

Armadillo-840 液晶モデル開発セット スタートアップガイド

A8401-D00Z

Version 1.2.0
2014/09/19

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-840 液晶モデル開発セットスタートアップガイド

株式会社アットマークテクノ

札幌本社

〒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

横浜営業所


〒221-0835 横浜市神奈川区鶴屋町3丁目 30-4 明治安田生命横浜西口ビル 7F
TEL 045-548-5651 FAX 050-3737-4597

製作著作 © 2013-2014 Atmark Techno, Inc.

Version 1.2.0
2014/09/19

目次

1. はじめに	8
1.1. 本書および関連ファイルのバージョンについて	8
1.2. 対象となる読者	9
1.3. 本書の構成	9
1.4. 表記について	9
1.4.1. フォント	9
1.4.2. 製品名と略称	9
1.4.3. コマンド入力例	9
1.4.4. アイコン	10
1.5. 謝辞	10
2. 注意事項	11
2.1. 製品本体開封についてのご注意	11
2.2. 評価ボードについてのご注意	11
2.3. 安全に関する注意事項	11
2.4. 取扱い上の注意事項	12
2.5. ソフトウェア使用に関する注意事項	13
2.6. 書込み禁止領域について	14
2.7. 保証について	14
2.8. 輸出について	14
2.9. 商標について	14
3. Armadillo の電源を入れる前に	15
3.1. 準備するもの	15
3.2. 開発/動作確認環境の構築	15
3.2.1. ATDE5 セットアップ	16
3.2.2. 取り外し可能デバイスの使用	20
3.2.3. コマンドライン端末(GNOME 端末)の起動	20
3.2.4. シリアル通信ソフトウェア(minicom)の使用	21
3.3. インターフェースレイアウト	22
3.3.1. Armadillo-840	22
3.3.2. Armadillo-840 拡張ボード 01(C コネクタ用)	23
3.4. 接続方法	25
3.5. ジャンパピンの設定について	27
3.6. スライドスイッチの設定について	28
3.7. vi エディタの使用方法	28
3.7.1. vi の起動	28
3.7.2. 文字の入力	28
3.7.3. カーソルの移動	29
3.7.4. 文字の削除	30
3.7.5. 保存と終了	30
4. 起動と終了	31
4.1. 起動	31
4.2. ログイン	36
4.3. 終了方法	36
5. 動作確認方法	38
5.1. ネットワーク	38
5.1.1. デフォルト状態のネットワーク設定	38
5.1.2. ネットワークの有効化、無効化	38
5.1.3. ネットワーク設定の変更方法	39
5.1.4. 接続を確認する	41
5.1.5. ファイアウォール	41

5.1.6. ネットワークアプリケーション	41
5.2. ビデオ	44
5.2.1. フレームバッファデバイスにテスト画像を出力	44
5.2.2. HDMI - フレームバッファデバイス /dev/fb0	46
5.2.3. LCD - フレームバッファデバイス /dev/fb1	48
5.3. オーディオ	49
5.3.1. サウンドを再生する	49
5.3.2. サウンドを録音する	50
5.4. ストレージ	51
5.4.1. ストレージの使用方法	51
5.4.2. ストレージのパーティション変更とフォーマット	53
5.5. AV コーデックミドルウェア	54
5.5.1. HDMI ディスプレイへの表示	55
5.5.2. LCD への表示 	56
5.6. LED	56
5.6.1. LED を点灯/消灯する	57
5.6.2. トリガを使用する	58
5.7. RTC	58
5.7.1. RTC に時刻を設定する	58
5.8. GPIO	59
5.8.1. 入出力方向を変更する	62
5.8.2. 入力レベルを取得する	62
5.8.3. 出力レベルを設定する	63
5.8.4. ユーザージャンパを使用する	63
5.9. ユーザースイッチ	64
5.9.1. イベントを確認する	65
5.10. タッチスクリーン	65
5.10.1. イベントを確認する	65
6. コンフィグ領域 - 設定ファイルの保存領域	67
6.1. コンフィグ領域の読出し	67
6.2. コンフィグ領域の保存	67
6.3. コンフィグ領域の初期化	67
7. ユーザー登録	69
7.1. 購入製品登録	69
7.1.1. 正規認証ファイルを取り出す手順	69

目次

3.1. GNOME 端末の起動	21
3.2. GNOME 端末のウィンドウ	21
3.3. minicom 設定方法	22
3.4. minicom 起動方法	22
3.5. minicom 終了確認	22
3.6. インターフェースレイアウト図	23
3.7. インターフェースレイアウト図	24
3.8. Armadillo-840 液晶モデルの接続例	26
3.9. スライドスイッチの設定	28
3.10. vi の起動	28
3.11. 入力モードに移行するコマンドの説明	29
3.12. 文字を削除するコマンドの説明	30
4.1. 起動ログ	31
4.2. 終了方法	37
5.1. デフォルト状態の/etc/config/interfaces	38
5.2. ネットワークインターフェース(eth0)の有効化	38
5.3. ネットワークインターフェース(eth0)の無効化	39
5.4. 固定 IP アドレス設定	40
5.5. DHCP 設定	40
5.6. DNS サーバーの設定	40
5.7. PING 確認	41
5.8. iptables	41
5.9. telnet でリモートログイン	42
5.10. ftp でファイル転送	43
5.11. Armadillo 上でアップロードされたファイルを確認	43
5.12. Armadillo トップページ	44
5.13. GStreamer のテスト画像	45
5.14. テスト画像を表示するコマンド	45
5.15. 自動起動されるデフォルトアプリケーション画面	46
5.16. LCD にテスト画像を表示するコマンド	48
5.17. テストサウンドの再生	50
5.18. サウンドの録音	50
5.19. 録音したファイルを再生	51
5.20. mount コマンド書式	52
5.21. ストレージのマウント	52
5.22. ストレージのアンマウント	53
5.23. fdisk コマンドによるパーティション変更	53
5.24. EXT3 ファイルシステムの構築	54
5.25. サンプル動画の取得	54
5.26. Photo Viewer の停止	55
5.27. サンプル動画の再生(HDMI ディスプレイ)	55
5.28. サンプル動画の再生(拡張ボード 01)	56
5.29. LED を点灯させる	57
5.30. LED を消灯させる	57
5.31. LED の状態を表示する	57
5.32. LED のトリガに timer を指定する	58
5.33. LED のトリガを表示する	58
5.34. システムクロックを設定	59
5.35. ハードウェアクロックを設定	59
5.36. GPIO の入出力方向を設定する(INPUT に設定)	62

5.37. GPIO の入出力方向を設定する (OUTPUT に設定)	62
5.38. GPIO の入力レベルを取得する	62
5.39. GPIO の出力レベルを設定する	63
5.40. ユーザージャンパの状態を取得する	63
5.41. ユーザースイッチ: イベントの確認	65
5.42. タッチスクリーン: イベントの確認	65
6.1. コンフィグ領域の読み出し方法	67
6.2. コンフィグ領域の保存方法	67
6.3. コンフィグ領域の初期化方法	68

表目次

1.1. 使用しているフォント	9
1.2. 製品名と略称の対応	9
1.3. 表示プロンプトと実行環境の関係	9
1.4. コマンド入力例での省略表記	10
3.1. ATDE5 の種類	16
3.2. ユーザー名とパスワード	19
3.3. 動作確認に使用する取り外し可能デバイス	20
3.4. シリアル通信設定	21
3.5. インターフェース内容	23
3.6. Armadillo-840 拡張ボード 01(C コネクタ用) インターフェース内容	24
3.7. ジャンパの機能	27
3.8. 入力モードに移行するコマンド	29
3.9. カーソルの移動コマンド	29
3.10. 文字の削除コマンド	30
3.11. 保存・終了コマンド	30
4.1. 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	36
5.1. デフォルト状態のネットワーク設定	38
5.2. 固定 IP アドレス設定例	39
5.3. TELNET でログイン可能なユーザ	41
5.4. ftp でログイン可能なユーザ	42
5.5. 輝度設定に使用する sysfs ファイル	48
5.6. ストレージデバイス	51
5.7. サンプル動画	54
5.8. LED - Armadillo-840	57
5.9. LED - 拡張ボード 01	57
5.10. trigger の種類	58
5.11. 時刻フォーマットのフィールド	59
5.12. 拡張インターフェース 1(Armadillo-840: CON7)の GPIO ディレクトリ	60
5.13. 拡張インターフェース 2(Armadillo-840: CON8)の GPIO ディレクトリ	61
5.14. direction の設定	62
5.15. ユーザージャンパの状態と取得できる値の対応	63
5.16. インプットデバイスファイルとイベントコード	64

1. はじめに

このたびは Armadillo-840 液晶モデル開発セットをお求めいただき、ありがとうございます。

Armadillo-840 液晶モデルは、「Armadillo-840」と「Armadillo-840 拡張ボード 01(C コネクタ用)」から構成されています。

Armadillo-840 は、ルネサスエレクトロニクス製 Cortex-A9 プロセッサ「R-Mobile A1」、DDR3 SDRAM、フラッシュメモリを中心に、HDMI、USB 2.0 ホストポート、Ethernet ポート、SD カードスロットなどを搭載し、且つ、拡張用コネクタには USB 2.0 ホスト/デバイスインターフェース、LCD インターフェース、カメラインターフェース、SD/SDIO インターフェース、SPI、GPIO などといった組み込みシステムに求められる機能を備える小型 CPU ボードです。

Armadillo-840 は、画面出力・表示機能に特化した組み込みプラットフォームとして利用することを想定して設計されています。ネットワークから受けたデータを FullHD で HDMI 対応ディスプレイに出力したり、Qt という GUI フレームワークを使ってユーザーインターフェースを構築することができます。開発セットには、Qt Creator という統合開発環境や開発に必要なソフトウェアが同梱されていますので、ご購入後すぐにシステム開発をスタートすることができます。

Armadillo-800 シリーズは標準 OS に Linux を採用していますので、Linux の豊富な (Qt のように高性能なフレームワークなどの) ソフトウェア資産を利用することができます。また、C や C++ などのプログラミング言語を使用し、オリジナルのプログラムを作成して動作させることも可能です。ソフトウェアのカスタマイズ方法については、「Armadillo-840 製品マニュアル」等を参照してください。

本書には、ご利用にあたっての注意事項や、ご購入時の状態で利用できるソフトウェアの機能について記載されています。Armadillo-840 液晶モデル開発セットがお手元に届きましたら、ハードウェアの動作確認、およびデフォルトソフトウェアの使用方法について確認いただくようお願いいたします。

尚、Armadillo-840 には、**ご購入ユーザーに限定して公開しているソフトウェアやハードウェア情報**があります。主な限定コンテンツを次に示します。

- ・ armhf アーキテクチャ用 OpenGL ES2 ライブラリ
- ・ AV コーデックミドルウェア
- ・ Armadillo-840 拡張ボード 01(C コネクタ用)の回路図

限定コンテンツを取得するには、「7. ユーザー登録」を参照してください。

以降、本書では他の Armadillo ブランド製品にも共通する記述については、製品名を Armadillo と表記します。

1.1. 本書および関連ファイルのバージョンについて

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。本書を読み始める前に、Armadillo サイトで最新版の情報をご確認ください。

Armadillo サイト - Armadillo-840 ドキュメント・ダウンロード

<http://armadillo.atmark-techno.com/armadillo-840/downloads>

1.2. 対象となる読者

- ・ ハードウェアの動作確認をされる方
- ・ ソフトウェアの基本的な使用方法の確認をされる方

上記以外の方でも、本書を有効に利用していただけただら幸いです。

1.3. 本書の構成

本書では、Armadillo-840 の基本的な使用方法について記載されています。

以下に主な項目を挙げます。

- ・ 接続方法
- ・ 起動と終了
- ・ 各種設定方法
- ・ 各種アプリケーションの使用方法

1.4. 表記について

1.4.1. 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

1.4.2. 製品名と略称

以降、本書では名称の長い製品名について以下の略称で表記します。

表 1.2 製品名と略称の対応

製品名	略称
Armadillo-840 拡張ボード 01 (C コネクタ用)	拡張ボード 01

1.4.3. 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 1.3 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

プロンプト	コマンドの実行環境
[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 1.4 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

1.4.4. 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。

1.5. 謝辞

Armadillo で使用しているソフトウェアの多くは 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. 製品本体開封についてのご注意

製品本体を開封する前に、以下の事項をご確認ください。



- ・ 本製品をご利用いただくには、あらかじめ「ソフトウェア使用許諾契約書」(本製品に同梱されている資料「はじめにお読みください」に記載)に同意いただくことが必要です。はじめに「ソフトウェア使用許諾契約書」をご確認いただき、同意の上で開封してください。

2.2. 評価ボードについてのご注意

評価ボード(「評価セット」の本体ボード、または「開発セット」に評価・開発用として同梱されたボード)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。

2.3. 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。

2.4. 取扱い上の注意事項

本製品に恒久的なダメージをあたえないよう、取扱い時には以下のような点にご注意ください。

破損しやすい箇所 BtoB コネクタ、FFC コネクタは破損しやすい部品になっています。無理に力を加えて破損することのないよう十分注意してください。

- 本製品の改造** 本製品に改造^[1]を行った場合は保証対象外となりますので十分ご注意ください。また、改造やコネクタ等の増設^[2]を行う場合は、作業前に必ず動作確認を行ってください。
- 電源投入時のコネクタ着脱** 本製品や周辺回路に電源が入っている状態で、活線挿抜対応インターフェース(LAN、HDMI、SD/SDIO、USB、マイク、ヘッドホン)以外へのコネクタ着脱は、絶対に行わないでください。
- 静電気** 本製品には CMOS デバイスを使用しており、静電気により破壊されるおそれがあります。本製品を開封するときは、低湿度状態にならないよう注意し、静電防止用マットの使用、導電靴や人体アースなどによる作業者の帯電防止対策、備品の放電対策、静電気対策を施された環境下で行ってください。また、本製品を保管する際は、静電気を帯びやすいビニール袋やプラスチック容器などは避け、導電袋や導電性の容器・ラックなどに収納してください。
- ラッチアップ** 電源および入出力からの過大なノイズやサージ、電源電圧の急激な変動等により、使用している CMOS デバイスがラッチアップを起こす可能性があります。いったんラッチアップ状態となると、電源を切断しないかぎりこの状態が維持されるため、デバイスの破損につながる可能性があります。ノイズの影響を受けやすい入出力ラインには、保護回路を入れることや、ノイズ源となる装置と共通の電源を使用しない等の対策をとることをお勧めします。
- 衝撃** 落下や衝撃などの強い振動を与えないでください。

2.5. ソフトウェア使用に関する注意事項

- 本製品に含まれるソフトウェアについて** 本製品の標準出荷状態でプリインストールされている Linux 対応ソフトウェアは、個別に明示されている（書面、電子データでの通知、口頭での通知を含む）場合を除き、オープンソースとしてソースコードが提供されています。再配布等の権利については、各ソースコードに記載のライセンス形態にしたがって、お客様の責任において行使してください。また、本製品に含まれるソフトウェア（付属のドキュメント等も含む）は、現状有姿（AS IS）にて提供します。お客様ご自身の責任において、使用用途・目的の適合について事前に十分な検討と試験を実施した上でお使いください。アットマークテクノは、当該ソフトウェアが特定の目的に適合すること、ソフトウェアの信頼性および正確性、ソフトウェアを含む本製品の使用による結果について、お客様に対し何らの保証も行いません。

パートナー等の協力により Armadillo ブランド製品向けに提供されているミドルウェア、その他各種ソフトウェアソリューションは、ソフトウェア毎にライセンスが規定されています。再頒布権等については、各ソフトウェアに付属する readme ファイル等をご参照ください。その他のバンドルソフトウェアについては、各提供元にお問い合わせください。

^[1]コネクタ非搭載箇所へのコネクタ等の増設は除く。

^[2]コネクタを増設する際にはマスキングを行い、周囲の部品に半田くず、半田ボール等付着しないよう十分にご注意ください。

2.6. 書込み禁止領域について



EEPROM のデータは、本製品に含まれるソフトウェアで使用しています。正常に動作しなくなる可能性があるため、書込みを行わないでください。また、意図的に書込みを行った場合は保証対象外となります。

2.7. 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>

2.8. 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。

2.9. 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、SDXC、microSD、microSDHC、microSDXC、SDIO ロゴは SD-3C, LLC の商標です。



- ・ HDMI、HDMI ロゴ、High-Definition Multimedia Interface は HDMI Licensing, LLC の登録商標です。



3. Armadillo の電源を入れる前に

3.1. 準備するもの

Armadillo を使用する前に、次のものを必要に応じて準備してください。

作業用 PC	Linux または Windows が動作し、ネットワークインターフェースと 1 つ以上の USB ポートを持つ PC です。「3.2. 開発/動作確認環境の構築」を参照して、作業用 PC 上に開発/動作確認環境を構築してください。
ネットワーク環境	Armadillo と作業用 PC をネットワーク通信ができるようにしてください。
HDMI 対応ディスプレイ	HDMI の動作を確認する場合に利用します。
SD カード	SD スロットの動作を確認する場合などに利用します。
USB マウスと USB キーボード	USB ホストの動作を確認したり、HDMI 対応ディスプレイに表示されたアプリケーションを操作する場合などに利用します。
tar.xz 形式のファイルを展開するソフトウェア	開発/動作確認環境を構築するために利用します。Linux では、tar ^[1] で展開できます。Windows では、7-Zip や Lhaz などが対応しています。7-Zip は、開発用 DVD に収録されています。
スピーカ又はヘッドホン	サウンドの再生を確認する場合に利用します。
マイク	サウンドの録音を確認する場合に利用します。

3.2. 開発/動作確認環境の構築

アットマークテクノ製品のソフトウェア開発や動作確認を簡単に行うために、VMware 仮想マシンのデータイメージを提供しています。この VMware 仮想マシンのデータイメージを ATDE (Atmark Techno Development Environment) と呼びます。ATDE の起動には仮想化ソフトウェアである VMware を使用します。ATDE のデータは、tar.xz 圧縮されています。環境に合わせたツールで展開してください。



仮想化ソフトウェアとして、VMware の他に Oracle VM VirtualBox が有名です。Oracle VM VirtualBox には以下の特徴があります。

- ・ GPL v2 (General Public License version 2) で提供されている^[2]
- ・ VMware 形式の仮想ディスク (.vmdk) ファイルに対応している

Oracle VM VirtualBox から ATDE を起動し、ソフトウェア開発環境として使用することができます。

ATDE は、バージョンにより対応するアットマークテクノ製品が異なります。Armadillo-840 に対応している ATDE は、ATDE5 (ATDE バージョン 5) です。

^[1]tar.xz 形式のファイルを展開するには Jxf オプションを指定します。

^[2]バージョン 3.x までは PUEL (VirtualBox Personal Use and Evaluation License) が適用されている場合があります。

ATDE5 は Debian GNU/Linux 7(コードネーム wheezy)をベースに、Armadillo-840 のソフトウェア開発を行うために必要なクロス開発ツールや、Armadillo-840 の動作確認を行うために必要なツールが事前にインストールされています。

3.2.1. ATDE5 セットアップ

3.2.1.1. VMware のインストール

ATDE5 を使用するためには、作業用 PC に VMware がインストールされている必要があります。VMware 社 Web ページ(<http://www.vmware.com/>)を参照し、利用目的に合う VMware 製品をインストールしてください。また、ATDE5 は tar.xz 圧縮されていますので、環境に合わせたツールで展開してください。



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。



VMware や ATDE5 が動作しないことを未然に防ぐため、使用する VMware のドキュメントから以下の項目についてご確認ください。

- ・ ホストシステムのハードウェア要件
- ・ ホストシステムのソフトウェア要件
- ・ ゲスト OS のプロセッサ要件

VMware のドキュメントは、VMware 社 Web ページ (<http://www.vmware.com/>)から取得することができます。

3.2.1.2. ATDE5 アーカイブの取得

「表 3.1. ATDE5 の種類」に示す ATDE5 のアーカイブのうちいずれか 1 つを作業用 PC にコピーします。ATDE5 のアーカイブは Armadillo サイト(<http://armadillo.atmark-techno.com>)または、開発セット付属の DVD から取得可能です。

表 3.1 ATDE5 の種類

ATDE5 アーカイブ	ベースの Debian GNU/Linux
atde5-[version]-amd64.tar.xz	64-bit PC(「amd64」)アーキテクチャ用 Debian GNU/Linux 7
atde5-[version]-i386.tar.xz	32-bit PC(「i386」)アーキテクチャ用 Debian GNU/Linux 7



作業用 PC の動作環境(ハードウェア、VMware、ATDE5 の種類など)により、ATDE5 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。

3.2.1.3. ATDE5 アーカイブの展開

ATDE5 のアーカイブを展開します。ATDE5 のアーカイブは、tar.xz 形式の圧縮ファイルです。

Windows での展開方法を「手順 3.1. Windows で ATDE5 のアーカイブ展開する」に、Linux での展開方法を「手順 3.2. Linux で tar.xz 形式のファイルを展開する」に示します。

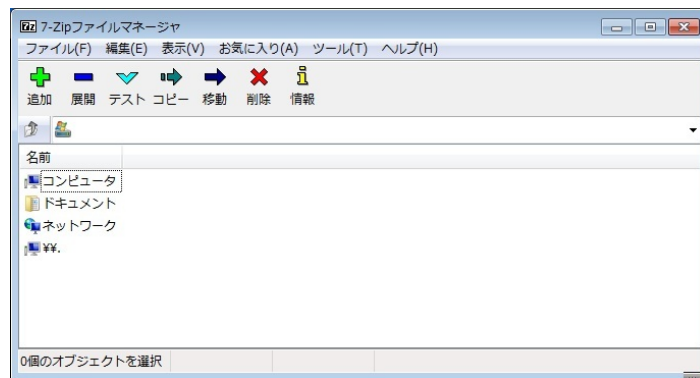
手順 3.1 Windows で ATDE5 のアーカイブ展開する

1. 7-Zip のインストール

7-Zip をインストールします。7-Zip は、圧縮解凍ソフト 7-Zip(<http://sevenzip.sourceforge.jp>)または、開発セット付属の DVD から取得可能です。

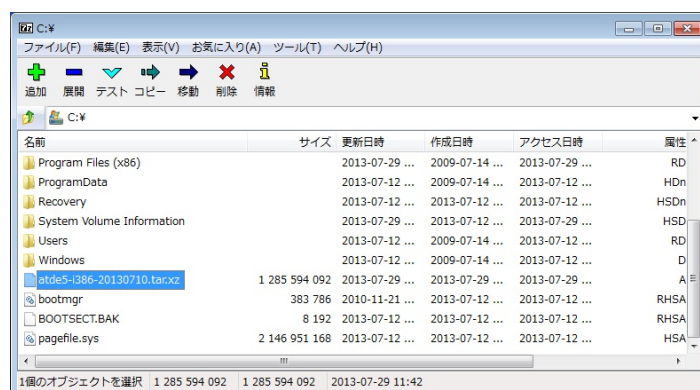
2. 7-Zip の起動

7-Zip を起動します。



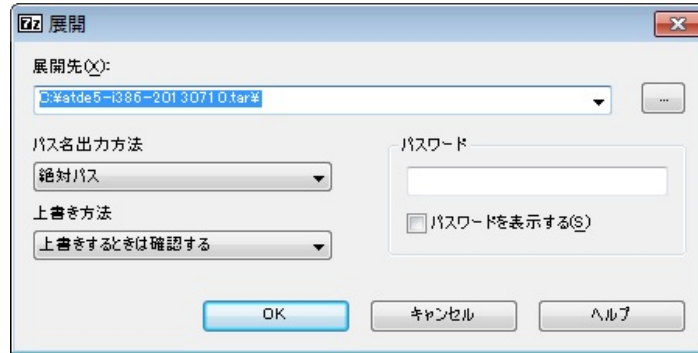
3. xz 圧縮ファイルの選択

xz 圧縮ファイルを展開して、tar 形式のファイルを出力します。tar.xz 形式のファイルを選択して、「展開」をクリックします。



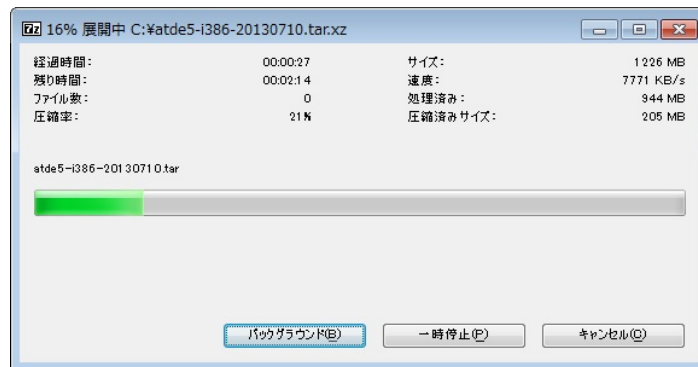
4. xz 圧縮ファイルの展開先の指定

「展開先」を指定して、「OK」をクリックします。



5. xz 圧縮ファイルの展開

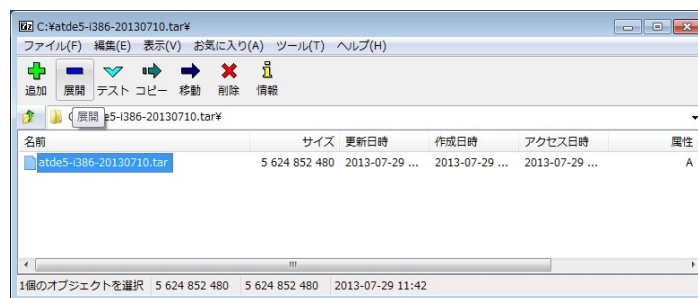
展開が始まります。



6. tar アーカイブファイルの選択

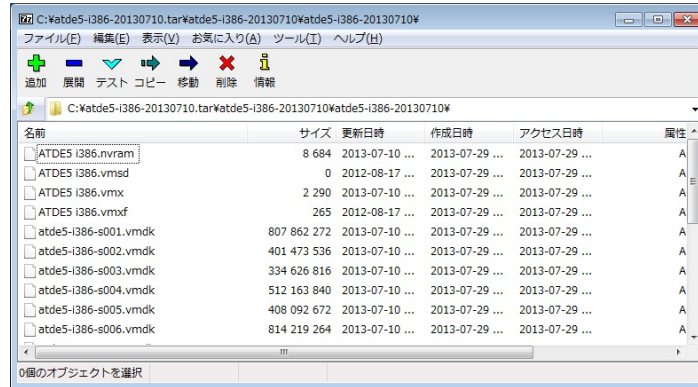
xz 圧縮ファイルの展開が終了すると、tar 形式のファイルが出力されます。

tar アーカイブファイルを出力したのと同様の手順で、tar アーカイブファイルから ATDE5 のデータイメージを出力します。tar 形式のファイルを選択して「展開」をクリックし、「展開先」を指定して、「OK」をクリックします。



7. 展開の完了確認

tar アーカイブファイルの展開が終了すると、ATDE5 アーカイブの展開は完了です。「展開先」に指定したフォルダに ATDE5 のデータイメージが出力されています。



手順 3.2 Linux で tar.xz 形式のファイルを展開する

1. tar.xz 圧縮ファイルの展開

tar の Jxf オプションを使用して tar.xz 圧縮ファイルを展開します。

```
[PC ~]$ tar Jxf atde5-i386-20130710.tar.xz
```

2. 展開の完了確認

tar.xz 圧縮ファイルの展開が終了すると、ATDE5 アーカイブの展開は完了です。atde5-i386-[*version*]ディレクトリに ATDE5 のデータイメージが出力されています。


```
[PC ~]$ ls atde5-i386-[version]/
ATDE5 i386.nvram      atde5-i386-s005.vmdk  atde5-i386-s013.vmdk
ATDE5 i386.vmsd      atde5-i386-s006.vmdk  atde5-i386-s014.vmdk
ATDE5 i386.vmx       atde5-i386-s007.vmdk  atde5-i386-s015.vmdk
ATDE5 i386.vmx      atde5-i386-s008.vmdk  atde5-i386-s016.vmdk
atde5-i386-s001.vmdk atde5-i386-s009.vmdk  atde5-i386-s017.vmdk
atde5-i386-s002.vmdk atde5-i386-s010.vmdk  atde5-i386.vmdk
atde5-i386-s003.vmdk atde5-i386-s011.vmdk
atde5-i386-s004.vmdk atde5-i386-s012.vmdk
```

3.2.1.4. ATDE5 の起動

ATDE5 のアーカイブを展開したディレクトリに存在する仮想マシン構成(.vmx)ファイルを VMware 上で開くと、ATDE5 を起動することができます。ATDE5 にログイン可能なユーザーを、「表 3.2. ユーザー名とパスワード」に示します^[3]。

表 3.2 ユーザー名とパスワード

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




ATDE に割り当てるメモリおよびプロセッサ数を増やすことで、ATDE をより快適に使用することができます。仮想マシンのハードウェア設定の変

^[3]特権ユーザーで GUI ログインを行うことはできません。

更方法については、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ください。

3.2.2. 取り外し可能デバイスの使用

VMware は、ゲスト OS (ATDE)による取り外し可能デバイス(USB デバイスや DVD など)の使用をサポートしています。デバイスによっては、ホスト OS (VMware を起動している OS)とゲスト OS で同時に使用することができません。そのようなデバイスをゲスト OS で使用するためには、ゲスト OS にデバイスを接続する操作が必要になります。



取り外し可能デバイスの使用方法については、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ください。

Armadillo-840 の動作確認を行うためには、「表 3.3. 動作確認に使用する取り外し可能デバイス」に示すデバイスをゲスト OS に接続する必要があります。

表 3.3 動作確認に使用する取り外し可能デバイス

デバイス	デバイス名
開発用 USB シリアル変換アダプタ(Armadillo-800 シリーズ対応)	Future Devices FT232R USB UART
作業用 PC の物理シリアルポート	シリアルポート

3.2.3. コマンドライン端末(GNOME 端末)の起動

ATDE5 で、CUI (Character-based User Interface)環境を提供するコマンドライン端末を起動します。ATDE5 で実行する各種コマンドはコマンドライン端末に入力し、実行します。コマンドライン端末にはいくつかの種類がありますが、ここでは GNOME デスクトップ環境に標準インストールされている GNOME 端末を起動します。

GNOME 端末を起動するには、「図 3.1. GNOME 端末の起動」のようにデスクトップ左上のメニューから「端末」を選択してください。



図 3.1 GNOME 端末の起動

「図 3.2. GNOME 端末のウィンドウ」のようにウィンドウが開きます。



図 3.2 GNOME 端末のウィンドウ

3.2.4. シリアル通信ソフトウェア(minicom)の使用

シリアル通信ソフトウェア(minicom)のシリアル通信設定を、「表 3.4. シリアル通信設定」のように設定します。また、minicom を起動する端末の横幅を 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。

表 3.4 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

minicom の設定を開始するには、「図 3.3. minicom 設定方法」のようにしてください。設定完了後、デフォルト設定(df1)に保存して終了します。

```
[ATDE ~]$ LANG=C minicom --setup
```

図 3.3 minicom 設定方法

minicom を起動させるには、「図 3.4. minicom 起動方法」のようにしてください。

```
[ATDE ~]$ LANG=C minicom --noinit --wrap --device /dev/ttyUSB0
```

図 3.4 minicom 起動方法



デバイスファイル名は、環境によって/dev/ttyS0 や/dev/ttyUSB1 など、本書の実行例とは異なる場合があります。

minicom を終了させるには、まず Ctrl+a に続いて q キーを入力します。その後、以下のように表示されたら「Yes」にカーソルを合わせて Enter キーを入力すると minicom が終了します。

```
+-----+
| Leave without reset? |
|   Yes      No      |
+-----+
```

図 3.5 minicom 終了確認



Ctrl+a に続いて z キーを入力すると、minicom のコマンドヘルプが表示されます。

3.3. インターフェースレイアウト

3.3.1. Armadillo-840

Armadillo-840 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。

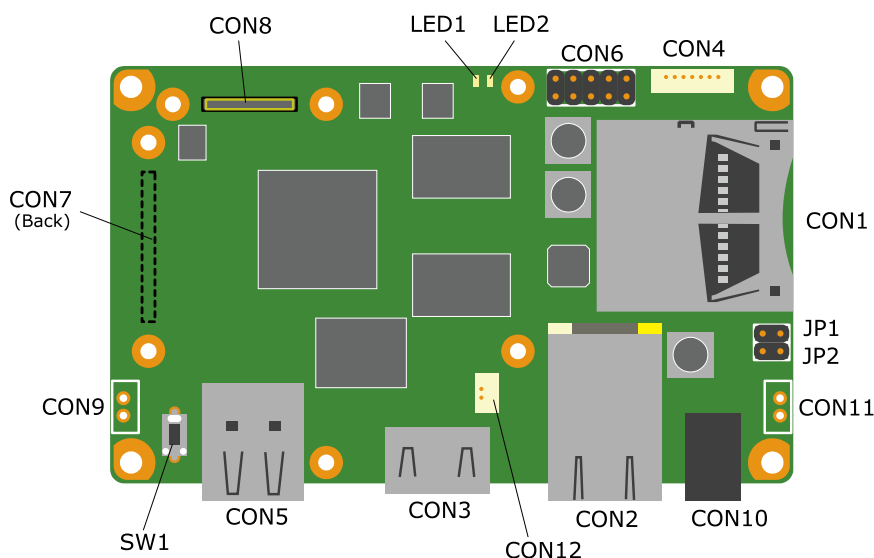


図 3.6 インターフェースレイアウト図

表 3.5 インターフェース内容

部品番号	インターフェース名	形状	備考
CON1	SD インターフェース	SD スロット	
CON2	LAN インターフェース	RJ-45 コネクタ	
CON3	HDMI インターフェース	HDMI Type-A コネクタ	
CON4	シリアルインターフェース	ピンヘッダ 7P(1.25mm ピッチ)	挿抜寿命: 50 回
CON5	USB インターフェース	USB Type-A コネクタ(2 段)	
CON6	JTAG インターフェース	ピンヘッダ 10P(2.54mm ピッチ)	
CON7	拡張インターフェース 1(C コネクタ)	BtoB コネクタ 100P(0.4mm ピッチ)	挿抜寿命: 30 回
CON8	拡張インターフェース 2(D コネクタ)	BtoB コネクタ 60P(0.4mm ピッチ)	挿抜寿命: 30 回
CON9	電源出力インターフェース	ピンヘッダ 2P(2.5mm ピッチ)	
CON10	電源入力インターフェース 1	DC ジャック	対応プラグ: EIAJ#2 ※CON11 と同時使用不可
CON11	電源入力インターフェース 2	ピンヘッダ 2P(2.5mm ピッチ)	※CON10 と電源ライン共通
CON12	RTC 外部バックアップ用電源入力インターフェース	ピンヘッダ 2P(1.25mm ピッチ)	挿抜寿命: 30 回
JP1	設定ジャンパ	ピンヘッダ 2P(2.54mm ピッチ)	オープン: OS 自動起動モード ショート: 保守モード
JP2			オープン: オンボードフラッシュメモリブート ショート: SD(CON1)ブート
LED1	ユーザー LED	LED(黄色、面実装)	
LED2			
SW1	リセットスイッチ	タクトスイッチ	

3.3.2. Armadillo-840 拡張ボード 01(C コネクタ用)

Armadillo-840 拡張ボード 01(C コネクタ用)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。

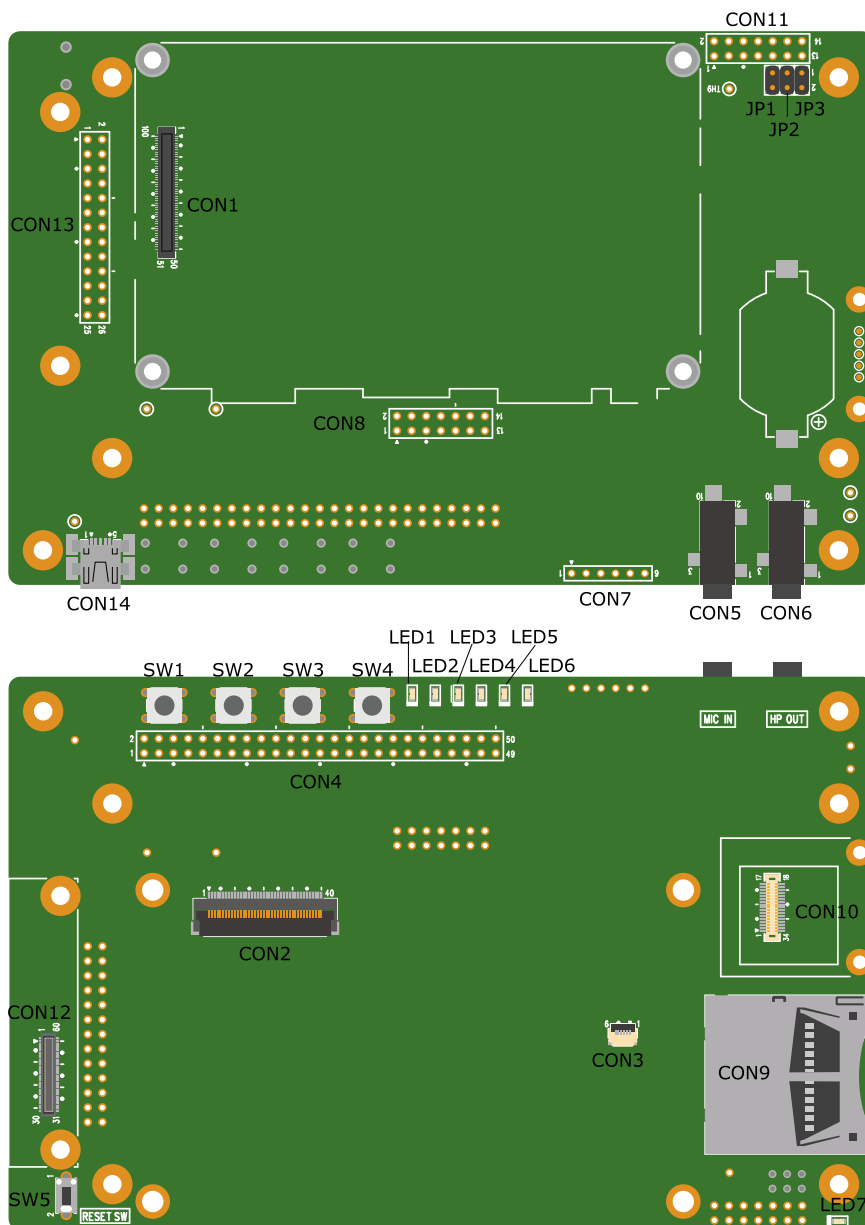


図 3.7 インターフェースレイアウト図

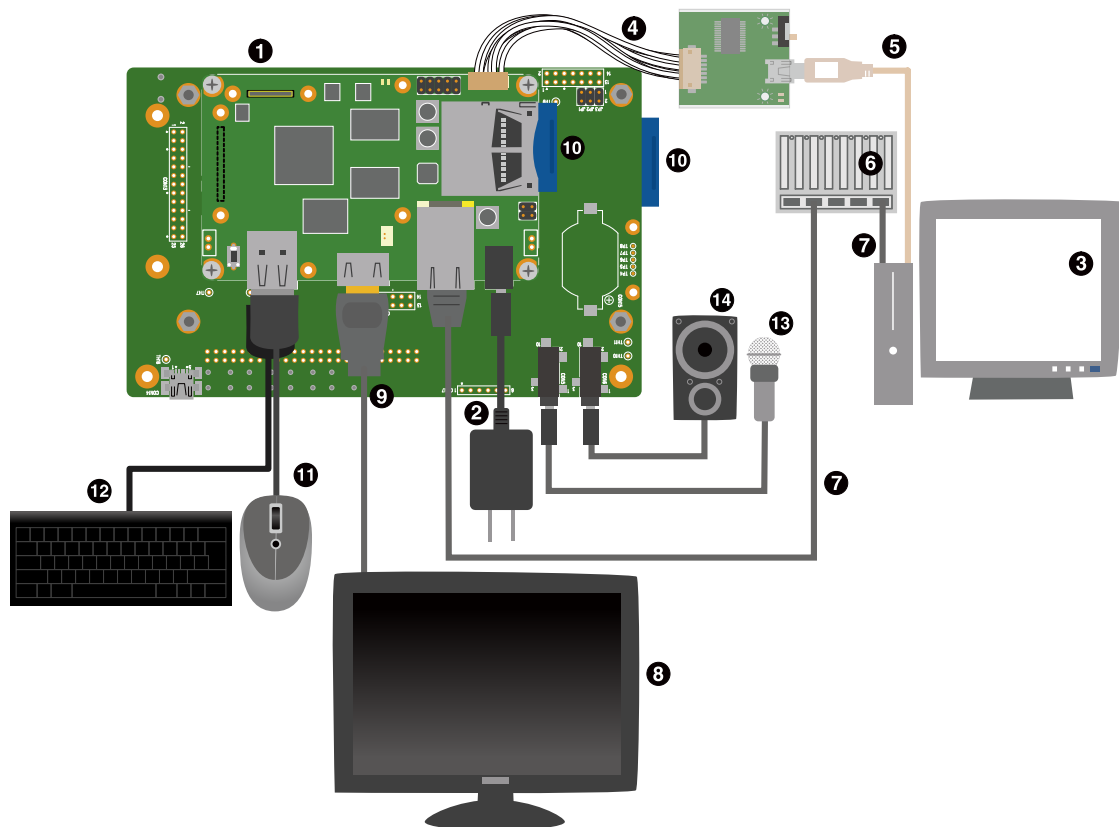
表 3.6 Armadillo-840 拡張ボード 01(C コネクタ用) インターフェース内容

部品番号	インターフェース名	形状	備考
CON1	Armadillo-840 インターフェース	BtoB コネクタ 100P(0.4mm ピッチ)	挿抜寿命: 30 回
CON2	LCD インターフェース	FFC コネクタ 40P(0.5mm ピッチ)	接続可能 LCD: SCF0500133GFR03/ DataImage 挿抜寿命: 20 回
CON3	タッチパネルインターフェース	FFC コネクタ 6P(0.5mm ピッチ)	接続可能 LCD: SCF0500133GFR03/ DataImage 挿抜寿命: 20 回
CON4	拡張インターフェース 1	ピンヘッダ 50P(2.54mm ピッチ)	コネクタ非搭載

部品番号	インターフェース名	形状	備考
CON5	マイク入力インターフェース	ミニジャック(φ3.5mm)	
CON6	ヘッドホン出力インターフェース	ミニジャック(φ3.5mm)	
CON7	オーディオライン/コンポジットビデオ出力インターフェース	ピンヘッダ 6P(2.54mm ピッチ)	コネクタ非搭載
CON8	拡張インターフェース 2	ピンヘッダ 14P(2.54mm ピッチ)	コネクタ非搭載
CON9	SD インターフェース	SD スロット	
CON10	WLAN インターフェース	BtoB コネクタ 34P(0.5mm ピッチ)	接続可能モジュール: AWL13-U00Z/アットマークテクノ 挿抜寿命: 50 回
CON11	拡張インターフェース 3	ピンヘッダ 14P(2.54mm ピッチ)	コネクタ非搭載
CON12	カメラインターフェース	BtoB コネクタ 60P(0.4mm ピッチ)	接続可能モジュール: OP-A810-CAM01-00/アットマークテクノ 挿抜寿命: 30 回
CON13	拡張インターフェース 4	ピンヘッダ 26P(2.54mm ピッチ)	コネクタ非搭載
CON14	USB インターフェース	USB mini B コネクタ	
JP1	ユーザージャンパ	ピンヘッダ 2P(2.54mm ピッチ)	
JP2	設定ジャンパ	ピンヘッダ 2P(2.54mm ピッチ)	
JP3			
SW1	ユーザースイッチ	タクトスイッチ	
SW2			
SW3			
SW4			
SW5	リセットスイッチ	タクトスイッチ	
LED1	ユーザー LED	LED(黄色、面実装)	
LED2			
LED3			
LED4			
LED5			
LED6	リセット LED	LED(黄色、面実装)	
LED7			

3.4. 接続方法

Armadillo-840 液晶モデルと周辺装置の接続例を次に示します。



- ❶ Armadillo-840 液晶モデル
- ❷ AC アダプタ (5V/2.0A EIAJ#2)^[4]
- ❸ 作業用 PC
- ❹ 開発用 USB シリアル変換アダプタ (Armadillo-800 シリーズ対応)^[4]
- ❺ USB2.0 ケーブル (A-miniB タイプ)^[4]
- ❻ LAN HUB
- ❼ LAN ケーブル
- ❽ HDMI 対応ディスプレイ
- ❾ HDMI ケーブル (A-A タイプ)^[4]
- ❿ SD カード
- ⓫ USB マウス
- ⓬ USB キーボード
- ⓭ スピーカー又はヘッドホン
- ⓮ マイク

図 3.8 Armadillo-840 液晶モデルの接続例

^[4]Armadillo-840 液晶モデル開発セット付属品



開発用 USB シリアル変換アダプタ(Armadillo-800 シリーズ対応)の取扱い上の注意

USB シリアル変換アダプタには電源投入順序があります。Armadillo-840 に接続する際は、以下の手順に従ってご使用ください。接続手順に従わない場合は、USB シリアル変換アダプタが故障する可能性がありますのでご注意ください。

1. 起動中の作業用 PC と USB シリアル変換アダプタを USB2.0 ケーブルで接続します。
2. シリアルインターフェース(Armadillo-840: CON4)に USB シリアル変換アダプタを接続します。
3. 上記接続を確認後、Armadillo-840 に電源を投入します。

また、Armadillo-840 に USB シリアル変換アダプタを接続した状態のまま、作業用 PC または USB シリアル変換アダプタから USB2.0 ケーブルを抜く場合や作業用 PC をシャットダウンする場合は、Armadillo-840 の電源が切断されていることを確認してから行ってください。

3.5. ジャンパピンの設定について

ジャンパの設定を変更することで、Armadillo-840 の動作を変更することができます。ジャンパの機能を「表 3.7. ジャンパの機能」に示します。

表 3.7 ジャンパの機能

ジャンパ	機能	動作
JP1	起動モード設定	オープン: OS を自動起動します。 ショート: ブートローダーを保守モードにします。
JP2	起動デバイス設定	オープン: オンボードフラッシュメモリのブートローダーを起動します。 ショート: SD カードのブートローダーを起動します。

各ジャンパは必要に応じて切り替えの指示があります。ここでは、全てのジャンパをオープンに設定しておきます。

ジャンパピンの位置は「図 3.6. インターフェースレイアウト図」で確認することができます。



ジャンパのオープン、ショートとは



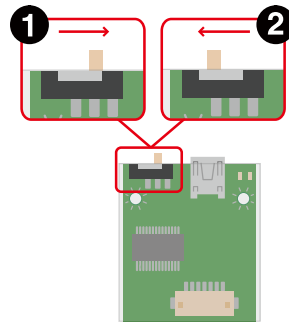
「オープン」とはジャンパピンにジャンパソケットを接続していない状態です。



「ショート」とはジャンパピンにジャンパソケットを接続している状態です。

3.6. スライドスイッチの設定について

開発用 USB シリアル変換アダプタ (Armadillo-800 シリーズ対応) のスライドスイッチには、Armadillo-840 の JP1 と同じ機能が割り当てられています。



- ❶ OS 自動起動モード
- ❷ 保守モード

図 3.9 スライドスイッチの設定

3.7. vi エディタの使用方法

vi エディタは、Armadillo に標準でインストールされているテキストエディタです。本書では、Armadillo の設定ファイルの編集などに vi エディタを使用します。

vi エディタは、ATDE にインストールされてる gedit や emacs などのテキストエディタとは異なり、モードを持っていることが大きな特徴です。vi のモードには、コマンドモードと入力モードがあります。コマンドモードの時に入力した文字はすべてコマンドとして扱われます。入力モードでは文字の入力ができます。

本章で示すコマンド例は ATDE で実行するよう記載していますが、Armadillo でも同じように実行することができます。

3.7.1. vi の起動

vi を起動するには、以下のコマンドを入力します。

```
[ATDE ~]# vi [file]
```

図 3.10 vi の起動

file にファイル名のパスを指定すると、ファイルの編集 (*file* が存在しない場合は新規作成) を行いません。vi はコマンドモードの状態です。

3.7.2. 文字の入力

文字を入力するにはコマンドモードから入力モードへ移行する必要があります。コマンドモードから入力モードに移行するには、「表 3.8. 入力モードに移行するコマンド」に示すコマンドを入力します。入力モードへ移行後は、キーを入力すればそのまま文字が入力されます。

表 3.8 入力モードに移行するコマンド

コマンド	動作
i	カーソルのある場所から文字入力を開始
a	カーソルの後ろから文字入力を開始

入力モードからコマンドモードに戻りたい場合は、ESC キーを入力することで戻ることができます。現在のモードが分からなくなった場合は、ESC キーを入力し、一旦コマンドモードへ戻ることにより混乱を防げます。



日本語変換機能を OFF に

vi のコマンドを入力する時は ATDE の日本語入力システム(Mozc)を OFF にしてください。日本語入力システムの ON/OFF は、半角/全角キーまたは、Shift+Space キーで行うことができます。

「i」、「a」それぞれのコマンドを入力した場合の文字入力の開始位置を「図 3.11. 入力モードに移行するコマンドの説明」に示します。

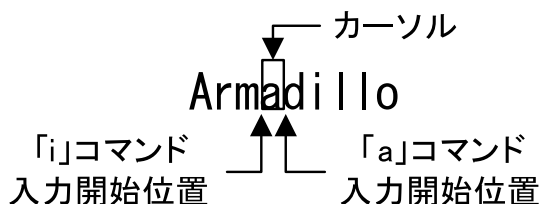



図 3.11 入力モードに移行するコマンドの説明



vi での文字削除

コンソールの環境によっては BS(Backspace)キーで文字が削除できず、「^H」文字が入力される場合があります。その場合は、「3.7.4. 文字の削除」で説明するコマンドを使用し、文字を削除してください。

3.7.3. カーソルの移動

方向キーでカーソルの移動ができますが、コマンドモードで「表 3.9. カーソルの移動コマンド」に示すコマンドを入力することでもカーソルを移動することができます。

表 3.9 カーソルの移動コマンド

コマンド	動作
h	左に 1 文字移動
j	下に 1 文字移動
k	上に 1 文字移動
l	右に 1 文字移動

3.7.4. 文字の削除

文字を削除する場合は、コマンドモードで「表 3.10. 文字の削除コマンド」に示すコマンドを入力します。

表 3.10 文字の削除コマンド

コマンド	動作
x	カーソル上の文字を削除
dd	現在行を削除

「x」コマンド、「dd」コマンドを入力した場合に削除される文字を「図 3.12. 文字を削除するコマンドの説明」に示します。

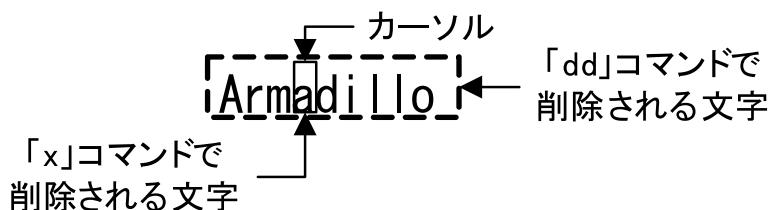


図 3.12 文字を削除するコマンドの説明

3.7.5. 保存と終了

ファイルの保存、終了を行うコマンドを「表 3.11. 保存・終了コマンド」に示します。

表 3.11 保存・終了コマンド

コマンド	動作
:q!	変更を保存せずに終了
:w [file]	ファイル名を file に指定して保存
:wq	ファイルを上書き保存して終了

保存と終了を行うコマンドは「:」(コロン)からはじまるコマンドを使用します。":"キーを入力すると画面下部にカーソルが移り入力したコマンドが表示されます。コマンドを入力した後 Enter キーを押すことで、コマンドが実行されます。


```
Booting Linux on physical CPU 0
Initializing cgroup subsys cpuset
Initializing cgroup subsys cpu
Linux version 3.4-at6 (atmark@atde5) (gcc version 4.6.3 (Debian 4.6.3-14atmark1)
) #1 PREEMPT Mon Jan 27 23:10:37 JST 2014
CPU: ARMv7 Processor [412fc093] revision 3 (ARMv7), cr=10c53c7d
CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing instruction cache
Machine: armadillo840
cma: CMA: reserved 128 MiB at 50000000
Memory policy: ECC disabled, Data cache writeback
bootconsole [early_ttySC2] enabled
Built 1 zonelists in Zone order, mobility grouping on. Total pages: 97536
Kernel command line: console=ttySC2,115200 earlyprintk=sh-sci.2,115200 mem=384M
PID hash table entries: 2048 (order: 1, 8192 bytes)
Dentry cache hash table entries: 65536 (order: 6, 262144 bytes)
Inode-cache hash table entries: 32768 (order: 5, 131072 bytes)
allocated 786432 bytes of page_cgroup
please try 'cgroup_disable=memory' option if you don't want memory cgroups
Memory: 384MB = 384MB total
Memory: 146336k/146336k available, 246880k reserved, 0K highmem
Virtual kernel memory layout:
   vector   : 0xffff0000 - 0xffff1000   (   4 kB)
   fixmap   : 0xffff0000 - 0xffffe000   ( 896 kB)
   vmalloc   : 0xd8800000 - 0xff000000   ( 616 MB)
   lowmem   : 0xc0000000 - 0xd8000000   ( 384 MB)
   pkmap    : 0xbfe00000 - 0xc0000000   (   2 MB)
   modules  : 0xbf000000 - 0xbfe00000   (  14 MB)
   .text    : 0xc0008000 - 0xc052b000   (5260 kB)
   .init    : 0xc052b000 - 0xc0551000   (  152 kB)
   .data    : 0xc0552000 - 0xc058cb60   (  235 kB)
   .bss     : 0xc058cb84 - 0xc05d6bf4   (  297 kB)
NR_IRQS:16 nr_irqs:16 16
sched_clock: 32 bits at 128 Hz, resolution 7812500ns, wraps every 3489660920ms
Console: colour dummy device 80x30
   sh_cmt_simple.10: used as clock source
   sh_cmt_simple.14: used for clock events
   sh_cmt_simple.14: used for periodic clock events
Calibrating delay loop... 1576.53 BogoMIPS (lpj=6156288)
pid_max: default: 32768 minimum: 301
Mount-cache hash table entries: 512
Initializing cgroup subsys cpuacct
Initializing cgroup subsys memory
Initializing cgroup subsys devices
Initializing cgroup subsys freezer
Initializing cgroup subsys blkio
CPU: Testing write buffer coherency: ok
hw perfevents: enabled with ARMv7 Cortex-A9 PMU driver, 7 counters available
Setting up static identity map for 0x403f3788 - 0x403f37bc
dummy:
NET: Registered protocol family 16
DMA: preallocated 256 KiB pool for atomic coherent allocations
pfc: r8a7740_pfc handling gpio 0 -> 858
gpiochip_add: registered GPIOs 0 to 858 on device: r8a7740_pfc
CON7: no extension board found.
L310 cache controller enabled
l2x0: 8 ways, CACHE_ID 0x410000c7, AUX_CTRL 0x42440000, Cache size: 262144 B
hw-breakpoint: found 5 (+1 reserved) breakpoint and 1 watchpoint registers.
hw-breakpoint: maximum watchpoint size is 4 bytes.
```



```
bio: create slab <bio-0> at 0
sdhi0: 3300 mV
SCSI subsystem initialized
usbcore: registered new interface driver usbfs
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
i2c-gpio i2c-gpio.2: using pins 106 (SDA) and 114 (SCL)
i2c-sh_mobile i2c-sh_mobile.0: Runtime PM disabled, clock forced on.
i2c-sh_mobile i2c-sh_mobile.0: I2C adapter 0 with bus speed 100000 Hz
i2c-sh_mobile i2c-sh_mobile.1: Runtime PM disabled, clock forced on.
i2c-sh_mobile i2c-sh_mobile.1: I2C adapter 1 with bus speed 100000 Hz
Linux video capture interface: v2.00
Advanced Linux Sound Architecture Driver Version 1.0.25.
Switching to clocksource sh_cmt_simple.10
  sh_cmt_simple.14: used for oneshot clock events
NET: Registered protocol family 2
IP route cache hash table entries: 4096 (order: 2, 16384 bytes)
TCP established hash table entries: 16384 (order: 5, 131072 bytes)
TCP bind hash table entries: 16384 (order: 4, 65536 bytes)
TCP: Hash tables configured (established 16384 bind 16384)
TCP: reno registered
UDP hash table entries: 256 (order: 0, 4096 bytes)
UDP-Lite hash table entries: 256 (order: 0, 4096 bytes)
NET: Registered protocol family 1
RPC: Registered named UNIX socket transport module.
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
RPC: Registered tcp NFSv4.1 backchannel transport module.
Trying to unpack rootfs image as initramfs...
rootfs image is not initramfs (junk in compressed archive); looks like an initrd
Freeing initrd memory: 105280K
audit: initializing netlink socket (disabled)
type=2000 audit(1.023:1): initialized
VFS: Disk quotas dquot_6.5.2
Dquot-cache hash table entries: 1024 (order 0, 4096 bytes)
squashfs: version 4.0 (2009/01/31) Phillip Lougher
NFS: Registering the id_resolver key type
nfs4filelayout_init: NFSv4 File Layout Driver Registering...
msgmni has been set to 747
Block layer SCSI generic (bsg) driver version 0.4 loaded (major 253)
io scheduler noop registered
io scheduler deadline registered
io scheduler cfq registered (default)
sh-mobile-hdmi sh-mobile-hdmi: Detected HDMI controller 0x1:0xd5
sh_mobile_lcdc_fb sh_mobile_lcdc_fb.1: Runtime PM disabled, clock forced on.
sh_mobile_lcdc_fb sh_mobile_lcdc_fb.1: registered sh_mobile_lcdc_fb.1/mainlcd as
  1920x1080 32bpp.
sh-dma-engine sh-dma-engine.0: Runtime PM disabled, clock forced on.
sh-dma-engine sh-dma-engine.1: Runtime PM disabled, clock forced on.
sh-dma-engine sh-dma-engine.2: Runtime PM disabled, clock forced on.
sh-dma-engine sh-dma-engine.3: Runtime PM disabled, clock forced on.
SuperH SCI(F) driver initialized
sh-sci sh-sci.0: Runtime PM disabled, clock forced on.
sh-sci.0: ttySC0 at MMIO 0xe6c40000 (irq = 132) is a scifa
console [ttySC2] enabled, bootconsole disabled
console [ttySC2] enabled, bootconsole disabled
sh-sci sh-sci.1: Runtime PM disabled, clock forced on.
sh-sci.1: ttySC1 at MMIO 0xe6c50000 (irq = 133) is a scifa
```

```
sh-sci sh-sci.2: Runtime PM disabled, clock forced on.
sh-sci.2: ttySC2 at MMIO 0xe6c60000 (irq = 134) is a scifa
sh-sci sh-sci.3: Runtime PM disabled, clock forced on.
sh-sci.3: ttySC3 at MMIO 0xe6c70000 (irq = 135) is a scifa
sh-sci sh-sci.4: Runtime PM disabled, clock forced on.
sh-sci.4: ttySC4 at MMIO 0xe6c80000 (irq = 136) is a scifa
sh-sci sh-sci.5: Runtime PM disabled, clock forced on.
sh-sci.5: ttySC5 at MMIO 0xe6cb0000 (irq = 137) is a scifa
sh-sci sh-sci.6: Runtime PM disabled, clock forced on.
sh-sci.6: ttySC6 at MMIO 0xe6cc0000 (irq = 138) is a scifa
sh-sci sh-sci.7: Runtime PM disabled, clock forced on.
sh-sci.7: ttySC7 at MMIO 0xe6cd0000 (irq = 139) is a scifa
sh-sci sh-sci.8: Runtime PM disabled, clock forced on.
sh-sci.8: ttySC8 at MMIO 0xe6c30000 (irq = 140) is a scifb
brd: module loaded
loop: module loaded
r8a7740_cec r8a7740_cec.0: Runtime PM disabled, clock forced on.
physmap platform flash device: 08000000 at 04000000
physmap-flash.0: Found 1 x16 devices at 0x0 in 16-bit bank. Manufacturer ID 0x00
0089 Chip ID 0x008967
Intel/Sharp Extended Query Table at 0x010A
Intel/Sharp Extended Query Table at 0x010A
Intel/Sharp Extended Query Table at 0x010A
Intel/Sharp Extended Query Table at 0x010A
Intel/Sharp Extended Query Table at 0x010A
Using buffer write method
Using auto-unlock on power-up/resume
cfi_cmdset_0001: Erase suspend on write enabled
Creating 6 MTD partitions on "physmap-flash.0":
0x000000000000-0x0000000040000 : "bootloader"
0x0000000040000-0x0000000080000 : "config"
0x0000000080000-0x00000000c0000 : "license"
0x00000000c0000-0x000000004c0000 : "firmware"
0x000000004c0000-0x000000008c0000 : "kernel"
0x000000008c0000-0x000000008000000 : "userland"
sh-eth sh-eth: Runtime PM disabled, clock forced on.
sh_mii: probed
Base address at 0xe9a00000, 00:11:0c:16:00:d2, IRQ 142.
pegasus: v0.6.14 (2006/09/27), Pegasus/Pegasus II USB Ethernet driver
usbcore: registered new interface driver pegasus
usbcore: registered new interface driver asix
usbcore: registered new interface driver smsc95xx
ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
rmobile-ehci-driver rmobile-ehci-driver: R-Mobile EHCI
rmobile-ehci-driver rmobile-ehci-driver: new USB bus registered, assigned bus nu
mber 1
rmobile-ehci-driver rmobile-ehci-driver: irq 266, io mem 0xc6701000
rmobile-ehci-driver rmobile-ehci-driver: USB 2.0 started, EHCI 1.00
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 2 ports detected
ohci_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver
rmobile-ohci-driver rmobile-ohci-driver: R-Mobile OHCI
rmobile-ohci-driver rmobile-ohci-driver: new USB bus registered, assigned bus nu
mber 2
rmobile-ohci-driver rmobile-ohci-driver: irq 266, io mem 0xc6700000
hub 2-0:1.0: USB hub found
hub 2-0:1.0: 2 ports detected
Initializing USB Mass Storage driver...
```

```
usbcore: registered new interface driver usb-storage
USB Mass Storage support registered.
mousedev: PS/2 mouse device common for all mice
rtc-s35390a 2-0030: rtc core: registered rtc-s35390a as rtc0
i2c /dev entries driver
uvcvideo: Unable to create debugfs directory
usbcore: registered new interface driver uvcvideo
USB Video Class driver (1.1.1)
sh_mobile_wdt sh_mobile_wdt.0: Runtime PM disabled, clock forced on.
device-mapper: ioctl: 4.22.0-ioctl (2011-10-19) initialised: dm-devel@redhat.com
sh_mobile_sdhi sh_mobile_sdhi.0: Runtime PM disabled, clock forced on.
sh_mobile_sdhi sh_mobile_sdhi.0: Platform OCR mask is ignored
sh_mobile_sdhi sh_mobile_sdhi.0: mmc0 base at 0xe6850000 clock rate 99 MHz
usbcore: registered new interface driver usbhid
usbhid: USB HID core driver
usbcore: registered new interface driver snd-usb-audio
fsi-pcm-audio sh_fsi2: Runtime PM disabled, clock forced on.
sh-mobile-hdmi sh-mobile-hdmi: SH Mobile HDMI Audio Codec
asoc: sh_mobile_hdmi-hifi <-> fsib-dai mapping ok
ip_tables: (C) 2000-2006 Netfilter Core Team
TCP: cubic registered
NET: Registered protocol family 17
VFP support v0.3: implementor 41 architecture 3 part 30 variant 9 rev 3
registered taskstats version 1
rtc-s35390a 2-0030: setting system clock to 2000-01-11 11:21:52 UTC (947589712)
ALSA device list:
 #0: FSI2B-HDMI
RAMDISK: ext2 filesystem found at block 0
RAMDISK: Loading 105281KiB [1 disk] into ram disk... done.
VFS: Mounted root (ext2 filesystem) on device 1:0.
Freeing init memory: 152K
Mounting proc: done
Starting fsck for root filesystem.
fsck 1.25 (20-Sep-2001)
/dev/ram0: clean, 1569/1976 files, 94050/105281 blocks
Checking root filesystem: done
Remounting root rw: done
Mounting usbfs: done
Mounting sysfs: done
Mounting tmpfs on /dev: done
Cleaning up system: done
Running local start scripts.
Creating mtd devnode: done
Loading /etc/config: done
sh_hdmi_hotplug: 4 callbacks suppressed
Starting udevd: done
Mounting devpts: done
Changing file permissions: done
Configure /home/ftp: done
Starting syslogd: done
Starting klogd: done
Mounting firmware on /opt/firmware: done
Mounting license on /opt/license: done
Mounting tmpfs on /tmp, /var/tmp: done
Mounting ramfs on /home/ftp/pub: done
Creating decoder firmware symlink: done
Creating encoder firmware symlink: done
Setting hostname: done
```

```
Starting PVR Server: done
Starting basic firewall: done
Configuring network interfaces: net eth0: attached phy 0 to driver SMSC LAN8710/
LAN8720
udhcpd (v1.20.2) started
Sending discover...
sh_hdmi_hotplug: 1 callbacks suppressed
PHY: sh-eth-ffffffff:00 - Link is Up - 100/Full
Sending discover...
Sending select for 192.168.1.100...
Lease of 192.168.1.100 obtained, lease time 86400
done
Starting inetd: done
Creating avahi.services: done
Starting avahi.daemon: done
Starting lighttpd: done
Starting sshd: failed
(sshd: you will be available to use after run '/etc/init.d/sshd keygen')
Running local start script (/etc/config/rc.local).
Starting photoviewer: done
load decoder firmware: done
acm_h264dec: H.264 Decoder of AV Codec Middleware
acm_aacdec: AAC Decoder of AV Codec Middleware

atmark-dist v1.33.0 (AtmarkTechno/Armadillo-840)
Linux 3.4-at6 [armv7l arch]

armadillo840-0 login:
```

図 4.1 起動ログ

4.2. 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 4.1. 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」に示すユーザでログインすることができます。

表 4.1 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
guest	(なし)	一般ユーザ

4.3. 終了方法

安全に終了させる場合は、次のようにコマンドを実行し、「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
[armadillo ~]#
System is going down for system reboot now.

Starting local stop scripts.
Syncing all filesystems: done
Unmounting all filesystems: done
The system is going down NOW!
Sent SIGTERM to all processes
Sent SIGKILL to all processes
Requesting system halt
System halted.
```

図 4.2 終了方法

SD カードなどのストレージをマウントしていない場合は、電源を切断し終了させることもできます。



ストレージ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。ストレージをアンマウントしてから電源を切断するようご注意ください。

5. 動作確認方法

5.1. ネットワーク

ここでは、ネットワークの設定方法やネットワークを利用するアプリケーションについて説明します。

5.1.1. デフォルト状態のネットワーク設定

ネットワーク設定は、`/etc/config/interfaces` に記述されています。デフォルト状態では、次のように設定されています。

表 5.1 デフォルト状態のネットワーク設定

インターフェース	種類	設定	起動時に有効化
lo	TCP/IP	ループバック	有効
eth0	TCP/IP	DHCP	有効
usb0	TCP/IP	手動	無効

```
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)

auto lo eth0
iface lo inet loopback
iface eth0 inet dhcp
iface usb0 inet manual
    up ifconfig usb0 up
    post-up zcip usb0 /etc/zcip.script > /dev/null
    down ifconfig usb0 down
```

図 5.1 デフォルト状態の/etc/config/interfaces



usb0 は USB ガジェットで利用することを想定した設定です。

5.1.2. ネットワークの有効化、無効化

有効化されていないインターフェースや一度無効化したインターフェースを再度有効化するには、以下のコマンドを使います。

```
[armadillo ~]# ifup eth0
```

図 5.2 ネットワークインターフェース(eth0)の有効化

有効化されているインターフェースを無効化するには、以下のコマンドを使います。設定を変更する前には、かならず無効化してください。

```
[armadillo ~]# ifdown eth0
```

図 5.3 ネットワークインターフェース(eth0)の無効化

コマンドの eth0 を usb0 など他のインターフェース名に変更することで、指定したインターフェースの操作をすることが可能です。

5.1.3. ネットワーク設定の変更方法

Armadillo のネットワーク設定の変更方法について説明します。



ネットワーク接続に関する不明な点については、ネットワークの管理者へ相談してください。

Armadillo 上の「/etc/config」以下にあるファイルを編集し、コンフィグ領域に保存することにより起動時のネットワーク設定を変更することができます。コンフィグ領域の保存については、「6. コンフィグ領域 – 設定ファイルの保存領域」を参照してください。



設定を変更する場合は、かならずネットワークを無効化してから行ってください。変更してからネットワークを無効化しても、「新しい設定」を無効化することになります。「古い設定」が無効化されるわけではありません。

5.1.3.1. 固定 IP アドレスに設定する

「表 5.2. 固定 IP アドレス設定例」に示す内容に設定変更するには、vi エディタで/etc/config/interfaces を、「図 5.4. 固定 IP アドレス設定」のように編集します。

表 5.2 固定 IP アドレス設定例

項目	設定
IP アドレス	192.168.10.10
ネットマスク	255.255.255.0
ネットワークアドレス	192.168.10.0
ブロードキャストアドレス	192.168.10.255
デフォルトゲートウェイ	192.168.10.1

```
[armadillo ~]# vi /etc/config/interfaces
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)

auto lo eth0
iface lo inet loopback
iface eth0 inet static
    address 192.168.10.10
    netmask 255.255.255.0
    network 192.168.10.0
    broadcast 192.168.10.255
    gateway 192.168.10.1
iface usb0 inet manual
    up ifconfig usb0 up
    post-up zcip usb0 /etc/zcip.script > /dev/null
    down ifconfig usb0 down
```

図 5.4 固定 IP アドレス設定

5.1.3.2. DHCP に設定する

DHCP に設定するには、vi エディタで/etc/config/interfaces を、次のように編集します。

```
[armadillo ~]# vi /etc/config/interfaces
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)

auto lo eth0
iface lo inet loopback
iface eth0 inet dhcp
iface usb0 inet manual
    up ifconfig usb0 up
    post-up zcip usb0 /etc/zcip.script > /dev/null
    down ifconfig usb0 down
```

図 5.5 DHCP 設定

5.1.3.3. DNS サーバーを指定する

DNS サーバーを指定する場合は、vi エディタで/etc/config/resolv.conf を編集します。

```
[armadillo ~]# vi /etc/config/resolv.conf
nameserver 192.168.10.1
```

図 5.6 DNS サーバーの設定



DHCP を利用している場合には、DHCP サーバーが DNS サーバーを通知する場合があります。この場合、/etc/config/resolv.conf は自動的に更新されます。

5.1.4. 接続を確認する

ここでは、変更した IP 設定で正常に通信が可能か確認します。設定を変更した後は、かならず変更したインターフェースを再度有効化してください。

同じネットワーク内にある通信機器と PING 通信を行います。下記の例では、通信機器が「192.168.10.20」という IP アドレスを持っていると想定しています。

```
[armadillo ~]# ping 192.168.10.20
```

図 5.7 PING 確認

5.1.5. ファイアーウォール

Armadillo では、簡易ファイアーウォールが動作しています。設定されている内容を参照するには、「図 5.8. iptables」のようにコマンド実行してください。

```
[armadillo ~]# iptables --list
```

図 5.8 iptables

5.1.6. ネットワークアプリケーション

工場出荷イメージで利用することができるネットワークアプリケーションについて説明します。



ATDE と Armadillo のネットワーク設定がデフォルト状態であることを想定して記述しています。ネットワーク設定を変更している場合は適宜読み換えてください。

5.1.6.1. TELNET

ATDE などの PC からネットワーク経由でログインし、リモート操作することができます。ログイン可能なユーザを次に示します。

表 5.3 TELNET でログイン可能なユーザ

ユーザ名	パスワード
guest	(なし)

TELNET を使用して ATDE から Armadillo にリモートログインする場合の例を、次に示します。

```
[ATDE ~]$ telnet 192.168.10.10 ❶
Trying 192.168.10.10...
Connected to 192.168.10.10.
Escape character is '^'.

atmark-dist v1.32.0 (AtmarkTechno/Armadillo-840)
Linux 3.4-at4 [armv7l arch]

armadillo840-0 login: guest ❷
[guest@armadillo ~]$
[guest@armadillo ~]$ su ❸
Password: ❹
[root@armadillo ~]#
[root@armadillo ~]# exit ❺
[guest@armadillo ~]$ exit ❻
Connection closed by foreign host.
[ATDE ~]$
```

- ❶ telnet の引数に Armadillo の IP アドレスを指定します。
- ❷ "guest"と入力するとログインすることができます。パスワードの入力は不要です。
- ❸ 特権ユーザーとなる場合には"su"コマンドを実行します。
- ❹ 特権ユーザーのデフォルトパスワードは"root"です。
- ❺ 特権トユーザーから guest ユーザーに戻る場合は、"exit"と入力します
- ❻ telnet を終了するにはもう一度"exit"を入力します

図 5.9 telnet でリモートログイン

5.1.6.2. FTP

ATDE などの PC からネットワーク経由でファイル転送することができます。次に示すユーザでログインすることができます。

表 5.4 ftp でログイン可能なユーザ

ユーザ名	パスワード
ftp	(なし)

ftp を使用して ATDE から Armadillo にファイルを転送する場合の例を、次に示します。

```
[ATDE ~]$ ls -l file
-rw-r--r-- 1 atmark atmark 1048576 Jan 1 12:00 file
[ATDE ~]$ ftp 192.168.10.10 ❶
Connected to 192.168.10.10.
220 localhost FTP server (GNU inetutils 1.4.1) ready.
Name (192.168.10.10:atmark): ftp
331 Guest login ok, type your name as password.
Password: ❷
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd pub ❸
250 CWD command successful.
ftp> put file ❹
local: file remote: file
200 PORT command successful.
150 Opening BINARY mode data connection for 'file'.
226 Transfer complete.
1048576 bytes sent in 0.14 secs (7399.5 kB/s)
ftp> quit ❺
221 Goodbye.
[ATDE ~]$
```

- ❶ ftp の引数に Armadillo の IP アドレスを指定します。
- ❷ ftp ユーザにパスワードが設定されていないため Enter キーを入力します。
- ❸ ファイル転送することができる pub ディレクトリに移動します。
- ❹ ファイルをアップロードします。ダウンロードする場合は"get"コマンドを使用します。
- ❺ ftp を終了する場合は"quit"と入力します。

図 5.10 ftp でファイル転送

ATDE から Armadillo にファイルをアップロードすると、/home/ftp/pub/ディレクトリ以下にファイルが作成されています。ダウンロードする場合も、同じディレクトリにファイルを配置してください。

```
[armadillo ~]# cd /home/ftp/pub/
[armadillo /home/ftp/pub]# ls
file
```

図 5.11 Armadillo 上でアップロードされたファイルを確認

5.1.6.3. HTTP サーバー

Armadillo では、HTTP サーバーが動作しています。ATDE などの PC の Web ブラウザから Armadillo の URL ([http://\[ArmadilloのIPアドレス\]/](http://[ArmadilloのIPアドレス]/)^[1] または、<http://armadillo840-0.local/>) にアクセスすると、Armadillo のトップページ(index.html)が表示されます。

[1] Armadillo の IP アドレスが 192.168.10.10 の場合、<http://192.168.10.10/> となります。

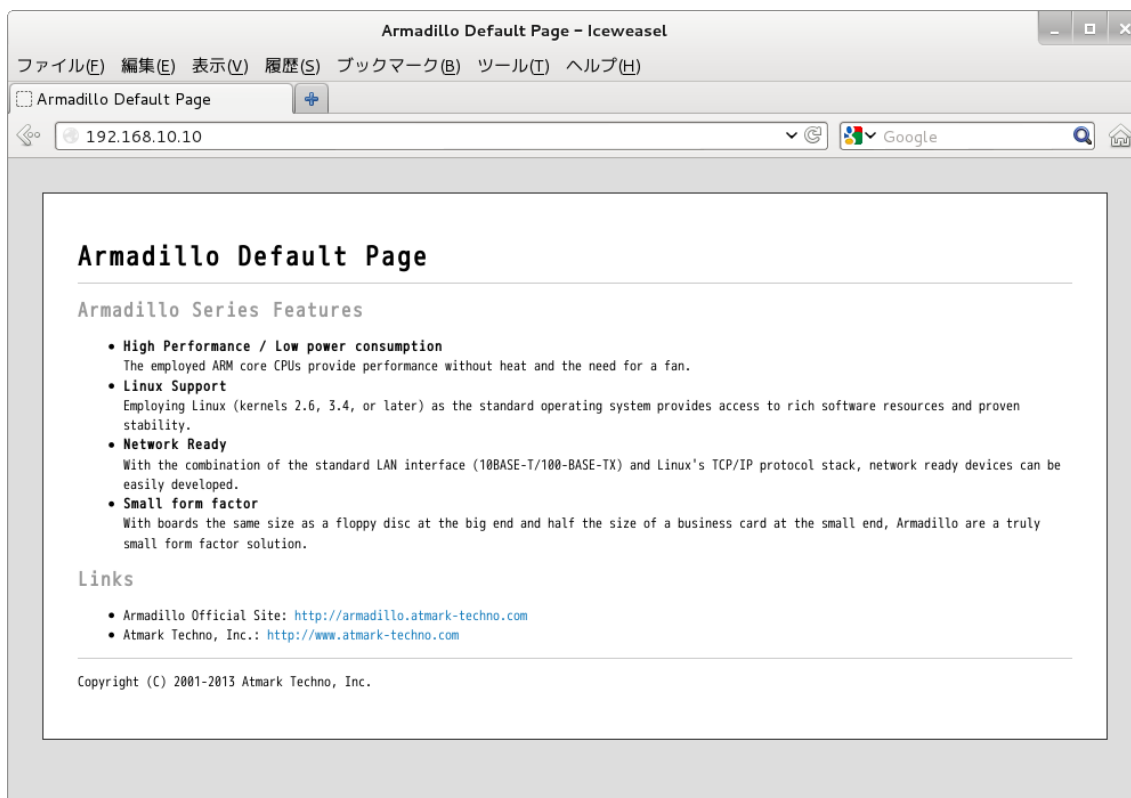


図 5.12 Armadillo トップページ

5.2. ビデオ

Armadillo-840 は画面出力インターフェースを搭載しています。これらのインターフェースは、フレームバッファデバイス(fb)として扱うことができます。

次に、標準状態で利用可能なフレームバッファデバイスを示します。

フレームバッファデバイス - /dev/fb0

HDMI インターフェース(Armadillo-840: CON3)

解像度: 1920 x 1080^[2]

カラーフォーマット: ARGB8888 (32bit)

フレームバッファデバイス - /dev/fb1

LCD インターフェース(拡張ボード 01: CON2)

解像度: 800 x 480

カラーフォーマット: ARGB8888 (32bit)

5.2.1. フレームバッファデバイスにテスト画像を出力

上述した利用可能なフレームバッファデバイスに、テスト画像を出力する方法について説明します。ここでは、テスト画像を生成するために GStreamer の「videotestsrc」を利用します。

^[2]接続する HDMI 対応ディスプレイによって異なる場合があります。

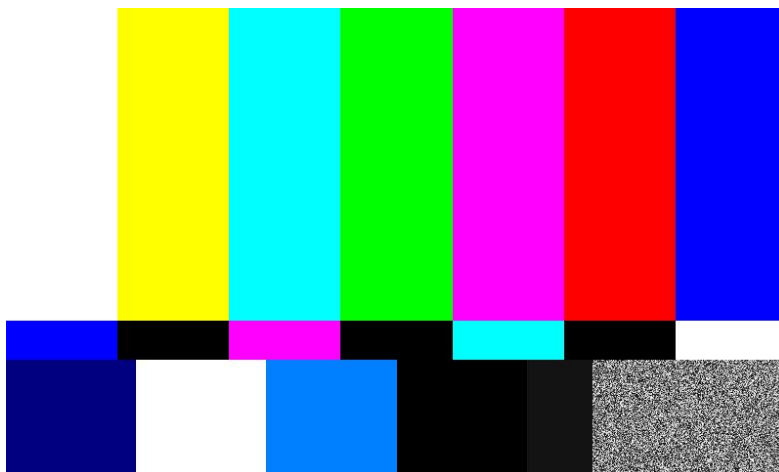


図 5.13 GStreamer のテスト画像

次のようにコマンドを実行すると、指定したフレームバッファデバイスにテスト画像が表示されます。テスト画像の表示を停止する場合は、Ctrl+c を入力してください。

```
[armadillo ~]# gst-launch-1.0 videotestsrc ! \
    "video/x-raw,width=1920,height=1080" ! \ ❶
    fbdevsink device=/dev/fb0 ❷
```

注) 本来は一行のコマンドとして実行します。

- ❶ width, height パラメータには、画面の解像度を指定します。
- ❷ device パラメータには、出力するフレームバッファデバイスを指定します。

図 5.14 テスト画像を表示するコマンド



ユーザーランドイメージ romfs-a840-v1.01.img 以前 (Atmark Dist v20131018 以前) では、次のようにコマンドを実行する必要があります。

```
[armadillo ~]# gst-launch-0.10 videotestsrc ! \
    "video/x-raw-rgb,width=1920,height=1080" ! \
    fbdevsink device=/dev/fb0
```

コマンドの違いは、インストールされている Gstreamer のバージョンによるものです。ユーザーランドイメージ romfs-a840-v1.01.img 以前 (Atmark Dist v20131018 以前) では Gstreamer0.10 がインストールされていましたが、ユーザーランドイメージ romfs-a840-v1.02.img 以降 (Atmark Dist v20140131 以降) では Gstreamer1.0 がインストールされています。



フレームバッファデバイスの解像度がわからない場合、次のように fbset コマンドを用いると現在設定されている解像度を表示することができます。

```
[armadillo ~]# fbset -fb /dev/fb0

mode "1920x1080-30"
    # D: 74.250 MHz, H: 33.750 kHz, V: 30.027 Hz
    geometry 1920 1080 1920 2160 32
    timings 13468 148 88 30 4 44 10
    accel false
    rgba 8/16,8/8,8/0,8/24
endmode
```

5.2.2. HDMI - フレームバッファデバイス /dev/fb0

Armadillo-840 の標準状態では、デフォルトアプリケーションが自動的に起動されるようになっています。このデフォルトアプリケーションは、フレームバッファデバイス /dev/fb0 に描画を行います。そのため、HDMI インターフェース(Armadillo-840: CON3)に HDMI 対応ディスプレイ(本節では単に「ディスプレイ」と称します) を接続し Armadillo-840 を起動した場合には、次のような画面が表示されます。

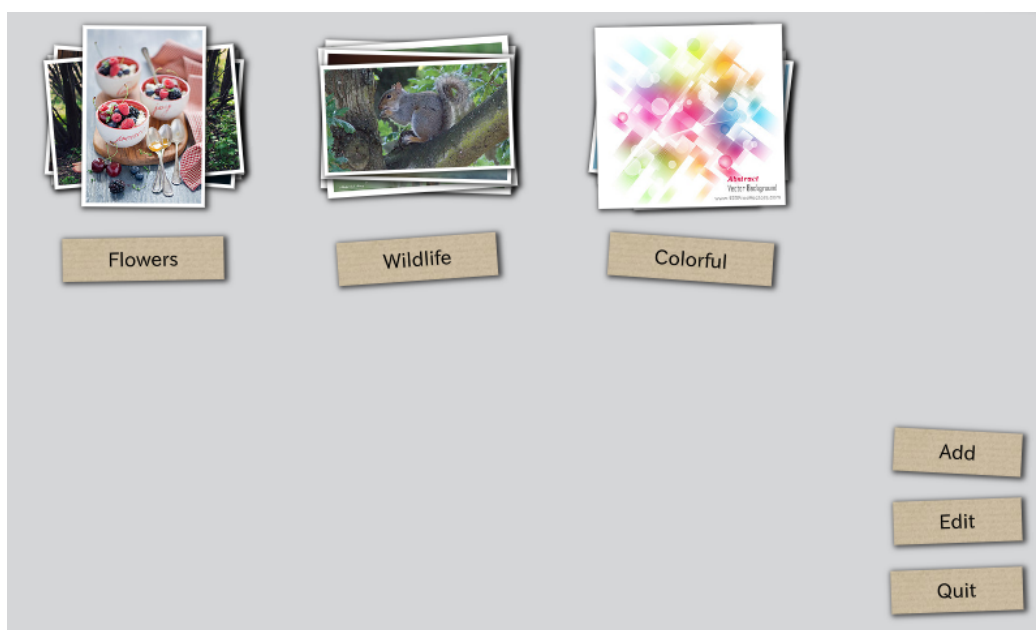


図 5.15 自動起動されるデフォルトアプリケーション画面

このデフォルトアプリケーションは、Qt を利用して作成された「Photo Viewer」というデモアプリケーションです。指定したキーワードに対応する写真を「Flickr」という写真共有サイトから取得します。スタックされた写真をクリックすると、指定したキーワードの写真が画面に広がります。インターネットに繋がっていない場合にはデータを取得できないため、「図 5.15. 自動起動されるデフォルトアプリケーション画面」のように写真を表示することができません。ネットワークの設定については、「5.1. ネットワーク」を確認してください。



Armadillo-840 では、ディスプレイによって自動的にビデオモードを変更する機能が搭載されています。この機能によりフレームバッファデバイス /dev/fb0 の解像度は、Armadillo-840 とディスプレイがサポートで

きる最大の解像度に設定されます。ディスプレイが接続されていない場合は、フレームバッファデバイスの解像度は FullHD (1920 x 1080 px) に設定されます。Armadillo-840 を起動した後に、FullHD に対応していないディスプレイを接続すると、デフォルトアプリケーションが認識している解像度とフレームバッファデバイスに設定されている解像度が異なる場合があります。正常に画面が表示できなくなることがあります。

正常に画面が表示されない場合は、デフォルトアプリケーション「Photo Viewer」を再起動させると解決することがあります。以下のようにコマンドを実行すると、アプリケーションを再起動させることができます。

```
[armadillo ~]# killall qmlscene
[armadillo ~]# /etc/config/rc.local
Starting photoviewer: done
```



利用するディスプレイによっては、ビデオモードの自動設定が完了した後であっても画像が表示されない場合があります。これは、ディスプレイが持つ表示可能なビデオモードを Armadillo-840 が再現できない場合があるためです。

画像が表示されない場合は、次のように該当箇所を変更してください。特定ビデオモードに対する排他処理機能を利用して、表示されないビデオモードを排除することができます。

```
[armadillo ~]# vi /etc/config/configure-fbmode.sh
PARAM=$3

MUST_VMODE_CHANGE=y
IGNORE_MODE_1='1920x1080p-60'
IGNORE_MODE_2='U:' ❶

fbmode_reconfigure() {
    # p_: path
    # s_: strings
```

❶ 'U:'が含まれるビデオモードを排他します。

変更後、次回起動時に設定が反映されるようにコンフィグ領域を保存します。

```
[armadillo ~]# flatfsd -s
```

5.2.3. LCD - フレームバッファデバイス /dev/fb1

Armadillo-840 の標準状態では、LCD インターフェース(拡張ボード 01: CON2)に対応するフレームバッファデバイス /dev/fb1 へ描画を行うアプリケーションが自動起動するように設定されていないため、LCD の画面は黒一色となります。

画面の出力を確認する場合は、「5.2.1. フレームバッファデバイスにテスト画像を出力」にも記載されている、次のようなコマンドを実行してください。

```
[armadillo ~]# gst-launch-1.0 videotestsrc ! \
    "video/x-raw,width=800,height=480" ! \
    fbdevsink device=/dev/fb1
```

図 5.16 LCD にテスト画像を表示するコマンド



ユーザーランドイメージ romfs-a840-v1.01.img 以前 (Atmark Dist v20131018 以前)では、次のようにコマンドを実行する必要があります。

```
[armadillo ~]# gst-launch videotestsrc ! \
    "video/x-raw-rgb,width=800,height=480" ! \
    fbdevsink device=/dev/fb1
```

コマンドの違いは、インストールされている Gstreamer のバージョンによるものです。ユーザーランドイメージ romfs-a840-v1.01.img 以前 (Atmark Dist v20131018 以前)では Gstreamer0.10 がインストールされていましたが、ユーザーランドイメージ romfs-a840-v1.02.img 以降 (Atmark Dist v20140131 以降)では Gstreamer1.0 がインストールされています。

5.2.3.1. バックライトの輝度調整

拡張ボード 01 に搭載された LCD のバックライトは、ソフトウェアで輝度を調整することができます。

LCD のバックライトは、バックライトクラスとして実装されています。バックライトの輝度を変更するには、/sys/class/backlight/pwm-backlight.0/ディレクトリ以下の次の表に示す sysfs ファイルを使用します。

表 5.5 輝度設定に使用する sysfs ファイル

ファイル	説明
brightness	0(消灯) ~ max_brightness(最高輝度)までの数値を書き込むことで輝度を変更することができます。
max_brightness	brightness に書きこむ数値の最大値(最高輝度 = 255)が読み出せます。

次に、バックライトの輝度を調整する場合のコマンド例を示します。

最高輝度を取得する

```
[armadillo ~]# cat /sys/class/backlight/pwm-backlight.0/max_brightness
255
```


消灯させる

```
[armadillo ~]# echo 0 > /sys/class/backlight/pwm-backlight.0/brightness
```

任意の輝度に変更する (ここでは「128」に設定)

```
[armadillo ~]# echo 128 > /sys/class/backlight/pwm-backlight.0/brightness
```

5.3. オーディオ

ここでは、サウンドの再生および録音の方法について説明します。

Linux でオーディオ機能を実現するには、ALSA^[3]と OSS^[4]の 2 つの方法があります。デフォルト設定では、ALSA によるオーディオ機能を提供しています。

利用可能な ALSA デバイスを次に示します。

ALSA デバイス - hw:0

HDMI オーディオインターフェース(Armadillo-840: CON3)
サンプリング周波数: 48k Hz
チャンネル数: 2
フォーマット: Signed 16/24 bit, Little-endian

ALSA デバイス - hw:1

モノラルマイク入力インターフェース(拡張ボード 01: CON5)
ステレオヘッドホン出力インターフェース(拡張ボード 01: CON6)
サンプリング周波数: 48k, 44.1k, 32k, 16k, 8k Hz
チャンネル数: 1 or 2
フォーマット: Signed 16/24 bit, Little-endian

5.3.1. サウンドを再生する

ここでは、テストサウンドを再生する方法を示します。テストサウンドには、Gstreamer の「audiotestsrc」を利用します。

次のようにコマンドを実行すると、ALSA デバイスに対応するオーディオ出力から正弦波(440Hz)の音が再生されます。テストサウンドの再生を停止する場合は、Ctrl+c を入力してください。

^[3]Advanced Linux Sound Architecture <http://alsa.sourceforge.net>

^[4]Open Sound System <http://developer.opensound.com/>

```
[armadillo ~]# gst-launch-1.0 audiotestsrc ! \
    "audio/x-raw,channels=2,rate=48000,width=16" ! \ ❶
    alsasink device=hw:0 ❷
```

注) 本来は一行のコマンドとして実行します。

- ❶ rate パラメータには、サンプリング周波数を指定します。
- ❷ device パラメータには、ALSA デバイスを指定します。

図 5.17 テストサウンドの再生



ユーザーランドイメージ romfs-a840-v1.01.img 以前 (Atmark Dist v20131018 以前) では、次のようにコマンドを実行する必要があります。

```
[armadillo ~]# gst-launch-0.10 audiotestsrc ! \
    "audio/x-raw-int,channels=2,rate=48000,width=16" ! \
    alsasink device=hw:0
```

コマンドの違いは、インストールされている Gstreamer のバージョンによるものです。ユーザーランドイメージ romfs-a840-v1.01.img 以前 (Atmark Dist v20131018 以前) では Gstreamer0.10 がインストールされていましたが、ユーザーランドイメージ romfs-a840-v1.02.img 以降 (Atmark Dist v20140131 以降) では Gstreamer1.0 がインストールされています。

5.3.2. サウンドを録音する

モノラルマイク入力(拡張ボード 01: CON5)に接続されたマイクから入力された音声を録音する方法を示します。ここでは、WAV (RIFF waveform Audio Format) ファイル形式で録音する例を示します。録音を停止する場合は、Ctrl+c を入力してください。

```
[armadillo ~]# gst-launch-1.0 alsasrc device=hw:1 ! \ ❶
    wavenc ! \ ❷
    filesink location=sample.wav ❸
```

注) 本来は一行のコマンドとして実行します。

- ❶ device パラメータには、ALSA デバイスを指定します。
- ❷ ソフトウェアエンコーダに「wavenc」を指定します。他のエンコーダを指定することも可能です。
- ❸ location パラメータには、保存するファイル名を指定します。

図 5.18 サウンドの録音



ユーザーランドイメージ romfs-a840-v1.01.img 以前 (Atmark Dist v20131018 以前) では、次のようにコマンドを実行する必要があります。

```
[armadillo ~]# gst-launch-0.10 alsasrc device=hw:1 ! \
                    wavenc ! \
                    filesink location=sample.wav
```

録音したファイルを再生するには、次のようにコマンドを実行します。ここでは、ステレオヘッドホン出力(拡張ボード 01: CON6) hw:1 に出力するように指定しています。

```
[armadillo ~]# gst-launch-1.0 filesrc location=sample.wav ! \
                    wavparse ! \
                    alsasink device=hw:1
```

図 5.19 録音したファイルを再生



ユーザーランドイメージ romfs-a840-v1.01.img 以前 (Atmark Dist v20131018 以前) では、次のようにコマンドを実行する必要があります。

```
[armadillo ~]# gst-launch-0.10 filesrc location=sample.wav ! \
                    wavparse ! \
                    alsasink device=hw:1
```

5.4. ストレージ

Armadillo-840 でストレージとして使用可能なデバイスを次に示します。

表 5.6 ストレージデバイス


デバイス種類	ディスクデバイス	先頭パーティション
USB フラッシュメモリ	/dev/sd*[a]	/dev/sd*1
SD/SDHC/SDXC カード	/dev/mmcblk*[b]	/dev/mmcblk*p1

[a]USB ハブを利用して複数の USB メモリを接続した場合は、認識された順に sda sdb sdc ... となります。

[b]拡張ボード 01 を接続して 2 つの SD/SDHC/SDXC カードを接続した場合は、認識された順に mmcblk0 mmcblk1 となります。

5.4.1. ストレージの使用方法

ここでは、SDHC カードを例にストレージの使用方法を説明します。以降の説明では、共通の操作が可能な場合に、SD/SDHC/SDXC カードを SD カードと表記します。



SDXC カードを使用する場合は、事前に「5.4.2. ストレージのパーティション変更とフォーマット」を参照してフォーマットを行う必要があります。これは、Linux カーネルが exFAT ファイルシステムを扱うことがで

きないためです。通常、購入したばかりの SDXC カードは exFAT ファイルシステムでフォーマットされています。

Linux では、アクセス可能なファイルやディレクトリは、一つの木構造にまとめられています。あるストレージデバイスのファイルシステムを、この木構造に追加することを、マウントするといいます。マウントを行うコマンドは、mount です。

mount コマンドの典型的なフォーマットは、次の通りです。

```
mount -t fstype device dir
```

図 5.20 mount コマンド書式

-t オプションに続く fstype には、ファイルシステムタイプを指定します^[5]。FAT32 ファイルシステムの場合は vfat^[6]、EXT3 ファイルシステムの場合は ext3 を指定します。

device には、ストレージデバイスのデバイスファイル名を指定します。SD カードのパーティション 1 の場合は /dev/mmcblk0p1、パーティション 2 の場合は /dev/mmcblk0p2 となります。

dir には、ストレージデバイスのファイルシステムをマウントするディレクトリを指定します。

SD スロットに SDHC カードを挿入した状態で「図 5.21. ストレージのマウント」に示すコマンドを実行すると、/mnt ディレクトリに SDHC カードのファイルシステムをマウントします。SD カード内のファイルは、/mnt ディレクトリ以下に見えるようになります。

```
[armadillo ~]# mount -t vfat /dev/mmcblk0p1 /mnt
```

図 5.21 ストレージのマウント



FAT32 ファイルシステムをマウントした場合、次の警告メッセージが表示される場合があります。

```
FAT-fs (mmcblk0p1): utf8 is not a recommended I/O charset for
FAT filesystems, filesystem will be case sensitive!
```

これは無視して構いません。UTF-8 ロケールでは結局はファイル名の表示を正しく処理できないためです。

ストレージを安全に取り外すには、アンマウントする必要があります。アンマウントを行うコマンドは、umount です。オプションとして、アンマウントしたいデバイスがマウントされているディレクトリを指定します。

^[5]ファイルシステムタイプの指定は省略可能です。省略した場合、mount コマンドはファイルシステムタイプを推測します。この推測は必ずしも適切なものとは限りませんので、事前にファイルシステムタイプが分かっている場合は明示的に指定してください。

^[6]通常、購入したばかりの SDHC カードは FAT32 ファイルシステムでフォーマットされています。

```
[armadillo ~]# umount /mnt
```

図 5.22 ストレージのアンマウント

5.4.2. ストレージのパーティション変更とフォーマット

通常、購入したばかりの SDHC カードや USB メモリは、一つのパーティションを持ち、FAT32 ファイルシステムでフォーマットされています。

パーティション構成を変更したい場合、fdisk コマンドを使用します。fdisk コマンドの使用例として、一つのパーティションで構成されている SD カードのパーティションを、2 つに分割する例を「図 5.23. fdisk コマンドによるパーティション変更」に示します。一度、既存のパーティションを削除してから、新たにプライマリパーティションを二つ作成しています。先頭のパーティションには 100MByte、二つめのパーティションに残りの容量を割り当てています。先頭のパーティションは/dev/mmcbkOp1、二つめは/dev/mmcbkOp2 となります。fdisk コマンドの詳細な使い方は、man ページ等を参照してください。

```
[armadillo ~]# fdisk /dev/mmcbk0
```

```
The number of cylinders for this disk is set to 62528.
```

```
There is nothing wrong with that, but this is larger than 1024,  
and could in certain setups cause problems with:
```

- 1) software that runs at boot time (e.g., old versions of LIL0)
- 2) booting and partitioning software from other OSs
(e.g., DOS FDISK, OS/2 FDISK)

```
Command (m for help): d
```

```
Selected partition 1
```

```
Command (m for help): n
```

```
Command action
```

```
  e  extended
```

```
  p  primary partition (1-4)
```

```
p
```

```
Partition number (1-4): 1
```

```
First cylinder (1-62528, default 1):
```

```
Using default value 1
```

```
Last cylinder or +size or +sizeM or +sizeK (1-62528, default 62528): +100M
```

```
Command (m for help): n
```

```
Command action
```

```
  e  extended
```

```
  p  primary partition (1-4)
```

```
p
```

```
Partition number (1-4): 2
```

```
First cylinder (3054-62528, default 3054):
```

```
Using default value 3054
```

```
Last cylinder or +size or +sizeM or +sizeK (3054-62528, default 62528):
```

```
Using default value 62528
```

```
Command (m for help): w
```

```
The partition table has been altered!
```

```
Calling ioctl() to re-read partition table.
```

```
mmcbk0: p1 p2
```

```
mmcblk0: p1 p2
Syncing disks.
```

図 5.23 fdisk コマンドによるパーティション変更

FAT32 ファイルシステムでストレージデバイスをフォーマットするには、mkfs.vfat コマンドを使用します。また、EXT2 や EXT3 ファイルシステムでフォーマットするには、mke2fs コマンドを使用します。SD カードのパーティション 1 を EXT3 ファイルシステムでフォーマットするコマンド例を、次に示します。

```
[armadillo ~]# mke2fs -j /dev/mmcblk0p1
```

図 5.24 EXT3 ファイルシステムの構築

5.5. AV コーデックミドルウェア

AV コーデックミドルウェアを使い、H.264/AVC 及び AAC でエンコードされている動画を再生する方法について説明します。動画は MP4(MPEG-4 Part 14)コンテナに格納されているものを使用します。



AV コーデックミドルウェアには、Atmark Dist v20140131 以降(ユーザーランドイメージ romfs-a840-v1.02.img 以降)、Linux カーネル v3.4-at6 以降(カーネルイメージ linux-a840-v1.02.img.gz 以降)で対応しています。それ以前のものを使用されている場合、本節で説明する動作確認を行う前にイメージを対応バージョンに書き換えてください。

動作確認に利用する動画はサイズが大きいため、Armadillo サイトから取得し、ストレージに保存します。「5.1. ネットワーク」を参照してネットワーク設定を行い、Armadillo からインターネットに接続できる状態にしておいてください^[7]。また、USB メモリや SD カード等のストレージを /mnt にマウントしているという前提で説明を行います。「5.4. ストレージ」を参照して適切なデバイスをマウントしておいてください。

表 5.7 サンプル動画

種類	ファイル名
Full HD サイズ 30 秒動画	big-buck-bunny-30sec-fullhd.mp4
800x480 サイズ 30 動画	big-buck-bunny-30sec-800x480.mp4

次のようにコマンドを実行し、Armadillo サイトから動画ファイルを取得してください。

```
[armadillo ~]# cd /mnt
[armadillo /mnt]# wget http://download.atmark-techno.com/sample/bbb/big-buck-bunny-30sec-
fullhd.mp4
[armadillo /mnt]# wget http://download.atmark-techno.com/sample/bbb/big-buck-
bunny-30sec-800x480.mp4
```

図 5.25 サンプル動画の取得

^[7]動画は開発セット付属の DVD にも収録されています。ネットワークに接続できない環境の場合、そちらをご利用ください。



30 秒動画に使われている Big Buck Bunny は、Creative Commons Attribution 3.0 Unported License で提供されています。(c) copyright 2008, Blender Foundation / www.bigbuckbunny.org



5.5.1. HDMI ディスプレイへの表示

HDMI インターフェース(Armadillo-840: CON3)に接続した HDMI 対応ディスプレイに動画を表示します。本節で示すコマンド例をそのまま実行するためには、HDMI 対応ディスプレイは Full HD サイズの表示に対応している必要があります。対応するフレームバッファデバイスは/dev/fb0 です。

サウンドについても、HDMI 対応ディスプレイに出力します。対応する ALSA デバイスは hw:0 です。

下記コマンドを実行しデフォルトアプリケーション「Photo Viewer」を停止しておいてください。

```
[armadillo ~]# killall qmlscene
```

図 5.26 Photo Viewer の停止

次のようにコマンドを実行すると動画が再生されます。動画の再生を途中で停止する場合は、Ctrl+c を入力してください。

```
[armadillo ~]# gst-launch-1.0 filesrc location=/mnt/big-buck-bunny-30sec-fullhd.mp4 \
! qtdemux name=demux0 \
demux0.audio_0 ! queue ! acmaacdec ! audioresample ! audio/x-raw,rate=48000,channels=2 \
! alsasink device=hw:0 \
demux0.video_0 ! queue ! acmh264dec ! acmfbdevsink device=/dev/fb0
Setting pipeline to PAUSED ...
Pipeline is PREROLLING ...
Pipeline is PREROLLED ...
Setting pipeline to PLAYING ...
New clock: GstAudioSinkClock
```

図 5.27 サンプル動画の再生(HDMI ディスプレイ)



このコマンドはとても長く間違えやすいので、コマンドを本マニュアルからシリアル通信プログラムにコピー&ペーストすることをお勧めします。コマンドプロンプト ([armadillo ~]#) はコピーしないよう注意してください。



HDMI 対応ディスプレイが Full HD サイズの表示に対応していない場合は、解像度を指定しなければ動画が再生されません。解像度が 1280 x 720 の場合のコマンド例を次に示します。

```
[armadillo ~]# gst-launch-1.0 filesrc \
  location=/mnt/big-buck-bunny-30sec-fullhd.mp4 \
  ! qtdemux name=demux0 \
  demux0.audio_0 ! queue ! acmaacdec ! audioresample \
  ! audio/x-raw,rate=48000,channels=2 \
  ! alsasink device=hw:0 \
  demux0.video_0 ! queue ! acmh264dec ! \
  video/x-raw,width=1280,height=720 \
  ! acmfbdevsink device=/dev/fb0
```

5.5.2. LCD への表示 液晶 モデル

LCD インターフェイス(拡張ボード 01: CON2)に接続された LCD に動画を表示します。対応するフレームバッファデバイスは/dev/fb1 です。

サウンドは、ステレオヘッドホン出力インターフェイス(拡張ボード 01: CON6)に接続されたスピーカ又はヘッドホンに出力します。対応する ALSA デバイスは hw:1 です。

次のようにコマンドを実行すると動画が再生されます。動画の再生を途中で停止する場合は、Ctrl+c を入力してください。

```
[armadillo ~]# gst-launch-1.0 filesrc location=/mnt/big-buck-bunny-30sec-800x480.mp4 \
  ! qtdemux name=demux0 \
  demux0.audio_0 ! queue ! acmaacdec ! audioresample ! audio/x-raw,rate=48000,channels=2 \
  ! alsasink device=hw:1 \
  demux0.video_0 ! queue ! acmh264dec ! acmfbdevsink device=/dev/fb1
Setting pipeline to PAUSED ...
Pipeline is PREROLLING ...
Pipeline is PREROLLED ...
Setting pipeline to PLAYING ...
New clock: GstAudioSinkClock
```

図 5.28 サンプル動画の再生(拡張ボード 01)



このコマンドはとて長く間違えやすいので、コマンドを本マニュアルからシリアル通信プログラムにコピー&ペーストすることをお勧めします。コマンドプロンプト ([armadillo ~]#) はコピーしないよう注意してください。

5.6. LED

Armadillo-840 の LED は、GPIO が接続されているためソフトウェアで制御することができます。利用しているデバイスドライバは LED クラスとして実装されているため、LED クラスディレクトリ以下のファイルによって LED の制御を行うことができます。LED クラスディレクトリと各 LED の対応を次に示します。

表 5.8 LED - Armadillo-840

LED クラスディレクトリ	説明	デフォルトトリガ
/sys/class/leds/LED1/	Armadillo-840: LED1	none
/sys/class/leds/LED2/	Armadillo-840: LED2	none

表 5.9 LED - 拡張ボード 01

LED クラスディレクトリ	説明	デフォルトトリガ
/sys/class/leds/EXT1/	拡張ボード 01: LED1	none
/sys/class/leds/EXT2/	拡張ボード 01: LED2	none
/sys/class/leds/EXT3/	拡張ボード 01: LED3	none
/sys/class/leds/EXT4/	拡張ボード 01: LED4	none
/sys/class/leds/EXT5/	拡張ボード 01: LED5	none
/sys/class/leds/EXT6/	拡張ボード 01: LED6	none

以降の説明では、任意の LED を示す LED クラスディレクトリを"/sys/class/leds/[LED]"のように表記します。

5.6.1. LED を点灯/消灯する

LED クラスディレクトリ以下の brightness ファイルへ値を書き込むことによって、LED の点灯/消灯を行うことができます。brightness に書き込む有効な値は 0~255 です。

brightness に 0 以外の値を書き込むと LED が点灯します。

```
[armadillo ~]# echo 1 > /sys/class/leds/[LED]/brightness
```

図 5.29 LED を点灯させる



Armadillo-840 の LED には輝度制御の機能が無いため、0 (消灯)、1~255 (点灯)の 2 つの状態のみ指定することができます。

brightness に 0 を書き込むと LED が消灯します。

```
[armadillo ~]# echo 0 > /sys/class/leds/[LED]/brightness
```

図 5.30 LED を消灯させる

brightness を読み出すと LED の状態が取得できます。

```
[armadillo ~]# cat /sys/class/leds/[LED]/brightness
0
```

図 5.31 LED の状態を表示する

5.6.2. トリガを使用する

LED クラスディレクトリ以下の trigger ファイルへ値を書き込むことによって LED の点灯/消灯にトリガを設定することができます。trigger に書き込む有効な値を次に示します。

表 5.10 trigger の種類

設定	説明
none	トリガを設定しません。
mmc0	SD カードのアクセスランプにします。
timer	任意のタイミングで点灯/消灯を行います。この設定にすることにより、LED クラスディレクトリ以下に delay_on, delay_off ファイルが出現し、それぞれ点灯時間, 消灯時間をミリ秒単位で指定します。
heartbeat ^[a]	心拍のように点灯/消灯を行います。工場出荷イメージでは設定することができません。
default-on ^[a]	主にカーネルから使用します。起動時に LED が点灯します。工場出荷イメージでは設定することができません。

^[a]カーネルコンフィギュレーションで該当トリガを有効にすると設定可能になります。

以下のコマンドを実行すると、LED が 2 秒点灯、1 秒消灯を繰り返します。

```
[armadillo ~]# echo timer > /sys/class/leds/[LED]/trigger
[armadillo ~]# echo 2000 > /sys/class/leds/[LED]/delay_on
[armadillo ~]# echo 1000 > /sys/class/leds/[LED]/delay_off
```

図 5.32 LED のトリガに timer を指定する

trigger を読み出すと LED のトリガが取得できます。"[]"が付いているものが現在のトリガです。

```
[armadillo ~]# cat /sys/class/leds/[LED]/trigger
[none] mmc0 timer
```

図 5.33 LED のトリガを表示する

5.7. RTC

Armadillo-840 には、カレンダー時計(Real Time Clock)が実装されています。電源を切断しても一定時間(平均 300 秒間、最小 60 秒間)時刻を保持することができます

電源が切断されても長時間時刻を保持させたい場合は、RTC 外部バックアップ用電源入力インターフェース(Armadillo-840: CON12)に外付けバッテリー(対応バッテリー例: CR2032 WK11)^[8]を接続することができます。

5.7.1. RTC に時刻を設定する

Linux の時刻には、Linux カーネルが管理するシステムクロックと、RTC が管理するハードウェアクロックの 2 種類があります。RTC に時刻を設定するためには、まずシステムクロックを設定します。その後、ハードウェアクロックをシステムクロックと一致させる手順となります。

システムクロックは、date コマンドを用いて設定します。date コマンドの引数には、設定する時刻を [MMDDhhmmCCYY.ss] というフォーマットで指定します。時刻フォーマットの各フィールドの意味を次に示します。

^[8]詳しくは、各 Armadillo 販売代理店にお問い合わせください。

表 5.11 時刻フォーマットのフィールド

フィールド	意味
MM	月
DD	日(月内通算)
hh	時
mm	分
CC	年の最初の 2 桁(省略可)
YY	年の最後の 2 桁(省略可)
ss	秒(省略可)

2013 年 1 月 23 日 4 時 56 分 00 秒に設定する例を示します。

```
[armadillo ~]# date ❶
Sat Jan 1 09:00:00 JST 2000
[armadillo ~]# date 012304562013.00 ❷
Wed Jan 23 04:56:00 JST 2013
[armadillo ~]# date ❸
Wed Jan 23 04:56:00 JST 2013
```

- ❶ 現在のシステムクロックを表示します。
- ❷ システムクロックを設定します。
- ❸ システムクロックが正しく設定されていることを確認します。

図 5.34 システムクロックを設定

システムクロックを設定後、ハードウェアクロックを hwclock コマンドを用いて設定します。

```
[armadillo ~]# hwclock ❶
Sat Jan 1 00:00:00 2000 0.000000 seconds
[armadillo ~]# hwclock --utc --systohc ❷
[armadillo ~]# hwclock --utc ❸
Wed Jan 23 04:56:10 2013 0.000000 seconds
```

- ❶ 現在のハードウェアクロックを表示します。
- ❷ ハードウェアクロックを協定世界時(UTC)で設定します。
- ❸ ハードウェアクロックが UTC で正しく設定されていることを確認します。

図 5.35 ハードウェアクロックを設定

5.8. GPIO

Armadillo-840 の GPIO は、generic GPIO として実装されています。GPIO クラスディレクトリ以下のファイルによって GPIO の制御を行うことができます。GPIO クラスディレクトリと GPIO の対応を次に示します。

表 5.12 拡張インターフェース 1(Armadillo-840: CON7)の GPIO ディレクトリ

ピン番号	GPIO ディレクトリ
CON7 2 ピン	/sys/class/gpio/gpio195
CON7 3 ピン	/sys/class/gpio/gpio196
CON7 4 ピン	/sys/class/gpio/gpio23
CON7 5 ピン	/sys/class/gpio/gpio21
CON7 6 ピン	/sys/class/gpio/gpio160
CON7 7 ピン	/sys/class/gpio/gpio197
CON7 8 ピン	/sys/class/gpio/gpio198
CON7 9 ピン	/sys/class/gpio/gpio194
CON7 10 ピン	/sys/class/gpio/gpio193
CON7 12 ピン	/sys/class/gpio/gpio62
CON7 13 ピン	/sys/class/gpio/gpio63
CON7 14 ピン	/sys/class/gpio/gpio64
CON7 15 ピン	/sys/class/gpio/gpio65
CON7 16 ピン	/sys/class/gpio/gpio61
CON7 17 ピン	/sys/class/gpio/gpio165
CON7 18 ピン	/sys/class/gpio/gpio164
CON7 19 ピン	/sys/class/gpio/gpio202
CON7 20 ピン	/sys/class/gpio/gpio102
CON7 21 ピン	/sys/class/gpio/gpio59
CON7 22 ピン	/sys/class/gpio/gpio60
CON7 25 ピン	/sys/class/gpio/gpio172
CON7 26 ピン	/sys/class/gpio/gpio173
CON7 27 ピン	/sys/class/gpio/gpio4
CON7 28 ピン	/sys/class/gpio/gpio3
CON7 29 ピン	/sys/class/gpio/gpio2
CON7 30 ピン	/sys/class/gpio/gpio0
CON7 31 ピン	/sys/class/gpio/gpio1
CON7 33 ピン	/sys/class/gpio/gpio66
CON7 34 ピン	/sys/class/gpio/gpio67
CON7 35 ピン	/sys/class/gpio/gpio68
CON7 36 ピン	/sys/class/gpio/gpio69
CON7 37 ピン	/sys/class/gpio/gpio70
CON7 38 ピン	/sys/class/gpio/gpio71
CON7 39 ピン	/sys/class/gpio/gpio72
CON7 40 ピン	/sys/class/gpio/gpio73
CON7 41 ピン	/sys/class/gpio/gpio74
CON7 42 ピン	/sys/class/gpio/gpio75
CON7 43 ピン	/sys/class/gpio/gpio97
CON7 44 ピン	/sys/class/gpio/gpio98
CON7 45 ピン	/sys/class/gpio/gpio99
CON7 46 ピン	/sys/class/gpio/gpio100
CON7 61 ピン	/sys/class/gpio/gpio13
CON7 62 ピン	/sys/class/gpio/gpio12
CON7 63 ピン	/sys/class/gpio/gpio9
CON7 64 ピン	/sys/class/gpio/gpio5
CON7 65 ピン	/sys/class/gpio/gpio20
CON7 66 ピン	/sys/class/gpio/gpio10
CON7 67 ピン	/sys/class/gpio/gpio8
CON7 68 ピン	/sys/class/gpio/gpio7
CON7 70 ピン	/sys/class/gpio/gpio40
CON7 71 ピン	/sys/class/gpio/gpio41
CON7 72 ピン	/sys/class/gpio/gpio42

ピン番号	GPIO ディレクトリ
CON7 73 ピン	/sys/class/gpio/gpio43
CON7 74 ピン	/sys/class/gpio/gpio44
CON7 75 ピン	/sys/class/gpio/gpio45
CON7 76 ピン	/sys/class/gpio/gpio46
CON7 77 ピン	/sys/class/gpio/gpio47
CON7 78 ピン	/sys/class/gpio/gpio48
CON7 79 ピン	/sys/class/gpio/gpio49
CON7 80 ピン	/sys/class/gpio/gpio50
CON7 81 ピン	/sys/class/gpio/gpio51
CON7 82 ピン	/sys/class/gpio/gpio52
CON7 83 ピン	/sys/class/gpio/gpio53
CON7 84 ピン	/sys/class/gpio/gpio54
CON7 85 ピン	/sys/class/gpio/gpio55
CON7 86 ピン	/sys/class/gpio/gpio56
CON7 87 ピン	/sys/class/gpio/gpio57
CON7 88 ピン	/sys/class/gpio/gpio58
CON7 90 ピン	/sys/class/gpio/gpio24
CON7 91 ピン	/sys/class/gpio/gpio25
CON7 92 ピン	/sys/class/gpio/gpio26
CON7 93 ピン	/sys/class/gpio/gpio178
CON7 94 ピン	/sys/class/gpio/gpio179
CON7 95 ピン	/sys/class/gpio/gpio180
CON7 96 ピン	/sys/class/gpio/gpio181
CON7 97 ピン	/sys/class/gpio/gpio182

表 5.13 拡張インターフェース 2(Armadillo-840: CON8)の GPIO ディレクトリ

ピン番号	GPIO ディレクトリ
CON8 4 ピン	/sys/class/gpio/gpio34
CON8 5 ピン	/sys/class/gpio/gpio33
CON8 6 ピン	/sys/class/gpio/gpio32
CON8 7 ピン	/sys/class/gpio/gpio31
CON8 8 ピン	/sys/class/gpio/gpio30
CON8 9 ピン	/sys/class/gpio/gpio29
CON8 10 ピン	/sys/class/gpio/gpio28
CON8 11 ピン	/sys/class/gpio/gpio27
CON8 13 ピン	/sys/class/gpio/gpio35
CON8 15 ピン	/sys/class/gpio/gpio38
CON8 16 ピン	/sys/class/gpio/gpio37
CON8 17 ピン	/sys/class/gpio/gpio39
CON8 19 ピン	/sys/class/gpio/gpio36
CON8 21 ピン	/sys/class/gpio/gpio158
CON8 22 ピン	/sys/class/gpio/gpio159
CON8 27 ピン	/sys/class/gpio/gpio199
CON8 28 ピン	/sys/class/gpio/gpio94
CON8 29 ピン	/sys/class/gpio/gpio93
CON8 30 ピン	/sys/class/gpio/gpio22
CON8 40 ピン	/sys/class/gpio/gpio195
CON8 41 ピン	/sys/class/gpio/gpio196
CON8 42 ピン	/sys/class/gpio/gpio23
CON8 44 ピン	/sys/class/gpio/gpio21
CON8 45 ピン	/sys/class/gpio/gpio160
CON8 46 ピン	/sys/class/gpio/gpio197
CON8 47 ピン	/sys/class/gpio/gpio198

ピン番号	GPIO ディレクトリ
CON8 48 ピン	/sys/class/gpio/gpio194
CON8 49 ピン	/sys/class/gpio/gpio193
CON8 50 ピン	/sys/class/gpio/gpio182
CON8 51 ピン	/sys/class/gpio/gpio181
CON8 52 ピン	/sys/class/gpio/gpio180
CON8 53 ピン	/sys/class/gpio/gpio179
CON8 54 ピン	/sys/class/gpio/gpio178
CON8 55 ピン	/sys/class/gpio/gpio26
CON8 56 ピン	/sys/class/gpio/gpio25
CON8 57 ピン	/sys/class/gpio/gpio24

以降の説明では、任意の GPIO を示す GPIO クラスディレクトリを"/sys/class/gpio/[GPIO]"のように表記します。

5.8.1. 入出力方向を変更する

GPIO ディレクトリ以下の direction ファイルへ値を書き込むことによって、入出力方向を変更することができます。direction に書き込む有効な値を次に示します。

表 5.14 direction の設定

設定	説明
high	入出力方向を OUTPUT に設定します。出力レベルの取得/設定を行うことができます。出力レベルは HIGH レベルになります。
out	入出力方向を OUTPUT に設定します。出力レベルの取得/設定を行うことができます。出力レベルは LOW レベルになります。
low	out を設定した場合と同じです。
in	入出力方向を INPUT に設定します。入力レベルの取得を行うことができますが設定はできません。

```
[armadillo ~]# echo in > /sys/class/gpio/[GPIO]/direction
```

図 5.36 GPIO の入出力方向を設定する(INPUT に設定)

```
[armadillo ~]# echo out > /sys/class/gpio/[GPIO]/direction
```

図 5.37 GPIO の入出力方向を設定する(OUTPUT に設定)

5.8.2. 入力レベルを取得する

GPIO ディレクトリ以下の value ファイルから値を読み出すことによって、入力レベルを取得することができます。"0"は LOW レベル、"1"は HIGH レベルを表わします。入力レベルの取得は入出力方向が INPUT, OUTPUT のどちらでも行うことができます。入出力方向が OUTPUT の時に読み出される値は、GPIO ピンの状態ではなく、自分が value ファイルに書き込んだ値となります。

```
[armadillo ~]# cat /sys/class/gpio/[GPIO]/value
0
```

図 5.38 GPIO の入力レベルを取得する

5.8.3. 出力レベルを設定する

GPIO ディレクトリ以下の value ファイルへ値を書き込むことによって、出力レベルを設定することができます。"0"は LOW レベル、"0"以外は HIGH レベルを表わします。出力レベルの設定は入出力方向が OUTPUT でなければ行うことはできません。

```
[armadillo ~]# echo 1 > /sys/class/gpio/[GPIO]/value
```

図 5.39 GPIO の出力レベルを設定する

5.8.4. ユーザージャンパを使用する

拡張ボード 01 のジャンパピン JP1 はユーザージャンパとして使用できます。ユーザージャンパは generic GPIO として実装されており、GPIO と同様の操作でユーザージャンパの入力レベルを取得することができます。対応する GPIO ディレクトリは /sys/class/gpio/gpio75 です。

ユーザージャンパの状態と取得できる値の対応表を次に示します。

表 5.15 ユーザージャンパの状態と取得できる値の対応

ユーザージャンパの状態	取得できる値
ショート	0
オープン	1

5.8.4.1. 状態を取得する

ユーザージャンパの状態を取得するには、direction を in に設定する必要があります。詳しい GPIO のアクセス方法については、「5.8.1. 入出力方向を変更する」および「5.8.2. 入力レベルを取得する」を参考にしてください。

ユーザージャンパの状態を取得する例を次に示します。

```
[armadillo ~]# echo in > /sys/class/gpio/gpio75/direction
[armadillo ~]# cat /sys/class/gpio/gpio75/value
1
```

図 5.40 ユーザージャンパの状態を取得する



ユーザージャンパを利用して、起動時に行う処理を変更することができます。ここでは例として、Qt サンプルアプリケーションの photoviewer の画面出力先を次のように変更してみます。

ユーザージャンパの状態	画面出力先
オープン	HDMI 対応ディスプレイ
ショート	LCD

/etc/config/rc.local を次のように編集します。

```
[armadillo ~]# vi /etc/config/rc.local
```

```
# First read /etc/profile
test -f /etc/profile && . /etc/profile

JP1=/sys/class/gpio/gpio75
echo in > ${JP1}/direction ❶
if [ $(cat ${JP1}/value) -eq 0 ]; then ❷
    export QT_QPA_EGLFS_DISPLAY=1 ❸
    export QT_QPA_EGLFS_WIDTH=800 ❹
    export QT_QPA_EGLFS_HEIGHT=480 ❺
fi

#
# Starting a default application
#
```

- ❶ 入出力方向を INPUT に設定します。
- ❷ 入力レベルを取得し、処理を分岐します。
- ❸ 画面出力先を LCD に設定します。
- ❹ 画面の幅を 800 ピクセルに設定します。
- ❺ 画面の高さを 480 ピクセルに設定します。

変更後、次回起動時に設定が反映されるようにコンフィグ領域を保存します。

```
[armadillo ~]# flatfsd -s
```

5.9. ユーザースイッチ

拡張ボード 01 に搭載されているユーザースイッチ(SW1~SW4)のボタン押し/リリースイベントを取得する方法について説明します。

ユーザースイッチのデバイスドライバは、インプットデバイスとして実装されています。そのため、インプットデバイスのデバイスファイルから各イベントを取得することができます。

Armadillo-840 で利用可能なユーザースイッチのインプットデバイスファイルと、各スイッチに対応したイベントコードを次に示します。

表 5.16 インプットデバイスファイルとイベントコード

ユーザースイッチ	インプットデバイスファイル	イベントコード
SW1	/dev/input/event1	116 (Power)
SW2		158 (Back)
SW3		139 (Menu)
SW4		102 (Home)



インプットデバイスは検出された順番にインデックスが割り振られます。USB デバイスなどを接続してインプットデバイスを追加している場合は、デバイスファイルのインデックスが異なる可能性があります。

5.9.1. イベントを確認する

ユーザースイッチのボタンプッシュ/リリースイベントは、インプットデバイスファイルから取得することができます。ここでは、「evtest」コマンドを利用してイベントを確認します。evtest を停止するには、Ctrl+c を入力してください。

```
[armadillo ~]# evtest /dev/input/event1
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100
Input device name: "gpio-keys"
Supported events:
  Event type 0 (Sync)
  Event type 1 (Key)
    Event code 102 (Home)
    Event code 116 (Power)
    Event code 139 (Menu)
    Event code 158 (Back)
Testing ... (interrupt to exit)
Event: time 946777243.784020, type 1 (Key), code 116 (Power), value 1 ❶
Event: time 946777243.784028, ----- Report Sync -----
Event: time 946777243.895528, type 1 (Key), code 116 (Power), value 0 ❷
Event: time 946777243.895534, ----- Report Sync -----
:
[armadillo ~]#
```

- ❶ SW1 のボタンプッシュイベントを検出したときの表示。
- ❷ SW1 のボタンリリースイベントを検出したときの表示。

図 5.41 ユーザースイッチ: イベントの確認

5.10. タッチスクリーン

拡張ボード 01 には、タッチパネル LCD が搭載されています。ソフトウェアでタッチイベントを取得することができます。

タッチスクリーンドライバは、インプットデバイスとして実装されています。そのため、インプットデバイスのデバイスファイルからタッチイベントを取得することができます。デバイスファイルは、/dev/input/event0 です。



インプットデバイスは検出された順番にインデックスが割り振られます。USB デバイスなどを接続してインプットデバイスを追加している場合は、デバイスファイルのインデックスが異なる可能性があります。

5.10.1. イベントを確認する

タッチイベントは、インプットデバイスファイルから取得することができます。ここでは、「evtest」コマンドを利用してイベントを確認します。evtest を停止するには、Ctrl+c を入力してください。

```
[armadillo ~]# evtest /dev/input/event0
Input driver version is 1.0.1
```

```
Input device ID: bus 0x18 vendor 0x0 product 0x0 version 0x0
Input device name: "st1232-touchscreen"
Supported events:
  Event type 0 (Sync)
  Event type 1 (Key)
  Event type 3 (Absolute)
    Event code 48 (Touch Major)
      Value      0
      Min        0
      Max       255
    Event code 53 (Position X)
      Value      0
      Min        0
      Max       799
    Event code 54 (Position Y)
      Value      0
      Min        0
      Max       479
Testing ... (interrupt to exit)
Event: time 946699216.584437, type 3 (Absolute), code 48 (Touch Major), value 73 ❶
Event: time 946699216.584446, type 3 (Absolute), code 53 (Position X), value 476 ❷
Event: time 946699216.584451, type 3 (Absolute), code 54 (Position Y), value 251 ❸
Event: time 946699216.584456, ----- Config Sync -----
Event: time 946699216.584460, ----- Report Sync -----
      :
[armadillo ~]#
```

- ❶ タッチされている楕円の直径を表すイベントの表示。
- ❷ タッチされているポイントの X 座標を表すイベントの表示。
- ❸ タッチされているポイントの Y 座標を表すイベントの表示。

図 5.42 タッチスクリーン: イベントの確認

6. コンフィグ領域 – 設定ファイルの保存領域

コンフィグ領域は、設定ファイルなどを保存しハードウェアのリセット後にもデータを保持することができるフラッシュメモリ領域です。コンフィグ領域からのデータの読出し、またはコンフィグ領域への書込みは、flatfsd コマンドを使用します。

6.1. コンフィグ領域の読出し

コンフィグ領域を読み出すには以下のコマンドを実行します。読み出されたファイルは、「/etc/config」ディレクトリに作成されます。

```
[armadillo ~]# flatfsd -r
```

図 6.1 コンフィグ領域の読出し方法



デフォルトのソフトウェアでは、起動時に自動的にコンフィグ領域の読出しを行うように設定されています。コンフィグ領域の情報が壊れている場合、「/etc/default」ディレクトリの内容が反映されます。

6.2. コンフィグ領域の保存

コンフィグ領域を保存するには以下のコマンドを実行します。保存されるファイルは、「/etc/config」ディレクトリ以下のファイルです。

```
[armadillo ~]# flatfsd -s
```

図 6.2 コンフィグ領域の保存方法



コンフィグ領域の保存をおこなわない場合、「/etc/config」ディレクトリ以下のファイルへの変更は電源遮断時に失われます。

6.3. コンフィグ領域の初期化

コンフィグ領域を初期化するには以下のコマンドを実行します。初期化時には、「/etc/default」ディレクトリ以下のファイルがコンフィグ領域に保存され、且つ「/etc/config」ディレクトリにファイルが複製されます。

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

図 6.3 コンフィグ領域の初期化方法

7. ユーザー登録

アットマークテクノ製品をご利用のユーザーに対して、購入者向けの限定公開データの提供や大切なお知らせをお届けするサービスなど、ユーザー登録すると様々なサービスを受けることができます。サービスを受けるためには、「アットマークテクノ ユーザーズサイト」にユーザー登録をする必要があります。

ユーザー登録すると次のようなサービスを受けることができます。

- ・ 製品仕様や部品などの変更通知の閲覧・配信
- ・ 購入者向けの限定公開データのダウンロード
- ・ 該当製品のバージョンアップに伴う優待販売のお知らせ配信
- ・ 該当製品に関する開発セミナーやイベント等のお知らせ配信

詳しくは、「アットマークテクノ ユーザーズサイト」をご覧ください。

アットマークテクノ ユーザーズサイト

<https://users.atmark-techno.com/>

7.1. 購入製品登録

ユーザー登録完了後に、購入製品登録することで、「購入者向けの限定公開データ^[1]」をダウンロードすることができるようになります。

Armadillo-840 購入製品登録

<https://users.atmark-techno.com/armadillo-840/register>

Armadillo-840 の購入製品登録を行うには、Armadillo-840 から取り出した「正規認証ファイル」をアットマークテクノ ユーザーズサイトからアップロードする必要があります。Armadillo-840 から正規認証ファイル(board-info.txt)を取り出す手順は以下の通りです。

7.1.1. 正規認証ファイルを取り出す手順

Armadillo にログインし、コマンドを実行すると正規認証ファイルが生成されます。そのファイルをお使いの Web ブラウザを使ってダウンロードしてください。

1. ATDE で minicom を立ち上げて、Armadillo-840 に root ユーザーでログインします。デバイスファイル名(/dev/ttyUSB0)は、ご使用の環境により ttyUSB1 や ttyS0、ttyS1 などになる場合があります。Armadillo に接続されているシリアルポートのデバイスファイルを指定してください。

```
atmark@atde5:~$ LANG=C minicom --noinit --wrap --device /dev/ttyUSB0
```

^[1] 拡張ボード 01 の回路図データや、ミドルウェアパッケージなど

```
armadillo840-0 login: root
Password:
[root@armadillo840-0 (ttySC2) ~]#
```

2. "get-board-info-a840"コマンドを実行して正規認証ファイル(board-info.txt)を作成します。

```
[root@armadillo840-0 (ttySC2) ~]# get-board-info-a840
[root@armadillo840-0 (ttySC2) ~]# ls
board-info.txt
[root@armadillo840-0 (ttySC2) ~]#
```

3. Armadillo 上で動いている WEB サーバーがアクセスできる場所に、正規認証ファイルを移動し、アクセス権限を変更します。

```
[root@armadillo840-0 (ttySC2) ~]# mv board-info.txt /home/www-data/
[root@armadillo840-0 (ttySC2) ~]# chmod a+r /home/www-data/board-info.txt
```

4. minicom を終了させ、お使いの Web ブラウザから、Armadillo の URL にアクセスしてください。下記どちらかの指定方法でアクセス可能です。

```
http://armadillo840-0.local/board-info.txt
http://[Armadillo の IP アドレス]/board-info.txt [2]
```

取り出した正規認証ファイルを「Armadillo-840 購入製品登録」ページの「正規認証ファイル」欄に指定し、アップロードしてください。

^[2] Armadillo の IP アドレスが 192.168.10.10 の場合、http://192.168.10.10/board-info.txt となります。

改訂履歴

バージョン	年月日	改訂内容
1.0.0	2013/08/09	・ 初版発行
1.1.0	2014/01/29	・ AV コーデックミドルウェアに対応 ・ 「表 5.14. direction の設定」の内容を修正 ・ 誤記、表記ゆれ修正
1.2.0	2014/09/19	・ 読みやすさ向上のための修正 ・ 誤記、表記ゆれ修正

Armadillo-840 液晶モデル開発セットスタートアップガイド
Version 1.2.0
2014/09/19

株式会社アットマークテクノ

札幌本社

〒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

横浜営業所

〒221-0835 横浜市神奈川区鶴屋町3丁目 30-4 明治安田生命横浜西口ビル 7F
TEL 045-548-5651 FAX 050-3737-4597
