数指定可能)

使用例

```
[Armadillo ~]# apt-get install gcc
```

apt-get remove [パッケージ名]

引数に指定したパッケージをアンインストールします。インストールされていない場合は何もありません。

引数 パッケージ名(複数指定可能)

使用例

```
[Armadillo ~]# apt-get remove apache2
```

apt-cache search [キーワード]

引数に指定したキーワードをパッケージ名または説明文に含むパッケージを検索します。

引数 キーワード(正規表現が使用可能)

使用例

```
[Armadillo ~]# apt-cache search "Bourne Again Shell"
bash-doc - Documentation and examples for the The GNU Bourne Again Shell
bash-static - The GNU Bourne Again Shell (static version)
bash - The GNU Bourne Again Shell
```

7.5.4. カメラを動作させる

カメラモジュールインターフェース(CON1)に搭載されたカメラを動作させてみましょう。カメラから映像を取得してLCDへ表示させるには、GStreamerを利用します。また、GStreamerからLCDへの直接描画はできないため、Xサーバーをインストールして描画させてみます。

Xサーバー(xserver-xorg-core)とGStreamerをインストールするには、次のようにコマンドを実行します。

```
[Armadillo ~]# apt-get install xserver-xorg-core gstreamer0.10
```

図 7.7 XサーバーとGStreamerのインストール

Xサーバーを起動するには次のようにコマンドを実行します。Xサーバーが起動するとLCDに「図 7.9. Xサーバー起動画面」のように表示されます。

```
[Armadillo ~]# X -retro &
```

図 7.8 Xサーバーの起動

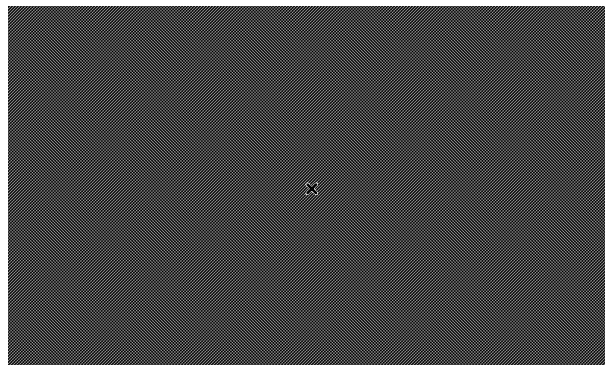


図 7.9 Xサーバー起動画面

GStreamerを使用して、カメラから取得した映像をLCDに出力します。次のようにコマンドを実行します。アプリケーションを停止するには、Ctrl+cを押下してください。

```
[Armadillo ~]# DISPLAY=:0 gst-launch-0.10 v4l2src ! "video/x-raw-yuv,width=800,height=480,format=(fourcc)NV12" ! ffmpegcolorspace ! autovideosink
```



図 7.10 カメラの映像を LCD に表示



LCD にカメラから取得した映像が表示されない場合は、Armadillo-800 EVA を再起動し、再度「図 7.8. X サーバーの起動」および「図 7.10. カメラの映像を LCD に表示」を実行してください。

7.5.5. オーディオを動作させる

オーディオを動作させてみましょう。オーディオの再生・録音にも GStreamer を使用します。インストールされていない場合は、「図 7.7. X サーバーと GStreamer のインストール」を参照してインストールしてください。

また、事前にモノラルマイク入力インターフェース(CON10)にマイクを、ステレオヘッドホン出力インターフェース(CON11)にスピーカやヘッドフォンを接続しておく必要があります。

Armadillo-800 EVA のオーディオ機能は、ALSA デバイスとして実装されています。オーディオデバイスは以下の機能を有しています。

サンプリング周波数 48k, 32k, 16k, 8k Hz

フォーマット Signed 16/24 bit, Little-endian

任意の音声ファイルを再生する手順を「図 7.11. 音声ファイルの再生」に示します。音声ファイルはカレントディレクトリ以下にあることを想定しています。ここで使用する音声ファイルは、サンプリングレートが 48kHz、ステレオの WAV ファイル(sample.wav)を想定しています。

```
[Armadillo ~]# ls
sample.wav
[Armadillo ~]# gst-launch-0.10 filesrc location=sample.wav ! decodebin ! audioconvert !
audioresample ! alsasink
```



図 7.11 音声ファイルの再生

マイクの音声をそのままスピーカに出力する手順を「図 7.12. マイク入力をスピーカに出力」に示します。

```
[Armadillo ~]# gst-launch-0.10 alsasrc ! 'audio/x-raw-int,channels=2,rate=48000,width=16' !
audioresample ! alsasink
```



図 7.12 マイク入力をスピーカに出力



音声ファイルのサンプルには、gnome-audio パッケージに含まれるファイルを利用することができます。パッケージに含まれる音声ファイル(WAV)

は、44.1kHz でサンプリングされているため、48kHz にリサンプリングする必要があります。

```
[Armadillo ~]# apt-get install gnome-audio resample
[Armadillo ~]# cp /usr/share/sounds/login.wav .
[Armadillo ~]# resample -to 48000 login.wav sample.wav
[Armadillo ~]# gst-launch-0.10 filesrc location=sample.wav ! decodebin !
audioconvert ! audioresample ! alsasink
```

↵

図 7.13 gnome-audio パッケージの音声ファイルを利用

8. 無線 LAN の使用方法

本章では、Armadillo-800 EVA に搭載されている無線 LAN モジュール Armadillo-WLAN(AWL13) を、Debian で使用する方法について記載します。

無線通信をおこなうためには以下の作業が必要となりますが、Armadillo-800 EVA に標準でインストールされている Debian では、udev によって 1.および 2.を自動的におこなうように設定することが可能です。

1. カーネルモジュールをロードする
2. ファームウェアをロードする
3. 無線設定をおこなう

8.1. 準備

8.1.1. Debian の起動

Debian を起動させます。起動手順については「7. Debian GNU/Linux を起動」を参照してください。

8.1.2. カーネルモジュール/ファームウェアロードの自動化

udev が AWL13 を検出した場合に、自動的にカーネルモジュールをロードできるようにするために、depmod コマンドを実行します。

```
[Armadillo ~]# ls /lib/modules/`uname -r`  
awl13  
[Armadillo ~]# depmod  
[Armadillo ~]# ls /lib/modules/`uname -r`  
awl13          modules.dep      modules.softdep  
modules.alias  modules.dep.bin  modules.symbols  
modules.alias.bin  modules.devname  modules.symbols.bin
```

depmod コマンド実行後に Armadillo-800 EVA を再起動すると、次回 Debian 起動時に自動的にカーネルモジュールがロードされます。カーネルモジュールがロードされると、udev が自動的にファームウェアのロードをおこないます。起動ログに以下のように表示されるため、カーネルモジュールおよびファームウェアのロードに成功したことが確認できます。

```
awl13: Version 3.0.0 Load.  
awl13: MAC is 00:1d:12:cf:29:65
```



depmod コマンド実行後、Armadillo-800 EVA を再起動せずに、udev に AWL13 を検出させたい場合は、以下のコマンドを実行してください。

```
[Armadillo ~]# udevadm trigger
```



カーネルモジュールがロードされない場合は、「5.3. ディップスイッチの設定」を参照して SW1(ディップスイッチ)の設定を確認してください。CON14(AWL13 モジュールインターフェース)と CON8(SD2 インターフェース)は排他になっており、どちらか一方しか使用することができません。AWL13 を使用する場合は、ディップスイッチの SDH11 設定(SW1.5)を OFF に設定してください。

8.2. 無線設定

インフラストラクチャモードおよびアドホックモードの無線設定について記載します。より詳細な設定方法については、「Armadillo-WLAN(AWL13) ソフトウェアマニュアル」を参照してください。



有線 LAN インターフェースを使用してネットワークに接続している場合、ネットワーク通信時に AWL13 が使用されない場合があります。確実に無線通信をおこないたい場合は、「図 7.5. ifdown コマンドによる有線 LAN の無効化」を参照して有線 LAN インターフェースを無効にしてください。



使用環境によっては、AWL13 が通信しづらい場合があります。そのような環境下では、外付けアンテナを取り付けることで問題が改善する場合があります。外付けアンテナに関する情報は、「Armadillo-WLAN(AWL13) ハードウェアマニュアル」をご覧ください。

8.2.1. 無線設定パラメータ

無線通信をおこなうための基本的な設定パラメータについて記載します。設定には、Wireless Tools^[1]に含まれる iwconfig/iwpriv コマンドを使用します。

SSID

アクセスポイントの識別子です。

ID 長 1～32 文字 (デフォルト: "wifi")

設定例 iwconfig awlan0 essid wifi

取得例 iwconfig awlan0

^[1]標準でインストールされています。

暗号化方式

WPA/WPA2 などの暗号化方式を設定、参照します。

値	設定値	暗号化方式
	"none"	暗号化無効
	"WEP64"	WEP(キー長: 64bits)
	"WEP128"	WEP(キー長: 128bits)
	"WPA-TKIP"	WPA-PSK(TKIP)
	"WPA-CCMP"	WPA-PSK(AES)
	"WPA-AES"	WPA-PSK(AES)
	"WPA2-TKIP"	WPA2-PSK(TKIP)
	"WPA2-CCMP"	WPA2-PSK(AES)
	"WPA2-AES"	WPA2-PSK(AES)
	"WPA-MIX"	WPA-PSK(TKIP, AES)
	"WPA2-MIX"	WPA-PSK(AES)
	"WPA/2-TKIP"	WPA-PSK(TKIP) WPA2-PSK(TKIP)
	"WPA/2-CCMP"	WPA-PSK(AES) WPA2-PSK(AES)
	"WPA/2-AES"	WPA-PSK(AES) WPA2-PSK(AES)
	"WPA/2-MIX"	WPA-PSK(TKIP, AES) WPA2-PSK(TKIP, AES)

設定例 `iwpriv awlan0 set_cryptmode WPA2-AES`

取得例 `iwpriv awlan0 get_cryptmode`

備考 アドホックモードでは、WPA/WPA2 には対応していません。

TKIP や、AES を指定すると、キー生成のために計算が行われるため、コマンド終了までに時間がかかります。

SSID や、事前共有キーを設定した場合にも、キーが再作成されコマンド終了までに時間がかかります。キー生成を 1 回で済ませるために、SSID とパスワードを設定した後に、暗号化方式の設定をすることをお勧めします。

事前共有キー(PSK)

WPA/WPA2 の PSK または、ネットワークパスワードです。

キー長 PSK の場合 64 文字の 16 進数列

パスワードの場合 8~63 文字の文字列

設定例 `iwpriv awlan0 set_psk PreSharedKey` (パスワードの場合)

通信モード

通信形態の種別です。

モード	"Managed"	インフラストラクチャ
	"Ad-Hoc"	アドホック
	"Auto"	無線 LAN 機能オフ (デフォルト)

設定例 `iwconfig awlan0 mode Managed`

取得例 `iwconfig awlan0`

備考 ファームウェアをロードした直後は、通信モードが Auto に設定されているため、無線機能がオフ状態となっています。無線通信を開始させる前には必ず "Managed" または "Ad-Hoc" に設定してください。

WEP キー

パケットを暗号化するとき使用する秘密鍵です。

キー長 WEP-64 ビットの場合は 10 文字、WEP-128 ビットの場合は 26 文字の 16 進数文字

設定例 `iwconfig awlan0 enc 1234567890`

取得例 `iwconfig awlan0`

注意 WEP キーのインデックス番号 1~3 には対応していません。

WEP キーの設定をすると、キー長に応じて**暗号化方式**が適切に設定されます。

チャンネル

使用する周波数帯域です。

チャンネル 1~13 (デフォルト: 11)

設定例 `iwconfig awlan0 channel 11`

8.2.2. インフラストラクチャモード設定例

AWL13 をインフラストラクチャモードに設定し、アクセスポイントに接続する設定例を示します。アクセスポイントの暗号化方式が WPA-PSK、または WPA2-PSK の場合は「8.2.2.1. インフラストラクチャモード: WPA-PSK/WPA2-PSK」を、WEP の場合は「8.2.2.2. インフラストラクチャモード: WEP」を参照してください。

8.2.2.1. インフラストラクチャモード: WPA-PSK/WPA2-PSK

WPA-PSK/WPA2-PSK に設定されたアクセスポイントに接続する例として、設定パラメータを「表 8.1. インフラストラクチャモード: WPA-PSK/WPA2-PSK パラメータ例」に、設定手順を「図 8.1. インフラストラクチャモード: WPA2-PSK(AES)設定手順」に示します。

表 8.1 インフラストラクチャモード: WPA-PSK/WPA2-PSK パラメータ例

項目	設定値
ESSID	myessid
パスフレーズ	mypassphrase
暗号化方式	WPA2-PSK(AES)

```
[Armadillo ~]# iwconfig wlan0 essid myessid
[Armadillo ~]# iwpriv wlan0 set_psk mypassphrase
[Armadillo ~]# iwpriv wlan0 set_cryptmode WPA2-AES
[Armadillo ~]# iwconfig wlan0 mode Managed
[Armadillo ~]# ifconfig wlan0 192.168.0.1 up
```

図 8.1 インフラストラクチャモード: WPA2-PSK(AES)設定手順

8.2.2.2. インフラストラクチャモード: WEP

WEP に設定されたアクセスポイントに接続する例として、設定パラメータを「表 8.2. インフラストラクチャモード: WEP パラメータ例」に、設定手順を「図 8.2. インフラストラクチャモード: WEP 設定手順」に示します。

表 8.2 インフラストラクチャモード: WEP パラメータ例

項目	設定値
ESSID	myessid
WEP キー(WEP-64 ビット)	1234567890

```
[Armadillo ~]# iwconfig wlan0 essid myessid
[Armadillo ~]# iwconfig wlan0 enc 1234567890
[Armadillo ~]# iwconfig wlan0 mode Managed
[Armadillo ~]# ifconfig wlan0 192.168.0.1 up
```

図 8.2 インフラストラクチャモード: WEP 設定手順

8.2.3. アドホックモード設定例

AWL13 をアドホックモードに設定し、アドホック端末接続する設定例を示します。



アドホックモードでは、WPA/WPA2 には対応していません。

8.2.3.1. アドホックモード: WEP-128 ビット

WEP に設定されたアクセスポイントに接続する例として、設定パラメータを「表 8.3. アドホックモード: WEP パラメータ例」に、設定手順を「図 8.3. アドホックモード: WEP 設定手順」に示します。

表 8.3 アドホックモード: WEP パラメータ例

項目	設定値
ESSID	myessid
WEP キー(WEP-128 ビット)	12345678901234567890123456
チャンネル	1

```
[Armadillo ~]# iwconfig awlan0 essid myessid
[Armadillo ~]# iwconfig awlan0 enc 12345678901234567890123456
[Armadillo ~]# iwconfig awlan0 mode Ad-Hoc
[Armadillo ~]# iwconfig rohm0 channel 1
[Armadillo ~]# ifconfig awlan0 192.168.0.1 up
```

図 8.3 アドホックモード: WEP 設定手順

9. SD ブートの活用

本章では、SD カードから直接起動(以降「SD ブート」と表記します)する手順を示します。SD ブートを活用すると、SD カードを取り替えることでシステムイメージを変更することができます。本章に示す手順を実行するためには、容量が 1GByte 以上の SD カードを必要とします。以下では、例として Debian GNU/Linux 6.0(コードネーム squeeze)を SD ブートする手順を示しますが、他の OS を SD ブートすることも可能です。

9.1. ブートディスクを作成

作業用 PC で、SD ブートをおこなうための SD カードを作成します。この SD カードをブートディスクと呼びます。本章で作成するブートディスクの構成については「表 9.1. ブートディスクの構成」を参照してください。

表 9.1 ブートディスクの構成

パーティション番号	ファイルシステム	配置するファイル
1	VFAT	ブートローダーイメージファイル
2	ext3	ルートファイルシステムおよびカーネルイメージファイル



ブートローダーイメージファイルをブートディスクに配置する場合、次の制約があります。

- ・ ブートディスクのパーティション 1 に配置すること
- ・ ルートディレクトリ直下に配置すること
- ・ ファイル名が"sdboot.bin"であること
- ・ パーティション ID が、0xb(Win95 FAT32)であること
- ・ ファイルシステムが FAT32 または FAT16(32MByte 以上)であること

本章に示す手順を実行した場合、問題になることはありませんが、独自のブートディスクを作成する場合は注意してください。



SD ブートは、CON7(SD インターフェース 1)にのみ対応しています。CON8 で SD ブートすることはできません。



ブートディスク作成時のコマンドは、ATDE で実行した場合の例です。Armadillo-800 EVA でも同様の手順でブートディスクを作成することができますがデバイスファイル名など一部異なるため、適宜読み替えてください。

9.1.1. ブートディスクの作成に必要なファイルの取得

「表 9.2. ブートディスクの作成に必要なファイル」に示す、ブートディスクの作成に必要なファイルを取得します。これらのファイルは Armadillo サイト (<http://armadillo.atmark-techno.com>) または、付属 DVD-ROM から取得可能です。

表 9.2 ブートディスクの作成に必要なファイル

ファイル	説明
loader-armadillo8x0-v[version].bin	ブートローダーイメージファイル
debian-squeeze_a800eva_[version].tar.gz	Debian GNU/Linux 6.0 ルートファイルシステムアーカイブ
linux-a800eva-[version].bin	カーネルイメージファイル

9.1.2. パーティションの作成

SD カードに 2 つのプライマリパーティションを作成します。

作業用 PC に SD カードを接続して「図 9.1. パーティション作成手順」のようにパーティションを作成します。

```
[PC ~]# fdisk /dev/sdb ❶
WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Command (m for help): o ❷
Building a new DOS disklabel with disk identifier 0x65314ac5.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Command (m for help): n ❸
Command action
e extended
p primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1790, default 1): ❹
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-1790, default 1790): +128M ❺

Command (m for help): n ❻
Command action
e extended
p primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (62-1790, default 62): ❼
Using default value 62
Last cylinder, +cylinders or +size{K,M,G} (62-1790, default 1790): ❽
```



```
Using default value 1790

Command (m for help): t 9
Partition number (1-4): 1
Hex code (type L to list codes): b
Changed system type of partition 1 to b (W95 FAT32)

Command (m for help): w 10
The partition table has been altered!

Calling ioctl() to re-read partition table.
sdb: p1 p2
Syncing disks.
[PC ~]#
```

- ① SD カードのパーティションテーブル操作を開始します。USB メモリなどを接続している場合は、SD カードのデバイスファイルが sdc や sdd など本実行例と異なる場合があります。
- ② 新しく DOS パーティションテーブルを作成します。
- ③ 新しくプライマリパーティション 1 を作成します。
- ④ 開始シリンダにはデフォルト値(先頭シリンダ)を使用するので、そのまま改行を入力してください。
- ⑤ 最終シリンダは、128MByte 分を指定します。
- ⑥ 新しくプライマリパーティション 2 を作成します。
- ⑦ 開始シリンダにはデフォルト値(プライマリパーティション 1 直後のシリンダ)を使用するので、そのまま改行を入力してください。
- ⑧ 最終シリンダにはデフォルト値(末尾シリンダ)を使用するので、そのまま改行を入力してください。
- ⑨ プライマリパーティション 1 のパーティションタイプを Win95 FAT32(0x0b)に変更します。
- ⑩ 変更を SD カードに書き込みます。

図 9.1 パーティション作成手順

パーティションが作成されていることを確認します。「図 9.2. パーティション確認手順」のように 2 つのパーティションが作成されていることを確認してください。

```
[PC ~]# fdisk -l /dev/sdb
Disk /dev/mmcblk1: 3983 MB, 3983540224 bytes
106 heads, 41 sectors/track, 1790 cylinders
Units = cylinders of 4346 * 512 = 2225152 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x61e03ae6

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            1           61     132532+    b   W95 FAT32
/dev/sdb2            62        1790     3757117   83   Linux
```

図 9.2 パーティション確認手順

9.1.3. ファイルシステムの構築

「9.1.2. パーティションの作成」で作成したそれぞれのパーティションにファイルシステムを構築します。

作業用 PC に SD カードを接続したまま、「図 9.3. ファイルシステム作成手順」のようにファイルシステムを作成します。



ファイルシステムの構築に `mkfs.vfat` コマンド (`dosfstools` パッケージ) を使用します。作業用 PC にインストールされていない場合は、次のようにインストールをおこなってください。

```
[PC ~]# apt-get update
[PC ~]# apt-get install dosfstools
```

```
[PC ~]# mkfs.vfat /dev/sdb1 ❶
mkfs.vfat 3.0.9 (31 Jan 2010)
[PC ~]# mkfs.ext3 /dev/sdb2 ❷
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
235248 inodes, 939279 blocks
46963 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=964689920
29 block groups
32768 blocks per group, 32768 fragments per group
8112 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

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

This filesystem will be automatically checked every 21 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
[PC ~]#
```

- ❶ プライマリパーティション 1 に VFAT ファイルシステムを構築します。
- ❷ プライマリパーティション 2 に ext3 ファイルシステムを構築します。

図 9.3 ファイルシステム作成手順

9.1.4. システムイメージの展開

「9.1.1. ブートディスクの作成に必要なファイルの取得」で取得したファイルを、「9.1.3. ファイルシステムの構築」でファイルシステムを構築した SD カードに展開します。

作業用 PC に SD カードを接続したまま、「図 9.4. システムイメージ展開手順」のようにファイルを展開します。以下のコマンド例では、「9.1.1. ブートディスクの作成に必要なファイルの取得」で取得したファイルはカレントディレクトリ以下にあることを想定しています。

```
[PC ~]# ls
debian-squeeze_a800eva_[version].tar.gz loader-armadillo8x0-v[version].bin
linux-a800eva-[version].bin
[PC ~]# mount -t vfat /dev/sdb1 /mnt
[PC ~]# cp ~/loader-armadillo8x0-v[version].bin /mnt/sdboot.bin ❶
[PC ~]# sync
[PC ~]# umount /mnt
[PC ~]# mount -t ext3 /dev/sdb2 /mnt
[PC ~]# tar zxf ~/debian-squeeze_a800eva_[version].tar.gz -C /mnt ❷
[PC ~]# cp ~/linux-a800eva-[version].bin /mnt/boot/Image.bin ❸
[PC ~]# sync
[PC ~]# umount /mnt
[PC ~]#
```

- ❶ ブートローダーイメージをブートディスクのパーティション 1 にコピーします。ファイル名は "sdboot.bin" にリネームする必要があります。
- ❷ Debian GNU/Linux 6.0 ルートファイルシステムアーカイブをブートディスクのパーティション 2 に展開します。
- ❸ カーネルイメージファイルをブートディスクのパーティション 2 の "/boot/" 以下にコピーします。ファイル名は、"Image.bin" または "linux.bin" にリネームする必要があります。

図 9.4 システムイメージ展開手順

9.1.5. 設定ファイルの編集

「9.1.4. システムイメージの展開」で展開した Debian GNU/Linux 6.0 ルートファイルシステムは、eMMC パーティション 2 をルートファイルシステムとして使用するよう設定されています。「図 9.4. システムイメージ展開手順」に示すように、"/dev/mmcblk0p2" を "/dev/mmcblk1p2" に修正します。

```
[PC ~]# mount -t ext3 /dev/sdb2 /mnt
[PC ~]# vi /mnt/etc/fstab
# <file system> <mount> <type> <options> <dump> <pass>
proc /proc proc defaults 0 0
/dev/mmcblk1p2 / ext3 defaults,errors=remount-ro 0 1
[PC ~]# sync
[PC ~]# umount /mnt
[PC ~]#
```

図 9.5 fstab の編集

9.2. ブートディスクから起動する

「9.1. ブートディスクを作成」で作成したブートディスクから起動します。Armadillo-800 EVA に電源を投入する前に以下の準備をおこないます。

- ・ SD スロット 1 (CON7) にブートディスクを接続します。
- ・ ブートディスクのブートローダーから起動するためにディップスイッチの起動デバイス設定を SDHIO (SW1.2 を ON、SW1.3 を OFF) に設定します。
- ・ ブートローダーを保守モードで起動するためにディップスイッチの起動モード設定を (SW1.1) を ON に設定します。

準備の完了後、電源を投入すると保守モードで起動します。「図 9.6. ブートディスクからの起動」を参照して、ブートディスクのカーネルおよびルートファイルシステムを使用するよう設定後、boot コマンドで起動してください。

```
hermit> setbootdevice mmcblk1p2
hermit> setenv console=ttySC1,115200 noinitrd rootwait root=/dev/mmcblk1p2
hermit> boot
```

図 9.6 ブートディスクからの起動

10. リカバリ手順

本章では、Armadillo-800 EVA に搭載された内蔵ストレージ(eMMC)の内容を工場出荷状態に戻す(リカバリする)方法を示します。本章に示す手順を実行するためには、容量が 1GByte 以上の SD カードを必要とします。

10.1. パーティション構成

Armadillo-800 EVA の内蔵ストレージのパーティション構成を示します。Armadillo-800 EVA の内蔵ストレージは、PC 同様にいくつかのパーティションに分割されています。そのため複数の OS をインストールできるようになっています。

工場出荷状態の内蔵ストレージのパーティション構成を「表 10.1. 内蔵ストレージのパーティション構成」に示します。

表 10.1 内蔵ストレージのパーティション構成

デバイスファイル名	サイズ	パーティションタイプ
/dev/mmcblkOp1	512MByte	0b (Win95 FAT32)
/dev/mmcblkOp2	約 5GByte	83 (Linux)
/dev/mmcblkOp3	1GByte	83 (Linux)
/dev/mmcblkOp4	1GByte	83 (Linux)

現行規格である MMC 4.3 では、eMMC デバイスに最大 2 つのブートパーティションを持つことが出来ます。Armadillo-800 EVA に搭載している eMMC には、2 つのブートパーティションが用意されています。

表 10.2 eMMC のブートパーティション

デバイスファイル名	サイズ
/dev/mmcblk0boot0	約 512kByte
/dev/mmcblk0boot1	約 512kByte

「表 10.1. 内蔵ストレージのパーティション構成」および「表 10.2. eMMC のブートパーティション」に示す通り、内蔵ストレージを 6 つの領域に分けて使用しています。それぞれの工場出荷状態での用途を「表 10.3. 内蔵ストレージ各領域の用途」に示します。

表 10.3 内蔵ストレージ各領域の用途

パーティション	用途
パーティション 1 (/dev/mmcblkOp1)	主に Android の SD カード領域に使用します。VFAT にフォーマットされています。
パーティション 2 (/dev/mmcblkOp2)	Debian GNU/Linux がインストールされています。/boot/直下にはカーネルイメージが格納されています。ext3 にフォーマットされています。
パーティション 3 (/dev/mmcblkOp3)	予約領域 ^[1] です。ext3 にフォーマットされています。
パーティション 4 (/dev/mmcblkOp4)	Android がインストールされています。/boot/直下にはカーネルイメージが格納されています。ext3 にフォーマットされています。
ブートパーティション 1 (/dev/mmcblk0boot0)	ブートローダーイメージが格納されています。

パーティション	用途
ブートパーティション 2 (/dev/mmcbk0boot1)	ブートローダーが使用するパラメータを保存しています。

[1]今後のアップデートで該当パーティション用のシステムイメージを公開する予定です。

10.2. eMMC 全体をリカバリする

eMMC 全体をリカバリする方法を示します。

10.2.1. リカバリディスクの作成

作業用 PC で、リカバリをおこなうための SD カードを作成します。この SD カードをリカバリディスクと呼びます。



リカバリディスク作成時のコマンドは、ATDE で実行した場合の例です。Armadillo-800 EVA でも同様の手順でリカバリディスクを作成することができますがデバイスファイル名など一部異なるため、適宜読み替えてください。

10.2.1.1. リカバリディスクの作成に必要なファイルの取得

「表 10.4. リカバリディスクの作成に必要なファイル」に示す、リカバリディスクの作成に必要なファイルを取得します。これらのファイルはアットマークテクノ ユーザーズサイト (<https://users.atmark-techno.com>) または、付属 DVD-ROM から取得可能です。



アットマークテクノ ユーザーズサイトからファイルを取得するためには、製品本体をご購入の上で「アットマークテクノ ユーザーズサイト」から「購入製品登録」をおこなう必要があります。

表 10.4 リカバリディスクの作成に必要なファイル

ファイル	説明
recovery-image_a800eva_[version].tar.gz	eMMC に書き込むためのイメージファイルなどが格納されています
recovery-system_a800eva_[version].tar.gz	リカバリを実行するプログラムなどが格納されています

10.2.1.2. パーティションの作成

SD カードに 2 つのプライマリパーティションを作成します。

作業用 PC に SD カードを接続して「図 10.1. パーティション作成手順」のようにパーティションを作成します。

```
[PC ~]# fdisk /dev/sdb ①
```

```
WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').
```

```

Command (m for help): o ②
Building a new DOS disklabel with disk identifier 0x65314ac5.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Command (m for help): n ③
Command action
e extended
p primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1790, default 1): ④
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-1790, default 1790): +512M ⑤

Command (m for help): n ⑥
Command action
e extended
p primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (243-1790, default 243): ⑦
Using default value 243
Last cylinder, +cylinders or +size{K,M,G} (243-1790, default 1790): ⑧
Using default value 1790

Command (m for help): t ⑨
Partition number (1-4): 1
Hex code (type L to list codes): b
Changed system type of partition 1 to b (W95 FAT32)

Command (m for help): w ⑩
The partition table has been altered!

Calling ioctl() to re-read partition table.
sdb: p1 p2
Syncing disks.
[PC ~]#

```

- ① SD カードのパーティショニングを開始します。USB メモリなどを接続している場合は、SD カードのデバイスファイルが sdc や sdd など本実行例と異なる場合があります。
- ② 新しく DOS パーティションテーブルを作成します。
- ③ 新しくプライマリパーティション 1 を作成します。
- ④ 開始シリンダにはデフォルト値(先頭シリンダ)を使用するので、そのまま改行を入力してください。
- ⑤ 最終シリンダは、512MByte 分を指定します。

- ⑥ 新しくプライマリパーティション 2 を作成します。
- ⑦ 開始シリンダにはデフォルト値(プライマリパーティション 1 直後のシリンダ)を使用するので、そのまま改行を入力してください。
- ⑧ 最終シリンダにはデフォルト値(末尾シリンダ)を使用するので、そのまま改行を入力してください。
- ⑨ プライマリパーティション 1 のパーティションタイプを Win95 FAT32(0x0b)に変更します。
- ⑩ 変更を SD カードに書き込みます。

図 10.1 パーティション作成手順

パーティションが作成されていることを確認します。「図 10.2. パーティション確認手順」のように 2 つのパーティションが作成されていることを確認してください。

```
[PC ~]# fdisk -l /dev/sdb
Disk /dev/sdb: 3983 MB, 3983540224 bytes
106 heads, 41 sectors/track, 1790 cylinders
Units = cylinders of 4346 * 512 = 2225152 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x65314ac5
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	242	525845+	b	W95 FAT32
/dev/sdb2		243	1790	3363804	83	Linux

図 10.2 パーティション確認手順

10.2.1.3. ファイルシステムの構築

「10.2.1.2. パーティションの作成」で作成したそれぞれのパーティションにファイルシステムを構築します。

作業用 PC に SD カードを接続したまま、「図 10.3. ファイルシステム作成手順」のようにファイルシステムを作成します。



ファイルシステムの構築に dosfstools を使用します。作業用 PC にインストールされていない場合は、次のようにインストールをおこなってください。

```
[PC ~]# apt-get update
[PC ~]# apt-get install dosfstools
```

```
[PC ~]# mkfs.vfat /dev/sdb1 ①
mkfs.vfat 3.0.9 (31 Jan 2010)
[PC ~]# mkfs.ext3 /dev/sdb2 ②
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
```



```

Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
210496 inodes, 840951 blocks
42047 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=864026624
26 block groups
32768 blocks per group, 32768 fragments per group
8096 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200

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

This filesystem will be automatically checked every 27 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[PC ~]#

```

- ❶ プライマリパーティション 1 に VFAT ファイルシステムを構築します。
- ❷ プライマリパーティション 2 に ext3 ファイルシステムを構築します。

図 10.3 ファイルシステム作成手順

10.2.1.4. リカバリイメージの展開

「10.2.1.1. リカバリディスクの作成に必要なファイルの取得」で取得したファイルを、「10.2.1.3. ファイルシステムの構築」でファイルシステムを構築した SD カードに展開します。

作業用 PC に SD カードを接続したまま、「図 10.4. ファイル展開手順」のようにファイルを展開します。以下のコマンド例では、「10.2.1.1. リカバリディスクの作成に必要なファイルの取得」で取得したファイルはカレントディレクトリ以下にあることを想定しています。

```

[PC ~]# ls
recovery-image_a800eva_[version].tar.gz  recovery-system_a800eva_[version].tar.gz
[PC ~]# mount -t vfat /dev/sdb1 /mnt
[PC ~]# cd /mnt
[PC /mnt]# tar xzf ~/recovery-image_a800eva_[version].tar.gz
[PC /mnt]# cd
[PC ~]# sync
[PC ~]# umount /mnt

[PC ~]# mount -t ext3 /dev/sdb2 /mnt
[PC ~]# cd /mnt
[PC /mnt]# tar xzf ~/recovery-system_a800eva_[version].tar.gz
[PC /mnt]# cd
[PC ~]# sync
[PC ~]# umount /mnt
[PC ~]#

```

図 10.4 ファイル展開手順

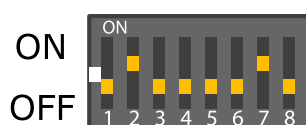
10.2.2. リカバリを実行する



以下に示す手順を実行すると、Armadillo-800 EVA に搭載された eMMC の全ての内容が工場出荷状態になります。一部のパーティションのみを工場出荷状態に戻したい場合は、「10.3. eMMC の特定ルートファイルシステムをリカバリする」または、「10.4. eMMC のブートローダーをリカバリする」を参照してください。

「10.2.1. リカバリディスクの作成」で作成したリカバリディスクを使用して、リカバリを実行します。Armadillo-800 EVA に電源を投入する前に以下の準備をおこないます。

- ・ ディップスイッチの SW1.2 と SW1.7 を ON に、その他を全て OFF に設定します。



- ・ SD スロット 1(CON7)にリカバリディスクを接続します。

準備の完了後、SW3 を押しながら電源を投入するとリカバリが開始されます。リカバリの進捗具合は、LED の点灯パターンで確認できます。

表 10.5 リカバリ進捗と LED の対応

LED3	LED4	LED5	LED6	リカバリ進捗
点滅	消灯	消灯	消灯	eMMC のユーザーパーティションをフォーマット中
消灯	点滅	消灯	消灯	eMMC のブートパーティションをフォーマット中
点滅	点灯	点灯	点灯	eMMC のパーティション 1 にシステムを構築中
点灯	点滅	点灯	点灯	eMMC のパーティション 2 にシステムを構築中
点灯	点灯	点滅	点灯	eMMC のパーティション 3 にシステムを構築中
点灯	点灯	点灯	点滅	eMMC のパーティション 4 にシステムを構築中
点灯	点灯	点灯	点灯	リカバリが正常に完了
点滅	点滅	点滅	点滅	リカバリが異常終了

全ての LED が点灯するとリカバリは完了です。全ての LED が点滅した場合、エラーが発生したことを示しています。手順を見直して再度リカバリを実行してください。リカバリ後、eMMC から起動するには、ディップスイッチの起動デバイス設定を eMMC(SW1.2 を OFF、SW1.3 を OFF)に設定し、SD カードを抜いた後電源を投入してください。

10.3. eMMC の特定ルートファイルシステムをリカバリする

eMMC の特定ルートファイルシステムのみをリカバリする方法を紹介します。「表 10.6. リカバリ対象のルートファイルシステム」をリカバリの対象とします。

表 10.6 リカバリ対象のルートファイルシステム

ルートファイルシステム	パーティション番号	ファイルシステム
Debian GNU/Linux	2	ext3
Android	4	ext3

事前に「9. SD ブートの活用」を参照して、Armadillo-800 EVA が SD ブートしている必要があります。

10.3.1. Debian GNU/Linux をリカバリする

eMMC のパーティション 2 に、Debian GNU/Linux をリカバリする方法を示します。

10.3.1.1. Debian GNU/Linux のリカバリに必要なファイルの取得

「表 10.7. Debian GNU/Linux のリカバリに必要なファイル」に示す、Debian GNU/Linux のリカバリに必要なファイルを取得します。これらのファイルは Armadillo サイト (<http://armadillo.atmark-techno.com>) または、付属 DVD-ROM から取得可能です。

表 10.7 Debian GNU/Linux のリカバリに必要なファイル

ファイル	説明
debian-squeeze_a800eva_[version].tar.gz	Debian GNU/Linux 6.0 のルートファイルシステムアーカイブ
linux-a800eva-[version].bin	カーネルイメージファイル

10.3.1.2. Debian GNU/Linux のリカバリを実行する

「図 10.5. Debian GNU/Linux のリカバリ手順」のように Debian GNU/Linux のリカバリを実行します。「10.3.1.1. Debian GNU/Linux のリカバリに必要なファイルの取得」で取得したファイルはカレントディレクトリ以下にあることを想定しています。

```
[armadillo ~]# ls
debian-squeeze_a800eva_[version].tar.gz  linux-a800eva-[version].bin
[armadillo ~]# mkfs.ext3 /dev/mmcblk0p2 ❶
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
318864 inodes, 1274944 blocks
63747 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=1308622848
39 block groups
32768 blocks per group, 32768 fragments per group
8176 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 24 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
[armadillo ~]# mount -t ext3 /dev/mmcblk0p2 /mnt ❷
[armadillo ~]# tar xzf ~/debian-squeeze_a800eva_[version].tar.gz -C /mnt ❸
[armadillo ~]# cp ~/linux-a800eva-[version].bin /mnt/boot/Image.bin ❹
[armadillo ~]# sync
[armadillo ~]# umount /mnt
[armadillo ~]#
```

❶ ext3 ファイルシステムを構築します。

- ② マウントします。
- ③ ルートファイルシステムアーカイブを展開します。
- ④ カーネルイメージファイルを/mnt/boot/以下にコピーします。ファイル名は"Image.bin"または"linux.bin"にリネームする必要があります。


図 10.5 Debian GNU/Linux のリカバリ手順

10.3.2. Android をリカバリする

eMMC のパーティション 4 に、Android をリカバリする方法を示します。

10.3.2.1. Android のリカバリに必要なファイルの取得

「表 10.8. Android のリカバリに必要なファイル」に示す、Android のリカバリに必要なファイルを取得します。Android のルートファイルシステムアーカイブはアットマークテクノ ユーザーズサイト (<https://users.atmark-techno.com>) または付属 DVD-ROM から、カーネルイメージファイルは Armadillo サイト (<http://armadillo.atmark-techno.com>) または、付属 DVD-ROM から取得可能です。



アットマークテクノ ユーザーズサイトからファイルを取得するためには、製品本体をご購入の上で「アットマークテクノ ユーザーズサイト」から「購入製品登録」をおこなう必要があります。

表 10.8 Android のリカバリに必要なファイル

ファイル	説明
android-2.3.7_a800eva_[version].tar.gz	Android 2.3.7 のルートファイルシステムアーカイブ
linux-a800eva-[version].bin	カーネルイメージファイル

10.3.2.2. Android のリカバリを実行する

「図 10.6. Android のリカバリ手順」のように Android のリカバリを実行します。「10.3.2.1. Android のリカバリに必要なファイルの取得」で取得したファイルはカレントディレクトリ以下にあることを想定しています。

```
[armadillo ~]# ls
android-2.3.7_a800eva_[version].tar.gz  linux-a800eva-[version].bin
[armadillo ~]# mkfs.ext3 /dev/mmcblk0p4 ①
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
61056 inodes, 244152 blocks
12207 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=251658240
8 block groups
32768 blocks per group, 32768 fragments per group
7632 inodes per group
```

```

Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 23 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
[armadillo ~]# mount -t ext3 /dev/mmcblk0p4 /mnt ❷
[armadillo ~]# tar zxf ~/android-2.3.7_a800eva_[version].tar.gz -C /mnt ❸
[armadillo ~]# mkdir /mnt/boot
[armadillo ~]# cp ~/linux-a800eva-[version].bin /mnt/boot/Image.bin ❹
[armadillo ~]# sync
[armadillo ~]# umount /mnt
[armadillo ~]#

```

- ❶ ext3 ファイルシステムを構築します。
- ❷ マウントします。
- ❸ ルートファイルシステムアーカイブを展開します。
- ❹ カーネルイメージファイルを/mnt/boot/以下にコピーします。ファイル名は"Image.bin"または"linux.bin"にリネームする必要があります。

図 10.6 Android のリカバリ手順

10.4. eMMC のブートローダーをリカバリする

eMMC のブートパーティション 0 に、ブートローダーをリカバリする方法を示します。事前に「9. SD ブートの活用」を参照して、Armadillo-800 EVA が SD ブートしている必要があります。

10.4.1. ブートローダーのリカバリに必要なファイルの取得

「表 10.9. ブートローダーのリカバリに必要なファイル」に示す、ブートローダーのリカバリに必要なファイルを取得します。これらのファイルは Armadillo サイト (<http://armadillo.atmark-techno.com>) または、付属 DVD-ROM から取得可能です。

表 10.9 ブートローダーのリカバリに必要なファイル

ファイル	説明
loader-armadillo8x0-v[version].bin	ブートローダーイメージファイル

10.4.2. ブートローダーのリカバリを実行する

「図 10.7. ブートローダーのリカバリ手順」のようにブートローダーのリカバリを実行します。「10.4.1. ブートローダーのリカバリに必要なファイルの取得」で取得したファイルはカレントディレクトリ以下にあることを想定しています。

```

[armadillo ~]# ls
loader-armadillo8x0-v[version].bin
[armadillo ~]# echo 0 > /sys/block/mmcblk0boot0/force_ro ❶
[armadillo ~]# cat ~/loader-armadillo8x0-v[version].bin > /dev/mmcblk0boot0 ❷

```

```
[armadillo ~]# sync  
[armadillo ~]#
```

- ① ブートパーティションのソフトライトプロテクトを解除します。
- ② ブートローダーイメージを書き込みます。

図 10.7 ブートローダーのリカバリ手順

11. 開発環境の準備

本章では、Armadillo-800 EVA のソフトウェア開発を行うための開発環境を作業用 PC に構築する方法について説明します。

作業用 PC が Debian GNU/Linux 6.0(コードネーム squeeze)^[1]の場合、クロス開発ツールをインストールすることで Armadillo-800 EVA のソフトウェア開発環境を構築することができます。クロス開発ツールをインストールする方法は、「11.2. クロス開発ツールのインストール方法」を参照してください。

作業用 PC が Windows または、Debian GNU/Linux 6.0(コードネーム squeeze)以外の Linux の場合、「ATDE(Atmark Techno Development Environment)」を使用することで Armadillo-800 EVA のソフトウェア開発環境を構築することができます。ATDE を使用する方法は、「11.1. ATDE の使用方法」を参照してください。

11.1. ATDE の使用方法

作業用 PC で、ATDE を使用する方法について記載します。

ATDE は、Armadillo の開発環境を提供する VMware 仮想マシンのデータイメージです。Armadillo-800 EVA のソフトウェア開発が可能な ATDE4 は、Debian GNU/Linux 6.0(コードネーム squeeze)をベースに、クロス開発ツールやその他の必要なツールが事前にインストールされています。

VMware の取得や、インストール方法については、VMware 社 Web ページ (<http://www.vmware.com/>)を参照してください。



VMware は、非商用利用限定で無償のものから、商用利用可能な有償のものまで複数の製品があります。製品ごとに異なるライセンス、エンドユーザー使用許諾契約書(EULA)が存在するため、十分に確認した上で利用目的に合う製品をご利用ください。

11.1.1. ATDE の取得

「表 11.1. ATDE4 の種類」に示す ATDE4 のアーカイブは Armadillo サイト (<http://armadillo.atmark-techno.com>)または、付属 DVD-ROM から取得可能です。

表 11.1 ATDE4 の種類

ATDE4 アーカイブ	ベースの Debian GNU/Linux
atde4-[version]-amd64.zip	64-bit PC(「amd64」)アーキテクチャ用 Debian GNU/Linux 6.0
atde4-[version]-i386.zip	32-bit PC(「i386」)アーキテクチャ用 Debian GNU/Linux 6.0

^[1]他の Linux でも開発はできますが、本書記載事項すべてが全く同じように動作するわけではありません。各作業はお使いの Linux 環境に合わせた形で自己責任のもと行ってください。

11.1.2. ATDE の起動

ATDE4 のアーカイブを展開したディレクトリに存在する atde4.vmx が、ATDE4 の仮想マシン構成ファイルです。このファイルを VMware 上で開くと、ATDE4 を起動することができます。ATDE4 にログイン可能なユーザーを、「表 11.2. ユーザ名とパスワード」に示します^[2]。

表 11.2 ユーザ名とパスワード

ユーザ名	パスワード	権限
atmark	atmark	一般ユーザー
root	root	特権ユーザー



以下の VMware で、ATDE4 が起動できることを確認しています。

- ・ VMware Player 3.1.5
- ・ VMware Workstation 7.1.5
- ・ VMware Workstation 8.0.1



作業用 PC の動作環境(ハードウェア、VMware、ATDE4 の種類など)により、ATDE4 が正常に動作しない可能性があります。VMware 社 Web ページ(<http://www.vmware.com/>)から、使用している VMware のドキュメントなどを参照して動作環境を確認してください。

11.2. クロス開発ツールのインストール方法

Debian GNU/Linux 6.0(コードネーム squeeze)が動作している作業用 PC に、armel アーキテクチャ用のクロス開発ツールを、Debian パッケージでインストールする方法について記載します。

11.2.1. クロス開発ツール Debian パッケージの取得

クロス開発ツールの Debian パッケージは Armadillo サイト(<http://armadillo.atmark-techno.com>)または、付属 DVD-ROM から取得可能です。

11.2.2. クロス開発ツール Debian パッケージのインストール

作業用 PC に、Debian パッケージをインストールします。以下では、「11.2.1. クロス開発ツール Debian パッケージの取得」で取得したクロス開発ツールの Debian パッケージが全て~/に配置されている場合を前提としています。

64-bit PC(「amd64」)アーキテクチャ用 Debian GNU/Linux 6.0 を使用している場合は「図 11.1. 64-bit PC 用クロス開発ツール Debian パッケージインストールコマンド」を、32-bit PC(「i386」)アーキテクチャ用 Debian GNU/Linux 6.0 を使用している場合は「図 11.2. 32-bit PC 用クロス開発ツール Debian パッケージインストールコマンド」を参照してください。

^[2]特権ユーザーで GUI ログインをおこなうことはできません。


```
[PC ~]$ sudo dpkg --install *_amd64.deb *_all.deb
```

図 11.1 64-bit PC 用クロス開発ツール Debian パッケージインストールコマンド

```
[PC ~]$ sudo dpkg --install *_i386.deb *_all.deb
```

図 11.2 32-bit PC 用クロス開発ツール Debian パッケージインストールコマンド



作業用 PC に既に同一ターゲットのアーキテクチャ用クロス開発環境がインストールされている場合、クロス開発環境をインストールする前に必ず既存のクロス開発環境をアンインストールするようにしてください。

12. カーネルのビルド

本章では、Linux カーネルのソースコードを作業用 PC でクロスビルドし、カーネルイメージファイルを作成する方法について記載します。

事前に「11. 開発環境の準備」を参照して、作業用 PC に開発環境が構築されている必要があります。

12.1. ソースアーカイブの取得

Linux カーネルのソースアーカイブ `linux-2.6.35-a800eva-[version].tar.gz` を取得します。ソースアーカイブは Armadillo サイト (<http://armadillo.atmark-techno.com>) または、付属 DVD-ROM から取得可能です。

12.2. ソースアーカイブの展開

取得したソースアーカイブを展開します。「図 12.1. ソースアーカイブの展開」のように作業してください。

```
[PC ~]# ls
linux-2.6.35-a800eva-[version].tar.gz
[PC ~]# tar zxvf linux-2.6.35-a800eva-[version].tar.gz
linux-2.6.35-a800eva-[version] linux-2.6.35-a800eva-[version].tar.gz
```

図 12.1 ソースアーカイブの展開

12.3. ビルド

展開したソースアーカイブをビルドして、イメージファイルを作成します。「図 12.2. カーネルのビルド」のように作業してください。

```
❶ [PC ~]$ cd linux-2.6.35-a800eva-[version]
❷ [PC ~/linux-2.6.35-a800eva-[version]]$ make ARCH=arm armadillo800eva_android_defconfig
❸ [PC ~/linux-2.6.35-a800eva-[version]]$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-
❹ [PC ~/linux-2.6.35-a800eva-[version]]$ ls arch/arm/boot/Image
Image
```

- ❶ ソースコードディレクトリに移動します
- ❷ デフォルトコンフィギュレーションを適用します
- ❸ ソースコードをビルドします
- ❹ イメージファイルが作成できたことを確認します

図 12.2 カーネルのビルド

13. SGX 用カーネルモジュールのビルド

本章では、SGX DDK(Driver Development Kit)のソースコードを作業用 PC でクロスビルドし、カーネルモジュールを作成する方法について記載します。



Kernel features など、カーネルの構成に大きく影響のあるコンフィグレーションを変更したカーネルイメージを Android で利用した場合、稀に SGX 用カーネルモジュールが動作しなくなる場合があります。カスタマイズをおこなったカーネルを使用する場合は、そのカーネルにあわせた SGX 用カーネルモジュールを作成する必要があります。

事前に 「11. 開発環境の準備」 を参照して、作業用 PC に開発環境が構築されている必要があります。

13.1. ソースアーカイブ取得

SGX DDK のソースコードアーカイブ `eurasia_km-[version].tar.gz` を取得します。ソースアーカイブは付属 DVD-ROM から取得可能です。

13.2. カーネルの準備

SGX 用カーネルモジュールをビルドするためには、ビルド済みのカーネルソースコードが必要です。「12. カーネルのビルド」 を参考にしてカーネルソースコードをビルドします。

13.3. ビルド

「13.1. ソースアーカイブ取得」 で取得したソースアーカイブを展開し、ビルド環境にあわせたパラメータを設定してビルドをおこないます。「13.2. カーネルの準備」 でビルドしたカーネルソースコードが `~/linux-2.6.35-a800eva` に配置されている場合、以下のコマンドを実行します。



SGX 用カーネルモジュールのビルドには `dos2unix` を使用します。作業用 PC にインストールされていない場合は、次のようにインストールをおこなってください。

```
[PC ~]# apt-get update
[PC ~]# apt-get install dos2unix
```

```

❶[PC ~]$ tar zxvf eurasia_km-[version].tar.gz
❷[PC ~]$ cd eurasia_km-[version]/eurasiacon/build/linux2/r8a7740_android
❸[PC ~/eurasia_km-[version]/eurasiacon/build/linux2/r8a7740_android]$ make ARCH=arm
KERNEL_CROSS_COMPILE=arm-linux-gnueabi- CROSS_COMPILE=arm-linux-gnueabi-
TARGET_PRODUCT=armadillo-800eva HAL_VARIANT=armadillo-800eva KERNELDIR=~/.linux-2.6.35-a800eva
(省略)
MODPOST 3 modules
CC      /home/atmark/eurasia_km-[version]/eurasiacon/binary2_r8a7740_android_release/target/
kbuild/bc_example.mod.o
LD [M]  /home/atmark/eurasia_km-[version]/eurasiacon/binary2_r8a7740_android_release/target/
kbuild/bc_example.ko
CC      /home/atmark/eurasia_km-[version]/eurasiacon/binary2_r8a7740_android_release/target/
kbuild/pvrsvkm.mod.o
LD [M]  /home/atmark/eurasia_km-[version]/eurasiacon/binary2_r8a7740_android_release/target/
kbuild/pvrsvkm.ko
CC      /home/atmark/eurasia_km-[version]/eurasiacon/binary2_r8a7740_android_release/target/
kbuild/shmobilelfb.mod.o
LD [M]  /home/atmark/eurasia_km-[version]/eurasiacon/binary2_r8a7740_android_release/target/
kbuild/shmobilelfb.ko
❹[PC ~/eurasia_km-[version]/eurasiacon/build/linux2/r8a7740_android]$ ls ../../../../
binary2_r8a7740_android_release/target/kbuild/*ko
../../../../binary2_r8a7740_android_release/target/kbuild/bc_example.ko
../../../../binary2_r8a7740_android_release/target/kbuild/pvrsvkm.ko
../../../../binary2_r8a7740_android_release/target/kbuild/shmobilelfb.ko

```

- ❶ SGX DDK のソースコードアーカイブを展開します
- ❷ ソースコードディレクトリに移動します
- ❸ ビルドを実行します
- ❹ 3つのカーネルモジュールが作成されます

図 13.1 SGX 用カーネルモジュールのビルド

13.4. インストール

SGX 用カーネルモジュールを、内蔵ストレージの Android システムにインストールします。事前に「9. SD ブートの活用」を参照して、Armadillo-800 EVA が SD ブートしている必要があります。

「13.3. ビルド」で作成した SGX 用カーネルモジュールはカレントディレクトリ以下にあることを想定しています。

```
[PC ~]$ ls
bc_example.ko  pvrsrvkm.ko  shmobilelfb.ko
❶ [PC ~]$ mount /dev/mmcblk0p4 /mnt
❷ [PC ~]$ cp *.ko /mnt/lib/modules/
[PC ~]$ sync
[PC ~]$ umount /mnt
```

- ❶ eMMC パーティション 4 をマウントします。
- ❷ eMMC パーティション 4 の /lib/modules/ 以下にカーネルモジュールをコピーします。

図 13.2 SGX 用カーネルモジュールのインストール

14. 無線 LAN(AWL13) 用 Linux デバイスドライバーのビルド

本章では、無線 LAN モジュール Armadillo-WLAN(AWL13)用のデバイスドライバーソースコードを作業用 PC でクロスビルドし、カーネルモジュールを作成する方法について記載します。

事前に「11. 開発環境の準備」を参照して、作業用 PC に開発環境が構築されている必要があります。

14.1. ソースアーカイブ取得

AWL13 用のデバイスドライバーソースアーカイブ `awl13-[version].tar.gz` を取得します。ソースアーカイブは Armadillo サイト (<http://armadillo.atmark-techno.com>) または、付属 DVD-ROM から取得可能です。

14.2. カーネルの準備

AWL13 用のデバイスドライバーソースコードをビルドするためには、ビルド済みのカーネルソースコードが必要です。「12. カーネルのビルド」を参照してカーネルソースコードをビルドします。以下のコンフィグレーションが有効になっている必要があります。^[1]

- ・ CONFIG_MMC
- ・ CONFIG_SYSFS
- ・ CONFIG_WIRELESS_EXT

14.3. ビルド

「14.1. ソースアーカイブ取得」で取得したソースアーカイブを展開し、ビルド環境にあわせたパラメータを設定しビルドをおこないます。「14.2. カーネルの準備」でビルドしたカーネルソースコードが `~/linux-2.6.35-a800eva-[version]` に配置されている場合、以下のコマンドを実行します。

^[1]Armadillo-800 EVA のデフォルトのコンフィグレーションでは有効になっています。

```
❶[PC ~]$ tar zxvf awl13-[version].tar.gz
❷[PC ~]$ cd awl13-[version]
❸[PC ~/awl13-[version]]$ AWL13_DEVICE=SDIO make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-
KERNELDIR=~/linux-2.6.35-a800eva-[version]
make -C ../linux-2.6.35-a800eva-[version]/ M=/home/atmark/aerial-awl13 modules
make[1]: ディレクトリ `~/home/atmark/linux-2.6.35-a800eva-[version]` に入ります
CC [M] /home/atmark/aerial-awl13/src/awl13_log.o
CC [M] /home/atmark/aerial-awl13/src/awl13_sdiodrv.o
CC [M] /home/atmark/aerial-awl13/src/awl13_ioctl.o
CC [M] /home/atmark/aerial-awl13/src/awl13_fw.o
CC [M] /home/atmark/aerial-awl13/src/awl13_sysfs.o
CC [M] /home/atmark/aerial-awl13/src/awl13_wid.o
LD [M] /home/atmark/aerial-awl13/src/awl13_sdio.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/atmark/aerial-awl13/src/awl13_sdio.mod.o
LD [M] /home/atmark/aerial-awl13/src/awl13_sdio.ko
make[1]: ディレクトリ `~/home/atmark/linux-2.6.35-a800eva-[version]` から出ます
❹[PC ~/awl13-[version]]$ ls src/awl13_sdio.ko
awl13_sdio.ko
```



- ❶ AWL13 のソースコードアーカイブを展開します
- ❷ ソースコードディレクトリに移動します
- ❸ ビルドを実行します
- ❹ カーネルモジュール `awl13_sdio.ko` が作成されます

図 14.1 AWL13 ドライバーのビルド



AWL13_DEVICE には、ホストインターフェースを指定します。Armadillo-800 EVA では、AWL13 が SDIO 起動モードで動作するように設定されています。

15. ブートローダーのビルド

本章では、ブートローダーのソースコードを作業用 PC でクロスビルドし、ブートローダーイメージファイルを作成する方法について記載します。

事前に「11. 開発環境の準備」を参照して、作業用 PC に開発環境が構築されている必要があります。

15.1. ソースアーカイブの取得

ブートローダーのソースアーカイブ `hermit-at-[version]-source.tar.gz` を取得します。ソースアーカイブは Armadillo サイト (<http://armadillo.atmark-techno.com>) または、付属 DVD-ROM から取得可能です。



Armadillo-800 EVA のブートローダーには、Hermit-At v3 系を採用しています。

15.2. ソースアーカイブの展開

取得したソースアーカイブを展開します。「図 15.1. ソースアーカイブの展開」のように作業してください。

```
[PC ~]# ls
hermit-at-[version]-source.tar.gz
[PC ~]# tar zxvf hermit-at-[version]-source.tar.gz
hermit-at-[version] hermit-at-[version]-source.tar.gz
```

図 15.1 ソースアーカイブの展開

15.3. ビルド

展開したソースアーカイブをビルドして、イメージファイルを作成します。「図 15.2. ブートローダーのビルド」のように作業してください。


```
❶ [PC ~]$ cd hermit-at-[version]
❷ [PC ~/hermit-at-[version]]$ make armadillo800eva_defconfig
❸ [PC ~/hermit-at-[version]]$ make
❹ [PC ~/hermit-at-[version]]$ ls src/target/armadillo8x0/loader-armadillo8x0-v[version].bin
src/target/armadillo8x0/loader-armadillo8x0-v[version].bin
```

- ❶ ソースコードディレクトリに移動します
- ❷ デフォルトコンフィギュレーションを適用します
- ❸ ソースコードをビルドします
- ❹ イメージファイルが作成できたことを確認します

図 15.2 ブートローダーのビルド

16. JTAG ICE を利用する

本章では ARM のデバッグをおこなうために、JTAG ICE を接続する方法について説明します。

16.1. 準備

16.1.1. JTAG ケーブルの接続

JTAG ICE のケーブルを、ARM JTAG インターフェース(ARM 標準 20 ピンコネクタ)に接続します。信号配列などの詳細な情報については、「18.6. CON6(ARM JTAG インターフェース)」を参照してください。

16.1.2. ディップスイッチの設定

ディップスイッチの JTAG 設定を ARM に設定します。「図 16.1. ディップスイッチの JTAG 設定 (ARM)」を参照して、JTAG 設定 1(SW1.7)を ON に、JTAG 設定 2(SW1.8)を OFF に設定してください。

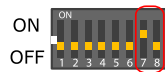


図 16.1 ディップスイッチの JTAG 設定(ARM)

16.2. 接続確認

「16.1. 準備」に従って設定されている場合に、CPU は以下のように見えます。

項目	値
デバイス ID	0x4BA00477
コマンド長	4

16.3. 各種デバッグへの対応について

お使いのデバッグが Armadillo-800 EVA に対応しているか等の情報につきましては、各メーカーにお問い合わせください。

17. Linux ドライバー 一覧

Armadillo-800 EVA を制御する Linux ドライバーのソースコードのパスや制御可能なデバイスを示します。Linux カーネルのソースアーカイブ `linux-2.6.35-a800eva-[version].tar.gz` は Armadillo サイト (<http://armadillo.atmark-techno.com>) または、付属 DVD-ROM から取得可能です。

ボード固有設定

ソースコード `arch/arm/mach-shmobile/board-armadillo800eva.c`

タイマードライバー

ソースコード `arch/arm/mach-shmobile/sh_cmt.c`

割り込みコントローラードライバー

ソースコード `arch/arm/mach-shmobile/intc-sh7740.c`

PWM ドライバー

ソースコード `arch/arm/mach-shmobile/tpu-pwm.c`

UART ドライバー

ソースコード `drivers/serial/sh-sci.c`

デバイスファイル `/dev/ttySC1 (CON22)`

Ethernet ドライバー

ソースコード `drivers/net/sh_eth.c`

デバイス `eth0 (CON23)`

MMC ホストドライバ

ソースコード `drivers/mmc/host/sh_mmcif.c`

デバイス `/dev/mmcblk0`

SD ホストドライバ

ソースコード `drivers/mmc/host/sh_mobile_sdhi.c`

デバイス `/dev/mmcblk1`

`/dev/mmcblk2`

USB ホストドライバ

ソースコード `drivers/usb/host/ehci-rmobile.c`

`drivers/usb/host/ohci-rmobile.c`

フレームバッファドライバ

ソースコード drivers/video/rmobile_lcdcfb.c
デバイス /dev/fb0 (CON17)
 /dev/fb1 (CON3 or CON14)

キャプチャーインターフェースドライバ

ソースコード drivers/media/video/sh_mobile_ceu_camera.c

カメラドライバ

ソースコード drivers/media/video/mt9t112.c
デバイス /dev/video0

リアルタイムクロックドライバ

ソースコード drivers/rtc/rtc-s35390a.c
デバイス /dev/rtc0

タッチスクリーンドライバ

ソースコード drivers/input/touchscreen/st1232.c
デバイス /dev/input/event1

ボタンスイッチキーボードドライバ

ソースコード drivers/input/keyboard/gpio_keys.c
デバイス /dev/input/event0

LCD バックライトドライバ

ソースコード drivers/video/backlight/pwm_glc.c
デバイス /sys/class/backlight/pwm-backlight.0

LED ドライバ

ソースコード drivers/leds/leds-gpio.c
デバイス /sys/class/leds/LED3 (LED3)
 /sys/class/leds/LED4 (LED4)
 /sys/class/leds/LED5 (LED5)
 /sys/class/leds/LED6 (LED6)

オーディオドライバ

ソースコード sound/soc/sh/fsi-wm8978.c

	sound/soc/sh/fsi-hdmi.c
	sound/soc/sh/fsi.c
	sound/soc/codecs/wm8978.c
デバイス	hw:0 (CON10/11/12/13)
	hw:1 (CON3)

18. インターフェース仕様

18.1. CON1(カメラモジュールインターフェース)

CON1 はカメラモジュール接続用インターフェースです。DF30FC-24DS-0.4V(ヒロセ電機)を実装しています。

- ・ 対応カメラモジュール： DCB-NSB55QFMRB-S-05-02(CRESYN)

表 18.1 CON1 信号配列

ピン番号	信号名	I/O	機能
1	CAM0_2V8_D6	In	データバス(bit6)、R-Mobile A1 の VIO_D6_0 ピンに接続
2	CAM0_2V8_D7	In	データバス(bit7)、R-Mobile A1 の VIO_D7_0 ピンに接続
3	NC	-	未接続
4	NC	-	未接続
5	CAM0_2V8_STANDBY	Out	スタンバイ信号、R-Mobile A1 の D22 ピンに接続
6	VCC_CAM2.8V	Power	電源(VCC_CAM2.8V)
7	VCC_CAM2.8V	Power	電源(VCC_CAM2.8V)
8	DGND	Power	電源(DGND)
9	CAM0_2V8_PCLK	In	クロック信号入力、R-Mobile A1 の VIO_CLK_0 ピンに接続
10	DGND	Power	電源(DGND)
11	CAM0_2V8_MCLK	Out	クロック信号出力、R-Mobile A1 の VIO_CKO_0 ピンに接続
12	CAM0_2V8_SCL	Out	I2C クロック、R-Mobile A1 の I2C_SCL_0 ピンに接続
13	CAM0_2V8_D3	In	データバス(bit3)、R-Mobile A1 の VIO_D3_0 ピンに接続
14	CAM0_2V8_D2	In	データバス(bit2)、R-Mobile A1 の VIO_D2_0 ピンに接続
15	CAM0_2V8_D1	In	データバス(bit1)、R-Mobile A1 の VIO_D1_0 ピンに接続
16	CAM0_2V8_D0	In	データバス(bit0)、R-Mobile A1 の VIO_D0_0 ピンに接続
17	CAM0_2V8_D4	In	データバス(bit4)、R-Mobile A1 の VIO_D4_0 ピンに接続
18	CAM0_2V8_D5	In	データバス(bit5)、R-Mobile A1 の VIO_D5_0 ピンに接続
19	CAM0_2V8_SDA	In/Out	I2C データ、R-Mobile A1 の I2C_SDA_0 ピンに接続
20	GPIO4_FLASH	In/Out	LED1 に接続(High: 点灯、Low: 消灯)
21	VCC_CAM1.8V	Power	電源(VCC_CAM1.8V)
22	CAM0_2V8_RESET#	Out	リセット信号、R-Mobile A1 の D23 ピンに接続
23	CAM0_2V8_VSYNC	In	VSYNC 信号、R-Mobile A1 の VIO_VD_0 ピンに接続
24	CAM0_2V8_HSYNC	In	HSYNC 信号、R-Mobile A1 の VIO_HD_0 ピンに接続

18.2. CON2(拡張バスインターフェース)

CON2 はバス拡張用のインターフェースです。コネクタは実装していません。

- ・ 基板側コネクタ例： XG4C-6431(オムロン)
- ・ 対向コネクタ例： XG4M-6431(オムロン)

表 18.2 CON2 信号配列

ピン番号	信号名	I/O	機能
1	VCC_3.3V	Power	電源(VCC_3.3V)
2	DGND	Power	電源(DGND)
3	A1_D0	In/Out	データバス(bit0)、R-Mobile A1 の D0 ピンに接続
4	A1_D1	In/Out	データバス(bit1)、R-Mobile A1 の D1 ピンに接続

ピン番号	信号名	I/O	機能
5	A1_D2	In/Out	データバス(bit2)、R-Mobile A1 の D2 ピンに接続
6	A1_D3	In/Out	データバス(bit3)、R-Mobile A1 の D3 ピンに接続
7	A1_D4	In/Out	データバス(bit4)、R-Mobile A1 の D4 ピンに接続
8	A1_D5	In/Out	データバス(bit5)、R-Mobile A1 の D5 ピンに接続
9	A1_D6	In/Out	データバス(bit6)、R-Mobile A1 の D6 ピンに接続
10	A1_D7	In/Out	データバス(bit7)、R-Mobile A1 の D7 ピンに接続
11	A1_D8	In/Out	データバス(bit8)、R-Mobile A1 の D8 ピンに接続
12	A1_D9	In/Out	データバス(bit9)、R-Mobile A1 の D9 ピンに接続
13	A1_D10	In/Out	データバス(bit10)、R-Mobile A1 の D10 ピンに接続
14	A1_D11	In/Out	データバス(bit11)、R-Mobile A1 の D11 ピンに接続
15	A1_D12	In/Out	データバス(bit12)、R-Mobile A1 の D12 ピンに接続
16	A1_D13	In/Out	データバス(bit13)、R-Mobile A1 の D13 ピンに接続
17	A1_D14	In/Out	データバス(bit14)、R-Mobile A1 の D14 ピンに接続
18	A1_D15	In/Out	データバス(bit15)、R-Mobile A1 の D15 ピンに接続
19	EXT_MMC_RST_B	In/Out	外部リセット信号、R-Mobile A1 の MEMC_NWE ピンに接続
20	A1_CS6A_B	In/Out	チップセレクト 6A、R-Mobile A1 の CS6A#ピンに接続
21	A1_CS5B_B	In/Out	チップセレクト 5B、R-Mobile A1 の CS5B#ピンに接続
22	A1_CS5A_B	In/Out	チップセレクト 5A、R-Mobile A1 の CS5A#ピンに接続
23	A1_CS0_B	In/Out	チップセレクト 0、R-Mobile A1 の CS0#ピンに接続
24	A1_RESETOUTS_B	In/Out	リセット信号、R-Mobile A1 の RESETOUTS#ピンに接続
25	A1_RD_B	In/Out	リード信号、R-Mobile A1 の RD#ピンに接続
26	A1_RDWR	In/Out	リードライト信号、R-Mobile A1 の RDWR ピンに接続
27	A1_WE1_B	In/Out	ライトイネーブル信号 1、R-Mobile A1 の WE1#ピンに接続
28	A1_WE0_B	In/Out	ライトイネーブル信号 0、R-Mobile A1 の WE0#ピンに接続
29	A1_DACK0_B	In/Out	DACK 信号 0、R-Mobile A1 の DACK_0 ピンに接続
30	A1_DREQ0_B	In/Out	DREQ 信号 0、R-Mobile A1 の DREQ_0 ピンに接続
31	A1_ICIORD	In/Out	ICIORD 信号、R-Mobile A1 の WE2#ピンに接続
32	A1_ICIOWR	In/Out	ICIOWR 信号、R-Mobile A1 の WE3#ピンに接続
33	A1_IOIS16_B	In/Out	IOIS16 信号、R-Mobile A1 の IOIS16#ピンに接続
34	A1_BSCMD	In/Out	BSCMD 信号、R-Mobile A1 の BSCMD ピンに接続
35	A1_CKO	In/Out	CKO 信号、R-Mobile A1 の CKO ピンに接続
36	DGND	Power	電源(DGND)
37	A1_A0	In/Out	アドレスバス(bit0)、R-Mobile A1 の A0 ピンに接続
38	A1_A1	In/Out	アドレスバス(bit1)、R-Mobile A1 の A1 ピンに接続
39	A1_A2	In/Out	アドレスバス(bit2)、R-Mobile A1 の A2 ピンに接続
40	A1_A3	In/Out	アドレスバス(bit3)、R-Mobile A1 の A3 ピンに接続
41	A1_A4	In/Out	アドレスバス(bit4)、R-Mobile A1 の A4 ピンに接続
42	A1_A5	In/Out	アドレスバス(bit5)、R-Mobile A1 の A5 ピンに接続
43	A1_A6	In/Out	アドレスバス(bit6)、R-Mobile A1 の A6 ピンに接続
44	A1_A7	In/Out	アドレスバス(bit7)、R-Mobile A1 の A7 ピンに接続
45	A1_A8	In/Out	アドレスバス(bit8)、R-Mobile A1 の A8 ピンに接続
46	A1_A9	In/Out	アドレスバス(bit9)、R-Mobile A1 の A9 ピンに接続
47	A1_A10	In/Out	アドレスバス(bit10)、R-Mobile A1 の A10 ピンに接続
48	A1_A11	In/Out	アドレスバス(bit11)、R-Mobile A1 の A11 ピンに接続
49	A1_A12	In/Out	アドレスバス(bit12)、R-Mobile A1 の A12 ピンに接続
50	A1_A13	In/Out	アドレスバス(bit13)、R-Mobile A1 の A13 ピンに接続
51	A1_A14	In/Out	アドレスバス(bit14)、R-Mobile A1 の A14 ピンに接続
52	A1_A15	In/Out	アドレスバス(bit15)、R-Mobile A1 の A15 ピンに接続
53	A1_A16	In/Out	アドレスバス(bit16)、R-Mobile A1 の A16 ピンに接続
54	A1_A17	In/Out	アドレスバス(bit17)、R-Mobile A1 の A17 ピンに接続
55	A1_A18	In/Out	アドレスバス(bit18)、R-Mobile A1 の A18 ピンに接続
56	A1_A19	In/Out	アドレスバス(bit19)、R-Mobile A1 の A19 ピンに接続
57	A1_A20	In/Out	アドレスバス(bit20)、R-Mobile A1 の A20 ピンに接続
58	A1_A21	In/Out	アドレスバス(bit21)、R-Mobile A1 の A21 ピンに接続

ピン番号	信号名	I/O	機能
59	A1_A22	In/Out	アドレスバス(bit22)、R-Mobile A1 の A22 ピンに接続
60	A1_A23	In/Out	アドレスバス(bit23)、R-Mobile A1 の A23 ピンに接続
61	A1_A24	In/Out	アドレスバス(bit24)、R-Mobile A1 の A24 ピンに接続
62	A1_A25	In/Out	アドレスバス(bit25)、R-Mobile A1 の A25 ピンに接続
63	VCC_3.3V	Power	電源(VCC_3.3V)
64	DGND	Power	電源(DGND)

18.3. CON3(HDMI 出カインターフェース)

CON3 は HDMI 出カインターフェースです。Type-A コネクタを実装しています。

表 18.3 CON3 信号配列

ピン番号	信号名	I/O	機能
1	TX2_P	Out	TMDS データ 2+, R-Mobile A1 の TODP2 ピンに接続
2	-	-	TMDS データ 2 シールド、DGND に接続
3	TX2_N	Out	TMDS データ 2-, R-Mobile A1 の TODN2 ピンに接続
4	TX1_P	Out	TMDS データ 1+, R-Mobile A1 の TODP1 ピンに接続
5	-	-	TMDS データ 1 シールド、DGND に接続
6	TX1_N	Out	TMDS データ 1-, R-Mobile A1 の TODN1 ピンに接続
7	TX0_P	Out	TMDS データ 0+, R-Mobile A1 の TODP0 ピンに接続
8	-	-	TMDS データ 0 シールド、DGND に接続
9	TX0_N	Out	TMDS データ 0-, R-Mobile A1 の TODN0 ピンに接続
10	TMC_P	Out	TMDS クロック+, R-Mobile A1 の TOCP ピンに接続
11	-	-	TMDS クロックシールド DGND に接続
12	TXC_N	Out	TMDS クロック-, R-Mobile A1 の TOCN ピンに接続
13	DDC_CEC	In/Out	CEC、R-Mobile A1 の HDMI_CEC ピンに接続
14	NC	-	未接続
15	DDC_SCL	In/Out	SCL、R-Mobile A1 の HDMI_SCL ピンに接続
16	DDC_SDA	In/Out	SDA、R-Mobile A1 の HDMI_SDA ピンに接続
17	DGND	Power	電源(DGND)
18	DDC_5.0V	Power	電源(DDC_5.0V)
19	DDC_HPD	In	ホットプラグ検出、R-Mobile A1 の HDMI_HPD ピンに接続

18.4. CON4(コンポジットビデオ出カインターフェース)

CON4 はコンポジットビデオ出カインターフェースです。RCA ジャック黄色を実装しています。


表 18.4 CON4 信号配列

ピン番号	信号名	I/O	機能
1	YCVBSOUT	Out	コンポジットビデオ出力、R-Mobile A1 の YCVBSOUT ピンに接続
2	DGNC	Power	電源(DGND)

18.5. CON5(H-UDI JTAG インターフェース)

CON5 は H-UDI JTAG インターフェースです。コネクタは実装していません。

- ・ 基板側コネクタ例： XG4C-1431(オムロン)
- ・ 対向コネクタ例： XG4M-1431(オムロン)



CON5 と CON6 の信号は共通となっており、CON6 と同時に利用することはできません。


表 18.5 CON5 信号配列

ピン番号	信号名	I/O	機能
1	A1_TCK	In	TCK 信号、R-Mobile A1 の TCK ピンに接続
2	A1_TRST_B	In	TRST 信号、R-Mobile A1 の TRST#ピンに接続
3	A1_TDO	Out	TDO 信号、R-Mobile A1 の TDO ピンに接続
4	A1_EDBGREQ/ ASEBRK_B	In	ASEBRK 信号、R-Mobile A1 の EDBGREQ ピンに接続
5	A1_TMS	In	TMS 信号、R-Mobile A1 の TMS ピンに接続
6	A1_TDI	In	TDI 信号、R-Mobile A1 の TDI ピンに接続
7	RESETP_B	In	リセット信号(Low: リセット状態、High: リセット解除)、R-Mobile A1 の RESETP#ピンに接続
8	NC	-	未接続
9	DGND	Power	電源(DGND)
10	DGND	Power	電源(DGND)
11	VCC_3.3V	Power	電源(VCC_3.3V)
12	DGND	Power	電源(DGND)
13	DGND	Power	電源(DGND)
14	DGND	Power	電源(DGND)

18.6. CON6(ARM JTAG インターフェース)

CON6 は ARM JTAG インターフェースです。XG4C-2031(オムロン)を実装しています。

- ・ 対向コネクタ例： XG4M-2031(オムロン)



CON5 と CON6 の信号は共通となっており、CON5 と同時に利用することはできません。

表 18.6 CON6 信号配列

ピン番号	信号名	I/O	機能
1	VCC_3.3V	Power	電源(VCC_3.3V)
2	VCC_3.3V	Power	電源(VCC_3.3V)
3	A1_TRST_B	In	TRST 信号、R-Mobile A1 の TRST#ピンに接続
4	DGND	Power	電源(DGND)
5	A1_TDI	In	TDI 信号、R-Mobile A1 の TDI ピンに接続
6	DGND	Power	電源(DGND)
7	A1_TMS	In	TMS 信号、R-Mobile A1 の TMS ピンに接続
8	DGND	Power	電源(DGND)
9	A1_TCK	In	TCK 信号、R-Mobile A1 の TCK ピンに接続
10	DGND	Power	電源(DGND)
11	A1_RTCK	Out	RTCK 信号、R-Mobile A1 の RTCK ピンに接続
12	DGND	Power	電源(DGND)
13	A1_TDO	Out	TDO 信号、R-Mobile A1 の TDO ピンに接続

ピン番号	信号名	I/O	機能
14	DGND	Power	電源(DGND)
15	SRST_IN_B_OD	In	リセット信号(Low: リセット状態、High: リセット解除)、R-Mobile A1 の RESETP#ピンに接続
16	DGND	Power	電源(DGND)
17	A1_EDBGREQ/ ASEBRK_B	In	DBGREQ 信号、R-Mobile A1 の EDBGREQ ピンに接続
18	DGND	Power	電源(DGND)
19	NC	-	R45 に抵抗を実装すると CON6 の 17 ピンと接続
20	DGND	Power	電源(DGND)

18.7. CON7(SD インターフェース 1)


CON7 は SD スロットです。CIM-H81N(ミツミ電機)を実装しています。

表 18.7 CON7 信号配列

ピン番号	信号名	I/O	機能
1	A1_SDHIO_D3	In/Out	SD データ(bit3)、R-Mobile A1 の SDHID3_0 ピンに接続
2	A1_SDHIO_CMD	In/Out	SD コマンド、R-Mobile A1 の SDHICMD_0 ピンに接続
3	DGND	Power	電源(DGND)
4	VCC_SD0	Power	電源(VCC_SD0)
5	A1_SDHIO_CLK	Out	SD クロック、R-Mobile A1 の SDHICLK_0 ピンに接続
6	DGND	Power	電源(DGND)
7	A1_SDHIO_D0	In/Out	SD データ(bit0)、R-Mobile A1 の SDHID0_0 ピンに接続
8	A1_SDHIO_D1	In/Out	SD データ(bit1)、R-Mobile AC2 の SDHID1_0 ピンに接続
9	A1_SDHIO_D2	In/Out	SD データ(bit2)、R-Mobile A1 の SDHID2_0 ピンに接続
10	A1_SDHIO_CD	In	SD カード検出(Low: カード挿入、High: カード抜去)、R-Mobile A1 の D20 ピンに接続
11	FG	Power	電源(DGND)
12	A1_SDHIO_WP	In	ライトプロテクト検出(Low: 書き込み可能、High: 書き込み不可能)、R-Mobile A1 の SDHIWP_0 ピンに接続
13	DGND	Power	電源(DGND)
14	DGND	Power	電源(DGND)
15	DGND	Power	電源(DGND)
18	DGND	Power	電源(DGND)

18.8. CON8(SD インターフェース 2)

CON8 は SD スロットです。CIM-H81N(ミツミ電機)を実装しています。



CON8 と CON14 の信号は共通となっており、同時に使用することはできません。

CON8 と CON14 で使用する各信号をどちらのインターフェースに接続させるかは、ディップスイッチ(SW1)の SDHI1 設定で選択できるようになっています。詳しくは、「表 4.3. ディップスイッチ(SW1)のスイッチの機能」または、「18.25. SW1(機能選択スイッチ)」を参照してください。

表 18.8 CON8 信号配列

ピン番号	信号名	I/O	機能
1	SDSLOT2_SDHI1_D3	In/Out	SD データ(bit3)、R-Mobile A1 の MEMC_AD11 ピンに接続

ピン番号	信号名	I/O	機能
2	SDSLOT2_SDHI1_CMD	In/Out	SD コマンド、R-Mobile A1 の MEMC_CS0 ピンに接続
3	DGND	Power	電源(DGND)
4	VCC_SD0	Power	電源(VCC_SD0)
5	SDSLOT2_SDHI1_CLK	In/Out	SD クロック、R-Mobile A1 の MEMC_INT ピンに接続
6	DGND	Power	電源(DGND)
7	SDSLOT2_SDHI1_D0	In/Out	SD データ(bit0)、R-Mobile A1 の MEMC_AD8 ピンに接続
8	SDSLOT2_SDHI1_D1	In/Out	SD データ(bit1)、R-Mobile A1 の MEMC_AD9 に接続
9	SDSLOT2_SDHI1_D2	In/Out	SD データ(bit2)、R-Mobile A1 の MEMC_AD10 ピンに接続
10	SDSLOT2_SDHI1_CD	In	SD カード検出(Low: カード挿入、High: カード抜去)、R-Mobile A1 の MEMC_AD12 ピンに接続
11	FG	Power	電源(DGND)
12	SDSLOT2_SDHI1_WP	In	ライトプロテクト検出(Low: 書き込み可能、High: 書き込み不可能)、R-Mobile A1 の MEMC_AD13 ピンに接続
13	DGND	Power	電源(DGND)
14	DGND	Power	電源(DGND)
15	DGND	Power	電源(DGND)
18	DGND	Power	電源(DGND)

18.9. CON9(RTC 外部バックアップインターフェース)

CON9 は RTC の外部バックアップインターフェースです。SMTU2032-LF.TR(Renata)を実装しています。

- ・ 対応電池： CR2032

表 18.9 CON9 信号配列

ピン番号	信号名	I/O	機能
1	BAT	Power	RTC のバックアップ用電源入力
2	DGND	Power	電源(DGND)

18.10. CON10~CON13(オーディオインターフェース)

CON10~CON13 はオーディオインターフェースです。オーディオ CODEC を経由して R-Mobile A1 に接続されています。

18.10.1. CON10(モノラルマイク入力インターフェース)

モノラルマイク入力インターフェースです。φ3.5mm ミニジャックを実装しています。

表 18.10 CON10 信号配列

ピン番号	信号名	I/O	機能
1	MIC_IN	In	マイク入力信号
2	AGND_DA	Power	電源(AGND_DA)
3	AGND_DA	Power	電源(AGND_DA)
5	AGND_DA	Power	電源(AGND_DA)

18.10.2. CON11(ステレオヘッドホン出力インターフェース)

CON11 はステレオヘッドホン出力インターフェースです。φ3.5mm ミニジャックを実装しています。

表 18.11 CON11 信号配列

ピン番号	信号名	I/O	機能
1	HP_OUT_L	Out	ヘッドホン出力(左チャンネル)
2	GPIO	In	ヘッドホン検知
3	HP_OUT_R	Out	ヘッドホン出力(右チャンネル)
5	AGND_DA	Power	電源(AGND_DA)

18.10.3. CON12(オーディオ ライン出力(L)インターフェース)

CON12 はオーディオ ライン出力(L)インターフェースです。RCA ジャック白色を実装しています。

表 18.12 CON12 信号配列

ピン番号	信号名	I/O	機能
1	LINE_OUT_L	Out	ライン出力(左チャンネル)
2	AGND_DA	Power	電源(AGND_DA)

18.10.4. CON13(オーディオ ライン出力(R)インターフェース)

CON13 はオーディオ ライン出力(R)インターフェースです。RCA ジャック赤色を実装しています。

表 18.13 CON13 信号配列

ピン番号	信号名	I/O	機能
1	LINE_OUT_R	Out	ライン出力(右チャンネル)
2	AGND_DA	Power	電源(AGND_DA)

18.11. CON14(AWL13 モジュールインターフェース)

CON14 は AWL13 モジュールインターフェースです。AXK6F34347YG(パナソニック電工)を実装しています。AWL13 の制御信号が接続されており、SDIO 起動モードで動作するように設定されています。^[1]



CON8 と CON14 の信号は共通となっており、同時に使用することはできません。

CON8 と CON14 で使用する各信号をどちらのインターフェースに接続させるかは、ディップスイッチ(SW1)の SDHI1 設定で選択できるようになっています。詳しくは、「表 4.3. ディップスイッチ(SW1)のスイッチの機能」または、「18.25. SW1(機能選択スイッチ)」を参照してください。

表 18.14 CON14 信号配列

ピン番号	信号名	I/O	機能
1	SDDATA1	In/Out	SDIO データ(bit1)、R-Mobile A1 の MEMC_AD9 ピンに接続
2	SDDATA0	In/Out	SDIO データ(bit0)、R-Mobile A1 の MEMC_AD8 ピンに接続
3	DGND	Power	電源(DGND)
4	DGND	Power	電源(DGND)
5	NC	-	未接続
6	NC	-	未接続

^[1]AWL13 モジュールの詳細につきましては、「Armadillo-WLAN(AWL13)ハードウェアマニュアル」を参照してください。

ピン番号	信号名	I/O	機能
7	SDCLK	Out	SDIO クロック、R-Mobile A1 の MEMC_INT ピンに接続
8	VCC_SD1	Power	電源(VCC_SD1)
9	NC	-	未接続
10	SDCMD	Out	SDIO コマンド、R-Mobile A1 の MEMC_CS0 ピンに接続
11	SDDATA3	In/Out	SDIO データ(bit3)、R-Mobile A1 の MEMC_AD11 ピンに接続
12	SDDATA2	In/Out	SDIO データ(bit2)、R-Mobile A1 の MEMC_AD10 ピンに接続
13	NC	-	未接続
14	NC	-	未接続
15	BOOT_SEL1	Out	SDIO 起動モード選択 (BOOT_SEL1: Low、BOOT_SEL0: High、HOST_SEL: High)
16	BOOT_SEL0	Out	
17	HOST_SEL	Out	
18	NC	-	未接続
19	NC	-	未接続
20	NC	-	未接続
21	NC	-	未接続
22	NC	-	未接続
23	NC	-	未接続
24	NC	-	未接続
25	NC	-	未接続
26	NC	-	未接続
27	NC	-	未接続
28	HRST	In	VCC_SD1 に接続
29	NC	-	未接続
30	NC	-	未接続
31	NC	-	未接続
32	NC	-	未接続
33	NC	-	未接続
34	NC	-	未接続

18.12. CON15(拡張インターフェース)

CON15 は拡張インターフェースです。コネクタは実装していません。

- ・ 基板側コネクタ例： XG4C-6031(オムロン)
- ・ 対向コネクタ例： XG4M-6031(オムロン)


表 18.15 CON15 信号配列

ピン番号	信号名	I/O	機能
1	VCC_5V	Power	電源(VCC_5V)
2	VCC_5V	Power	電源(VCC_5V)
3	VCC_3.3V	Power	電源(VCC_3.3V)
4	VCC_3.3V	Power	電源(VCC_3.3V)
5	DGND	Power	電源(DGND)
6	DGND	Power	電源(DGND)
7	A1_VIO0_CK0	In/Out	R-Mobile A1 の VIO_CK0_0 ピンに接続
8	A1_VIO0_CLK	In/Out	R-Mobile A1 の VIO_CLK_0 ピンに接続
9	A1_VIO0_HD	In/Out	R-Mobile A1 の VIO_HD_0 ピンに接続
10	A1_VIO0_VD	In/Out	R-Mobile A1 の VIO_VD_0 ピンに接続
11	A1_VIO0_FIELD	In/Out	R-Mobile A1 の VIO_FIELD_0 ピンに接続
12	A1_VIO0_D0	In/Out	R-Mobile A1 の VIO_D0_0 ピンに接続
13	A1_VIO0_D1	In/Out	R-Mobile A1 の VIO_D1_0 ピンに接続
14	A1_VIO0_D2	In/Out	R-Mobile A1 の VIO_D2_0 ピンに接続
15	A1_VIO0_D3	In/Out	R-Mobile A1 の VIO_D3_0 ピンに接続

ピン番号	信号名	I/O	機能
16	A1_VIO0_D4	In/Out	R-Mobile A1 の VIO_D4_0 ピンに接続
17	A1_VIO0_D5	In/Out	R-Mobile A1 の VIO_D5_0 ピンに接続
18	A1_VIO0_D6	In/Out	R-Mobile A1 の VIO_D6_0 ピンに接続
19	A1_VIO0_D7	In/Out	R-Mobile A1 の VIO_D7_0 ピンに接続
20	A1_VIO0_STANDBY	In/Out	R-Mobile A1 の D22 ピンに接続
21	A1_VIO0_RST_B	In/Out	R-Mobile A1 の D23 ピンに接続
22	DGND	Power	電源(DGND)
23	A1_VIO1_CLK	In/Out	R-Mobile A1 の SCIFA_RXD_0 ピンに接続
24	A1_VIO1_HD	In/Out	R-Mobile A1 の D29 ピンに接続
25	A1_VIO1_VD	In/Out	R-Mobile A1 の SCIFA_TXD_0 ピンに接続
26	A1_VIO1_D0	In/Out	R-Mobile A1 の VIO_D8_0 ピンに接続
27	A1_VIO1_D1	In/Out	R-Mobile A1 の VIO_D9_0 ピンに接続
28	A1_VIO1_D2	In/Out	R-Mobile A1 の VIO_D10_0 ピンに接続
29	A1_VIO1_D3	In/Out	R-Mobile A1 の VIO_D11_0 ピンに接続
30	A1_VIO1_D4	In/Out	R-Mobile A1 の VIO_D12_0 ピンに接続
31	A1_VIO1_D5	In/Out	R-Mobile A1 の VIO_D13_0 ピンに接続
32	A1_VIO1_D6	In/Out	R-Mobile A1 の VIO_D14_0 ピンに接続
33	A1_VIO1_D7	In/Out	R-Mobile A1 の VIO_D15_0 ピンに接続
34	A1_VIO1_STANDBY	In/Out	R-Mobile A1 の SCIFA_RTS_0#ピンに接続
35	A1_VIO1_RST_B	In/Out	R-Mobile A1 の SCIFA_CTS_0#ピンに接続
36	CAM_DISABLE	In	カメラのイネーブル/ディスエーブルの切り替え信号 (Low: イネーブル、High: ディスエーブル)
37	VCC_3.3V	Power	電源(VCC_3.3V)
38	DGND	Power	電源(DNGD)
39	A1_STP0_IPCLK	In/Out	R-Mobile A1 の STP_IPCLK_0 ピンに接続
40	A1_STP0_IPSYNC	In/Out	R-Mobile A1 の SCIFB_TXD ピンに接続
41	A1_STP0_IPEN	In/Out	R-Mobile A1 の SCIFB_RXD ピンに接続
42	A1_STP0_IPD0	In/Out	R-Mobile A1 の SCIFB_SCK ピンに接続
43	A1_STP0_IPD1	In/Out	R-Mobile A1 の MEMC_AD1 ピンに接続
44	A1_STP0_IPD2	In/Out	R-Mobile A1 の SCIFA_RTS_1#ピンに接続
45	A1_STP0_IPD3	In/Out	R-Mobile A1 の MEMC_AD2 ピンに接続
46	A1_STP0_IPD4	In/Out	R-Mobile A1 の MEMC_AD3 ピンに接続
47	A1_STP0_IPD5	In/Out	R-Mobile A1 の SCIFA_CTS_1#ピンに接続
48	A1_STP0_IPD6	In/Out	R-Mobile A1 の MEMC_WAIT ピンに接続
49	A1_STP0_IPD7	In/Out	R-Mobile A1 の MEMC_NOE ピンに接続
50	A1_SIM_D	In/Out	R-Mobile A1 の SCIFA_SCK_2 ピンに接続
51	A1_SIM_CLK	In/Out	R-Mobile A1 の MEMC_ADV ピンに接続
52	A1_SIM_RST	In/Out	R-Mobile A1 の MEMC_CS1 ピンに接続
53	A1_PORT15	In/Out	R-Mobile A1 の FMISOIBT ピンに接続
54	A1_PORT14	In/Out	R-Mobile A1 の FMISOILR ピンに接続
55	RESETP_B	In/Out	R-Mobile A1 の RESETP#ピンに接続
56	NC	-	未接続
57	VCC_3.3V	Power	電源(VCC_3.3V)
58	DGND	Power	電源(DGND)
59	A1_SCL	In/Out	R-Mobile A1 の I2C_SCL_0 ピンに接続
60	A1_SDA	In/Out	R-Mobile A1 の I2C_SDA_0 ピンに接続

18.13. CON16(LCD インターフェース 1)

CON16 は LCD インターフェースです。コネクタは実装していません。



CON16 と CON17 の信号は共通となっており、同時に使用することはできません。

- ・ 基板側コネクタ例： XG4C-5031(オムロン)

表 18.16 CON16 信号配列


ピン番号	信号名	I/O	機能
1	VCC_5V	Power	電源(VCC_5V)
2	VCC_5V	Power	電源(VCC_5V)
3	VCC_3.3V	Power	電源(VCC_3.3V)
4	VCC_3.3V	Power	電源(VCC_3.3V)
5	DGND	Power	電源(DGND)
6	DGND	Power	電源(DGND)
7	A1_LCD0_D16	Out	LCD データ(bit16)、R-Mobile A1 の LCDD16_0 ピンに接続
8	A1_LCD0_D17	Out	LCD データ(bit17)、R-Mobile A1 の LCDD17_0 ピンに接続
9	A1_LCD0_D18	Out	LCD データ(bit18)、R-Mobile A1 の LCDD18_0 ピンに接続
10	LCD0_D19	Out	LCD データ(bit19)、R-Mobile A1 の DBGMD20 ピンに接続
11	LCD0_D20	Out	LCD データ(bit20)、R-Mobile A1 の DBGMD21 ピンに接続
12	LCD0_D21	Out	LCD データ(bit21)、R-Mobile A1 の DBGMDT0 ピンに接続
13	LCD0_D22	Out	LCD データ(bit22)、R-Mobile A1 の DBGMDT2 ピンに接続
14	LCD0_D23	Out	LCD データ(bit23)、R-Mobile A1 の DBGMDT1 ピンに接続
15	A1_LCD0_D8	Out	LCD データ(bit8)、R-Mobile A1 の LCDD8_0 ピンに接続
16	A1_LCD0_D9	Out	LCD データ(bit9)、R-Mobile A1 の LCDD9_0 ピンに接続
17	A1_LCD0_D10	Out	LCD データ(bit10)、R-Mobile A1 の LCDD10_0 ピンに接続
18	A1_LCD0_D11	Out	LCD データ(bit11)、R-Mobile A1 の LCDD11_0 ピンに接続
19	A1_LCD0_D12	Out	LCD データ(bit12)、R-Mobile A1 の LCDD12_0 ピンに接続
20	A1_LCD0_D13	Out	LCD データ(bit13)、R-Mobile A1 の LCDD13_0 ピンに接続
21	A1_LCD0_D14	Out	LCD データ(bit14)、R-Mobile A1 の LCDD14_0 ピンに接続
22	A1_LCD0_D15	Out	LCD データ(bit15)、R-Mobile A1 の LCDD15_0 ピンに接続
23	A1_LCD0_D0	Out	LCD データ(bit0)、R-Mobile A1 の LCDD0_0 ピンに接続
24	A1_LCD0_D1	Out	LCD データ(bit1)、R-Mobile A1 の LCDD1_0 ピンに接続
25	A1_LCD0_D2	Out	LCD データ(bit2)、R-Mobile A1 の LCDD2_0 ピンに接続
26	A1_LCD0_D3	Out	LCD データ(bit3)、R-Mobile A1 の LCDD3_0 ピンに接続
27	A1_LCD0_D4	Out	LCD データ(bit4)、R-Mobile A1 の LCDD4_0 ピンに接続
28	A1_LCD0_D5	Out	LCD データ(bit5)、R-Mobile A1 の LCDD5_0 に接続
29	A1_LCD0_D6	Out	LCD データ(bit6)、R-Mobile A1 の LCDD6_0 ピンに接続
30	A1_LCD0_D7	Out	LCD データ(bit7)、R-Mobile A1 の LCDD7_0 ピンに接続
31	DGND	Power	電源(DGND)
32	A1_LCD0_DCK	Out	DCK 信号、R-Mobile A1 の LCDDCK_0 ピンに接続
33	A1_LCD0_HSYN	Out	HSYNC 信号、R-Mobile A1 の LCDHSYN_0 ピンに接続
34	A1_LCD0_VSYN	Out	VSYNC 信号、R-Mobile A1 の LCDVSYN_0 ピンに接続
35	A1_LCD0_DISP	Out	DISP 信号、R-Mobile A1 の LCDDISP_0 ピンに接続
36	A1_LCD0_DON	Out	DON 信号、R-Mobile A1 の LCDDON_0 ピンに接続
37	A1_LCD0_VEPWC	Out	VEPWC 信号、R-Mobile A1 の LCDVEPWC_0 ピンに接続
38	A1_LCD0_VCPWC	Out	VCPWC 信号、R-Mobile A1 の LCDVCPWC_0 ピンに接続
39	A1_LCD0_LCLK	Out	LCLK 信号、R-Mobile A1 の D24 ピンに接続
40	A1_LCD0_LED_CONT	Out	LED_CONT 信号、R-Mobile A1 の MEMC_BUSCLK ピンに接続
41	NC	-	未接続
42	NC	-	未接続
43	NC	-	未接続

ピン番号	信号名	I/O	機能
44	NC	-	未接続
45	A1_SDA0	In/Out	SDA 信号、R-Mobile A1 の I2C_SDA_0 ピンに接続
46	A1_SCL0	In/Out	SCL 信号、R-Mobile A1 の I2C_SCL_0 ピンに接続
47	TP_INT	In	割り込み信号、R-Mobile A1 の FMSICK ピンに接続
48	TP_RST_B	Out	リセット信号、R-Mobile A1 の D21 ピンに接続
49	VCC_3.3V	Power	電源(VCC_3.3V)
50	DGND	Power	電源(DGND)

18.14. CON17(LCD インターフェース 2)

CON17 は LCD インターフェースです。XF2M-5015-1A(オムロン)を実装しています。

- ・ 対応 LCD モジュール：AM-800480L1TMQW-T00H(AMPIRE)



CON16 と CON17 の信号は共通となっており、同時に使用することはできません。

表 18.17 CON17 信号配列

ピン番号	信号名	I/O	機能
1	VCC_5V	Power	電源(VCC_5V)
2	VCC_5V	Power	電源(VCC_5V)
3	VCC_5V	Power	電源(VCC_5V)
4	NC	-	電源(VCC_3.3V)
5	VCC_3.3V	Power	電源(VCC_3.3V)
6	VCC_3.3V	Power	電源(VCC_3.3V)
7	NC	-	未接続
8	DGND	Power	電源(DGND)
9	DGND	Power	電源(DGND)
10	TP_RST_B	Out	リセット信号、R-Mobile A1 の D21 ピンに接続
11	TP_INT	In	割り込み信号、R-Mobile A1 の FMSICK ピンに接続
12	DGND	Power	電源(DGND)
13	A1_SCL0	In/Out	SCL 信号、R-Mobile A1 の I2C_SCL_0 ピンに接続
14	A1_SDA0	In/Out	SDA 信号、R-Mobile A1 の I2C_SDA_0 ピンに接続
15	DGND	Power	電源(DGND)
16	A1_LCD0_LED_CONT	Out	LED_CONT 信号、R-Mobile A1 の MEMC_BUSCLK ピンに接続
17	DGND	Power	電源(DGND)
18	A1_LCD0_DON	Out	DON 信号、R-Mobile A1 の LCDDON_0 ピンに接続
19	A1_LCD0_DISP	Out	DISP 信号、R-Mobile A1 の LCDDISP_0 ピンに接続
20	A1_LCD0_VSYN	Out	VSYNC 信号、R-Mobile A1 の LCDVSYN_0 ピンに接続
21	A1_LCD0_HSYN	Out	HSYNC 信号、R-Mobile A1 の LCDHSYN_0 ピンに接続
22	A1_LCD0_DCK	Out	DCK 信号、R-Mobile A1 の LCDDCK_0 ピンに接続
23	DGND	Power	電源(DGND)
24	A1_LCD0_D7	Out	LCD データ(bit7)、R-Mobile A1 の LCDD7_0 ピンに接続
25	A1_LCD0_D6	Out	LCD データ(bit6)、R-Mobile A1 の LCDD6_0 ピンに接続
26	A1_LCD0_D5	Out	LCD データ(bit5)、R-Mobile A1 の LCDD5_0 ピンに接続
27	A1_LCD0_D4	Out	LCD データ(bit4)、R-Mobile A1 の LCDD4_0 ピンに接続
28	A1_LCD0_D3	Out	LCD データ(bit3)、R-Mobile A1 の LCDD3_0 ピンに接続
29	A1_LCD0_D2	Out	LCD データ(bit2)、R-Mobile A1 の LCDD2_0 ピンに接続
30	A1_LCD0_D1	Out	LCD データ(bit1)、R-Mobile A1 の LCDD1_0 ピンに接続

ピン番号	信号名	I/O	機能
31	A1_LCD0_D0	Out	LCD データ(bit0)、R-Mobile A1 の LCDD0_0 ピンに接続
32	DGND	Power	電源(DGND)
33	A1_LCD0_D15	Out	LCD データ(bit15)、R-Mobile A1 の LCDD15_0 ピンに接続
34	A1_LCD0_D14	Out	LCD データ(bit14)、R-Mobile A1 の LCDD14_0 ピンに接続
35	A1_LCD0_D13	Out	LCD データ(bit13)、R-Mobile A1 の LCDD13_0 ピンに接続
36	A1_LCD0_D12	Out	LCD データ(bit12)、R-Mobile A1 の LCDD12_0 ピンに接続
37	A1_LCD0_D11	Out	LCD データ(bit11)、R-Mobile A1 の LCDD11_0 ピンに接続
38	A1_LCD0_D10	Out	LCD データ(bit10)、R-Mobile A1 の LCDD10_0 ピンに接続
39	A1_LCD0_D9	Out	LCD データ(bit9)、R-Mobile A1 の LCDD9_0 ピンに接続
40	A1_LCD0_D8	Out	R-Mobile A1 の LCDD8_0 ピンに接続
41	DGND	Power	電源(DGND)
42	LCD0_D23	Out	LCD データ(bit23)、R-Mobile A1 の DBGMDT1 ピンに接続
43	LCD0_D22	Out	LCD データ(bit22)、R-Mobile A1 の DBGMDT2 ピンに接続
44	LCD0_D21	Out	LCD データ(bit21)、R-Mobile A1 の DBGMDT0 ピンに接続
45	LCD0_D20	Out	LCD データ(bit20)、R-Mobile A1 の DBGMD21 ピンに接続
46	LCD0_D19	Out	LCD データ(bit19)、R-Mobile A1 の DBGMD20 ピンに接続
47	A1_LCD0_D18	Out	LCD データ(bit18)、R-Mobile A1 の LCDD18_0 ピンに接続
48	A1_LCD0_D17	Out	LCD データ(bit17)、R-Mobile A1 の LCDD17_0 ピンに接続
49	A1_LCD0_D16	Out	LCD データ(bit16)、R-Mobile A1 の LCDD16_0 ピンに接続
50	DGND	Power	電源(DGND)

18.15. CON19(電源入力インターフェース)

CON19 は電源を供給する DC ジャックです。AC アダプターのジャック形状は EIAJ RC-5320A 準拠 (電圧区分 2) です。



図 18.1 AC アダプターの極性マーク

表 18.18 CON19 信号配列

ピン番号	信号名	I/O	機能
1	VCC_5V	Power	電源(VCC_5V)
2	DGND	Power	電源(DGND)
3	DGND	Power	電源(DGND)
4	DGND	Power	電源(DGND)

18.16. CON20(USB インターフェース 1)

CON20 は USB ホストインターフェースです。Type A コネクタを実装しています。



CON20 と CON24 の信号は共通となっており、同時に使用することはできません。

CON20 と CON24 で使用する各信号をどちらのインターフェースに接続させるかは、ディップスイッチ(SW1)の USB0 設定で選択できるようになっています。詳しくは、「表 4.3. ディップスイッチ(SW1)のスイッチの機能」または、「18.25. SW1(機能選択スイッチ)」を参照してください。

表 18.19 CON20 信号配列

ピン番号	信号名	I/O	機能
1	VCC_5V	Power	電源(VCC_5V)
2	A1_DM0	In/Out	USB マイナス側信号、R-Mobile A1 の DM_0 ピンに接続
3	A1_DP0	In/Out	USB プラス側信号、R-Mobile A1 の DP_0 ピンに接続
4	DGND	Power	電源(DGND)
5	DGND	Power	電源(DGND)
6	DGND	Power	電源(DGND)
7	DGND	Power	電源(DGND)

18.17. CON21(USB インターフェース 2)

USB ホストインターフェースです。Type A コネクタを実装しています。

表 18.20 CON21 信号配列

ピン番号	信号名	I/O	機能
1	VCC_5V	Power	電源(VCC_5V)
2	A1_DM1	In/Out	USB マイナス側信号、R-Mobile A1 の DM_1 ピンに接続
3	A1_DP1	In/Out	USB プラス側信号、R-Mobile A1 の DP_1 ピンに接続
4	DGND	Power	電源(DGND)
5	DGND	Power	電源(DGND)
6	DGND	Power	電源(DGND)
7	DGND	Power	電源(DGND)

18.18. CON22(シリアルインターフェース)

非同期(調歩同期)シリアルインターフェースです。D-Sub 9 ピンを実装しています。RS232C トランシーバを経由して R-Mobile A1 と接続されています。

表 18.21 CON22 信号配列

ピン番号	信号名	I/O	機能
1	NC	-	未接続
2	UART_RXD	In	UART 受信データ
3	UART_TXD	Out	UART 送信データ
4	NC	-	未接続
5	DGND	Power	電源(DGND)
6	RTS	Out	常時 High
7	CTS	-	未接続
8	NC	-	未接続
9	NC	-	未接続

18.19. CON23(LAN インターフェース)

10BASE-T/100BASE-TX の LAN インターフェースです。RJ45 コネクタを実装しています。イーサネット PHY を経由して R-Mobile A1 と接続されています。

表 18.22 CON23 信号配列

ピン番号	信号名	I/O	機能
1	TX+	Out	差動のツイストペア送信出力(+)
2	TX-	Out	差動のツイストペア送信出力(-)
3	RX+	In	差動のツイストペア受信入力(+)

ピン番号	信号名	I/O	機能
4	-	-	75Ω 終端、CON23(5ピン)とコネクタ内部で接続
5	-	-	75Ω 終端、CON23(4ピン)とコネクタ内部で接続
6	RX-	In	差動のツイストペア受信入力(-)
7	-	-	75Ω 終端、CON23(8ピン)とコネクタ内部で接続
8	-	-	75Ω 終端、CON23(7ピン)とコネクタ内部で接続

18.20. CON24(USB インターフェース 3)

CON24 は USB デバイスインターフェースです。Type B コネクタを実装しています。



CON20 と CON24 の信号は共通となっており、同時に使用することはできません。

CON20 と CON24 で使用する各信号をどちらのインターフェースに接続させるかは、ディップスイッチ(SW1)の USB0 設定で選択できるようになっています。詳しくは、「表 4.3. ディップスイッチ(SW1)のスイッチの機能」または、「18.25. SW1(機能選択スイッチ)」を参照してください。

表 18.23 CON24 信号配列

ピン番号	信号名	I/O	機能
1	A1_VBUS	Power	電源(A1_VBUS)、R-Mobile A1 の VBUS ピンに接続
2	A1_DM0	In/Out	USB マイナス側信号、R-Mobile A1 の DM_0 ピンに接続
3	A1_DP0	In/Out	USB プラス側信号、R-Mobile A1 の DP_0 ピンに接続
4	DGND	Power	電源(DGND)
5	FG	Power	電源(DGND)
6	FG	Power	電源(DGND)
7	FG	Power	電源(DGND)

18.21. LED1(カメラ LED)

LED1 はカメラモジュールの FLASH LED です。

表 18.24 LED1 の挙動

LED	名称(色)	説明
LED1	LED(黄色)	カメラモジュールの GPIO4_FLASH に接続 (Low: 消灯、High: 点灯)

18.22. LED2(電源 LED)

LED2 は電源 LED です。VCC_3.3V が供給されると点灯します。

表 18.25 LED の挙動

LED	名称(色)	点灯	消灯
LED2	電源 LED(緑色)	VCC_3.3V が供給されていることを示します。	VCC_3.3V が供給されていないことを示します。

18.23. LED3~LED6(ユーザー LED)

LED3~LED6 はユーザー LED です。

表 18.26 LED3～LED6 の挙動

LED	名称(色)	説明
LED3	ユーザー LED(黄色)	R-Mobile A1 の FRB ピンに接続(Low: 消灯、High: 点灯)
LED4	ユーザー LED(黄色)	R-Mobile A1 の CS4#ピンに接続(Low: 消灯、High: 点灯)
LED5	ユーザー LED(黄色)	R-Mobile A1 の CS2#ピンに接続(Low: 消灯、High: 点灯)
LED6	ユーザー LED(黄色)	R-Mobile A1 の WAIT#ピンに接続(Low: 消灯、High: 点灯)

18.24. LED7、LED8(LAN LED)

LED7、LED8 は LAN インターフェース(CON23)のステータス LED です。

表 18.27 LED7、LED8 の挙動

LED	名称(色)	点灯	消灯	点滅
LED7	LINK/ACT LED(緑色)	LAN ケーブルが接続されており、リンクが確立されている。	LAN ケーブルが接続されており、リンクが確立されていない。	データを送受信している。
LED8	SPEED LED(黄色)	100BASE-TX のリンクが確立されている。	LAN ケーブルが接続されている場合、10BASE-T のリンクが確立されている。または、リンクが確立されていない。	-

18.25. SW1 (機能選択スイッチ)

SW1 はディップスイッチ 8 接点です。スイッチを OFF に設定すると High、ON に設定すると Low が各信号に入力されます。

表 18.28 SW1 信号配列

ピン番号	信号名	機能
1	A1_PORT101_JP2	R-Mobile A1 の FCE0#ピンに接続
2	A1_MD2	R-Mobile A1 の MD2 ピンに接続
3	A1_MD3	R-Mobile A1 の MD3 ピンに接続
4	MMC_DISABLE	eMMC と拡張バスインターフェース(CON2)の切り替え (ON: eMMC、OFF: 拡張バスインターフェース)
5	SDSL0T2_ENABLE	SD インターフェース 2(CON8)と AWL13 モジュール(CON14)の切り替え (ON: SD インターフェース 2、OFF: AWL13 モジュール)
6	USB_DEVICE_MODE	USB デバイスインターフェース(CON24)と USB ホストインターフェース(CON20)の切り替え (ON: USB デバイスインターフェース、OFF: USB ホストインターフェース)
7	DBGMD20	R-Mobile A1 の DBGMD20 ピンに接続
8	DBGMD21	R-Mobile A1 の DBGMD21 ピンに接続

18.26. SW2(リセットスイッチ)

SW2 はリセットスイッチです。タクトスイッチを実装しています。基板上のリセット IC に接続されています。

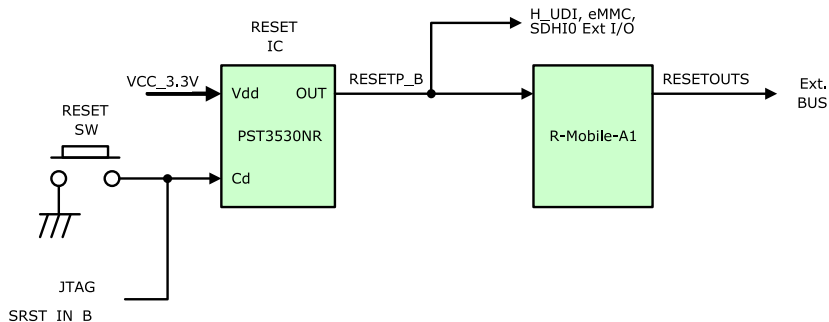


図 18.2 リセットブロック図

表 18.29 SW2 の機能

SW	機能
SW2	リセット(押された状態: リセット状態、押されていない状態: リセット解除)

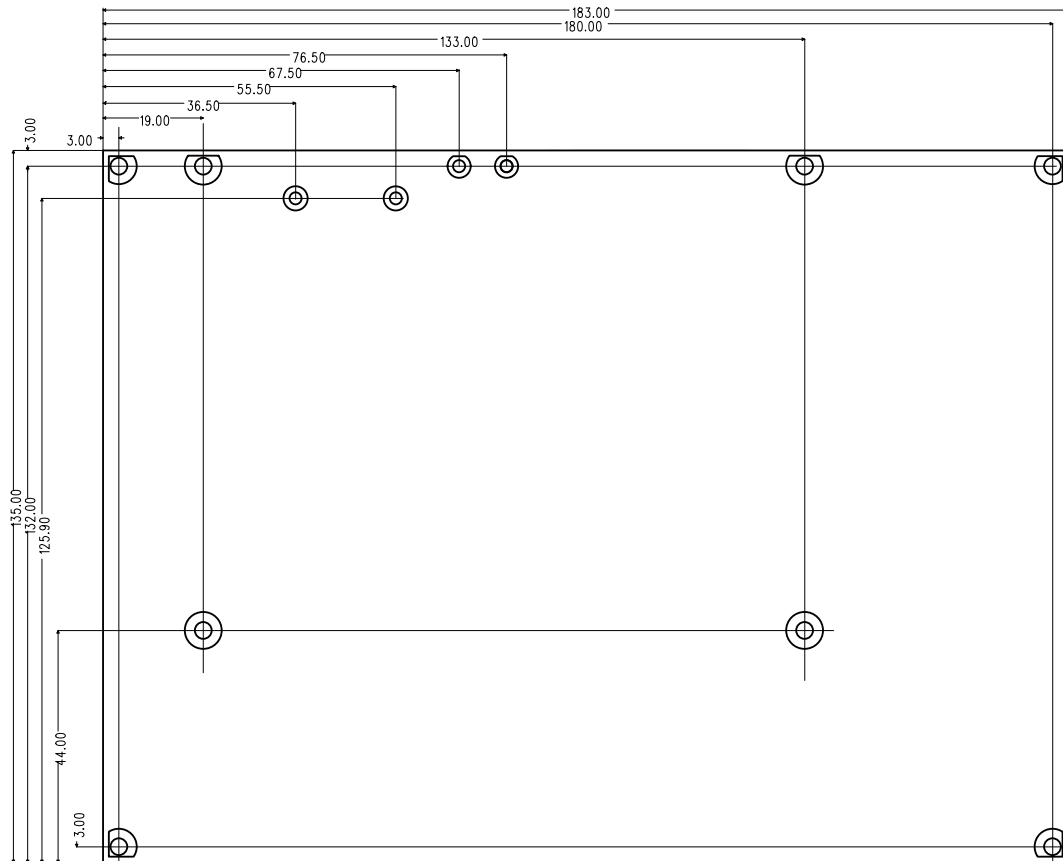
18.27. SW3～SW6(ユーザースイッチ)

SW3～SW6 はユーザースイッチです。タクトスイッチを実装しています。スイッチが押されている時は Low、押されていない時は High が各ピンに入力されます。

表 18.30 SW3～SW6 の機能

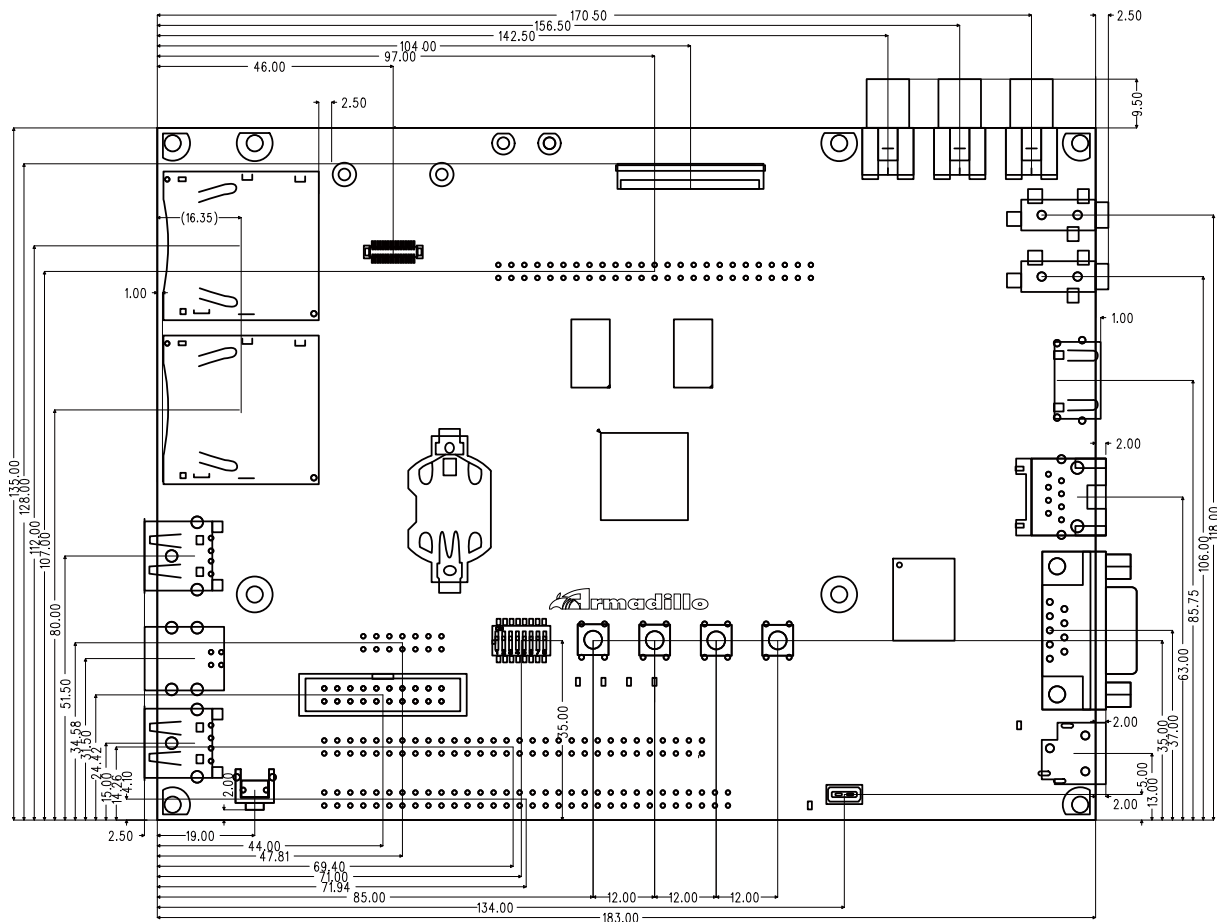
SW	説明
SW3	R-Mobile A1 の MEMC_AD6 ピンに接続
SW4	R-Mobile A1 の MEMC_AD7 ピンに接続
SW5	R-Mobile A1 の MEMC_AD4 ピンに接続
SW6	R-Mobile A1 の MEMC_AD5 ピンに接続

19. 基板形状図



[Unit : mm]

図 19.1 基板形状および固定穴寸法



[Unit : mm]

図 19.2 コネクタ中心寸法

付録 A Hermit-At ブートローダー

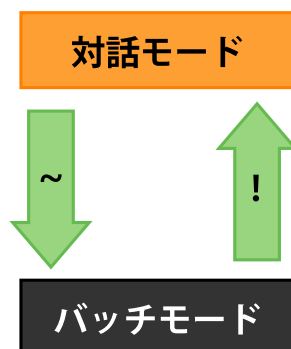
Hermit-At は、アットマークテクノ製品に採用されている高機能ダウンローダー兼ブートローダー^[1]です。Armadillo を保守モードで起動すると、Hermit-At ブートローダーのプロンプトが表示されます。プロンプトからコマンドを入力することにより、フラッシュメモリの書き換えや、Linux カーネルパラメーターの設定等 Hermit-At ブートローダーの様々な機能を使用することができます。ここでは、代表的な機能について説明します。



Hermit-AT のモード

Hermit-AT には、2つのモードがあります。コマンドプロンプトを表示して対話的に動作する「対話モード」と、Hermit-AT ダウンローダと通信するための「バッチモード」です。バッチモードではコマンドプロンプトの表示や入力した文字の表示を行いませんが、コマンドの実行は可能です。

起動直後の Hermit-AT は必ず対話モードになっています。対話モードからバッチモードに移行するにはチルダ「~」を、バッチモードから対話モードに移行するにはエクスクラメーションマーク「!」を入力します。



Hermit-AT ダウンローダと通信を行った場合は、バッチモードに移行します。これは通信を確立するために Hermit-AT ダウンローダがチルダを送信するためです。

対話モードからバッチモードに移行したり、バッチモード中に入力したコマンドが成功した場合などは以下のように表示されます。

```
+OK
```

A.1. version

バージョン情報を表示するコマンドです。

^[1]Armadillo-800 EVA では、Hermit-At のブートローダー機能のみ利用可能です。


```
構文 : version
```

図 A.1 version 構文

A.1.1. version 使用例

```
hermit> version  
Hermit-At v3.0.0 (Armadillo-800 EVA) compiled at 22:22:10, Dec 21 2011
```

図 A.2 version の使用例

A.2. info

ボード情報を表示するコマンドです。

```
構文 : info
```

図 A.3 info 構文

A.2.1. info 使用例

```
hermit> info  
Board Type: 0x08000000  
Revision: 0x00000003  
Lot: 0x00000001  
Serial Number: 0  
Boot Mode: 0x00000038 (eMMC)  
Jumper: ON  
Tact-SW: OFF, OFF, OFF, OFF  
ORIG MAC-1: 00:11:0c:00:00:00
```

図 A.4 info の使用例

A.3. mac

MAC アドレスを表示するコマンドです。

```
構文 : mac
```

図 A.5 mac 構文

A.3.1. mac 使用例

```
hermit> mac
00:11:0c:00:00:00
```

図 A.6 mac の使用例

A.4. setenv と clearenv

Linux カーネルパラメーターを設定するコマンドです。setenv で設定されたパラメータは、Linux カーネルブート時にカーネルに渡されます。clearenv を実行すると、設定がクリアされます。このパラメータは、内蔵ストレージに保存され再起動後も設定は有効となります。

構文：setenv [カーネルパラメーター]...
 説明：カーネルパラメーターを設定します。オプションを指定せずに実行すると、現在の設定を表示します。

構文：clearenv
 説明：設定されているオプションをクリアします。

図 A.7 setenv/clearenv 構文

A.4.1. setenv/clearenv 使用例

```
hermit> setenv console=ttySC1,115200
hermit> setenv
1: console=ttySC1,115200
hermit> clearenv
hermit> setenv
hermit>
```

図 A.8 setenv と clearenv の使用例

A.4.2. Linux カーネルパラメーター

Linux カーネルパラメーターの例を、「表 A.1. よく使用される Linux カーネルパラメーター」に示します。その他のオプションについては、[linux-2.6/Documentation/kernel-parameters.txt](#) を参照してください。

表 A.1 よく使用される Linux カーネルパラメーター

オプション	説明
console	カーネルコンソールとして使用するデバイスを指示します。
root	ルートファイルシステム関連の設定を指示します。
rootdelay	ルートファイルシステムをマウントする前に指定秒間待機します。
rootwait	ルートファイルシステムがアクセス可能になるまで待機します。
noinitrd	カーネルが起動した後に initrd データがどうなるのかを指示します。
nfsroot	NFS を使用する場合に、ルートファイルシステムの場所や NFS オプションを指示します。

A.5. setbootdevice

Linux カーネルを格納しているブートデバイスを指定するコマンドです。この設定は内蔵ストレージに保存され、再起動後も設定は有効となります。

構文：setbootdevice mmcblk0pN

説明：内蔵ストレージのパーティション N の /boot/ ディレクトリに置かれたカーネルイメージを RAM に展開してブートします

↵

構文：setbootdevice mmcblk1pN

説明：SD カード(CON7)のパーティション N の /boot/ ディレクトリに置かれたカーネルイメージを RAM に展開してブートします

↵

図 A.9 setbootdevice 構文

A.5.1. setbootdevice の使用例

内蔵ストレージのパーティション 4 の /boot/ ディレクトリに置かれたカーネルイメージでブートするには、「図 A.10. ブートデバイスに内蔵ストレージのパーティション 4 を指定する」のようにコマンドを実行します。

```
hermit> setbootdevice mmcblk0p4
```

図 A.10 ブートデバイスに内蔵ストレージのパーティション 4 を指定する

SD カード(CON7)のパーティション 1 に格納されたカーネルイメージでブートするには、「図 A.11. ブートデバイスに SD カードを指定する」のようにコマンドを実行します。

```
hermit> setbootdevice mmcblk1p1
```

図 A.11 ブートデバイスに SD カードを指定する

A.6. frob

指定したアドレスのデータを読み込む、または、変更することができるモードに移行するコマンドです。

表 A.2 frob コマンド

frob コマンド	説明
peek [addr]	指定されたアドレスから 32bit のデータを読み出します。
peek16 [addr]	指定されたアドレスから 16bit のデータを読み出します。
peek8 [addr]	指定されたアドレスから 8bit のデータを読み出します。
poke [addr] [value]	指定されたアドレスに 32bit のデータを書き込みます。
poke16 [addr] [value]	指定されたアドレスに 16bit のデータを書き込みます。
poke8 [addr] [value]	指定されたアドレスに 8bit のデータを書き込みます。

A.7. boot

setbootdevice で指定されたブートデバイスから Linux カーネルをブートするコマンドです。

```
構文 : boot
```

図 A.12 boot 構文

A.7.1. boot 使用例

```
hermit> boot
mmcscd: SD card at address 0x00000001
mmcscd: M8G2FA 1048576KiB
gendisk: /dev/mmcblk0p4: start=0x000f4280, size=0x001dc0c0
gendisk: Image.bin is found. (4390496 Bytes)
Copying      kernel...done.           ❶
Doing console=ttySC1,115200
Doing noinitrd
Doing rootwait
Doing root=/dev/mmcblk0p4
Doing init=/init                       ❷
Linux version 2.6.35.7 (atmark@atde4) (gcc version 4.4.5 (Debian 4.4.5-8) )
#1 PREEMPT Wed Dec 21 22:37:47 JST 2011  ❸
CPU: ARMv7 Processor [412fc093] revision 3 (ARMv7), cr=10c53c7f
CPU: VIPT nonaliasing data cache, VIPT nonaliasing instruction cache
Machine: Armadillo-800EVA
:
:
```

図 A.13 boot の使用例

- ❶ カーネルイメージ RAM 上に展開しています。
- ❷ setenv でカーネルパラメーターを設定している場合、ここで表示されます。ここまでは Hermit-At が表示しています。
- ❸ カーネルがブートされ、カーネルの起動ログが表示されます。

改訂履歴

バージョン	年月日	改訂内容
1.0.0	2012/01/05	・ 初版発行

Armadillo-800 EVA 製品マニュアル
Version 1.0.0
2012/01/05

株式会社アットマークテクノ

060-0035 札幌市中央区北 5 条東 2 丁目 AFT ビル TEL 011-207-6550 FAX 011-207-6570
