

# Armadillo-EVA 1500 製品マニュアル

AE1500-D00Z

Version 1.1.2  
2015/07/27

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-EVA 1500 製品マニュアル

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 © 2014-2015 Atmark Techno, Inc.

Version 1.1.2  
2015/07/27

---

# 目次

1. はじめに .....	15
1.1. ユーザー限定コンテンツ .....	15
1.2. 本書および関連ファイルについて .....	15
1.3. 本書の構成 .....	15
1.4. 表記について .....	16
1.4.1. フォント .....	16
1.4.2. コマンド入力例 .....	16
1.4.3. アイコン .....	16
1.5. 謝辞 .....	17
2. 注意事項 .....	18
2.1. 製品本体開封についてのご注意 .....	18
2.2. 評価ボードについてのご注意 .....	18
2.3. 安全に関する注意事項 .....	18
2.4. 取扱い上の注意事項 .....	19
2.5. ソフトウェア使用に関する注意事項 .....	20
2.6. 書込み禁止領域について .....	20
2.7. 保証について .....	21
2.8. 輸出について .....	21
2.9. 商標について .....	21
3. 製品概要 .....	22
3.1. ボード概要 .....	22
3.2. 仕様 .....	22
3.3. ブロック図 .....	24
3.4. インターフェースレイアウト .....	25
3.5. ソフトウェア構成 .....	26
3.6. メモリマップ .....	27
3.7. パーティション構成 .....	27
4. 電源を入れる前に .....	29
4.1. 準備するもの .....	29
4.2. 開発/動作確認環境の構築 .....	29
4.2.1. ATDE5 セットアップ .....	30
4.2.2. 取り外し可能デバイスの使用 .....	34
4.2.3. コマンドライン端末(GNOME 端末)の起動 .....	34
4.2.4. シリアル通信ソフトウェア(minicom)の使用 .....	35
4.3. 組み立て .....	36
4.4. 接続方法 .....	37
4.5. 機能選択スイッチの設定について .....	39
5. 起動と終了 .....	40
5.1. 起動 .....	40
5.2. ログイン .....	46
5.3. 終了方法 .....	46
6. 各種機能を使用する .....	48
6.1. ネットワーク .....	48
6.2. 時刻設定 .....	49
6.3. パッケージ管理 .....	50
6.4. GStreamer .....	52
6.4.1. GStreamer のインストール .....	52
6.4.2. GStreamer の使用 .....	53
6.5. X サーバー .....	53
6.5.1. X サーバーのインストール .....	53

6.5.2. X サーバーの起動 .....	54
6.6. ストレージ .....	55
6.6.1. ストレージの使用法 .....	56
6.6.2. ストレージのパーティション変更とフォーマット .....	57
6.7. ビデオ .....	58
6.7.1. ディスプレイに画像を表示する .....	58
6.7.2. カメラ画像をディスプレイに表示する .....	59
6.7.3. ディスプレイのビデオモードを設定する .....	60
6.8. オーディオ .....	60
6.8.1. サウンドを再生する .....	61
6.8.2. サウンドを録音する .....	61
6.9. マルチメディアパッケージ .....	62
6.9.1. Linux カーネルの起動オプションを設定する .....	64
6.9.2. 動画を再生する .....	64
6.9.3. 2つの動画を同時に再生する .....	66
6.10. シリアル .....	68
6.10.1. シリアルコンソールとして使用する .....	68
6.11. 温度センサー .....	69
6.12. LED .....	69
6.12.1. LED を点灯/消灯する .....	70
6.12.2. トリガを使用する .....	70
6.13. ユーザースイッチ .....	71
6.13.1. イベントを確認する .....	71
7. Linux カーネル仕様 .....	73
7.1. Device Tree .....	73
7.2. デフォルトコンフィギュレーション .....	73
7.3. デフォルト起動オプション .....	73
7.4. Linux ドライバ一覧 .....	74
7.4.1. Armadillo-EVA 1500 .....	74
7.4.2. ピンコントロール .....	74
7.4.3. タイマー .....	75
7.4.4. SPI フラッシュメモリ .....	75
7.4.5. UART .....	76
7.4.6. Ethernet .....	77
7.4.7. MMC ホスト .....	77
7.4.8. SD ホスト .....	78
7.4.9. USB 3.0 ホスト .....	78
7.4.10. USB 2.0 ホスト .....	79
7.4.11. USB ファンクション .....	80
7.4.12. デジタル HD 出力 .....	81
7.4.13. コンポジットビデオ入力 .....	82
7.4.14. アナログオーディオ .....	83
7.4.15. マルチメディアパッケージ .....	84
7.4.16. SATA .....	85
7.4.17. PCI Express .....	86
7.4.18. 温度センサー .....	86
7.4.19. リアルタイムクロック .....	87
7.4.20. LED .....	87
7.4.21. ユーザースイッチ .....	88
7.4.22. I2C .....	89
7.4.23. LCD .....	90
7.4.24. LVDS .....	91
8. ブートローダー仕様 .....	93

8.1. 起動シーケンス .....	93
8.2. 2nd ブートローダーイメージの選択 .....	94
8.3. Hermit-At の起動モード .....	94
8.4. Hermit-At の機能 .....	95
8.4.1. コンソールの指定方法 .....	95
8.4.2. Linux カーネルイメージの指定方法 .....	96
8.4.3. Linux カーネルの起動オプション .....	96
9. ビルド手順 .....	98
9.1. Linux カーネルをビルドする .....	98
9.2. ブートローダーをビルドする .....	99
9.2.1. IPL をビルドする .....	99
9.2.2. Hermit-At をビルドする .....	99
9.3. ビルドしたイメージのインストール .....	100
9.3.1. SPI フラッシュメモリへの書き込み方法 .....	100
9.3.2. eMMC への書き込み方法 .....	105
10. SD ブートの活用 .....	106
10.1. ブートディスクの作成 .....	106
10.2. ルートファイルシステムの構築 .....	110
10.3. Linux カーネルイメージの配置 .....	112
10.4. SD ブートの実行 .....	113
11. リカバリ手順 .....	115
11.1. パーティション構成 .....	115
11.1.1. eMMC のパーティション構成 .....	115
11.1.2. SPI フラッシュメモリのパーティション構成 .....	116
11.2. 内蔵ストレージ全体をリカバリする .....	116
11.2.1. リカバリディスクの作成 .....	116
11.2.2. リカバリを実行する .....	121
11.3. eMMC の Debian GNU/Linux ルートファイルシステムをリカバリする .....	121
11.3.1. Debian GNU/Linux のリカバリに必要なファイルの取得 .....	121
11.3.2. Debian GNU/Linux のリカバリを実行する .....	122
11.4. SPI フラッシュメモリの特定パーティションをリカバリする .....	123
11.4.1. SPI フラッシュメモリのリカバリに必要なファイルの取得 .....	123
11.4.2. ブートローダーのリカバリを実行する .....	123
12. JTAG ICE を利用する .....	125
12.1. 準備 .....	125
12.2. 機能選択スイッチの設定 .....	125
12.3. 接続確認 .....	125
12.4. 各種デバッグへの対応について .....	125
13. ハードウェア仕様 .....	126
13.1. プロセッサ .....	126
13.1.1. 特長 .....	126
13.1.2. モード設定 .....	126
13.2. RAM .....	129
13.3. eMMC .....	129
13.4. SPI フラッシュメモリ .....	129
13.5. LAN(Ethernet) .....	130
13.6. 無線 LAN .....	130
13.7. シリアル(UART) .....	131
13.8. USB シリアル .....	132
13.9. SD .....	132
13.9.1. CON4 .....	132
13.9.2. CON6 .....	133
13.10. USB 2.0 .....	134

13.11. USB 3.0 .....	135
13.12. PCI Express .....	136
13.13. シリアル ATA .....	137
13.14. コンポジットビデオ入力 .....	138
13.15. LCD .....	138
13.16. LVDS .....	139
13.17. デジタル HD 出力 .....	140
13.18. オーディオ .....	141
13.19. 拡張インターフェース .....	142
13.19.1. CON2 .....	142
13.19.2. CON3 .....	143
13.19.3. CON15 .....	144
13.19.4. CON22 .....	144
13.19.5. CON24 .....	145
13.19.6. CON30 .....	145
13.20. I2C .....	146
13.21. クロック .....	147
13.22. スイッチ .....	147
13.22.1. 機能選択スイッチ .....	148
13.22.2. ユーザースイッチ .....	151
13.22.3. SD ブートデバイス選択スイッチ .....	151
13.22.4. 電源スイッチ .....	151
13.22.5. リセットスイッチ .....	152
13.23. LED .....	152
13.23.1. シリアル LED .....	152
13.23.2. ユーザー LED .....	152
13.23.3. 電源 LED .....	153
13.23.4. 警告 LED .....	153
13.24. JTAG .....	154
13.25. リアルタイムクロック .....	154
13.26. 電源 .....	155
13.26.1. 電源回路の構成 .....	155
13.26.2. 電源シーケンス .....	156
13.27. リセット .....	157
14. 電氣的仕様 .....	158
14.1. 絶対最大定格 .....	158
14.2. 推奨動作条件 .....	158
15. インターフェース仕様 .....	159
15.1. インターフェースレイアウト .....	159
15.2. CON1 JTAG インターフェース .....	161
15.3. CON2 拡張インターフェース .....	162
15.4. CON3 拡張インターフェース .....	163
15.5. CON4 SD インターフェース .....	164
15.6. CON5 WLAN インターフェース .....	164
15.7. CON6 SD インターフェース .....	165
15.8. CON7 LAN インターフェース .....	165
15.9. CON8 USB 2.0 ホストインターフェース .....	166
15.10. CON9 USB 2.0 ファンクションインターフェース .....	166
15.11. CON10 USB 3.0 ホストインターフェース .....	166
15.12. CON11 SATA インターフェース .....	166
15.13. CON12 miniPCIe インターフェース .....	167
15.14. CON13 電源出力インターフェース .....	168
15.15. CON14 USB シリアルインターフェース .....	168

15.16. CON15 拡張インターフェース .....	168
15.17. CON16 シリアルインターフェース .....	169
15.18. CON17 デジタル HD 出力インターフェース .....	169
15.19. CON18 ヘッドホンインターフェース .....	169
15.20. CON21 マイク/ライン入力インターフェース .....	170
15.21. CON22 拡張インターフェース .....	170
15.22. CON23 LCD インターフェース .....	170
15.23. CON24 拡張インターフェース .....	172
15.24. CON25 LVDS インターフェース .....	172
15.25. CON26 コンポジットビデオ入力インターフェース .....	173
15.26. CON30 拡張インターフェース .....	173
15.27. CON31 RTC 外部バックアップインターフェース .....	174
15.28. CON32 電源入力インターフェース .....	174
15.29. CON33 電源入力インターフェース .....	175
15.30. CON34 電源出力インターフェース .....	175
15.31. JP2、JP3 バイアスジャンパ .....	175
15.32. SW1、SW2 機能選択スイッチ .....	175
15.33. SW3～SW6 ユーザースイッチ .....	176
15.34. SW7 電源スイッチ .....	176
15.35. SW8 リセットスイッチ .....	176
15.36. SW9 SD ブートデバイス選択スイッチ .....	176
15.37. LED1、LED2 シリアル LED .....	176
15.38. LED4～LED7 ユーザー LED .....	177
15.39. LED8 電源 LED .....	177
15.40. LED9 警告 LED .....	177
16. 基板形状図 .....	178
17. オプション品 .....	180
17.1. Armadillo-EVA 1500 LCD オプションセット .....	180
17.1.1. 概要 .....	180
17.1.2. 組み立て .....	181
17.2. Armadillo-WLAN モジュール(AWL13) .....	182
17.2.1. 概要 .....	182
17.2.2. 組み立て .....	183
18. Howto .....	185
18.1. Linux カーネルイメージをカスタマイズする .....	185
18.2. Armadillo-EVA 1500 LCD 拡張ボードを使用する .....	186
18.2.1. 接続方法 .....	187
18.2.2. ビルド手順 .....	187
18.2.3. インストール .....	188
18.2.4. 動作確認方法 .....	188
18.3. Armadillo-WLAN(AWL13)を使用する .....	190
18.3.1. 接続方法 .....	190
18.3.2. ビルド手順 .....	191
18.3.3. AWL13 デバイスドライバのインストール .....	192
18.3.4. ファームウェアのロード .....	192
18.3.5. 無線設定 .....	192
18.3.6. 動作確認方法 .....	195
18.4. USB ガジェットを使用する .....	196
18.4.1. 接続方法 .....	196
18.4.2. ビルド手順 .....	197
18.4.3. インストール .....	198
18.4.4. 動作確認方法 .....	198
19. ユーザー登録 .....	200

---

19.1. 購入製品登録 .....	200
19.1.1. 正規認証ファイルを取り出す手順 .....	200



## 目次

3.1. Armadillo-EVA 1500 .....	22
3.2. ブロック図 .....	24
3.3. インターフェースレイアウト図 .....	25
4.1. GNOME 端末の起動 .....	35
4.2. GNOME 端末のウィンドウ .....	35
4.3. minicom 設定方法 .....	36
4.4. minicom 起動方法 .....	36
4.5. minicom 終了確認 .....	36
4.6. Armadillo-EVA 1500 の組み立て .....	37
4.7. Armadillo-EVA 1500 の接続例 .....	38
4.8. 機能選択スイッチの設定 .....	39
5.1. AC アダプタ接続前の状態 .....	40
5.2. 起動ログ .....	40
5.3. 終了方法 .....	47
6.1. ifup コマンドによる有線 LAN の有効化 .....	48
6.2. ifdown コマンドによる有線 LAN の無効化 .....	49
6.3. システムクロックを設定 .....	50
6.4. ハードウェアクロックを設定 .....	50
6.5. GStreamer のインストール .....	53
6.6. テスト画像をファイルに保存 .....	53
6.7. X サーバーのインストール .....	54
6.8. X サーバーの起動 .....	54
6.9. X サーバー起動画面 .....	54
6.10. mount コマンド書式 .....	56
6.11. ストレージのマウント .....	56
6.12. ストレージのアンマウント .....	57
6.13. fdisk コマンドによるパーティション変更 .....	57
6.14. EXT3 ファイルシステムの構築 .....	58
6.15. JPEG 画像の表示 .....	59
6.16. gnome-backgrounds パッケージの画像ファイルを利用 .....	59
6.17. カメラ画像の表示 .....	59
6.18. カメラ画像の表示画面 .....	60
6.19. 解像度の変更 .....	60
6.20. カラーフォーマットの変更 .....	60
6.21. 再生パスのルーティング設定 .....	61
6.22. テストサウンドの再生 .....	61
6.23. 録音パスのルーティング設定 .....	61
6.24. サウンドの録音 .....	62
6.25. 録音したファイルを再生 .....	62
6.26. サンプル動画の取得 .....	63
6.27. 動画再生の実行イメージ .....	66
6.28. 温度の取得 .....	69
6.29. LED を点灯させる .....	70
6.30. LED を消灯させる .....	70
6.31. LED の状態を表示する .....	70
6.32. LED のトリガに timer を指定する .....	71
6.33. LED のトリガを表示する .....	71
6.34. ユーザースイッチ: イベントの確認 .....	71
8.1. hermit コマンドのヘルプを表示 .....	95
8.2. TFTP サーバー上の Linux カーネルイメージを指定する .....	96

9.1. 書き込み制限を外す .....	102
9.2. 書き込みを制限する .....	102
9.3. hermit コマンドのヘルプ .....	103
9.4. tftpd 例 .....	104
9.5. Debian GNU/Linux システムへのカーネルイメージのインストール .....	105
10.1. 自動マウントされた SD カードのアンマウント .....	106
10.2. fstab の編集 .....	111
10.3. ルートファイルシステムの起動設定 .....	114
10.4. Linux カーネルの起動設定 .....	114
11.1. パーティション作成手順 .....	117
11.2. パーティション確認手順 .....	118
11.3. ファイルシステム作成手順 .....	119
11.4. ファイル展開手順 .....	120
11.5. Firmware イメージファイルの配置 .....	120
11.6. Debian GNU/Linux のリカバリ手順 .....	122
11.7. SPI フラッシュメモリのリカバリ手順 .....	124
13.1. RZ/G1M 概要 .....	126
13.2. eMMC 周辺の構成 .....	129
13.3. SPI フラッシュメモリ周辺の構成 .....	130
13.4. LAN 周辺の構成 .....	130
13.5. WLAN インターフェース(CON5)周辺の構成 .....	131
13.6. シリアル(CON16)周辺の構成 .....	132
13.7. USB シリアル(CON14)周辺の構成 .....	132
13.8. SD スロット(CON4)周辺の構成 .....	133
13.9. SD スロット(CON6)周辺の構成 .....	134
13.10. USB 2.0 周辺の構成 .....	135
13.11. USB 3.0 周辺の構成 .....	136
13.12. PCI Express 周辺の構成 .....	137
13.13. シリアル ATA 周辺の構成 .....	138
13.14. コンポジットビデオ入力周辺の構成 .....	138
13.15. LCD 周辺の構成 .....	139
13.16. LVDS 周辺の構成 .....	140
13.17. Digital HD 周辺の構成 .....	141
13.18. AUDIO CODEC 周辺の構成 .....	142
13.19. JP2、JP3 周辺回路 .....	142
13.20. CON2 周辺の構成 .....	143
13.21. CON3 周辺の構成 .....	144
13.22. CON15 周辺の構成 .....	144
13.23. CON22 周辺の構成 .....	145
13.24. CON24 周辺の構成 .....	145
13.25. CON30 周辺の構成 .....	146
13.26. I2C 周辺の構成 .....	147
13.27. 機能選択スイッチの形状 .....	148
13.28. ユーザースイッチ周辺の構成 .....	151
13.29. SD ブートデバイス選択スイッチ周辺の構成 .....	151
13.30. 電源スイッチ周辺の構成 .....	152
13.31. リセットスイッチ周辺の構成 .....	152
13.32. ユーザー LED 周辺の構成 .....	153
13.33. 電源 LED 周辺の構成 .....	153
13.34. 警告 LED 周辺の構成 .....	154
13.35. JTAG 周辺の構成 .....	154
13.36. リアルタイムクロック周辺の構成 .....	155
13.37. 電源回路の構成 .....	156

---

13.38. 電源シーケンス .....	157
13.39. リセット回路の構成 .....	157
15.1. インターフェースレイアウト(A面) .....	159
15.2. インターフェースレイアウト(B面) .....	161
15.3. ミニプラグ(ヘッドホン)のピンアサイン .....	170
15.4. ミニプラグ(マイク)のピンアサイン .....	170
15.5. ACアダプターの極性マーク .....	174
16.1. 基板形状および固定穴寸法 .....	178
16.2. コネクタ中心寸法 .....	179
17.1. Armadillo-EVA 1500 LCD オプションセット .....	180
17.2. Armadillo-EVA 1500 と LCD の組み立て(信号確認) .....	181
17.3. Armadillo-EVA 1500 と LCD の組み立て(コンパクト) .....	182
17.4. Armadillo-WLAN モジュール(AWL13) .....	182
17.5. AWL13 の組み立て .....	184
18.1. タッチスクリーンのイベント確認 .....	189
18.2. インフラストラクチャモード: WPA2-PSK(AES)設定手順 .....	194
18.3. インフラストラクチャモード: WPA2-PSK(AES)設定確認手順 .....	194
18.4. インフラストラクチャモード: WEP 設定手順 .....	195
18.5. インフラストラクチャモード: WEP 設定確認手順 .....	195

## 表目次

1.1. 使用しているフォント .....	16
1.2. 表示プロンプトと実行環境の関係 .....	16
1.3. コマンド入力例での省略表記 .....	16
3.1. 仕様 .....	22
3.2. インターフェース内容 .....	25
3.3. Armadillo-EVA 1500 で利用可能なソフトウェア .....	26
3.4. SPI フラッシュメモリ メモリマップ .....	27
3.5. Linux 起動時の DDR3-SDRAM メモリマップ .....	27
3.6. eMMC パーティション構成 .....	28
4.1. ATDE5 の種類 .....	31
4.2. ユーザー名とパスワード .....	34
4.3. 動作確認に使用する取り外し可能デバイス .....	34
4.4. シリアル通信設定 .....	35
4.5. 機能選択スイッチの設定 .....	39
5.1. 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 .....	46
6.1. 時刻フォーマットのフィールド .....	50
6.2. エレメントの種類 .....	53
6.3. ストレージデバイス .....	55
6.4. サンプル動画 .....	63
6.5. 工場出荷状態のシリアル設定 .....	68
6.6. LED - Armadillo-EVA 1500 .....	69
6.7. trigger の種類 .....	70
6.8. インプットデバイスファイルとイベントコード .....	71
7.1. Linux カーネル主要設定 .....	73
7.2. Linux カーネルのデフォルト起動オプション .....	73
7.3. 代表的なガジェットドライバ .....	80
7.4. キーコード .....	88
7.5. I2C デバイス .....	89
8.1. スイッチと 2nd ブートローダーの対応 .....	94
8.2. Hermit-At 起動モード .....	94
8.3. Hermit-At の保守モードコマンド一覧 .....	95
8.4. コンソール指定子とログ出力先 .....	96
8.5. Linux カーネルイメージ指定子 .....	96
8.6. Linux カーネルの起動オプションの一例 .....	96
9.1. イメージの配置可能場所一覧 .....	100
9.2. フラッシュメモリの書き換え方法 .....	101
9.3. パーティションのデフォルト状態での書き込み制限の有無と対応するイメージファイル名 .....	101
9.4. パーティションと MTD クラスディレクトリの対応 .....	102
9.5. パーティションとオプションの対応 .....	105
10.1. ブートディスクの作成に使用するファイル .....	106
10.2. ブートディスクの制約 .....	107
10.3. ブートディスクの構成例 .....	107
10.4. ルートファイルシステムの構築に使用するファイル .....	110
10.5. ディスクデバイスの割り当て .....	112
10.6. ブートディスクの作成に使用するファイル .....	112
10.7. ブートローダーが Linux カーネルを検出可能な条件 .....	112
11.1. eMMC のパーティション構成 .....	115
11.2. eMMC のブートパーティション .....	115
11.3. eMMC 各領域の用途 .....	115
11.4. SPI フラッシュメモリのパーティション構成 .....	116

11.5. SPI フラッシュメモリ 各領域の用途 .....	116
11.6. リカバリディスクの作成に必要なファイル .....	117
11.7. リカバリ進捗と LED の対応 .....	121
11.8. Debian GNU/Linux のリカバリに必要なファイル .....	122
11.9. SPI フラッシュメモリのリカバリに必要なファイル .....	123
11.10. パーティションのデフォルト状態での書き込み制限の有無と対応する MTD ノード .....	124
13.1. MD0 - 自走モード/ステップアップモードの設定 .....	127
13.2. MD1、MD2、MD3 - ブートデバイスの設定 .....	127
13.3. MD4 - CS0 空間サイズ設定 .....	127
13.4. MD6、MD7 - Master boot processor の設定 .....	127
13.5. MD8 - CS0 空間データバス幅の設定 .....	127
13.6. MD9 - 水晶発振器/水晶発振子の設定 .....	127
13.7. MD10、MD11、MD12、MD20、MD21、MDT0、MDT1 - JTAG の機能設定 .....	128
13.8. MD13、MD14 - 周波数モードの設定 .....	128
13.9. MD19 - DDR3-SDRAM バスクロックの設定 .....	128
13.10. MD22、MD27、MD28 - DDR3-SDRAM のインターフェース設定 .....	128
13.11. MD23 - SATA0/USB3.0 の設定 .....	128
13.12. MD24 - SATA1/PCIe の設定 .....	128
13.13. MD ピンの出荷時の設定 .....	129
13.14. LAN コネクタ LED .....	130
13.15. シリアル LED .....	132
13.16. I2C バスの接続 .....	146
13.17. 水晶発振器、水晶振動子 .....	147
13.18. 機能選択スイッチの機能 .....	148
13.19. SW1.1、SW1.2、SW1.3 - USB 3.0、SATA/miniPCIe の設定 .....	148
13.20. SW1.4 - 自走モード/ステップアップモード設定 .....	148
13.21. SW1.5、SW1.6 - JTAG 設定 .....	149
13.22. SW1.7、SW1.8 - IPL 起動デバイス設定 .....	149
13.23. SW2.1 - ビデオ入力設定 .....	149
13.24. SW2.3 - シリアル(UART)設定 .....	149
13.25. SW2.4 - USB 2.0 設定 .....	150
13.26. SW2.5 - SD/WLAN 設定 .....	150
13.27. SW2.6 - SPI 設定 .....	150
13.28. SW2.7 - Hermit-At 起動デバイス設定 .....	150
13.29. SW2.8 - Hermit-At 起動デバイス設定 .....	150
13.30. リアルタイムクロック仕様 .....	154
14.1. 絶対最大定格 .....	158
14.2. 推奨動作条件 .....	158
15.1. 搭載コネクタ、スイッチ型番一覧(A 面) .....	159
15.2. 搭載コネクタ、スイッチ型番一覧(B 面) .....	161
15.3. CON1 信号配列 .....	161
15.4. CON2 信号配列 .....	162
15.5. CON3 信号配列 .....	163
15.6. CON4 信号配列 .....	164
15.7. CON4 カード検出、ライトプロテクト .....	164
15.8. CON5 信号配列 .....	164
15.9. CON6 信号配列 .....	165
15.10. CON6 カード検出、ライトプロテクト .....	165
15.11. CON7 信号配列 .....	165
15.12. CON8 信号配列 .....	166
15.13. CON9 信号配列 .....	166
15.14. CON10 信号配列 .....	166
15.15. CON11 信号配列 .....	166

15.16. CON12 信号配列 .....	167
15.17. CON13 信号配列 .....	168
15.18. CON14 信号配列 .....	168
15.19. CON15 信号配列 .....	168
15.20. CON16 信号配列 .....	169
15.21. CON17 信号配列 .....	169
15.22. CON18 信号配列 .....	169
15.23. CON21 信号配列 .....	170
15.24. CON22 信号配列 .....	170
15.25. CON23 信号配列 .....	170
15.26. CON24 信号配列 .....	172
15.27. CON25 信号配列 .....	172
15.28. CON26 信号配列 .....	173
15.29. CON30 信号配列 .....	173
15.30. CON31 信号配列 .....	174
15.31. CON32 信号配列 .....	174
15.32. CON33 信号配列 .....	175
15.33. CON34 信号配列 .....	175
15.34. バイアスジャンパの接続 .....	175
15.35. 機能選択スイッチ(SW1)の接続 .....	175
15.36. 機能選択スイッチ(SW2)の接続 .....	175
15.37. ユーザースイッチの接続 .....	176
15.38. 電源スイッチの接続 .....	176
15.39. リセットスイッチの接続 .....	176
15.40. SD ブートデバイス選択スイッチの接続 .....	176
15.41. シリアル LED の接続 .....	176
15.42. ユーザー LED の接続 .....	177
15.43. 電源 LED の接続 .....	177
15.44. 警告 LED の接続 .....	177
17.1. Armadillo-EVA 1500 関連のオプション品 .....	180
17.2. タッチパネル LCD の仕様 .....	180
17.3. AWL13 の仕様 .....	183
18.1. 輝度設定に使用する sysfs ファイル .....	189
18.2. インフラストラクチャモード: WPA-PSK/WPA2-PSK パラメーター例 .....	193
18.3. インフラストラクチャモード: WEP パラメーター例 .....	195

# 1. はじめに

---

このたびは Armadillo-EVA 1500 をお求めいただき、ありがとうございます。

Armadillo-EVA 1500 製品マニュアル(以下、本書と表記します)には、ご利用にあたっての注意事項や、ご購入時のソフトウェアの状態、ハードウェア・ソフトウェアをカスタマイズする場合に必要な情報などが記載されています。

## 1.1. ユーザー限定コンテンツ

Armadillo-EVA 1500 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。限定コンテンツを取得するには、「19. ユーザー登録」を参照してください。

## 1.2. 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-EVA 1500 ドキュメント・ダウンロード

<http://armadillo.atmark-techno.com/armadillo-eva-1500/downloads>

## 1.3. 本書の構成

本書は次のように構成されています。

◆ **はじめにお読みください。**

「1. はじめに」、「2. 注意事項」

◆ **Armadillo-EVA 1500 の仕様を紹介します。**

「3. 製品概要」

◆ **工場出荷状態のソフトウェアの使い方や、動作を確認する方法を紹介します。**

「4. 電源を入れる前に」、「5. 起動と終了」、「6. 各種機能を使用する」

◆ **工場出荷状態のソフトウェア仕様について紹介します。**

「7. Linux カーネル仕様」、「8. ブートローダー仕様」

◆ **システム開発に必要な情報を紹介します。**

「9. ビルド手順」、「10. SD ブートの活用」、「11. リカバリ手順」、「12. JTAG ICE を利用する」

◆ ハードウェアの仕様や、カスタマイズをする場合に必要な情報を紹介します。

「13. ハードウェア仕様」、「14. 電氣的仕様」、「15. インターフェース仕様」、「16. 基板形状図」、「17. オプション品」

◆ ソフトウェアのカスタマイズ方法や、オプション品の利用方法について紹介します。

「18. Howto」

◆ ご購入ユーザーに限定して公開している情報の紹介やユーザー登録について紹介します。

「19. ユーザー登録」

## 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. 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

### 1.4.3. 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。

## 1.5. 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<sup>[1]</sup>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<sup>[2]</sup>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、デジタルHD出力、SD、USB、マイク、ヘッドホン、コンポジットビデオ入力)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。

## 2.6. 書込み禁止領域について



SPI フラッシュメモリの ipl パーティション (アドレス 0x00000000-0x0000ffff) にはブートローダーが格納されています。正

<sup>[1]</sup>コネクタ非搭載箇所へのコネクタ等の増設は除く。

<sup>[2]</sup>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。

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。



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

- ・ 当社製品は、原則として日本国内での使用を想定して開発・製造されています。
- ・ 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。
- ・ 当社製品を輸出するときは、輸出者の責任において、日本国および関係する諸外国の輸出関連法令に従い、必要な手続きを行っていただきますようお願いいたします。
- ・ 日本国およびその他関係諸国による制裁または通商停止を受けている国家、組織、法人または個人に対し、当社製品を輸出、販売等することはできません。
- ・ 当社製品および関連技術は、大量破壊兵器の開発等の軍事目的、その他国内外の法令により製造・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。



# 3. 製品概要

## 3.1. ボード概要

「Armadillo-EVA 1500」は ARM Cortex-A15 を 2 コア搭載したルネサスエレクトロニクス製アプリケーションプロセッサ「RZ/G1M」の評価ボードです。USB 3.0、PCI Express、シリアル ATA 等、多彩な周辺インターフェースに対応しています。

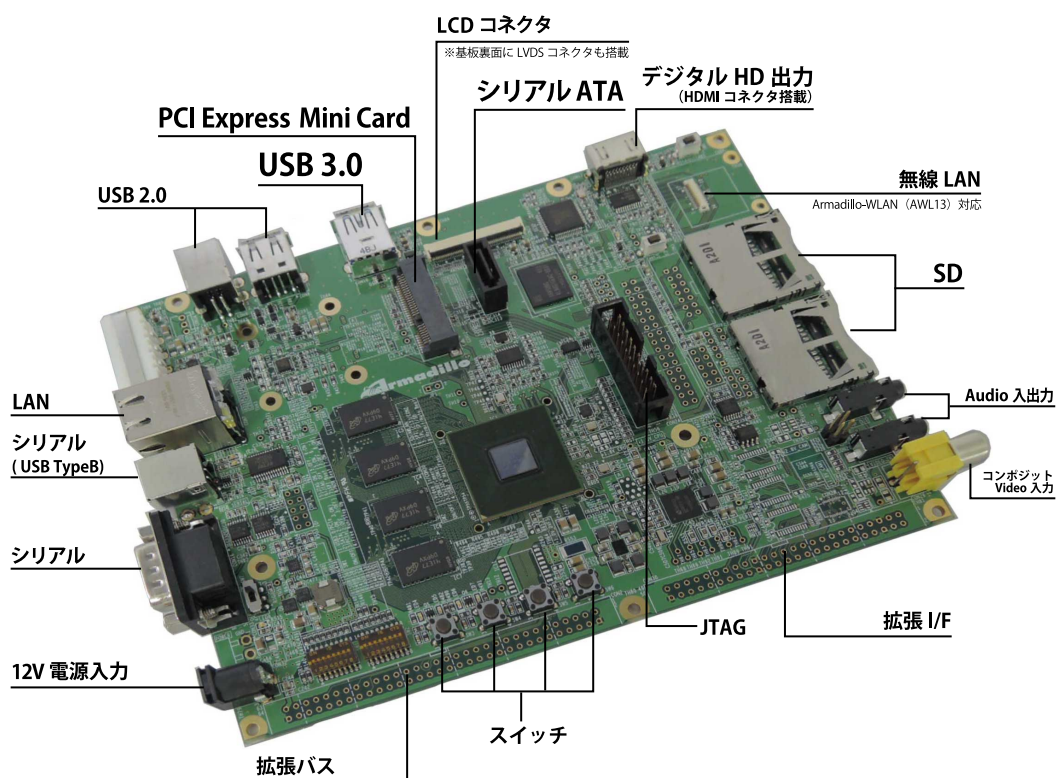


図 3.1 Armadillo-EVA 1500

## 3.2. 仕様

Armadillo-EVA 1500 の主な仕様は次のとおりです。

表 3.1 仕様

プロセッサ	ルネサスエレクトロニクス製 RZ/G1M (R8A77430HA01BG) ARM Cortex-A15 デュアルコア 浮動小数点コプロセッサ(VFPv3)搭載 メディアプロセッシングエンジン(NEON)搭載 PowerVR SGX544 MP2
システムクロック	CPU コアクロック: 1.5GHz 源発振クロック: 20MHz
RAM	デュアルチャンネル DDR3L-1600 SDRAM: 2GByte バス幅 32bit x 2

フラッシュメモリ	eMMC: 8GByte SPI フラッシュメモリ: 8MByte <sup>[a]</sup>
LAN(Ethernet)	RJ-45 x 1 10BASE-T/100BASE-TX、AUTO-MDIX 対応
無線 LAN	Armadillo-WLAN(AWL13)搭載可能 <sup>[b][c]</sup>
シリアル(UART)	D-Sub9 ピン(RS232C レベル) x 1、USB Type B(USB シリアル) x 1
汎用入出力(GPIO)	最大 142bit <sup>[d]</sup>
SD/MMC	SD スロット x 2 UHS-I(SDR50、最大クロック周波数 97.5MHz) <sup>[b]</sup> 、UHS-I(SDR104、最大クロック周波数 156MHz)
USB	USB Type A x 2、USB Type B x 1 USB 3.0 Host(Super Speed) <sup>[e]</sup> 、USB 2.0 Host(High Speed) <sup>[f]</sup> 、USB 2.0 Function(High Speed) <sup>[f]</sup>
PCIe	PCI Express Mini Card スロット x 1 <sup>[e][g]</sup> PCI Express Base Specification Rev2.0
シリアル ATA	SATA コネクタ x 1 <sup>[g]</sup> Serial ATA Standard Rev.3.1
ビデオ	CMOS 24bit LCD 出力 LVDS LCD 出力 デジタル HD 出力(HDMI Type A) コンポジットビデオ入力 RCA ピンジャック
オーディオ	ヘッドホン用ミニジャック マイク/ライン入力用ミニジャック デジタル HD 出力(HDMI Type A)
拡張インターフェース	GPIO、UART、I2C、I2S、SPI、Camera、CAN、PWM
拡張バス	16bit データバス、20bit アドレスバス(最大 27bit) <sup>[a]</sup>
カレンダー時計	リアルタイムクロック 外部バックアップ用電源入力コネクタ搭載
JTAG	20 ピン、2.54mm ピッチ(ARM 標準コネクタ)
スイッチ	機能選択スイッチ x 2、ユーザースイッチ x 4、リセットスイッチ x 1、SD ブートデバイス選択スイッチ x 1、電源スイッチ x 1
LED	ユーザー LED x 4、電源 LED x 1、警告 LED x 1
電源電圧	DC 12V±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 x 135mm(突起部を除く)

<sup>[a]</sup>SPI フラッシュメモリと拡張バスのアドレスピンは排他利用となります。

<sup>[b]</sup>SD スロット (SDR50 側)と無線 LAN コネクタは排他利用となります。

<sup>[c]</sup>Armadillo-EVA 1500 評価セットに Armadillo-WLAN(AWL13)は同梱されていません。

<sup>[d]</sup>他の機能を無効化して優先的に設定した場合の数となります。

<sup>[e]</sup>PCIe と USB 3.0 は排他利用となります。

<sup>[f]</sup>USB 2.0 Host と USB 2.0 Function は排他利用となります。

<sup>[g]</sup>PCIe とシリアル ATA は排他利用となります。

### 3.3. ブロック図

Armadillo-EVA 1500 のブロック図は次のとおりです。

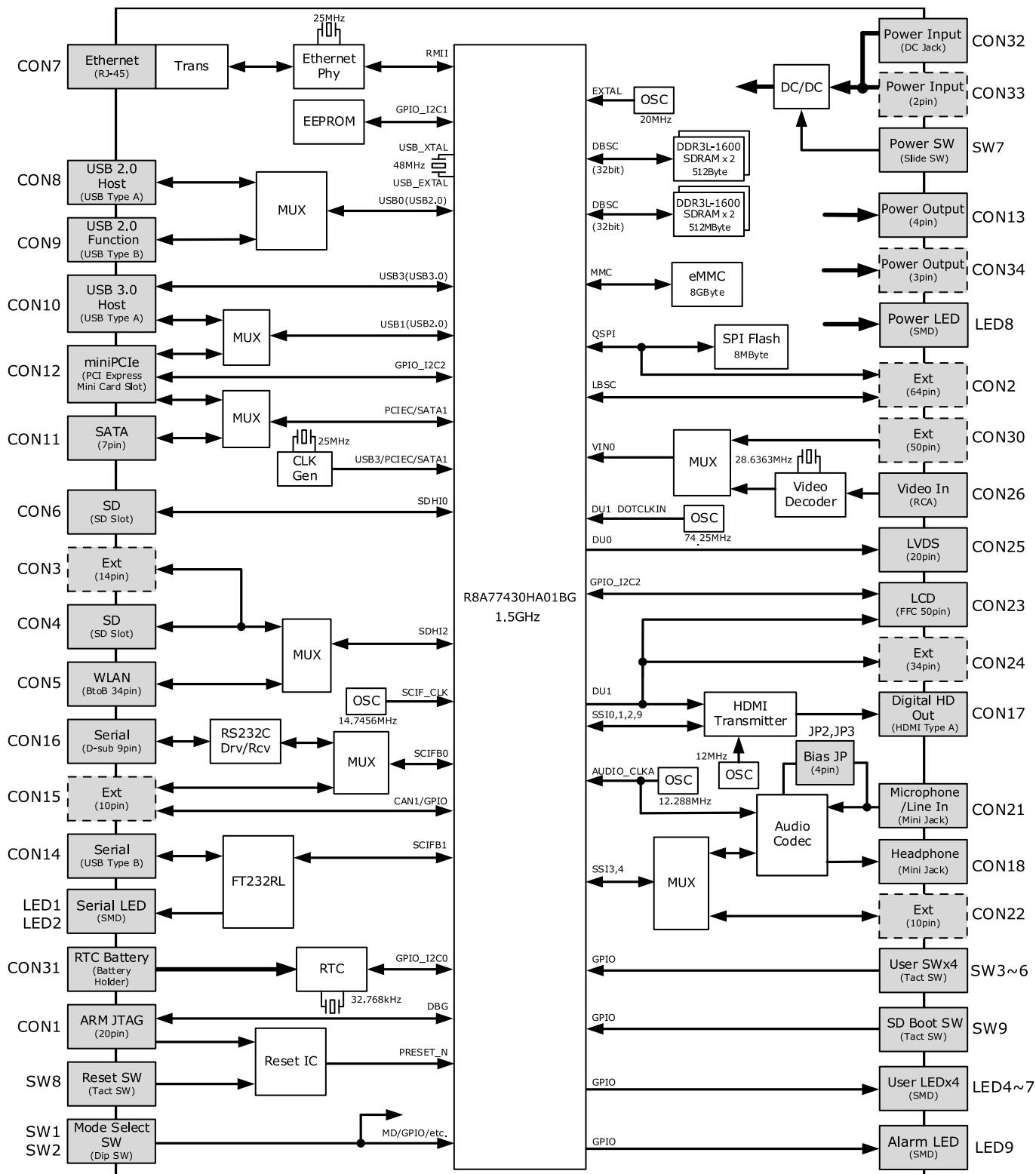


図 3.2 ブロック図



### 3.4. インターフェースレイアウト

Armadillo-EVA 1500 のインターフェースレイアウトは次のとおりです。各インターフェースの配置場所等を確認してください。

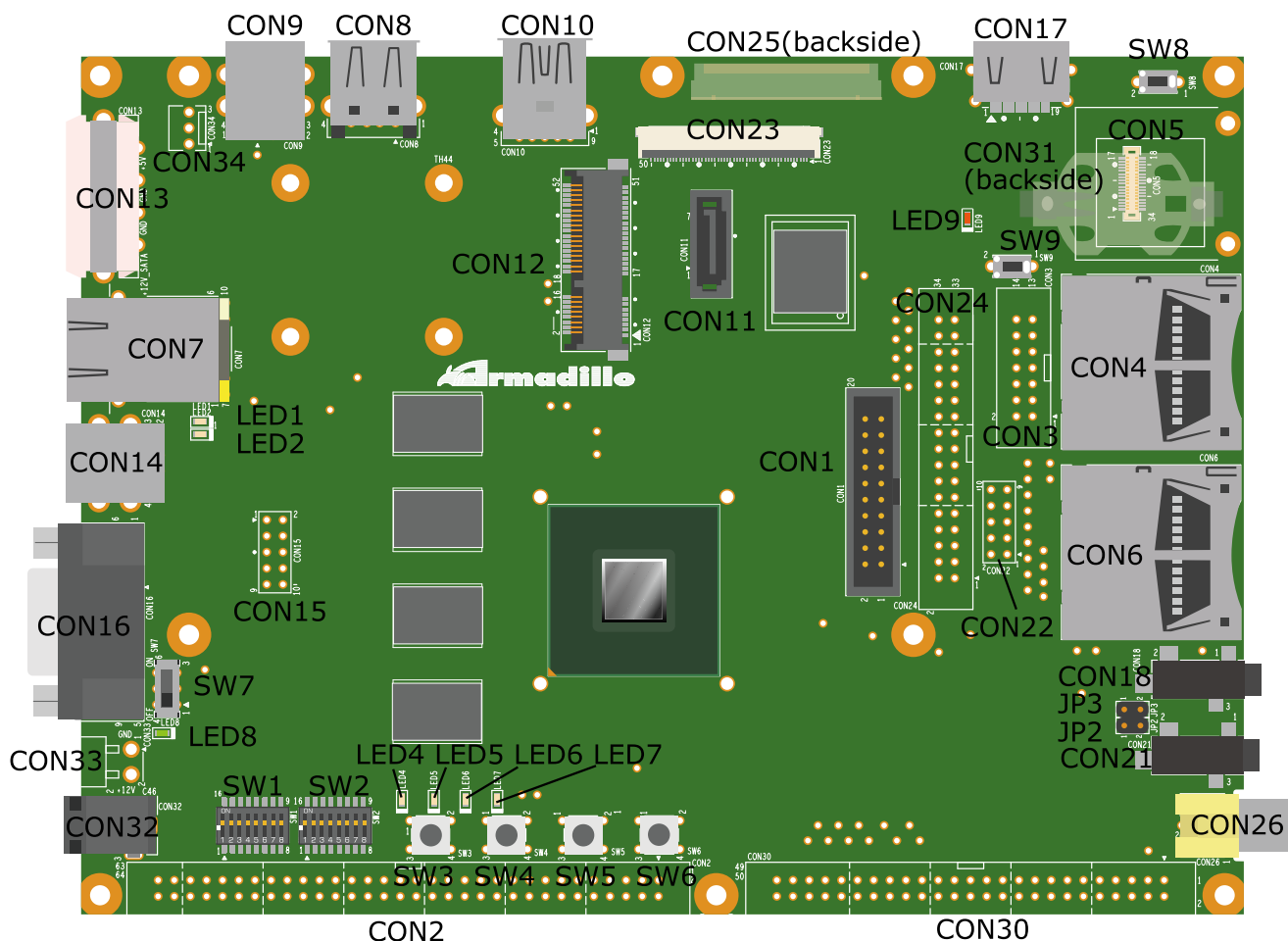


図 3.3 インターフェースレイアウト図

表 3.2 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	JTAG インターフェース	ピンヘッダ 20 ピン(2.54mm ピッチ)	
CON2	拡張インターフェース	ピンヘッダ 64 ピン(2.54mm ピッチ)	コネクタ非搭載
CON3	拡張インターフェース	ピンヘッダ 14 ピン(2.54mm ピッチ)	コネクタ非搭載
CON4	SD インターフェース	SD スロット	
CON5	WLAN インターフェース	BtoB コネクタ 34 ピン(0.5mm ピッチ)	挿抜寿命： 50 回
CON6	SD インターフェース	SD スロット	
CON7	LAN インターフェース	RJ-45 コネクタ	
CON8	USB 2.0 ホストインターフェース	USB 2.0 Type A コネクタ	
CON9	USB 2.0 ファンクションインターフェース	USB 2.0 Type B コネクタ	
CON10	USB 3.0 ホストインターフェース	USB 3.0 Type A コネクタ	
CON11	SATA インターフェース	SATA コネクタ 7 ピン	
CON12	miniPCIe インターフェース	PCI Express Mini Card スロット	

部品番号	インターフェース名	形状	備考
CON13	電源出カインターフェース	ペリフェラル電源コネクタ 4 ピン(5.08mm ピッチ)	
CON14	USB シリアルインターフェース	USB 2.0 Type B コネクタ	
CON15	拡張インターフェース	ピンヘッダ 10 ピン(2.54mm ピッチ)	コネクタ非搭載
CON16	シリアルインターフェース	D-Sub9 ピン(オス)	
CON17	デジタル HD 出力インターフェース	HDMI Type A コネクタ	
CON18	ヘッドホンインターフェース	ミニジャック(φ3.5mm)	
CON21	マイク/ライン入力インターフェース	ミニジャック(φ3.5mm)	
CON22	拡張インターフェース	ピンヘッダ 10 ピン(2.54mm ピッチ)	コネクタ非搭載
CON23	LCD インターフェース	FFC コネクタ 50 ピン(0.5mm ピッチ)	挿抜寿命: 20 回
CON24	拡張インターフェース	ピンヘッダ 34 ピン(2.54mm ピッチ)	コネクタ非搭載
CON25	LVDS インターフェース	ライトアングルコネクタ 20 ピン(1.25mm ピッチ)	
CON26	コンポジットビデオ入力インターフェース	RCA ジャック(黄色)	
CON30	拡張インターフェース	ピンヘッダ 50 ピン(2.54mm ピッチ)	コネクタ非搭載
CON31	RTC 外部バックアップインターフェース	電池ボックス	対応電池: CR2032
CON32	電源入力インターフェース	DC ジャック	対応プラグ: 内径 2.1mm 外径 5.5mm
CON33	電源入力インターフェース	ピンヘッダ 2 ピン(3.96mm ピッチ)	コネクタ非搭載
CON34	電源出カインターフェース	ピンヘッダ 3 ピン(2.54mm ピッチ)	コネクタ非搭載
JP2	バイアスジャンパ	ピンヘッダ 2 ピン(2.54mm ピッチ)	
JP3			
SW1	機能選択スイッチ	ディップスイッチ(8 接点)	
SW2			
SW3			
SW4			
SW5			
SW6			
SW7	電源スイッチ	スライドスイッチ	
SW8	リセットスイッチ	タクトスイッチ	
SW9	SD ブートデバイス選択スイッチ	タクトスイッチ	
LED1	シリアル LED	LED(黄色、面実装)	
LED2			
LED4	ユーザー LED	LED(黄色、面実装)	
LED5			
LED6			
LED7			
LED8	電源 LED	LED(緑色、面実装)	
LED9	警告 LED	LED(赤色、面実装)	

### 3.5. ソフトウェア構成

Armadillo-EVA 1500 で動作するソフトウェアの構成は次のとおりです。

表 3.3 Armadillo-EVA 1500 で利用可能なソフトウェア

ソフトウェア	説明
IPL	1st ブートローダーです。2nd ブートローダーである Hermit-At を起動させる機能を持っています。工場出荷状態では 1st ブートローダーイメージは SPI フラッシュメモリに配置されています。
Hermit-At	2nd ブートローダーです。Linux カーネルを起動させる機能の他に、ダウンローダーと協調動作を行い SPI フラッシュメモリを書き替える機能など様々な機能を持っています。工場出荷状態では 2nd ブートローダーイメージは SPI フラッシュメモリに配置されていますが、IPL の機能により SD カードに配置することもできます。

ソフトウェア	説明
Linux カーネル	バージョン 3.x 系の Linux カーネルです。工場出荷状態では Linux カーネルイメージは eMMC に配置されていますが、Hermit-At の機能により SD カードに配置することもできます。
Debian GNU/Linux	Debian Project によって作成された Linux ディストリビューションです。パッケージ管理システムを備えているため、Debian Project が提供する豊富なソフトウェアパッケージを簡単に追加することができます。工場出荷状態では Debian GNU/Linux のルートファイルシステムは eMMC に配置されていますが、Linux カーネルがサポートしている SD カードなどのストレージデバイスに配置することもできます。

### 3.6. メモリマップ

Armadillo-EVA 1500 の SPI フラッシュメモリのメモリマップは次のとおりです。

表 3.4 SPI フラッシュメモリ メモリマップ

アドレス	パーティション名	サイズ	工場出荷状態で書き込まれているソフトウェア
0x00000000   0x0000FFFF	ipl	64kByte	IPL ブートローダーイメージ
0x00010000   0x0008FFFF	bootloader	512kByte	Hermit-At ブートローダーイメージ
0x00090000   0x000CFFFF	fdt	256kByte	Device Tree Blob
0x000D0000   0x007FFFFF	firmware	7.1875MByte	armhf アーキテクチャ用 OpenGL ES2 ライブラリ <sup>[a]</sup> armhf アーキテクチャ用 OMX メディアコンポーネント

<sup>[a]</sup>2014 年 10 月現在では使用することができません。

Armadillo-EVA 1500 の Linux 起動時における DDR3-SDRAM のメモリマップは次のとおりです<sup>[1]</sup>。

表 3.5 Linux 起動時の DDR3-SDRAM メモリマップ

物理アドレス	サイズ	説明
0x40000000   0x6FFFFFFF	768MByte	Linux カーネルが使用
0x70000000   0x77FFFFFF	128MByte	予約領域(ルネサスエレクトロニクス製 BSP で使用)
0x78000000   0x7FFFFFFF	128MByte	マルチメディアパッケージが使用



linux-3.4-ae1500-at2 以降で、「表 3.5. Linux 起動時の DDR3-SDRAM メモリマップ」に示す通りのメモリマップになっています。linux-3.4-ae1500-at1 では、Linux カーネルが 1024MByte 全てを使用していました。

### 3.7. パーティション構成

Armadillo-EVA 1500 の eMMC のパーティション構成は次のとおりです。

<sup>[1]</sup>Linux カーネルからは 1024MByte を越えるメモリにアクセスすることができません。

表 3.6 eMMC パーティション構成

パーティション	サイズ	ファイルシステム	工場出荷状態で書き込まれているソフトウェア
パーティション 1 (/dev/mmcblk0p1)	約 5GByte	ext3	Debian GNU/Linux カーネルイメージ(/boot/Image.bin)
パーティション 2 (/dev/mmcblk0p2)	2GByte	ext3	無し
ブートパーティション 1 (/dev/mmcblk0boot0)	4MByte	無し	無し
ブートパーティション 2 (/dev/mmcblk0boot1)	4MByte	無し	無し

## 4. 電源を入れる前に

### 4.1. 準備するもの

Armadillo-EVA 1500 に電源を入れる前に、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。「4.2. 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。
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 <sup>[1]</sup> 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。
ネットワーク環境	Armadillo-EVA 1500 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	デジタル HD 出力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 メモリ	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。
スピーカまたはヘッドホン	サウンドの再生を確認する場合に利用します。スピーカーを利用する場合は、別途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	サウンドの録音を確認する場合に利用します。
NTSC 対応カメラ	コンポジットビデオ入力の動作を確認する場合に利用します。
シリアル ATA 対応ハードディスク	シリアル ATA の動作を確認する場合に利用します。別途、SATA ケーブル、電源ケーブルも必要となります。
シリアルクロスケーブル	RS232C レベルのシリアル通信を確認する場合に利用します。



Armadillo-EVA 1500 に搭載されている SATA 電源コネクタは、IDE 用電源コネクタ 4 ピン(オス)です。SATA 対応ハードディスク等と接続する場合、電源変換ケーブルが必要となります。接続ケーブルの一例は次のとおりです。

- ・ メーカー: サンワサプライ
- ・ 型番: TK-PWSATA2

### 4.2. 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

<sup>[1]</sup>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。

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<sup>[2]</sup>
- ・ 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-EVA 1500 に対応している ATDE は、ATDE5 (ATDE バージョン 5)です。

ATDE5 は Debian GNU/Linux 7(コードネーム wheezy)をベースに、Armadillo-EVA 1500 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-EVA 1500 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。

## 4.2.1. ATDE5 セットアップ

### 4.2.1.1. 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。

<sup>[2]</sup>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。

### 4.2.1.2. ATDE5 アーカイブの取得

「表 4.1. 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、評価セット付属の DVD から取得可能です。

表 4.1 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。

### 4.2.1.3. 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 4.1. Windows で ATDE5 のアーカイブ展開する」に、Linux での展開方法を「手順 4.2. Linux で tar.xz 形式のファイルを展開する」に示します。

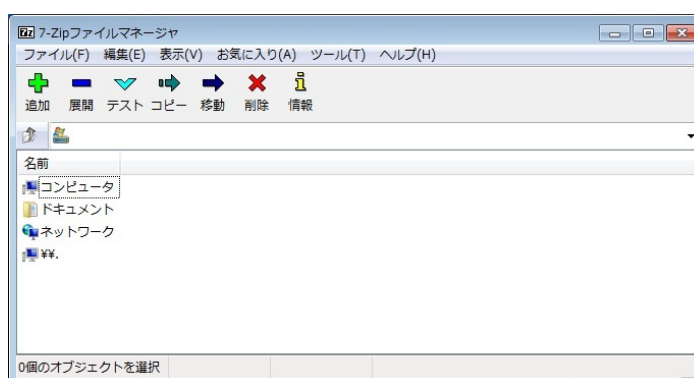
#### 手順 4.1 Windows で ATDE5 のアーカイブ展開する

##### 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、評価セット付属の DVD から取得可能です。

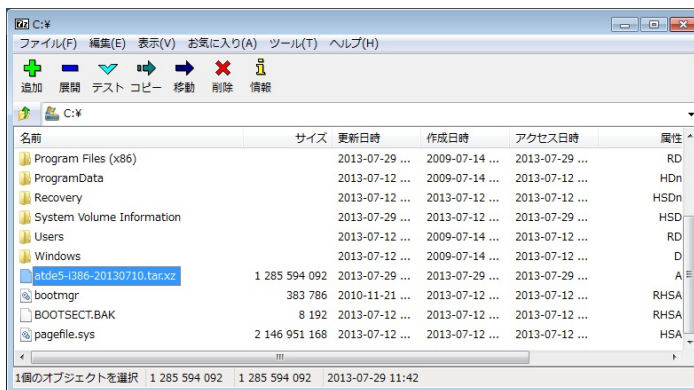
##### 2. 7-Zip の起動

7-Zip を起動します。



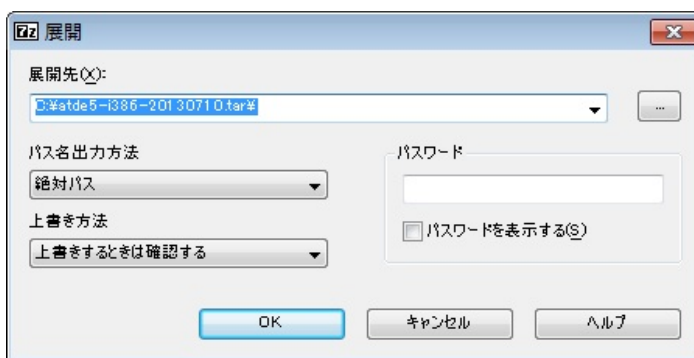
##### 3. xz 圧縮ファイルの選択

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。



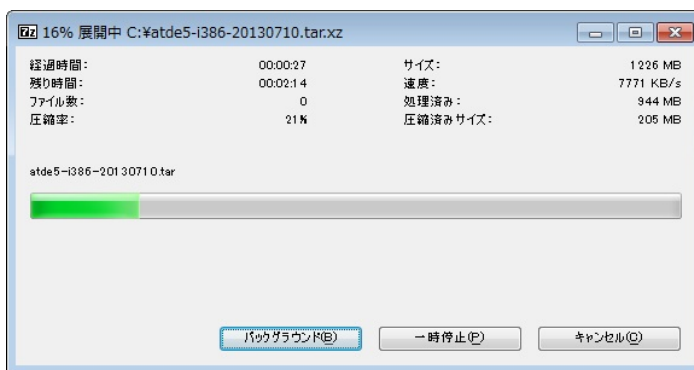
#### 4. xz 圧縮ファイルの展開先の指定

「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。

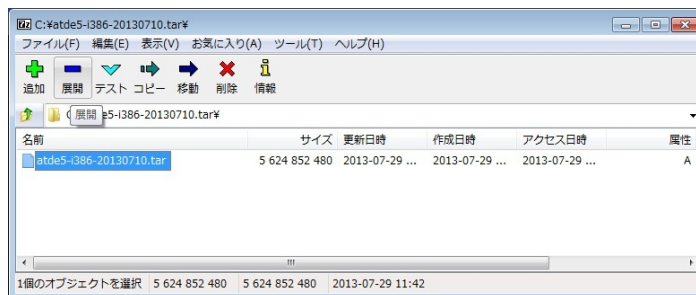


#### 6. tar アーカイブファイルの選択

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。

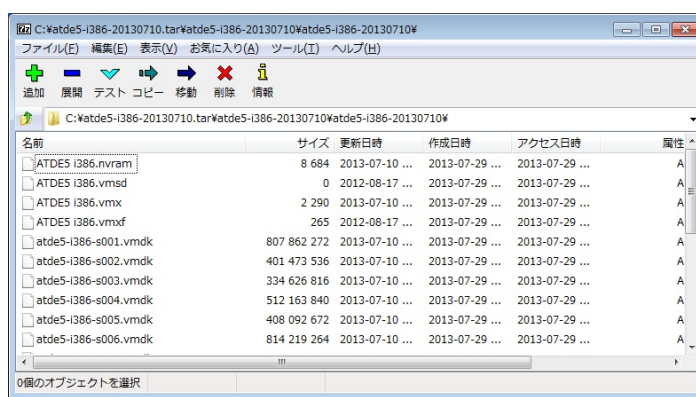
tar アーカイブファイルを出力したのと同様の手順で、tar アーカイブファイルから ATDE5 のデータイメージを出力します。tar 形式のファイルを選択して「展開」をクリックし、「展開先」を指定して、「OK」をクリックします。





## 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 4.2 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
```

#### 4.2.1.4. 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 4.2. ユーザー名とパスワード」に示します<sup>[3]</sup>。

表 4.2 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。

#### 4.2.2. 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-EVA 1500 の動作確認を行うためには、「表 4.3. 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 4.3 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-EVA 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

#### 4.2.3. 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 4.1. 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。

<sup>[3]</sup>特権ユーザーで GUI ログインを行うことはできません。



図 4.1 GNOME 端末の起動

「図 4.2. GNOME 端末のウィンドウ」のようにウィンドウが開きます。



図 4.2 GNOME 端末のウィンドウ

#### 4.2.4. 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 4.4. 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 4.4 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 4.3. 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(df)に保存して終了します。

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

### 図 4.3 minicom 設定方法

minicom を起動させるには、「図 4.4. 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 4.4 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。「図 4.5. minicom 終了確認」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      |
+-----+
```

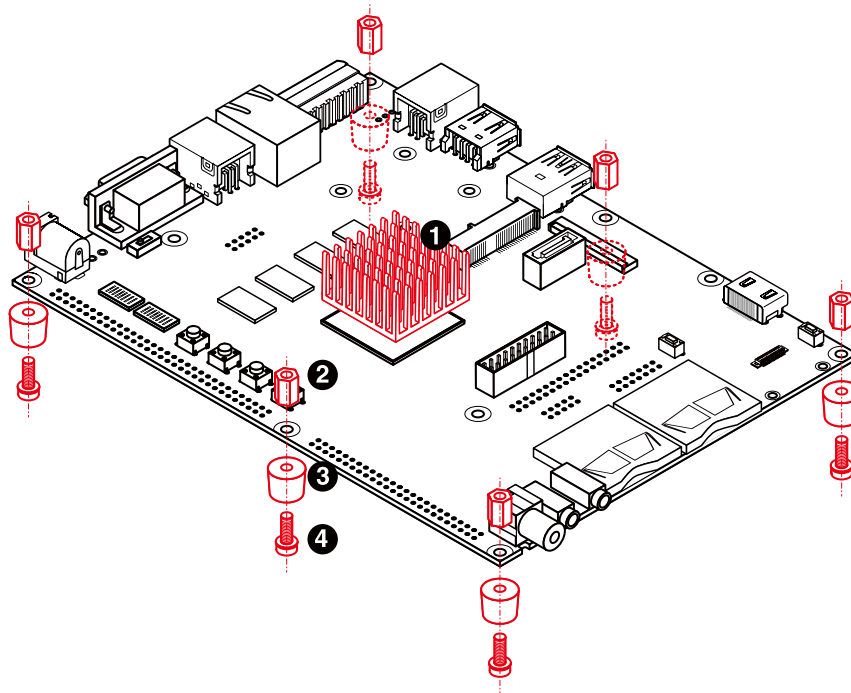
### 図 4.5 minicom 終了確認



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。

## 4.3. 組み立て

「図 4.6. Armadillo-EVA 1500 の組み立て」を参考に組み立ててください。ヒートシンク、スペーサ、ねじ、ゴム足は、Armadillo-EVA 1500 評価セットに同梱されています。

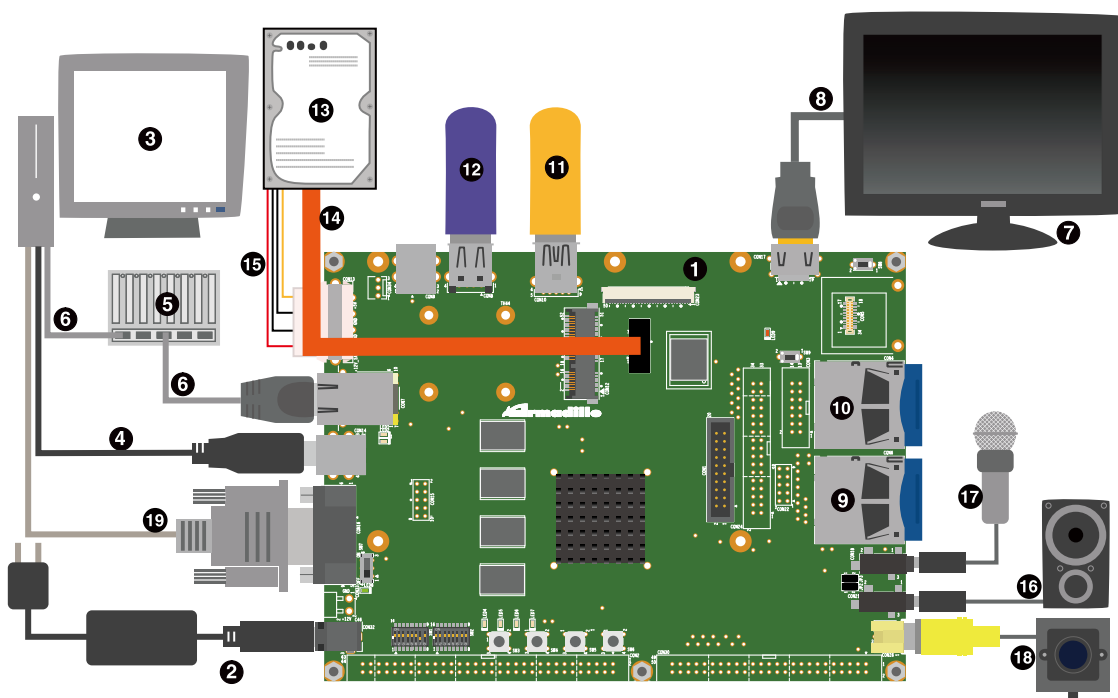


- ❶ ヒートシンク
- ❷ 金属スペーサ(M3、L=8mm)×6
- ❸ ゴム足×6
- ❹ なべ小ねじ(M3、L=10mm)×6

図 4.6 Armadillo-EVA 1500 の組み立て

## 4.4. 接続方法

「図 4.7. Armadillo-EVA 1500 の接続例」を参考に周辺機器を接続してください。



- ❶ Armadillo-EVA 1500
- ❷ AC アダプタ(12V/5A)<sup>[4]</sup>
- ❸ 作業用 PC
- ❹ USB 2.0 ケーブル(A-B タイプ)<sup>[4]</sup>
- ❺ LAN HUB
- ❻ LAN ケーブル
- ❼ HDMI 対応ディスプレイ
- ❽ HDMI ケーブル(A-A タイプ)
- ❾ SD カード(UHS-I SDR104 対応)
- ❿ SD カード(UHS-I SDR50 対応)
- ⓫ USB メモリ(USB 3.0 対応)
- ⓬ USB メモリ(USB 2.0 対応)
- ⓭ SATA 対応ハードディスク
- ⓮ SATA ケーブル
- ⓯ SATA 電源ケーブル
- ⓰ スピーカー
- ⓱ マイク
- ⓲ NTSC 対応カメラ

<sup>[4]</sup>Armadillo-EVA 1500 評価セットに同梱されているもの

⑱ シリアルクロスケーブル

図 4.7 Armadillo-EVA 1500 の接続例

## 4.5. 機能選択スイッチの設定について

機能選択用のディップスイッチを 2 つ搭載しています。出荷時の設定は「表 4.5. 機能選択スイッチの設定」のとおりです。選択できる機能の詳細については「13.22.1. 機能選択スイッチ」をご確認ください。

表 4.5 機能選択スイッチの設定

部品番号	1	2	3	4	5	6	7	8
SW1	ON	ON	ON	ON	ON	ON	OFF	ON
SW2	ON	ON	ON	ON	ON	ON	ON	OFF

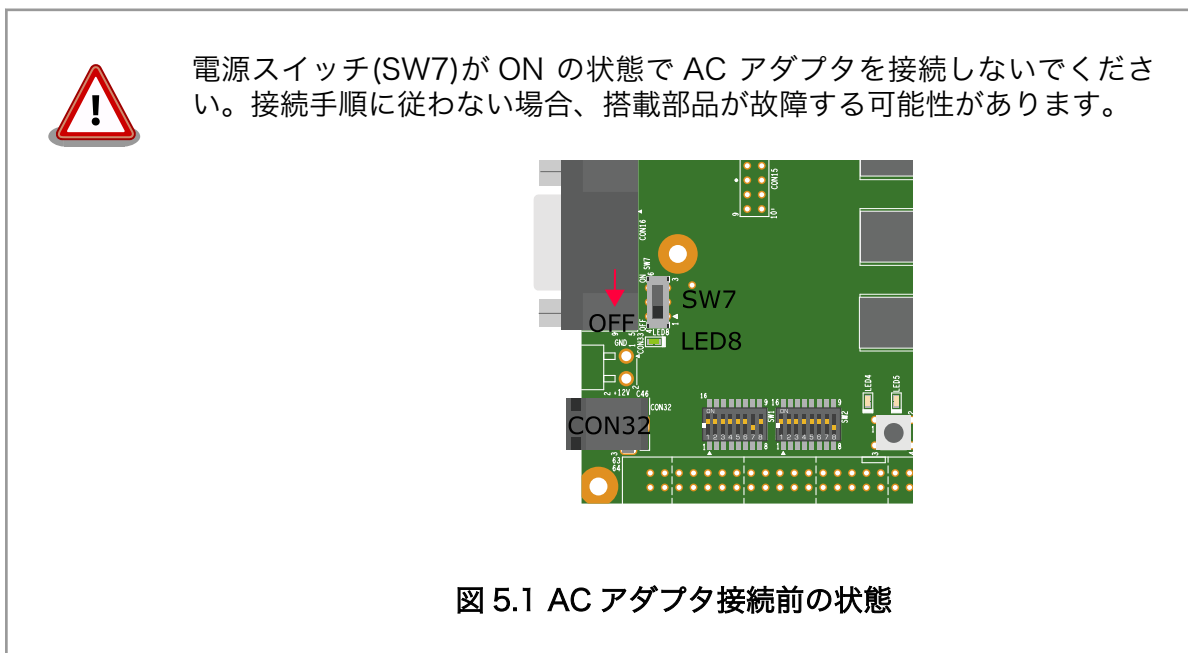


図 4.8 機能選択スイッチの設定

## 5. 起動と終了

### 5.1. 起動

AC アダプタを接続してから電源スイッチ(SW7)を ON にしてください。電源が供給されると、電源 LED(LED8)が点灯します。



起動すると、USB シリアルインターフェース(CON14)から次のようなログがシリアル通信ソフトウェアに表示されます。

```
Hermit-At v4.1.0 (Armadillo-EVA 1500) compiled at 13:55:20, Oct 21 2014
mmc0: SD card at address 0x00000001
mmc0: M8G1GC 1048576KiB
gendisk: /dev/mmcblk0p1: start=0x00000010, size=0x00b2d070
gendisk: Image.bin is found. (6694660 Bytes)
Copying kernel...done.
Uncompressing fdt.done.
Doing console=ttySC3,115200
Doing mem=768M
Doing noinitrd
Doing rootwait
Doing root=/dev/mmcblk0p1
Booting Linux on physical CPU 0x0
Initializing cgroup subsys cpuset
Initializing cgroup subsys cpu
Linux version 3.4.81-ae1500-at2 (atmark@atde5) (gcc version 4.6.3 (Debian 4.6.3-
14atmark1) ) #1 SMP PREEMPT Mon Oct 27 18:11:13 JST 2014
CPU: ARMv7 Processor [413fc0f2] revision 2 (ARMv7), cr=10c53c7d
CPU: PIPT / VIPT nonaliasing data cache, PIPT instruction cache
Machine: armadilloeva1500, model: Armadillo-EVA 1500
```



```
cma: CMA: reserved 128 MiB at 66000000
Memory policy: ECC disabled, Data cache writealloc
PERCPU: Embedded 7 pages/cpu @c0cc9000 s6144 r8192 d14336 u32768
Built 1 zonelists in Zone order, mobility grouping on. Total pages: 195072
Kernel command line: console=ttySC3,115200 mem=768M noinitrd rootwait root=/dev/
mmcbk0p1
PID hash table entries: 4096 (order: 2, 16384 bytes)
Dentry cache hash table entries: 131072 (order: 7, 524288 bytes)
Inode-cache hash table entries: 65536 (order: 6, 262144 bytes)
allocated 1572864 bytes of page_cgroup
please try 'cgroup_disable=memory' option if you don't want memory cgroups
Memory: 768MB = 768MB total
Memory: 639612k/639612k available, 146820k reserved, 8192K highmem
Virtual kernel memory layout:
  vector   : 0xffff0000 - 0xffff1000   (  4 kB)
  fixmap   : 0xfff00000 - 0xfffe0000   ( 896 kB)
  vmalloc  : 0xf0000000 - 0xff000000   (240 MB)
  lowmem   : 0xc0000000 - 0xef800000   (760 MB)
  pkmap    : 0xbfe00000 - 0xc0000000   (  2 MB)
  modules  : 0xbf000000 - 0xbfe00000   ( 14 MB)
    .text   : 0xc0008000 - 0xc05f3000   (6060 kB)
    .init   : 0xc05f3000 - 0xc0624800   ( 198 kB)
    .data   : 0xc0626000 - 0xc066a6e0   ( 274 kB)
    .bss    : 0xc066a704 - 0xc06c1444   ( 348 kB)
Preemptible hierarchical RCU implementation.
NR_IRQS:16 nr_irqs:16 16
Architected local timer running at 10.00MHz (virt).
Switching to timer-based delay loop
sched_clock: ARM arch timer >56 bits at 10000kHz, resolution 100ns
sched_clock: 32 bits at 100 Hz, resolution 10000000ns, wraps every 4294967286ms
Console: colour dummy loop device 80x30
Calibrating delay loop (skipped), value calculated using timer frequency.. 20.00
  BogoMIPS (lpj=100000)
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
CPU0: thread -1, cpu 0, socket 0, mpidr 80000000
hw perfevents: enabled with ARMv7 Cortex-A15 PMU driver, 7 counters available
Setting up static identity map for 0x40463dd0 - 0x40463e28
CPU1: Booted secondary processor
CPU1: thread -1, cpu 1, socket 0, mpidr 80000001
Brought up 2 CPUs
SMP: Total of 2 processors activated (40.00 BogoMIPS).
devtmpfs: initialized
pinctrl core: initialized pinctrl subsystem
dummy:
NET: Registered protocol family 16
DMA: preallocated 256 KiB pool for atomic coherent allocations
sh-pfc pfc-r8a7791: r8a77910_pfc support registered
renesas_irqc renesas_irqc.0: driving 10 irqs
hw-breakpoint: found 5 (+1 reserved) breakpoint and 4 watchpoint registers.
hw-breakpoint: maximum watchpoint size is 8 bytes.
bio: create slab <bio-0> at 0
```

```
gpio_rcar gpio_rcar.0: driving 32 GPIOs
gpio_rcar gpio_rcar.1: driving 32 GPIOs
gpio_rcar gpio_rcar.2: driving 32 GPIOs
gpio_rcar gpio_rcar.3: driving 32 GPIOs
gpio_rcar gpio_rcar.4: driving 32 GPIOs
gpio_rcar gpio_rcar.5: driving 32 GPIOs
gpio_rcar gpio_rcar.6: driving 32 GPIOs
gpio_rcar gpio_rcar.7: driving 32 GPIOs
pwm_rcar pwm_rcar.0: registering PWM0
pwm_rcar pwm_rcar.1: registering PWM1
pwm_rcar pwm_rcar.2: registering PWM2
pwm_rcar pwm_rcar.3: registering PWM3
pwm_rcar pwm_rcar.4: registering PWM4
pwm_rcar pwm_rcar.5: registering PWM5
pwm_rcar pwm_rcar.6: registering PWM6
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.7: using pins 238 (SDA) and 237 (SCL)
i2c-gpio i2c-gpio.8: using pins 240 (SDA) and 239 (SCL)
i2c-gpio i2c-gpio.9: using pins 71 (SDA) and 70 (SCL)
i2c-sh_mobile i2c-sh_mobile.6: I2C adapter 6 with bus speed 100000 Hz (L/H=32/2b
)
Linux media interface: v0.10
Linux video capture interface: v2.00
vgaarb: loaded
sh_cmt sh_cmt.0: used for clock events
Advanced Linux Sound Architecture Driver Version 1.0.25.
Switching to clocksource arch_sys_counter
NET: Registered protocol family 2
IP route cache hash table entries: 32768 (order: 5, 131072 bytes)
TCP established hash table entries: 131072 (order: 8, 1048576 bytes)
TCP bind hash table entries: 65536 (order: 7, 786432 bytes)
TCP: Hash tables configured (established 131072 bind 65536)
TCP: reno registered
UDP hash table entries: 512 (order: 2, 16384 bytes)
UDP-Lite hash table entries: 512 (order: 2, 16384 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.
audit: initializing netlink socket (disabled)
type=2000 audit(0.170:1): initialized
highmem bounce pool size: 64 pages
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...
jffs2: version 2.2. (NAND) © 2001-2006 Red Hat, Inc.
msgmni has been set to 1489
Block layer SCSI generic (bsg) driver version 0.4 loaded (major 252)
io scheduler noop registered
io scheduler deadline registered
io scheduler cfq registered (default)
Serial: 8250/16550 driver, 4 ports, IRQ sharing disabled
```

```

SuperH (H)SCI(F) driver initialized
sh-sci.0: ttySC0 at MMIO 0xe6c40000 (irq = 176) is a scifa
sh-sci.1: ttySC1 at MMIO 0xe6c50000 (irq = 177) is a scifa
sh-sci.2: ttySC2 at MMIO 0xe6c20000 (irq = 180) is a scifb
sh-sci.3: ttySC3 at MMIO 0xe6c30000 (irq = 181) is a scifb
console [ttySC3] enabled
sh-sci.4: ttySC4 at MMIO 0xe6ce0000 (irq = 182) is a scifb
sh-sci.5: ttySC5 at MMIO 0xe6c60000 (irq = 183) is a scifa
sh-sci.6: ttySC6 at MMIO 0xe6e60000 (irq = 184) is a scif
sh-sci.7: ttySC7 at MMIO 0xe6e68000 (irq = 185) is a scif
sh-sci.8: ttySC8 at MMIO 0xe62c0000 (irq = 186) is a hscif
sh-sci.9: ttySC9 at MMIO 0xe62c8000 (irq = 187) is a hscif
sh-sci.10: ttySC10 at MMIO 0xe6e58000 (irq = 54) is a scif
sh-sci.11: ttySC11 at MMIO 0xe6ea8000 (irq = 55) is a scif
sh-sci.12: ttySC12 at MMIO 0xe6ee0000 (irq = 56) is a scif
sh-sci.13: ttySC13 at MMIO 0xe6ee8000 (irq = 57) is a scif
sh-sci.14: ttySC14 at MMIO 0xe6c70000 (irq = 61) is a scifa
sh-sci.15: ttySC15 at MMIO 0xe6c78000 (irq = 62) is a scifa
sh-sci.16: ttySC16 at MMIO 0xe6c80000 (irq = 63) is a scifa
sh-sci.17: ttySC17 at MMIO 0xe62d0000 (irq = 53) is a hscif
scsi0 : sata_rcar
ata1: SATA max UDMA/133 irq 137
scsi1 : sata_rcar
ata2: SATA max UDMA/133 irq 138
m25p80 spi0.0: n25q064 (8192 Kbytes)
Creating 4 MTD partitions on "m25p80":
0x000000000000-0x000000010000 : "ipl"
0x000000010000-0x000000090000 : "bootloader"
0x000000090000-0x0000000d0000 : "fdt"
0x0000000d0000-0x000000080000 : "firmware"
qspi qspi.0: probed
spi_sh_msiof spi_sh_msiof.1: master is unqueued, this is deprecated
spi_sh_msiof spi_sh_msiof.2: master is unqueued, this is deprecated
spi_sh_msiof spi_sh_msiof.3: master is unqueued, this is deprecated
sh_mii: probed
Base address at 0xee700000, 00:11:0c:1a:00:00, IRQ 194.
ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
ehci-platform ehci-platform.0: Generic Platform EHCI Controller
ehci-platform ehci-platform.0: new USB bus registered, assigned bus number 1
ehci-platform ehci-platform.0: irq 140, io mem 0xee081000
ehci-platform ehci-platform.0: USB 2.0 started, EHCI 1.00
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 1 port detected
ehci-platform ehci-platform.1: Generic Platform EHCI Controller
ehci-platform ehci-platform.1: new USB bus registered, assigned bus number 2
ehci-platform ehci-platform.1: irq 145, io mem 0xee0c1000
ehci-platform ehci-platform.1: USB 2.0 started, EHCI 1.00
hub 2-0:1.0: USB hub found
hub 2-0:1.0: 1 port detected
ohci_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver
ohci-platform ohci-platform.0: Generic Platform OHCI Controller
ohci-platform ohci-platform.0: new USB bus registered, assigned bus number 3
ohci-platform ohci-platform.0: irq 140, io mem 0xee080000
hub 3-0:1.0: USB hub found
hub 3-0:1.0: 1 port detected
ohci-platform ohci-platform.1: Generic Platform OHCI Controller
ohci-platform ohci-platform.1: new USB bus registered, assigned bus number 4
ohci-platform ohci-platform.1: irq 145, io mem 0xee0c0000

```

```
hub 4-0:1.0: USB hub found
hub 4-0:1.0: 1 port detected
xhci-hcd xhci-hcd.0: xHCI Host Controller
xhci-hcd xhci-hcd.0: new USB bus registered, assigned bus number 5
xhci-hcd xhci-hcd.0: irq 133, io mem 0xee000000
ata1: link resume succeeded after 1 retries
hub 5-0:1.0: USB hub found
ata2: link resume succeeded after 1 retries
hub 5-0:1.0: 1 port detected
xhci-hcd xhci-hcd.0: xHCI Host Controller
xhci-hcd xhci-hcd.0: new USB bus registered, assigned bus number 6
hub 6-0:1.0: USB hub found
hub 6-0:1.0: 1 port 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
input: st1232-touchscreen as /devices/platform/i2c-gpio.9/i2c-9/9-0055/input/input0
rtc-s35390a 7-0030: rtc core: registered rtc-s35390a as rtc0
i2c /dev entries driver
i2c-rcar i2c-rcar_h2.0: probed
i2c-rcar i2c-rcar_h2.1: probed
i2c-rcar i2c-rcar_h2.2: probed
i2c-rcar i2c-rcar_h2.3: probed
i2c-rcar i2c-rcar_h2.4: probed
i2c-rcar i2c-rcar_h2.5: probed
lirc_dev: IR Remote Control driver registered, major 250
IR NEC protocol handler initialized
IR RC5(x) protocol handler initialized
IR RC6 protocol handler initialized
IR JVC protocol handler initialized
IR Sony protocol handler initialized
ata1: SATA link down (SStatus 0 SControl 300)
IR RC5 (streamzap) protocol handler initialized
IR SANYO protocol handler initialized
ata2: SATA link down (SStatus 0 SControl 300)
IR MCE Keyboard/mouse protocol handler initialized
IR LIRC bridge handler initialized
i2c-core: driver [adv7180] using legacy suspend method
i2c-core: driver [adv7180] using legacy resume method
soc-camera-pdrv soc-camera-pdrv.2: Probing soc-camera-pdrv.2
vin vin.2: VIN Unit driver attached to camera 0
adv7180 8-0020: chip found @ 0x40 (i2c-gpio8)
vin vin.2: VIN Unit driver detached from camera 0
uvcvideo: Unable to create debugfs directory
usbcore: registered new interface driver uvcvideo
USB Video Class driver (1.1.1)
MMNGR: C0 from 0x78000000 to 0x7fffffff
MMNGR: MXSAAR0(0x00780080)
vspm (Oct 27 2014 18:09:47) driver installed
fdpm (Oct 27 2014 18:09:45) driver for M2 installed
[drm] Initialized drm 1.1.0 20060810
Console: switching to colour frame buffer device 240x67
rcar-du rcar-du-r8a7791: fb0: frame buffer device
rcar-du rcar-du-r8a7791: registered panic notifier
[drm] Supports vblank timestamp caching Rev 1 (10.10.2010).
[drm] No driver support for vblank timestamp query.
```

```
[drm] Initialized rcar-du 1.0.0 20130110 on minor 0
PVR_K: SGX use system memory.
pvrsrvkm pvrsrvkm: start latency exceeded, new value 5700 ns
dc_linuxfb - Found usable fbdev device ():
  range (physical) = 0x66100000-0x678bb000
  size (bytes)     = 0x17bb000
  xres x yres     = 1920x1080
  xres x yres (v) = 1920x3240
  img pix fmt     = 20
  num buffers     = 3
rcar_thermal rcar_thermal: 1 sensor proved
device-mapper: ioctl: 4.22.0-ioctl (2011-10-19) initialised: dm-devel@redhat.com
sh_mmcif sh_mmcif.0: driver version 2010-04-28
mmc0: new high speed MMC card at address 0001
mmcblk0: mmc0:0001 M8G1GC 7.28 GiB
mmcblk0boot0: mmc0:0001 M8G1GC partition 1 4.00 MiB
mmcblk0boot1: mmc0:0001 M8G1GC partition 2 4.00 MiB
  mmcblk0: p1 p2
  mmcblk0boot1: unknown partition table
  mmcblk0boot0: unknown partition table
sh_mobile_sdhi sh_mobile_sdhi.0: mmc1 base at 0xee100000 clock rate 0 MHz
sh_mobile_sdhi sh_mobile_sdhi.1: mmc2 base at 0xee140000 clock rate 0 MHz
usbcore: registered new interface driver usbhid
usbhid: USB HID core driver
cs42l52 8-004a: Cirrus Logic CS42L52, Revision: 03
input: CS42L52 Beep Generator as /devices/platform/i2c-gpio.8/i2c-8/8-004a/input/input1
asoc: cs42l52 <-> scu-ssi-dai mapping ok
ip_tables: (C) 2000-2006 Netfilter Core Team
TCP: cubic registered
NET: Registered protocol family 17
Registering the dns_resolver key type
VFP support v0.3: implementor 41 architecture 4 part 30 variant f rev 0
input: gpio-keys as /devices/platform/gpio-keys/input/input2
rtc-s35390a 7-0030: setting system clock to 2000-01-01 01:18:20 UTC (946689500)
ALSA device list:
  #0: armadilloeva1500-cs42l52
kjournald starting. Commit interval 5 seconds
EXT3-fs (mmcblk0p1): mounted filesystem with ordered data mode
VFS: Mounted root (ext3 filesystem) readonly on device 179:1.
devtmpfs: mounted
Freeing init memory: 196K
INIT: version 2.88 booting
[info] Using makefile-style concurrent boot in runlevel S.
[....] Starting the hotplug events dispatcher: udevdudevd[996]: starting version
  175
. ok
[....] Synthesizing the initial hotplug events...done.
[....] Waiting for /dev to be fully populated...vin vin.2: VIN Unit driver attached to camera 0
adv7180 8-0020: Not detect any video input signal
vin vin.2: VIN Unit driver detached from camera 0
done.
[....] Activating swap...done.
[....] Checking root file system...fsck from util-linux 2.20.1
/dev/mmcblk0p1: clean, 30086/732960 files, 201350/1464846 blocks
done.
EXT3-fs (mmcblk0p1): using internal journal
```

```
[....] Cleaning up temporary files... /tmp. ok
[info] Loading kernel module loop.
libkmod: ERROR ../libkmod/libkmod.c:554 kmod_search_moddep: could not open moddep
file '/lib/modules/3.4.81-ae1500-at2/modules.dep.bin'
[....] Activating lvm and md swap...done.
[....] Checking file systems...fsck from util-linux 2.20.1
done.
[....] Mounting local filesystems...done.
[....] Activating swapfile swap...done.
[....] Cleaning up temporary files.... ok
[....] Setting kernel variables ...done.
[....] Configuring network interfaces...done.
[....] Starting rpcbind daemon...rpcbind: cannot create socket for udp6
rpcbind: cannot create socket for tcp6
. ok
[....] Starting NFS common utilities: statd idmapd. ok
[....] Cleaning up temporary files.... ok
[....] Setting up ALSA...done.
[....] Setting up X socket directories... /tmp/.X11-unix /tmp/.ICE-unix. ok
INIT: Entering runlevel: 2
[info] Using makefile-style concurrent boot in runlevel 2.
[....] Starting rpcbind daemon...[....] Already running.. ok
[....] Starting enhanced syslogd: rsyslogd. ok
[....] Starting NFS common utilities: statd idmapd. ok
[....] Starting deferred execution scheduler: atd. ok
[....] Starting periodic command scheduler: cron. ok
[....] Starting system message bus: dbus. ok
[....] Starting MTA: exim4. ok
ALERT: exim paniclog /var/log/exim4/paniclog has non-zero size, mail system poss
ibly broken

Debian GNU/Linux 7 debian ttySC3

debian login:
```

図 5.2 起動ログ

## 5.2. 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 5.1. 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 5.1 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	guest	一般ユーザ

## 5.3. 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てください。「System halted.」と表示されたのを確認したら、電源スイッチ(SW7)を OFF にした後、AC アダプタを抜いて電源を切断してください。

```
[armadillo ~]# halt
Broadcast message from root@debian (ttySC3) (Mon Jul 21 04:40:20 2014):

The system is going down for system halt NOW!
INIT: Switching to runlevel: 0
INIT: Sending processes the TERM signal
[info] Using makefile-style concurrent boot in runlevel 0.
[....] Stopping MTA:. ok
ALERT: exim paniclog /var/log/exim4/paniclog has non-zero size, mail system possibly broken
[....] Stopping deferred execution scheduler: atd. ok
[....] Shutting down ALSA...done.
[....] Asking all remaining processes to terminate...done.
[....] All processes ended within 2 seconds...done.
[....] Stopping enhanced syslogd: rsyslogd. ok
rpcbind: rpcbind terminating on signal. Restart with "rpcbind -w"
[....] Stopping rpcbind daemon... ok
[....] Deconfiguring network interfaces...done.
[....] Stopping NFS common utilities: idmapd statd. ok
[info] Saving the system clock.
[info] Hardware Clock updated to Mon Jul 21 04:40:25 JST 2014.
[....] Deactivating swap...done.
[....] Unmounting local filesystems...done.
[info] Will now halt.
System halted.
```

図 5.3 終了方法



まれに **halt** コマンド実行後に出力されるログが、「System halted.」と表示される前に停止してしまう場合があります。1 分程度待ってもログが停止した状態が続く場合は、そのまま電源スイッチ(SW7)を OFF にしてください。

## 6. 各種機能を使用する

本章では、Debian で Armadillo-EVA 1500 の機能を使用する方法について記載します。本章に記載のある手順を実行するためには、事前に「5. 起動と終了」を参照し、起動およびログインが完了している必要があります。

### 6.1. ネットワーク

有線 LAN を使用する方法について記載します。

工場出荷状態では有線 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 6.1. 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/eth0/00:11:0c:xx:xx:xx
Sending on   LPF/eth0/00:11:0c:xx:xx:xx
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 ❷
eth0      Link encap:Ethernet  HWaddr 00:11:0c:xx:xx:xx
          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
```

- ❶ 有線 LAN インターフェースを有効化します。工場出荷状態では 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ます。
- ❷ 有線 LAN インターフェースの状態を表示します。

図 6.1 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 6.2. 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/eth0/00:11:0c:xx:xx:xx
Sending on   LPF/eth0/00:11:0c:xx:xx:xx
Sending on   Socket/fallback
DHCPRELEASE on eth0 to 172.16.0.1 port 67
[armadillo ~]# ifconfig eth0 ②
eth0      Link encap:Ethernet  HWaddr 00:11:0c:xx:xx:xx
          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 6.2 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行います。工場出荷状態では、DHCP サーバーから 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
```

## 6.2. 時刻設定

時刻を設定する方法について記載します。時刻には、Linux カーネルが管理するシステムクロックと、リアルタイムクロックが管理するハードウェアクロックの 2 種類があります。ハードウェアクロックに時刻を設定すると、電源を切断しても一定時間時刻を保持することができます。

ハードウェアクロック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。その後に、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。

表 6.1 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)

2014 年 1 月 23 日 4 時 56 分 00 秒に設定する例を次に示します。

```
[armadillo ~]# date ❶
Sat Jan 1 09:00:00 JST 2000
[armadillo ~]# date 012304562014.00 ❷
Thu Jan 23 04:56:00 JST 2014
[armadillo ~]# date ❸
Thu Jan 23 04:56:00 JST 2014
```

- 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 6.3 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
Thu Jan 23 04:56:15 2014 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システムクロックと一致させます。ハードウェアクロックは協定世界時(UTC)で設定されます。
- 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 6.4 ハードウェアクロックを設定

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 外部バックアップインターフェース(CON31)に外付けバッテリー(CR2032 等)を接続することができます。

## 6.3. パッケージ管理

パッケージ管理システム APT(Advanced Packaging Tool)を使用して、パッケージを管理する方法について記載します。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が、APT 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。

工場出荷状態では、APT はインターネット上の Debian サイト (HTTP サーバー) 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<sup>[1]</sup>。そのため、APT を使用するためにはネットワークを有効化し、インターネットに接続できる状態にしておく必要があります。

ネットワークを有効化する方法については、「6.1. ネットワーク」を参照してください。



システムクロックが大幅にずれた状態で、APT 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「6.2. 時刻設定」を参照してシステムクロックを合わせてください。

### 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
```

<sup>[1]</sup>/etc/apt/sources.list で設定しています。記述ルールなどについては、sources.list のマニュアルページを参照してください。

### 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しません。

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
```

## 6.4. GStreamer

GStreamer のインストールと使用方法について記載します。

GStreamer はオープンソースのマルチメディアのフレームワークです。動画や音声などを使用するマルチメディアアプリケーションの基礎となる機能を提供します。

GStreamer には、動作確認用のテストツールとして **gst-launch-0.10** コマンドが用意されています。**gst-launch-0.10** を使用すると、プログラムを作成することなく GStreamer の機能を使用することができます。

次に示す機能の使用には、**gst-launch-0.10** を利用します。

- ・「6.7. ビデオ」
- ・「6.8. オーディオ」

### 6.4.1. GStreamer のインストール

GStreamer のインストールには APT を使用します。APT の使用方法については、「6.3. パッケージ管理」を参照してください。

APT を使用して 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 gstreamer0.10-tools gstreamer0.10-plugins-base gstreamer0.10-
plugins-good gstreamer0.10-alsa
```



図 6.5 GStreamer のインストール

## 6.4.2. GStreamer の使用

本書では、GStreamer を **gst-launch-0.10** コマンドを使用して利用します。

GStreamer では、特定の機能を持ったエレメントをリンクすることによって、目的とする機能を実現します。**gst-launch-0.10** では、エクスクラメーションマーク「!」によってエレメントをリンクします。

エレメントは、マルチメディアデータを入力または出力することのできるパッドを持ちます。エレメントは、エレメントが持つパッドによって 3 種類に分類されます。

表 6.2 エレメントの種類

エレメント	説明
ソースエレメント	出力用のパッドのみを持ちます。
フィルタエレメント	入力用と出力用のパッドをそれぞれ 1 つ以上持ちます。
シンクエレメント	入力用のパッドのみを持ちます。

各エレメントをそれぞれ 1 つずつ持った **gst-launch-0.10** の使用例を次に示します。

```
[armadillo ~]# gst-launch-0.10 videotestsrc ! pngenc ! filesink location=sample.png
```

図 6.6 テスト画像をファイルに保存

videotestsrc は、テスト画像を生成するソースエレメントです。pngenc は、入力画像を PNG 形式にエンコードするフィルタエレメントです。filesink は、ファイルにデータを出力するシンクエレメントです。この 3 つのエレメントをリンクすることで、テスト画像を PNG 形式でファイルに出力します。

GStreamer のより詳細な情報については、GStreamer 公式サイト (<http://gstreamer.freedesktop.org/>)を参照してください。

## 6.5. X サーバー

X サーバーのインストールと起動方法について記載します。

次に示す機能の使用には、X サーバーを利用します。

- ・「6.7. ビデオ」

### 6.5.1. X サーバーのインストール

X サーバーのインストールには APT を使用します。APT の使用方法については、「6.3. パッケージ管理」を参照してください。

APT を使用して X サーバー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
```

図 6.7 X サーバーのインストール

### 6.5.2. X サーバーの起動

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。

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

図 6.8 X サーバーの起動

X サーバーが起動するとディスプレイに次のように起動画面が表示されます。

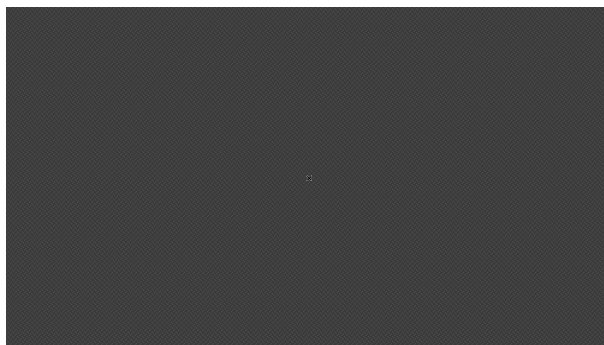


図 6.9 X サーバー起動画面



X サーバーの起動オプション "-retro" を指定する目的は、起動画面によって X サーバーの正常起動を確認しやすくするためです。不要であれば指定する必要はありません。



ディスプレイに表示された X サーバー起動画面のちらつきが気になる場合は、次のようにコマンドを実行してください。

```
[armadillo ~]# X -wr &
```

X サーバーの起動オプション "-retro" を指定した場合は網目模様の背景が表示されますが、"-wr"を指定した場合は単色白色の背景が表示されます。

## 6.6. ストレージ

ストレージを使用する方法について記載します。Armadillo-EVA 1500 では、内蔵ストレージ(eMMC)の他に SD カードや USB メモリ、SATA 接続の SSD などの外部ストレージを使用することができます。外部ストレージを使用することで、容易に作業用 PC などとデータ転送を行うことができます。

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 6.3 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	備考
eMMC	/dev/mmcblk0	/dev/mmcblk0p1	内蔵ストレージ。容量 8GB。
USB フラッシュメモリ	/dev/sd*[a]	/dev/sd*1	外部ストレージ。USB 3.0 ホストインターフェース (CON10)または USB 2.0 ホストインターフェース (CON8)に接続。
SD/SDHC/SDXC カード	/dev/mmcblk*[b]	/dev/mmcblk*p1	外部ストレージ。SD インターフェース(CON6)または SD インターフェース(CON4)に接続。
SSD/HDD	/dev/sd*[a]	/dev/sd*1	外部ストレージ。SATA インターフェース(CON11)に接続。

[a]ストレージが認識された順に sda sdb sdc ... となります。

[b]ストレージが認識された順に mmcblk0 mmcblk1 となります。



工場出荷状態の eMMC はいくつかのパーティションに分割されており、それぞれのパーティションに OS やルートファイルシステムなどのソフトウェアが格納されています。eMMC をストレージとして使用する場合は、使用しているソフトウェアが格納されたパーティションへの書き込みを行わないようご注意ください。



SD インターフェース(CON4)と WLAN インターフェース(CON5)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。どちらのインターフェースを使用するかは、機能選択スイッチの SD/WLAN 設定 (SW2.5)で選択できるようになっています。



USB 2.0 ホストインターフェース(CON8)と USB 2.0 デバイスインターフェース(CON9)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。どちらのインターフェースを使用するかは、機能選択スイッチの USB 2.0 設定(SW2.4)で選択できるようになっています。



SATA インターフェース(CON11)と miniPCIe インターフェース(CON12)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。どちらのインターフェースを使用するかは、機能選択スイッチの USB 3.0、SATA/miniPCIe 設定(SW1.2、SW1.3)で選択できるようになっています。

## 6.6.1. 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「6.6.2. 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、次のとおりです。

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

図 6.10 mount コマンド書式

-t オプションに続く device には、ファイルシステムタイプを指定します<sup>[2]</sup>。FAT32 ファイルシステムの場合は vfat<sup>[3]</sup>、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/mmcblk1p1、パーティション 2 の場合は /dev/mmcblk1p2 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 6.11. 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/mmcblk1p1 /mnt
```

図 6.11 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 IO charset for
FAT filesystems, filesystem will be case sensitive!
```

<sup>[2]</sup>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。

<sup>[3]</sup>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。



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。

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

図 6.12 ストレージのアンマウント

## 6.6.2. 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 6.13. `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/mmcbk1p1`、次は `/dev/mmcbk1p2` 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/mmcbk1
```

```
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.
mmcblk1: p1 p2
mmcblk1: p1 p2
Syncing disks.

```

図 6.13 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/mmcblk1p1
```

図 6.14 EXT3 ファイルシステムの構築

## 6.7. ビデオ

ディスプレイへの画像出力およびカメラからの画像入力を行う方法について記載します。

Linux では、画像出力インターフェースをフレームバッファデバイスとして、画像入力インターフェースを V4L2(Video for Linux Two)デバイスとして利用できます。標準状態で利用可能なそれぞれのデバイスを次に示します。

### フレームバッファデバイス - /dev/fb0

デジタル HD 出インターフェース(CON17)  
 解像度: 1920 x 1080<sup>[4]</sup>  
 カラーフォーマット: ARGB8888 (32bit)

### V4L2 デバイス - /dev/video0

コンポジットビデオ入インターフェース(CON26)  
 解像度: 720 x 480  
 カラーフォーマット: YUYV (16bit)

### 6.7.1. ディスプレイに画像を表示する

ディスプレイに画像を表示させる方法について記載します。画像の表示には、GStreamer を利用します。インストールされていない場合は、「6.4.1. 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、GStreamer からフレームバッファへの直接描画はできないため、X サーバーを利用します。事前に「6.5. X サーバー」を参照して X サーバーを起動してください。

任意の JPEG 画像をディスプレイに表示する手順を次に示します。

<sup>[4]</sup>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。

```
[armadillo ~]# DISPLAY=:0 gst-launch-0.10 filesrc location=sample.jpg ! jpegdec ! freeze !
ffmpegcolorspace ! autovideosink
```



図 6.15 JPEG 画像の表示



画面出力インターフェースはデータの入力が一定時間無かった場合、信号を出力しない blank 状態になります。blank 状態を解除するには下記のコマンドを実行してください。

```
[armadillo ~]# echo 0 > /sys/class/graphics/fb0/blank
```



JPEG ファイルのサンプルには、gnome-backgrounds 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。JPEG ファイルをディスプレイの解像度に合わせて表示するコマンド例を次に示します。

```
[armadillo ~]# apt-get install gnome-backgrounds imagemagick
[armadillo ~]# cp /usr/share/backgrounds/gnome/BlueMarbleWest.jpg .
[armadillo ~]# convert -geometry 1920x1080! BlueMarbleWest.jpg
sample.jpg
[armadillo ~]# X -retro &
[armadillo ~]# DISPLAY=:0 gst-launch-0.10 filesrc location=sample.jpg !
jpegdec ! freeze ! ffmpegcolorspace ! autovideosink
```



図 6.16 gnome-backgrounds パッケージの画像ファイルを利用

## 6.7.2. カメラ画像をディスプレイに表示する

カメラから取得した画像をディスプレイに表示させる方法について記載します。画像の表示には、GStreamer を利用します。インストールされていない場合は、「6.4.1. 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、GStreamer からフレームバッファへの直接描画はできないため、X サーバーを利用します。事前に「6.5. X サーバー」を参照して X サーバーを起動してください。

カメラから取得した画像をディスプレイに表示させる手順を次に示します。

```
[armadillo ~]# DISPLAY=:0 gst-launch-0.10 v4l2src ! "video/x-raw-yuv,width=720,height=480" !
ffmpegcolorspace ! autovideosink
```



図 6.17 カメラ画像の表示

「6.5. X サーバー」を実行すると、ディスプレイに次のように画面が表示されます。

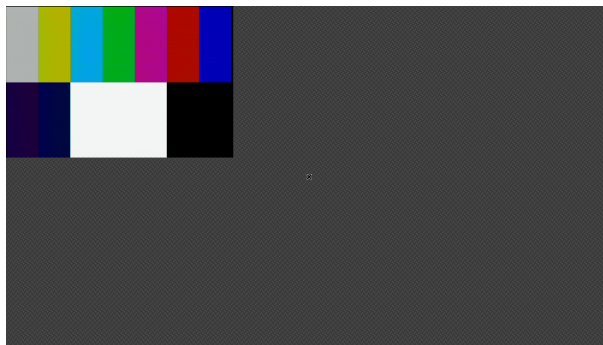


図 6.18 カメラ画像の表示画面

### 6.7.3. ディスプレイのビデオモードを設定する

ディスプレイの解像度やカラーフォーマットなどを設定する方法について記載します。

ディスプレイの解像度を 640 x 480 に、走査方式をプログレッシブに設定するには「図 6.19. 解像度の変更」のようにコマンドを実行します。fbset 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 ~]# fbset -xres 640 -yres 480 -laced 0
```

図 6.19 解像度の変更



ディスプレイがサポートしていない解像度や、ドットクロックが 100MHz を超える解像度に設定することはできません。

ディスプレイのカラーフォーマットを ARGB1555 に設定するには次のようにコマンドを実行します。

```
[armadillo ~]# fbset -depth 16 -rgba 5,5,5,1
```

図 6.20 カラーフォーマットの変更

Armadillo-EVA 1500 がサポートしているカラーフォーマットは次の通りです。

カラーフォーマット	色深度
RGB565	16
ARGB1555	16
XRGB8888	32
ARGB8888	32

## 6.8. オーディオ

サウンドの再生および録音を行う方法について記載します。

Linux では、オーディオ機能を ALSA<sup>[5]</sup>デバイスとして利用できます。標準状態で利用可能な ALSA デバイスを次に示します。

### ALSA デバイス - hw:0

マイク/ライン入力インターフェース(CON21)  
 ヘッドホン出力インターフェース(CON18)  
 サンプリング周波数: 96k, 48k, 32k, 16k, 8k Hz  
 チャンネル数: 1 or 2  
 フォーマット: Signed/Unsigned 16/18/20/24 bit, Little-endian

## 6.8.1. 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。

サウンドを再生するには、事前に再生パスのルーティング設定を行う必要があります。次のようにコマンドを実行します。

```
[armadillo ~]# amixer cset name='SSI3 Control' 1
```

図 6.21 再生パスのルーティング設定

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。

```
[armadillo ~]# gst-launch-0.10 audiotestsrc ! \  

  "audio/x-raw-int,channels=2,rate=48000,width=16" ! \ ①  

  alsasink device=hw:0 ②
```

- ① rate パラメータには、サンプリング周波数を指定します。
- ② device パラメータには、ALSA デバイスを指定します。

図 6.22 テストサウンドの再生

## 6.8.2. サウンドを録音する

マイク/ライン入力インターフェース(CON21)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。

サウンドを録音するには、事前に録音パスのルーティング設定を行う必要があります。次のようにコマンドを実行します。

```
[armadillo ~]# amixer cset name='SSI4 Control' 1  

[armadillo ~]# amixer cset name='PGA AIN3L Switch' 1  

[armadillo ~]# amixer cset name='PGA AIN3R Switch' 1
```

図 6.23 録音パスのルーティング設定

<sup>[5]</sup>Advanced Linux Sound Architecture <http://alsa.sourceforge.net>

```
[armadillo ~]# gst-launch-0.10 alsasrc device=hw:0 ! \ ❶
                    wavenc ! \ ❷
                    filesink location=sample.wav ❸
```

- 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 6.24 サウンドの録音

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、ヘッドホン出力インターフェース(CON18) hw:0 に出力するように指定しています。

```
[armadillo ~]# gst-launch-0.10 filesrc location=sample.wav ! \
                    wavparse ! \
                    alsasink device=hw:0
```

図 6.25 録音したファイルを再生

## 6.9. マルチメディアパッケージ

マルチメディアパッケージを使い、H.264/AVC 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。



マルチメディアパッケージは、次に示すソフトウェアのバージョンで対応しました。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ソフトウェアを対応バージョンに更新してください。

ソフトウェア	対応バージョン
Hermit-At	v4.1.0 以降(Hermit-At ブートローダーイメージ loader-armadilloeva1500-v4.1.0.bin 以降)
Linux カーネル	v3.4-ae1500-at2 以降(Linux カーネルイメージ linux-ae1500-v1.01.bin 以降)
Debian GNU/Linux	v20141022 以降(Debian GNU/Linux ルートファイルシステムアーカイブ debian-wheezy-armhf_ae1500_20141022.tar.gz 以降)
ファームウェア	v1.00 以降(ファームウェアイメージ squashfs-ae1500-firmware-v1.00.img 以降)



マルチメディアパッケージを利用するためには、Debian GNU/Linux に次に示すバージョンのパッケージがインストールされている必要があります。

パッケージ	バージョン
gstreamer0.10-openmax	0.10.1atmark1
gstreamer0.10-plugins-good	0.10.31-3atmark1
libdrm2	2.4.56-1atmark1
libdrm2-test	2.4.56-1atmark1
media-ctl	0.0.1-1 以降
mmp-armadilloeva1500	1.0.0 以降

パッケージの削除やアップデートをしてしまった場合は、該当バージョンのパッケージを再インストールしてください。各パッケージは、Armadillo サイトまたは評価セット付属の DVD-ROM から取得可能です。

動作確認に利用する動画は、Armadillo サイトから取得してストレージに保存します。「6.1. 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<sup>[6]</sup>。

表 6.4 サンプル動画

ファイル名	説明
big_buck_bunny_1080p-5.1ch-3min.mp4	Full HD(1920x1080)サイズ 約3分
big_buck_bunny_720p-5.1ch-3min.mp4	HD(1280x720)サイズ 約3分
tears_of_steel_720p-2ch-4min.mp4	HD(1280x720)サイズ 約4分

以降の例では、動画を内蔵ストレージ(eMMC)に保存します。USB フラッシュメモリや SD カードを利用する場合は「6.6. ストレージ」を参照し、動画を保存するディレクトリを適宜読み替えてください。

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 ~]# wget http://download.atmark-techno.com/sample/movie/
big_buck_bunny_1080p-5.1ch-3min.mp4
[armadillo ~]# wget http://download.atmark-techno.com/sample/movie/
big_buck_bunny_720p-5.1ch-3min.mp4
[armadillo ~]# wget http://download.atmark-techno.com/sample/movie/
tears_of_steel_720p-2ch-4min.mp4
```



図 6.26 サンプル動画の取得



Big Buck Bunny は、Creative Commons Attribution 3.0 Unported (CC BY 3.0) ライセンス下にあります。(c) copyright 2008, Blender Foundation / [www.bigbuckbunny.org](http://www.bigbuckbunny.org)





Tears of Steel は、Creative Commons Attribution 3.0 Unported (CC BY 3.0) ライセンス下にあります。(cc) Blender Foundation | [mango.blender.org](http://mango.blender.org)

<sup>[6]</sup>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。サンプル動画と動画サイズは次の通りです。



### 6.9.1. Linux カーネルの起動オプションを設定する

マルチメディアパッケージを利用するために、Linux カーネルの起動オプションを設定します。マルチメディアパッケージが使用するメモリを Linux カーネルに管理させないようにするため、Linux カーネルが利用可能なメモリを 768MByte に制限します。



すでに Linux カーネルの起動オプションを設定済みの場合は、以降の設定を行う必要はありません。Linux カーネルの起動オプションは SPI フラッシュメモリに保存されるためです。



マルチメディアパッケージが使用するメモリは 128MByte であるため、Linux カーネルが利用可能なメモリを 896MByte 以下に制限すると利用可能になります。ここでは、今後リリースされるソフトウェア互換性を重視して 768MByte に制限しています。メモリマップについては「表 3.5. Linux 起動時の DDR3-SDRAM メモリマップ」を参照してください。

#### 手順 6.1 マルチメディアパッケージを利用するための Linux カーネルの起動オプション設定

1. Linux カーネルの起動オプションを設定するため、Hermit-At ブートローダーを保守モードで起動します。機能選択スイッチの Hermit-At 起動モード設定(SW2.8)を ON に設定して電源を投入します。
2. 次のように Linux カーネルの起動オプションを設定します。

```
hermit> setenv console=ttySC3,115200 mem=768M noinitrd rootwait root=/dev/mmcblk0p1
```

### 6.9.2. 動画を再生する

ディスプレイに動画をフルスクリーンで再生する方法について記載します。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 1080i の表示に対応している必要があります。

ディスプレイに Full HD サイズ(1920x1080 画素)のサンプル動画を再生する例を次に示します。

#### 手順 6.2 ディスプレイに Full HD サイズのサンプル動画を再生

1. メディアデバイス(/dev/media0)の設定をリセットします。

```
[armadillo ~]# media-ctl -d /dev/media0 -r
```

2. メディアデバイスにリンク情報を設定します。



```
[armadillo ~]# media-ctl -d /dev/media0 -l 'vsp1.2 rpf.0':1 -> 'vsp1.2 uds.0':0 [1]
[armadillo ~]# media-ctl -d /dev/media0 -l 'vsp1.2 uds.0':1 -> 'vsp1.2 wpf.0':0 [1]
[armadillo ~]# media-ctl -d /dev/media0 -l 'vsp1.2 wpf.0':1 -> 'vsp1.2 lif':0 [1]
```

3. 動画のサイズを設定します。サンプル動画は Full HD サイズであるため、"1920x1080"を設定します。

```
[armadillo ~]# media-ctl -d /dev/media0 -V 'vsp1.2 rpf.0':0 [fmt:AYUV32/1920x1080]
[armadillo ~]# media-ctl -d /dev/media0 -V 'vsp1.2 rpf.0':1 [fmt:AYUV32/1920x1080]
[armadillo ~]# media-ctl -d /dev/media0 -V 'vsp1.2 uds.0':0 [fmt:AYUV32/1920x1080]
```

4. ディスプレイへの描画サイズを設定します。インターレース方式で描画するため垂直解像度には 1080 画素の 1/2 を使用し、"1920x540"を設定します。

```
[armadillo ~]# media-ctl -d /dev/media0 -V 'vsp1.2 uds.0':1 [fmt:AYUV32/1920x540]
[armadillo ~]# media-ctl -d /dev/media0 -V 'vsp1.2 wpf.0':0 [fmt:AYUV32/1920x540]
[armadillo ~]# media-ctl -d /dev/media0 -V 'vsp1.2 wpf.0':1 [fmt:ARGB32/1920x540]
[armadillo ~]# media-ctl -d /dev/media0 -V 'vsp1.2 lif':0 [fmt:ARGB32/1920x540]
```

5. DRM(Direct Rendering Manager)の動作モードを設定します。出力モードに 1080i を示す"1920x1080i"を、入力モードに Full HD サイズを示す"1920x1080"を設定します。

```
[armadillo ~]# modetest -M rcar-du -s 9@7:1920x1080i@AR24 -d \
-P '7@18:1920x1080@XR24' &
```

6. ディスプレイに動画を再生します。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-0.10 filesrc location=./big_buck_bunny_1080p-5.1ch-3min.mp4 \
! qtdemux ! omx_h264dec buffer-without-copy=true \
! v4l2sink device=/dev/media0 -e \
"vsp1.2 rpf.0 input" buffer-without-copy=true
```




**modetest** コマンドの詳細な使い方は help を参照してください。

```
[armadillo ~]# modetest -h
```

-s および-P オプションで指定する Armadillo-EVA 1500 固有の ID を次に示します。

ID	値	説明
crtc_id	6	DU0(Display Unit 0)
	7	DU1(Display Unit 1)
source_id	18	VSPD0(フォーマット: XR24)
	19	VSPD1(フォーマット: XR24)
connector_id	9	デジタル HD 出力
	11	LVDS



これら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。

### 6.9.3. 2つの動画を同時に再生する

ディスプレイに2つの動画を同時に再生する方法について記載します。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 720p の表示に対応している必要があります。

ディスプレイに HD サイズ(1280x720 画素)のサンプル動画を 640x360 画素にダウンスケールして再生する例を次に示します。

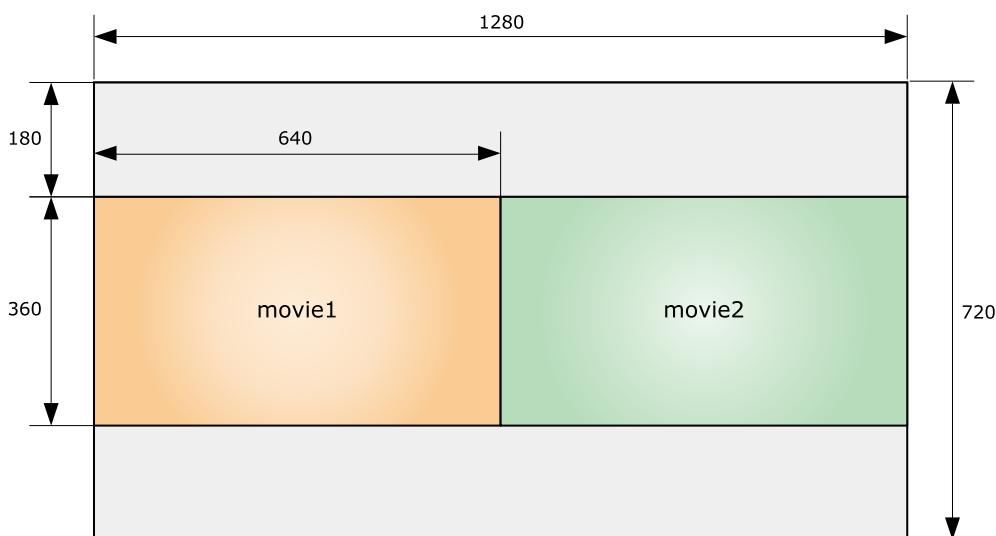



図 6.27 動画再生の実行イメージ



以降の例に示すディスプレイのビデオモードを、インターレース方式として読み替えると正常に動作しない場合があります。インターレース方式では、フルスクリーンの動画再生にのみ対応しています。

インターレース方式でフルスクリーンの動画再生を行う場合は、「6.9.2. 動画を再生する」を参照してください。

ディスプレイにサンプル動画を2つ同時に再生する例を次に示します。各コマンドのより詳細な説明については、「6.9.2. 動画を再生する」を参照してください。

### 手順 6.3 ディスプレイに HD サイズのサンプル動画を2つ再生

1. movie1 の再生に該当するメディアデバイス(/dev/media0)と DRM の動作モードを設定します。

```
[armadillo ~]# media-ctl -d /dev/media0 -r
[armadillo ~]# media-ctl -d /dev/media0 -l "vsp1.2 rpf.0":1 -> "vsp1.2 uds.0":0 [1]'
[armadillo ~]# media-ctl -d /dev/media0 -l "vsp1.2 uds.0":1 -> "vsp1.2 wpf.0":0 [1]'
[armadillo ~]# media-ctl -d /dev/media0 -l "vsp1.2 wpf.0":1 -> "vsp1.2 lif":0 [1]'
[armadillo ~]# media-ctl -d /dev/media0 -V "vsp1.2 rpf.0":0 [fmt:AYUV32/1280x720]'
[armadillo ~]# media-ctl -d /dev/media0 -V "vsp1.2 rpf.0":1 [fmt:AYUV32/1280x720]'
[armadillo ~]# media-ctl -d /dev/media0 -V "vsp1.2 uds.0":0 [fmt:AYUV32/1280x720]'
[armadillo ~]# media-ctl -d /dev/media0 -V "vsp1.2 uds.0":1 [fmt:AYUV32/640x360]'
[armadillo ~]# media-ctl -d /dev/media0 -V "vsp1.2 wpf.0":0 [fmt:AYUV32/640x360]'
[armadillo ~]# media-ctl -d /dev/media0 -V "vsp1.2 wpf.0":1 [fmt:ARGB32/640x360]'
[armadillo ~]# media-ctl -d /dev/media0 -V "vsp1.2 lif":0 [fmt:ARGB32/640x360]'
[armadillo ~]# modetest -M rcar-du -s 9@7:1280x720@AR24 -d \
-P '7@18:640x360+0+180@XR24' &
```

2. movie2 の再生に該当するメディアデバイス(/dev/media1)と DRM の動作モードを設定します。

```
[armadillo ~]# media-ctl -d /dev/media1 -r
[armadillo ~]# media-ctl -d /dev/media1 -l "vsp1.3 rpf.0":1 -> "vsp1.3 uds.0":0 [1]'
[armadillo ~]# media-ctl -d /dev/media1 -l "vsp1.3 uds.0":1 -> "vsp1.3 wpf.0":0 [1]'
[armadillo ~]# media-ctl -d /dev/media1 -l "vsp1.3 wpf.0":1 -> "vsp1.3 lif":0 [1]'
[armadillo ~]# media-ctl -d /dev/media1 -V "vsp1.3 rpf.0":0 [fmt:AYUV32/1280x720]'
[armadillo ~]# media-ctl -d /dev/media1 -V "vsp1.3 rpf.0":1 [fmt:AYUV32/1280x720]'
[armadillo ~]# media-ctl -d /dev/media1 -V "vsp1.3 uds.0":0 [fmt:AYUV32/1280x720]'
[armadillo ~]# media-ctl -d /dev/media1 -V "vsp1.3 uds.0":1 [fmt:AYUV32/640x360]'
[armadillo ~]# media-ctl -d /dev/media1 -V "vsp1.3 wpf.0":0 [fmt:AYUV32/640x360]'
[armadillo ~]# media-ctl -d /dev/media1 -V "vsp1.3 wpf.0":1 [fmt:ARGB32/640x360]'
[armadillo ~]# media-ctl -d /dev/media1 -V "vsp1.3 lif":0 [fmt:ARGB32/640x360]'
[armadillo ~]# modetest -M rcar-du -s 9@7:1280x720@AR24 -d \
-P '7@19:640x360+640+180@XR24' &
```

3. ディスプレイに動画を再生します。動画の再生を途中で停止する場合は、killall コマンドで **gst-launch-0.10** を終了させてください。

```
[armadillo ~]# gst-launch-0.10 filesrc location=./big_buck_bunny_720p-5.1ch-3min.mp4 \
! qtdemux ! omx_h264dec buffer-without-copy=true \
! v4l2sink device=`media-ctl -d /dev/media0 -e \
"vsp1.2 rpf.0 input"` buffer-without-copy=true & \
gst-launch-0.10 filesrc location=./movie/tears_of_steel_720p-2ch-4min.mp4 \
! qtdemux ! omx_h264dec buffer-without-copy=true \
! v4l2sink device=`media-ctl -d /dev/media1 -e \
"vsp1.3 rpf.0 input"` buffer-without-copy=true &
```



動画の描画位置は、**modetest** コマンドの-P オプションで指定しています。詳しくは **modetest** の help を参照してください。

```
[armadillo ~]# modetest -h
```



これら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。

## 6.10. シリアル

シリアルインターフェースを使用する方法について記載します。工場出荷状態の Armadillo-EVA 1500 で使用可能なシリアルインターフェースは次の通りです。

表 6.5 工場出荷状態のシリアル設定

インターフェース	デバイスファイル(ATDE5)	デバイスファイル(Armadillo-EVA 1500)
USB シリアルインターフェース(CON14)	/dev/ttyUSB0	/dev/ttySC3
シリアルインターフェース(CON16)	/dev/ttyS0	/dev/ttySC2



「表 6.5. 工場出荷状態のシリアル設定」に示すデバイスファイルは、「4.4. 接続方法」に従って接続された場合のものです。異なる接続をしている場合は適宜読み替えてください。

### 6.10.1. シリアルコンソールとして使用する

シリアルの動作を確認するには、シリアルコンソールを起動させます。ATDE のシリアル通信ソフトウェア(minicom)を用いることで、シリアル経由で Armadillo にログインすることができます。



Armadillo-EVA 1500 の工場出荷イメージでは、USB シリアルインターフェース(CON14)が標準でシリアルコンソールとして使用できるようになっています。

シリアルインターフェース(CON16)をシリアルコンソールとして使用する手順を次に示します。

#### 手順 6.4 シリアルコンソールとして使用

1. ATDE で minicom を起動します。シリアルデバイスには/dev/ttyS0 を指定します。

```
[ATDE ~]$ minicom -o -w -D /dev/ttyS0
```

- Armadillo で getty を起動します。シリアルデバイスには ttySC2<sup>[7]</sup>を指定します。

```
[armadillo ~]# getty -L 115200 ttySC2 &
```

ATDE の minicom にログインプロンプトが表示されます。ユーザー「guest」でログインすることができます。



次に示すように/etc/securetty に端末(シリアルデバイス)を登録すると、特権ユーザー「root」でログインすることが可能になります。

```
[armadillo ~]# echo ttySC2 >> /etc/securetty
```

## 6.11. 温度センサー

プロセッサ(RZ/G1M)の、温度を取得する方法について記載します。利用しているデバイスドライバは Thermal クラスとして実装されているため、Thermal クラスディレクトリ以下のファイルによって温度を取得することができます。

次のようにプロセッサの温度を取得することができます。

```
[armadillo ~]# cat /sys/class/thermal/thermal_zone0/temp
55000 ❶
```

- ❶ 温度はミリ°C の単位で表示されます。この例では 55°C を示しています。

図 6.28 温度の取得

温度センサーのより詳細な仕様については、「7.4.18. 温度センサー」を参照してください。

## 6.12. LED

LED を点灯および消灯する方法について記載します。

Armadillo-EVA 1500 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 6.6 LED - Armadillo-EVA 1500

LED クラスディレクトリ	説明	デフォルトトリガ
/sys/class/leds/LED4/	LED4	none
/sys/class/leds/LED5/	LED5	none
/sys/class/leds/LED6/	LED6	none
/sys/class/leds/LED7/	LED7	none

<sup>[7]</sup>/dev/を指定する必要はありません。

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。

### 6.12.1. 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 6.29 LED を点灯させる



Armadillo-EVA 1500 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 6.30 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 6.31 LED の状態を表示する

### 6.12.2. 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 6.7 trigger の種類

設定	説明
none	トリガを設定しません。
mmc0	eMMC のアクセスランプにします。
mmc1	SD カード(CON6)のアクセスランプにします。
mmc2	SD カード(CON4)のアクセスランプにします。
timer	任意のデューティー比で繰り返し点灯/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 <sup>[a]</sup>	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。

設定	説明
default-on <sup>[a]</sup>	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。

<sup>[a]</sup>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。

次に示す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 6.32 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 mmc1 mmc2 timer
```

図 6.33 LED のトリガを表示する

## 6.13. ユーザースイッチ

ユーザースイッチのイベントを取得する方法について記載します。Armadillo-EVA 1500 のユーザースイッチはインプットデバイスとして実装されています。ユーザースイッチとして使用可能なスイッチと、インプットデバイスファイルの対応を次に示します。

表 6.8 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
SW3	/dev/input/event2	4
SW4		5
SW5		6
SW6		7



イベントデバイスファイルの番号は、検出された順番に割り振られます。そのため、USB キーボードなど他のインプットデバイスが起動時に検出されると、デバイスファイルのインデックスが変わる可能性があります。

### 6.13.1. イベントを確認する

ユーザースイッチのイベントを取得するには、次のようにコマンドを実行します。**evtest** を停止するには、Ctrl+c を押下してください。

```
[armadillo ~]# evtest /dev/input/event2
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 (EV_SYN)
  Event type 1 (EV_KEY)
```

```
Event code 4 (KEY_3)
Event code 5 (KEY_4)
Event code 6 (KEY_5)
Event code 7 (KEY_6)
Properties:
Testing ... (interrupt to exit)
Event: time 946835461.742809, type 1 (EV_KEY), code 4 (KEY_3), value 1 ❶
Event: time 946835461.742830, ----- SYN_REPORT -----
Event: time 946835461.892802, type 1 (EV_KEY), code 4 (KEY_3), value 0 ❷
Event: time 946835461.892818, ----- SYN_REPORT -----
:
[armadillo ~]#
```

- ❶ SW3 のボタンプッシュイベントが検出された場合の表示。
- ❷ SW3 のボタンリリースイベントが検出された場合の表示。

図 6.34 ユーザースイッチ: イベントの確認



# 7. Linux カーネル仕様

本章では、工場出荷状態の Armadillo-EVA 1500 の Linux カーネルの仕様について説明します。

## 7.1. Device Tree

工場出荷状態では、SPI フラッシュメモリの fdt パーティションに DTB(Device Tree Blob) が書き込まれています。DTB は DTS(Device Tree Source) を DTC(Device Tree Compiler) でバイナリ形式に変換したファイルです。

DTS は、Linux カーネルソースアーカイブ(linux-3.4-ae1500-[VERSION].tar.gz)に含まれる arch/arm/boot/dts/r8a7791-armadilloeva1500.dts です。

工場出荷状態の eMMC に書き込まれている Linux カーネルイメージは、Device Tree を使用したマシンのセットアップのみをサポートしています。従来のマシンタイプ ID および ATAG を使用したマシンのセットアップを行うことはできませんので、オリジナルのブートローダーを作成する場合は注意してください。

## 7.2. デフォルトコンフィギュレーション

工場出荷状態の eMMC に書き込まれている Linux カーネルイメージは、デフォルトコンフィギュレーションを適用してビルドされています。Armadillo-EVA 1500 用のデフォルトコンフィギュレーションが記載されているファイルは、Linux カーネルソースアーカイブ(linux-3.4-ae1500-[VERSION].tar.gz)に含まれる arch/arm/configs/armadilloeva1500\_defconfig です。

armadilloeva1500\_defconfig で有効になっている主要な設定を「表 7.1. Linux カーネル主要設定」に示します。

表 7.1 Linux カーネル主要設定

コンフィグ	説明
NO_HZ	Tickless System (Dynamic Ticks)
HIGH_RES_TIMERS	High Resolution Timer Support
SMP	Symmetric Multi-Processing
PREEMPT	Preemptible Kernel (Low-Latency Desktop)
AEABI	Use the ARM EABI to compile the kernel
VFP	VFP-format floating point maths
NEON	Advanced SIMD (NEON) Extension support
BINFMT_ELF	Kernel support for ELF binaries

## 7.3. デフォルト起動オプション

工場出荷状態の Armadillo-EVA 1500 の Linux カーネルの起動オプションについて説明します。デフォルト状態では、次のように設定されています。

表 7.2 Linux カーネルのデフォルト起動オプション

起動オプション	説明
console=ttySC3,115200	起動ログなどが出力されるイニシャルコンソールに ttySC3(CON14)を、ポーレートに 115200bps を指定します。
mem=768M	Linux カーネルが利用可能なメモリを 768MByte に制限します <sup>[a]</sup> 。

起動オプション	説明
noinitrd	initrd(initial ramdisk)を使用しません。
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。
root=/dev/mmcblk0p1	ルートファイルシステムに eMMC の第 1 パーティションを指定します。

<sup>[a]</sup>メモリマップについては「表 3.5. Linux 起動時の DDR3-SDRAM メモリマップ」を参照してください。

## 7.4. Linux ドライバ一覧

Armadillo-EVA 1500 で利用することができるデバイスドライバについて説明します。各ドライバで利用しているソースコードの内主要なファイルのパスや、コンフィギュレーションに必要な情報、およびデバイスファイルなどについて記載します。

### 7.4.1. Armadillo-EVA 1500

Armadillo-EVA 1500 の初期化手順やハードウェアの構成情報などが定義されています。新規にデバイスを追加する場合などに変更を加えます。

#### 関連するソースコード

```
arch/arm/mach-shmobile/board-armadilloeva1500.c
arch/arm/mach-shmobile/setup-r8a7791.c
arch/arm/mach-shmobile/setup-r-car-gen2.c
arch/arm/mach-shmobile/smp-r8a7791.c
arch/arm/mach-shmobile/platsmp-apmu.c
arch/arm/mach-shmobile/platsmp-rst.c
arch/arm/mach-shmobile/pm-r8a7791.c
arch/arm/mach-shmobile/clock-r8a7791.c
```

#### カーネルコンフィギュレーション

```
System Type --->
  *** SH-Mobile System Type ***
[*] R-Car M2 (R8A77910) <CONFIG_ARCH_R8A7791>
  *** SH-Mobile Board Type ***
[*] Armadillo-EVA 1500 board <CONFIG_MACH_ARMADILLOEVA1500>
```

### 7.4.2. ピンコントロール

Armadillo-EVA 1500 のプロセッサである RZ/G1M のピンマルチプレクステーブルなどが定義されています。

#### 関連するソースコード

```
drivers/pinctrl/sh-pfc/pfc-r8a7791.c
drivers/pinctrl/sh-pfc/sh-pfc.c
drivers/pinctrl/sh-pfc/core.c
drivers/pinctrl/sh-pfc/gpio.c
drivers/pinctrl/sh-pfc/pinctrl.c
```

## カーネルコンフィギュレーション

```

Device Drivers --->
  Pin controllers --->
    *- SuperH PFC GPIO support                                <CONFIG_GPIO_SH_PFC>
    
```

### 7.4.3. タイマー

Armadillo-EVA 1500 では、カーネル内部のクロックソースに RZ/G1M の CMT0(Compare Match Timer 0) を利用しています。

#### 関連するソースコード

drivers/clocksource/sh\_cmt.c

#### カーネルコンフィギュレーション

```

System Type --->
  *** SH-Mobile System Configuration ***
  Timer and clock configuration --->
    [*] CMT timer driver                                <CONFIG_SH_TIMER_CMT>
    
```

### 7.4.4. SPI フラッシュメモリ

Armadillo-EVA 1500 では、SPI フラッシュメモリを制御するソフトウェアとして MTD(Memory Technology Device) を利用しています。MTD のキャラクタデバイスまたはブロックデバイスを經由して、ユーザーランドからアクセスすることができます。

#### 関連するソースコード

drivers/mtd/mtdcore.c  
drivers/mtd/mtdchar.c  
drivers/mtd/mtdblock.c  
drivers/mtd/devices/m25p80.c

#### デバイスファイル

デバイスファイル	デバイスタイプ	対応するパーティション名
/dev/mtd0	キャラクタ	ipl
/dev/mtdblock0	ブロック	
/dev/mtd1	キャラクタ	bootloader
/dev/mtdblock1	ブロック	
/dev/mtd2	キャラクタ	fdt
/dev/mtdblock2	ブロック	
/dev/mtd3	キャラクタ	firmware
/dev/mtdblock3	ブロック	

## カーネルコンフィギュレーション

```

Device Drivers --->
<*> Memory Technology Device (MTD) support --->                                <CONFIG_MTD>
  [*] Command line partition table parsing                                     <CONFIG_MTD_CMDLINE_PARTS>
  [*] Read-only switching user interface support                             <CONFIG_MTD_RO_IF>
  *** User Modules And Translation Layers ***
<*> Direct char device access to MTD devices                                <CONFIG_MTD_CHAR>
-- Common interface to block layer for MTD 'trnslation...                 <CONFIG_MTD_BLKDEVS>
<*> Caching block device access to MTD devices                             <CONFIG_MTD_BLOCK>
  Self-contained MTD device drivers --->
  <*> Support most SPI Flash chips (AT26DF, M25P, W25X, ...) <CONFIG_MTD_M25P80>
    
```

### 7.4.5. UART

Armadillo-EVA 1500 のシリアルは、RZ/G1M の SCIFB (Serial Communications Interface with FIFO B) を利用しています。

Armadillo-EVA 1500 の標準状態で利用可能なポートは、SCIFB1(CON14) および SCIFB0(CON16) です。

#### フォーマット

- データビット長: 7 or 8 ビット
- ストップビット長: 1 or 2 ビット
- パリティ: 偶数 or 奇数 or なし
- フロー制御: CTS/RTS or XON/XOFF or なし
- 最大ボーレート: 1Mbps

#### 関連するソースコード

drivers/tty/serial/serial\_core.c  
 drivers/tty/serial/sh-sci.c

#### デバイスファイル

シリアルインターフェース	デバイスファイル
SCIFB1	/dev/ttySC3
SCIFB0	/dev/ttySC2

## カーネルコンフィギュレーション

```

System Type --->
  *** SH-Mobile Board Type ***
  [*] USE SCIFB-0                                     <CONFIG_SERIAL_SH_SCI_USE_SCIFB0>
  [*] USE SCIFB-1                                     <CONFIG_SERIAL_SH_SCI_USE_SCIFB1>
Device Drivers --->
  Character devices --->
  Serial drivers --->
  *** Non-8250 serial port support ***
  <*> SuperH SCI(F) serial port support                                <CONFIG_SERIAL_SH_SCI>
  (18) Maximum number of SCI(F) serial ports                         <SERIAL_SH_SCI_NR_UARTS>
  [*] Support for console on SuperH SCI(F)                           <SERIAL_SH_SCI_CONSOLE>
  [ ] DMA support                                                    <SERIAL_SH_SCI_DMA>
    
```

## 7.4.6. Ethernet

Armadillo-EVA 1500 の Ethernet(LAN)は、RZ/G1M の Ether(Ethernet MAC Controller)を利用しています。

### 機能

通信速度: 100Mbps(100BASE-TX), 10Mbps(10BASE-T)  
 通信モード: Full-Duplex(全二重), Half-Duplex(半二重)  
 Auto Negotiation サポート  
 Auto MDIX サポート

### 関連するソースコード

drivers/net/ethernet/renesas/sh\_eth.c

### ネットワークデバイス

eth0

### カーネルコンフィギュレーション

```

Device Drivers --->
[*] Network device support --->                                <CONFIG_NETDEVICES>
  -- Network core driver support                                <CONFIG_NET_CORE>
  -- Generic Media Independent Interface device support        <CONFIG_MII>
[*] Ethernet driver support --->                                <CONFIG_ETHERNET>
  <*> Renesas SuperH Ethernet support                          <CONFIG_SH_ETH>
  -- PHY Device support and infrastructure --->                <CONFIG_PHYLIB>
  <*> Drivers for SMSC PHYs                                    <CONFIG_SMSC_PHY>
  -- Support for bitbanged MDIO buses                          <CONFIG_MDIO_BITBANG>

```

## 7.4.7. MMC ホスト

Armadillo-EVA 1500 の MMC ホストは、RZ/G1M の MMC(Multi Media Card Interface) を利用しています。Armadillo-EVA 1500 には、SAMSUNG 製 eMMC(KLM8G1GEAC)が搭載されています。

### 機能

バス幅: 8bit  
 スピードモード: High Speed モード対応  
 活線挿抜非対応

### デバイスファイル

/dev/mmcblk0

### 関連するソースコード

drivers/mmc/host/sh\_mmcif.c  
 drivers/mmc/card/block.c  
 drivers/mmc/card/queue.c

## カーネルコンフィギュレーション

```

Device Drivers --->
<*> MMC/SD/SDIO card support ---> <CONFIG_MMC>
    *** MMC/SD/SDIO Card Drivers ***
    <*> MMC block device driver <CONFIG_MMC_BLOCK>
    (8) Number of minors per block device <CONFIG_MMC_BLOCK_MINORS>
    [*] Use bounce buffer for simple hosts <CONFIG_MMC_BLOCK_BOUNCE>
    *** MMC/SD/SDIO Host Controller Drivers ***
    <*> SuperH Internal MMCIF support <CONFIG_MMC_SH_MMCIF>
    
```

### 7.4.8. SD ホスト

Armadillo-EVA 1500 の SD ホストは、RZ/G1M の SDHI(SD card Host Interface)を利用しています。

Armadillo-EVA 1500 の標準状態で利用可能なポートは、SDHI0(CON6) および SDHI2(CON4 または CON5) です。

#### 機能

- カードタイプ: SD / SDHC / SDXC / SDIO
- バス幅: 1bit or 4bit
- スピードモード: Default Speed, High Speed, Ultra High Speed Phase I(UHS-I)<sup>[1]</sup>
- カードディテクトサポート
- ライトプロテクトサポート

#### デバイスファイル

メモ리카ードの場合は、カードを認識した順番で/dev/mmcblkN (N は'1'または'2')となります。I/O カードの場合は、ファンクションに応じたデバイスファイルとなります。

#### 関連するソースコード

- drivers/mmc/host/sh\_mobile\_sdhi.c
- drivers/mmc/host/tmio\_mmc\_dma.c
- drivers/mmc/host/tmio\_mmc\_pio.c

## カーネルコンフィギュレーション

```

Device Drivers --->
<*> MMC/SD/SDIO card support ---> <CONFIG_MMC>
    *** MMC/SD/SDIO Host Controller Drivers ***
    <*> SH-Mobile SDHI SD/SDIO controller support <CONFIG_MMC_SDHI>
    
```

### 7.4.9. USB 3.0 ホスト

Armadillo-EVA 1500 の USB 3.0 ホストは、RZ/G1M の USB3.0 Host Controller を利用しています。

<sup>[1]</sup>SDHI0 は SDR50 および SDR104(最大 SD クロック: 156 MHz) に対応。SDHI2 は SDR50(最大 SD クロック: 97.5 MHz) に対応。

Armadillo-EVA 1500 の標準状態では、USB 3.0 Type-A コネクタ(CON10) を USB 3.0 ホストとして利用することができます。カーネルをコンフィギュレーションし、USB 3.0 ホストドライバを無効化することで USB 2.0 ホストとして利用することもできます。

### 機能

Universal Serial Bus 3.0 Specification Revision 1.0 準拠  
 転送レート: USB 3.0 Super Speed(5Gbps), High-Speed(480Mbps), Full-Speed(12Mbps), Low-Speed(1.5Mbps)

### デバイスファイル

メモリデバイスの場合は、デバイスを認識した順番で/dev/sdN (N は'a'からの連番)となります。I/O デバイスの場合は、ファンクションに応じたデバイスファイルとなります。

### 関連するソースコード

```
drivers/usb/host/xhci-rcar.c
drivers/usb/host/xhci.c
drivers/usb/host/xhci-mem.c
drivers/usb/host/xhci-ring.c
drivers/usb/host/xhci-hub.c
drivers/usb/host/xhci-dbg.c
drivers/usb/host/xhci-pci.c
```

### ファームウェア

firmware/UU3DRD1FW\_2005L.dlmem

### カーネルコンフィギュレーション

```
Device Drivers --->
Generic Driver Options --->
  <*> Userspace firmware loading support <CONFIG_FW_LOADER>
  [*] Include in-kernel firmware blobs in kernel binary <CONFIG_FIRMWARE_IN_KERNEL>
  (UU3DRD1FW_2005L.dlmem) External firmware blobs to build into the kernel binary
  <CONFIG_EXTRA_FIRMWARE>
  (firmware) Firmware blobs root directory <CONFIG_EXTRA_FIRMWARE_DIR>
  [*] USB support ---> <CONFIG_USB_SUPPORT>
  <*> Support for Host-side USB <CONFIG_USB>
  *** Miscellaneous USB options ***
  [*] USB device filesystem <CONFIG_USB_DEVICEFS>
  [*] USB device class-devices <CONFIG_USB_DEVICE_CLASS>
  *** USB Host Controller Drivers ***
  <*> xHCI HCD (USB 3.0) support (EXPERIMENTAL) <CONFIG_USB_XHCI_HCD>
```

## 7.4.10. USB 2.0 ホスト

Armadillo-EVA 1500 の USB 2.0 ホストは、RZ/G1M の USB2.0 Host (EHCI/OHCI) を利用しています。

Armadillo-EVA 1500 の標準状態では、USB Type-A コネクタ(CON8) を USB 2.0 ホストとして利用することができます。カーネルをコンフィギュレーションし、USB 2.0 ファンクションドライバを有効化することで USB Type-B コネクタ(CON9)を USB 2.0 ファンクションとして利用することもできます。

## 機能

Universal Serial Bus Specification Revision 2.0 準拠  
 Open Host Controller Interface (OHCI) Specification for USB Rev 1.0a 準拠  
 Enhanced Host Controller Interface (EHCI) Specification for USB Rev 1.0 準拠  
 転送レート: USB 2.0 High-Speed(480Mbps), Full-Speed(12Mbps), Low-Speed(1.5Mbps)

## デバイスファイル

メモリデバイスの場合は、デバイスを認識した順番で/dev/sdN (N は'a'からの連番)となります。  
 I/O デバイスの場合は、ファンクションに応じたデバイスファイルとなります。

## 関連するソースコード

drivers/usb/host/ehci-rcar.c  
 drivers/usb/host/ohci-rcar.c  
 drivers/usb/host/ehci-hcd.c  
 drivers/usb/host/ohci-hcd.c

## カーネルコンフィギュレーション

```

Device Drivers --->
[*] USB support --->                                <CONFIG_USB_SUPPORT>
    <*> Support for Host-side USB                      <CONFIG_USB>
        *** Miscellaneous USB options ***
    [*] USB device filesystem                          <CONFIG_USB_DEVICEFS>
    [*] USB device class-devices                      <CONFIG_USB_DEVICE_CLASS>
        *** USB Host Controller Drivers ***
    <*> EHCI HCD (USB 2.0) support                     <CONFIG_USB_EHCI_HCD>
    [*] Root Hub Transaction Translators              <CONFIG_USB_EHCI_ROOT_HUB_TT>
    [*] Improved Transaction Translator scheduling    <CONFIG_USB_EHCI_TT_NEWSCHED>
    <*> OHCI HCD support                              <CONFIG_USB_OHCI_HCD>
    
```

## 7.4.11. USB ファンクション

Armadillo-EVA 1500 の USB ファンクションは、RZ/G1M の HS-USB(USB High-Speed Module) および USB-DMAC(USB High-Speed DMAC)を利用しています。

Armadillo-EVA 1500 の標準状態では、USB ファンクションを利用することができません。USB ファンクションを利用するには、カーネルをコンフィギュレーションし、USB 2.0 ファンクションドライバの有効化<sup>[2]</sup>およびガジェットドライバの選択をおこないます。

Armadillo-EVA 1500 とホスト機器を接続する場合は、USB Type-B コネクタ(CON9) に USB ケーブルを接続して利用します。

Linux カーネルに用意されている代表的なガジェットドライバを次に示します。

表 7.3 代表的なガジェットドライバ

ガジェット	機能
Ethernet Gadget	接続されるとホスト機器間でネットワーク通信させることができます
Serial Gadget	接続されるとホスト機器間でシリアル通信させることができます

<sup>[2]</sup>USB 2.0 ファンクションドライバを有効化すると、USB 2.0 ホストは利用できなくなくなります。



### 関連するソースコード

drivers/usb/gadget/r8a66597-udc.c

### カーネルコンフィギュレーション

```

Device Drivers --->
[*] USB support ---> <CONFIG_USB_SUPPORT>
    *** USB Miscellaneous drivers ***
    <*> USB Gadget Support ---> <CONFIG_USB_GADGET>
    <*> USB Peripheral Controller (Renesas R8A66597 USB Peripheral Controller) --->
        (X) Renesas R8A66597 USB Peripheral Controller <CONFIG_USB_R8A66597>
        [*] on-chip USBHS type with extended bulk endpoints support
            <CONFIG_USB_R8A66597_TYPE_BULK_PIPES_12>
    <*> USB Gadget Drivers (Ethernet Gadget (with CDC Ethernet support)
        (X) Ethernet Gadget (with CDC Ethernet support) <CONFIG_USB_ETH>
        ( ) Serial Gadget (with CDC ACM and CDC OBEX support) <CONFIG_USB_G_SERIAL>
        :
    *** OTG and related infrastructure ***
    <*> GPIO based peripheral-only VBUS sensing 'transceiver' <CONFIG_USB_GPIO_VBUS>
    
```

## 7.4.12. デジタル HD 出力

Armadillo-EVA 1500 のデジタル HD 出力は、RZ/G1M の DU1(Display Unit 1) を利用しています。ディスプレイへの画像出力をサポートしています。HDMI トランスミッターには、Analog Devices 製 ADV7511W が採用されています。ADV7511W は I2C-GPIO1(I2C ノード: 8-0039) に接続されています。音声出力および CEC(Consumer Electronics Control)は非サポートです。

Linux では、フレームバッファおよび DRM(Direct Rendering Manager) アクセスをサポートしています。

### 機能

最大解像度<sup>[3]</sup>: 1920 x 1080 ピクセル  
 カラーフォーマット<sup>[4]</sup>: RGB565, ARGB1555, XRGB1555, XRGB8888, ARGB8888, YUYV, UYUV, NV12, NV16, NV21  
 ダブルバッファ, トリプルバッファサポート

### デバイスファイル

フレームバッファデバイス: /dev/fb0  
 DRM デバイス: /dev/dri/card0

### 関連するソースコード

drivers/gpu/drm/rcar-du/  
 drivers/gpu/drm/i2c/adv7511\_core.c

<sup>[3]</sup>ドットクロックが 100MHz を超えるビデオモードには対応できません  
<sup>[4]</sup>YUV フォーマットは DRM アクセスのみ対応しています

## カーネルコンフィギュレーション

```

Device Drivers --->
  Graphics support --->
    <*> Direct Rendering Manager (XFree86 4.1.0 and higher DRI support) --->
      <CONFIG_DRM>
    <*> ADV7511 encoder <CONFIG_DRM_ADV7511>
    [ ] ADV7511 DVI output mode <CONFIG_DRM_ADV7511_DVI_OUTPUT>
    [*] Enable FBDev CRTC select <CONFIG_DRM_FBDEV_CRTC>
    (1) FBDev CRTC number <CONFIG_DRM_FBDEV_CRTC_NUM>
    <*> DRM Support for R-Car Display Unit <CONFIG_DRM_RCAR_DU>
    [ ] R-Car DU LCD Support <CONFIG_DRM_RCAR_LCD>
    
```

### 7.4.13. コンポジットビデオ入力

Armadillo-EVA 1500 のコンポジットビデオ入力は、RZ/G1M の VIN2(Video Input Module 2) を利用しています。ビデオデコーダには Analog Devices 製 ADV7180 が採用されています。ADV7180 は、I2C-GPIO1(I2C ノード: 8-0020) に接続されています。

Linux では、V4L2(Video for Linux Two) アクセスをサポートしています。

#### 機能

インターフェース: ITU-R BT.656  
 VIN 入力フォーマット: UYVY  
 VIN 出力フォーマット: YUYV

#### デバイスファイル

/dev/video0<sup>[5]</sup>

#### 関連するソースコード

drivers/media/video/vin.c  
 drivers/media/video/adv7180.c  
 drivers/media/video/v4l2-dev.c  
 drivers/media/video/v4l2-ioctl.c  
 drivers/media/video/v4l2-device.c  
 drivers/media/video/v4l2-fh.c  
 drivers/media/video/v4l2-event.c  
 drivers/media/video/v4l2-ctrls.c  
 drivers/media/video/v4l2-subdev.c  
 drivers/media/video/v4l2-int-device.c

<sup>[5]</sup>USB カメラなどを接続して V4L2 デバイスを追加している場合は、番号が異なる可能性があります

## カーネルコンフィギュレーション

```

Device Drivers --->
  <*> Multimedia support --->                                <CONFIG_MEDIA_SUPPORT>
    *** Multimedia core support ***
  <*> Video For Linux                                        <CONFIG_VIDEO_DEV>
    *** Multimedia drivers ***
  [*] Video capture adapters --->                            <CONFIG_VIDEO_CAPTURE_DRIVERS>
    Encoders, decoders, sensors and other helper chips --->
      *** Video decoders ***
    <*> Analog Devices ADV7180 decoder                        <CONFIG_VIDEO_ADV7180>
  [*] V4L platform devices --->                              <CONFIG_V4L_PLATFORM_DRIVERS>
    <*> SoC camera support                                    <CONFIG_SOC_CAMERA>
    <*> VIN Unit Interface driver                             <CONFIG_VIDEO_VIN>
    
```

### 7.4.14. アナログオーディオ

Armadillo-EVA 1500 のアナログオーディオは、RZ/G1M の SSIU(Serial Sound Interface Unit)、ADG(Audio Clock Generator) および SCU(Sampling Rate Converter Unit)を利用しています。オーディオコーデックには、Cirrus Logic 製 CS42L52 が採用されています。CS42L52 は I2C-GPIO1 (I2C ノード: 8-004a) に接続されています。

#### 機能

- サンプリング周波数: 8000, 16000, 32000, 48000, 96000
- チャンネル数: 1 or 2
- フォーマット: Signed/Unsigned 16/18/20/24 bit, Little-endian
- 再生(Playback), 録音(Capture)サポート
- ミキサーサポート

#### オーディオデバイス

hw:0

#### 関連するソースコード

- sound/soc/sh/armadilloeva1500.c
- sound/soc/sh/scu\_pcm.c
- sound/soc/sh/scu\_dai.c
- sound/soc/codecs/cs42l52.c

## カーネルコンフィギュレーション

```

Device Drivers --->
  <*> Sound card support --->                                <CONFIG_SOUND>
    <*> Advanced Linux Sound Architecture --->                <CONFIG_SND>
      [*] Support old ALSA API                                <CONFIG_SND_SUPPORT_OLD_API>
    <*> ALSA for SoC audio support --->                        <CONFIG_SND_SOC>
      SoC Audio support for SuperH --->
    <*> SCU sound support on Armadillo-EVA 1500                <CONFIG_SND_SCU_ARMADILLOEVA1500>
    *- Armadillo-EVA 1500 CS42L52 sound support                <CONFIG_SND_ARMADILLOEVA1500_CS42L52>
    
```

## 7.4.15. マルチメディアパッケージ

Armadillo-EVA 1500 のマルチメディア機能(H.264/AVC 動画デコード)は、RZ/G1M の次に示す機能を利用しています。

- ・ VCP3
- ・ VSP1
- ・ VPC(Video Processing Unit Cache)
- ・ S3CTRL(S3 Cache Controller)
- ・ FDP1(Fine Display Processor)

マルチメディアパッケージドライバは、次に示す Linux カーネルが管理しないメモリを使用します。

物理アドレス	サイズ
0x78000000   0x7FFFFFFF	128MByte

### 機能

最大ビットレート	40M bit/sec
最大フレームレート	60p, 60i
入力フォーマット	H.264 Elementary Stream(Annex B Byte stream format 準拠)
出力フォーマット	<ul style="list-style-type: none"> <li>・ YUV420 Semi Planar</li> <li>・ YUV420 Planar 形式</li> </ul>
サポートプロファイル/レベル	<ul style="list-style-type: none"> <li>・ Baseline Profile Level 3</li> <li>・ Constrained Baseline Profile Level 4.2</li> <li>・ Main Profile Level 4.2</li> <li>・ High Profile Level 4.2</li> </ul>
プロファイル共通非サポートツール	<ul style="list-style-type: none"> <li>・ ASO (Arbitrary Slice Ordering)</li> <li>・ FMO (Flexible Macroblock Ordering)</li> <li>・ RS (Redundant Slices)</li> </ul>
ピクチャ構造	<ul style="list-style-type: none"> <li>・ フレーム構造</li> <li>・ フィールド構造</li> </ul>
画像サイズ	<ul style="list-style-type: none"> <li>・ 80x80~1920x1080 [a] [b]</li> </ul>

[a] プログレッシブシーケンスの場合は、水平 2 画素、垂直 2 ライン単位で設定可能

[b] インターレースシーケンスの場合は、水平 2 画素、垂直 4 ライン単位で設定可能

### デバイスファイル

V4L2 デバイス: /dev/video1~ /dev/video16, /dev/v4l-subdev0~ /dev/v4l-subdev23  
 メディアデバイス: /dev/media0, /dev/media1

### 関連するソースコード

drivers/media/video/rcar-mmp/  
 drivers/media/video/vsp1/

## カーネルコンフィギュレーション

```

Device Drivers --->
  <*> Multimedia support --->
    [*] Memory-to-memory multimedia devices --->
      <*> Renesas VSP1 Video Processing Engine <CONFIG_VIDEO_RENESAS_VSP1>
    [*] Video capture adapters --->
      Encoders, decoders, sensors and other helper chips --->
        *** Video and audio decoders ***
        <*> MMP Support for R-Car Display Unit <CONFIG_VIDEO_RCAR_MMP>
        <*> FDP Manager support for R8A7791 <VIDEO_RCAR_MMP_FDPM>
        <*> VSP Manager support for R8A7791 <VIDEO_RCAR_MMP_VSPM>
        <*> UVCS support for R8A7791 <VIDEO_RCAR_MMP_UVCS>
        <*> S3CTL support for R8A7791 <VIDEO_RCAR_MMP_S3CTL>
        <*> MMNGR support for R8A7791 <VIDEO_RCAR_MMP_MMNGR>
    
```

### 7.4.16. SATA

Armadillo-EVA 1500 の SATA は、RZ/G1M の Serial-ATA を利用しています。

#### 機能

Serial ATA Revision 3.1 準拠  
 転送レート: 3.0Gbps(Gen2), 1.5Gbps(Gen1)  
 接続形式: 1 チャンネル, Master  
 NCQ(Native Command Queuing)非サポート  
 活線挿抜非サポート

#### デバイスファイル

HDD や SSD などの場合は、デバイスを認識した順番で/dev/sdN (N は'a'からの連番)となります。  
 CD や DVD などの場合は、デバイスを認識した順番で/dev/srN (N は'O'からの連番)となります。

#### 関連するソースコード

```

drivers/ata/sata_rcar.c
drivers/ata/libata-sff.c
drivers/scsi/sd.c
drivers/scsi/sd_dif.c
drivers/scsi/sr.c
drivers/scsi/sr_ioctl.c
drivers/scsi/sr_vendor.c
    
```

## カーネルコンフィギュレーション

```

Device Drivers --->
  SCSI device support --->
    <*> SCSI disk support <CONFIG_BLK_DEV_SD>
    <*> SCSI CDROM support <CONFIG_BLK_DEV_SR>
  <*> Serial ATA and Parallel ATA drivers ---> <CONFIG_ATA>
    *** Controllers with non-SFF native interface ***
    [*] ATA SFF support <CONFIG_ATA_SFF>
    *** SFF controllers with custom DMA interface ***
    [*] ATA BMDMA support <CONFIG_ATA_BMDMA>
    *** SATA SFF controllers with BMDMA ***
    <*> Renesas R-Car SATA support <CONFIG_SATA_RCAR>
    
```

### 7.4.17. PCI Express

Armadillo-EVA 1500 の PCI Express は、RZ/G1M の PCIEC を利用しています。

#### 機能

PCI Express Base Specification Revision 2.0 (December 20, 2006) サブセットに対応  
 リンク幅: x1  
 転送レート: 2.5GT/s, 5.0GT/s  
 割り込み通知方式: INTx, MSI, MSI-X

#### 関連するソースコード

drivers/pci/host/pci-rcar.c

#### カーネルコンフィギュレーション

```

Bus support --->
  [*] PCI support <CONFIG_PCI>
  [*] Message Signaled Interrupts (MSI and MSI-X) <CONFIG_PCI_MSI>
  PCI host controller drivers --->
    [*] Renesas R-Car PCIe controller <CONFIG_PCI_RCAR>
    
```

### 7.4.18. 温度センサー

Armadillo-EVA 1500 の温度センサー は、RZ/G1M の THS/TSC(Thermal Sensor) を利用しています。

温度センサーの測定温度が 80°C 以上になった場合、警告 LED(LED9) が点灯します。80°C 未満になった場合は警告 LED が消灯します。また、90°C 以上になった場合は Linux カーネルが /sbin/poweroff コマンドを実行し、システムを停止します。

#### 機能

分解能: 5°C  
 測定範囲: -40°C ~ 125°C

### sysfs Thermal クラスディレクトリ

/sys/class/thermal/thermal\_zone0

### 関連するソースコード

drivers/thermal/rcar\_thermal.c  
drivers/thermal/thermal\_sys.c

### カーネルコンフィギュレーション

```

Device Drivers --->
  <*> Generic Thermal sysfs driver --->                                <CONFIG_THERMAL>
      --- Generic Thermal sysfs driver
  <*> Renesas R-Car thermal driver                                     <CONFIG_RCAR_THERMAL>
    
```

## 7.4.19. リアルタイムクロック

Armadillo-EVA 1500 には、セイコーインスツル(SII)製 S-35390A が搭載されています。S-35390A は、I2C-GPIO0 (I2C ノード: 7-0030) に接続されています。

### 機能

アラーム割り込みサポート

### デバイスファイル

/dev/rtc0

### 関連するソースコード

drivers/rtc/rtc-s35390a.c  
drivers/rtc/class.c  
drivers/rtc/rtc-dev.c  
drivers/rtc/rtc-sysfs.c

### カーネルコンフィギュレーション

```

Device Drivers --->
[*] Real Time Clock --->                                            <CONFIG_RTC_CLASS>
  [*] Set system time from RTC on startup and resume                <CONFIG_RTC_HCTOSYS>
  (rtc0) RTC used to set the system time                             <CONFIG_RTC_HCTOSYS_DEVICE>
      *** RTC interfaces ***
  [*] /sys/class/rtc/rtcN (sysfs)                                    <CONFIG_RTC_INTF_SYSFS>
  [*] /proc/driver/rtc (procfs for rtc0)                            <CONFIG_RTC_INTF_PROC>
  [*] /dev/rtcN (character devices)                                 <CONFIG_RTC_INTF_DEV>
      *** I2C RTC drivers ***
  <*> Seiko Instruments S-35390A                                     <CONFIG_RTC_DRV_S35390A>
    
```

## 7.4.20. LED

Armadillo-EVA 1500 に搭載されている制御可能な LED は、GPIO が接続されています。Linux では、GPIO 接続用 LED ドライバ(leds-gpio)で制御することができます。

Armadillo-EVA 1500 には、LED4～LED7 が実装されています。

### sysfs LED クラスディレクトリ

```
/sys/class/leds/LED4
/sys/class/leds/LED5
/sys/class/leds/LED6
/sys/class/leds/LED7
```

### 関連するソースコード

```
drivers/leds/leds-gpio.c
drivers/leds/led-class.c
drivers/leds/led-triggers.c
drivers/leds/ledtrig-timer.c
```

### カーネルコンフィギュレーション

```
Device Drivers --->
[*] LED Support --->
    <*> LED Class Support <CONFIG_NEW_LEDS>
    *** LED drivers *** <CONFIG_LEDS_CLASS>
    <*> LED Support for GPIO connected LEDs <CONFIG_LEDS_GPIO>
    [*] LED Trigger support <CONFIG_LEDS_TRIGGERS>
    *** LED Triggers ***
    <*> LED Timer Trigger <CONFIG_LEDS_TRIGGER_TIMER>
```

## 7.4.21. ユーザースイッチ

Armadillo-EVA 1500 に搭載されているユーザースイッチは、GPIO が接続されています。Linux では、GPIO 接続用キーボードドライバ(gpio\_keys)で制御されます。

ユーザースイッチにはそれぞれにキーコードが割り当てられています。

表 7.4 キーコード

ユーザースイッチ	キーコード	イベントコード
SW3	KEY_3	4
SW4	KEY_4	5
SW5	KEY_5	6
SW6	KEY_6	7

### デバイスファイル

```
/dev/input/event2[6]
```

### 関連するソースコード

```
drivers/input/keyboard/gpio_keys.c
drivers/input/input.c
drivers/input/evdev.c
```

<sup>[6]</sup>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、番号が異なる可能性があります



## カーネルコンフィギュレーション

```

Device Drivers --->
  Input device support --->
    *- Generic input layer (needed for keyboard, mouse, ...) <CONFIG_INPUT>
    <*> Event interface <CONFIG_INPUT_EVDEV>
        *** Input Device Drivers ***
    [*] Keyboards ---> <CONFIG_INPUT_KEYBOARD>
        <*> GPIO Buttons <CONFIG_KEYBOARD_GPIO>
    
```

### 7.4.22. I2C

Armadillo-EVA 1500 の I2C インターフェースは、GPIO を利用した I2C バスドライバ(i2c-gpio)を利用しています。

Armadillo-EVA 1500 で利用している I2C バスと、接続される I2C デバイスを次に示します。

表 7.5 I2C デバイス

I2C バス	I2C デバイス	
	アドレス	デバイス名
I2C-GPIO0	0x30 (0x31~0x37 も予約済み), 0x5a	S-35390A リアルタイムクロック, DA9063 パワーマネジメント IC
I2C-GPIO1	0x4a	CS42L52 オーディオコーデック
I2C-GPIO2	0x55	ST1232 タッチスクリーンコントローラ <sup>[a]</sup>

<sup>[a]</sup>LCD インターフェース(CON23)に Armadillo-EVA 1500 LCD 拡張ボードを接続した場合

Armadillo-EVA 1500 では標準状態では、CONFIG\_I2C\_CHARDEV が有効となっているためユーザードライバで I2C デバイスを制御することができます。ユーザードライバを利用する場合は、Linux カーネルで I2C デバイスに対応するデバイスドライバを無効にする必要があります。

### デバイスファイル

- /dev/i2c-7 (I2C-GPIO0)
- /dev/i2c-8 (I2C-GPIO1)
- /dev/i2c-9 (I2C-GPIO2)

### 関連するソースコード

- drivers/i2c/busses/i2c-gpio.c
- drivers/i2c/i2c-core.c
- drivers/i2c/i2c-dev.c

## カーネルコンフィギュレーション

```

Device Drivers --->
<*> I2C support ---> <CONFIG_I2C>
  <*> I2C device interface <CONFIG_I2C_CHARDEV>
  [*] Autoselect pertinent helper modules <CONFIG_I2C_HELPER_AUTO>
      I2C Hardware Bus support --->
        <*> GPIO-based bitbanging I2C <CONFIG_I2C_GPIO>
    
```

## 7.4.23. LCD

Armadillo-EVA 1500 の LCD インターフェースは、RZ/G1M の DU1 (Display Unit 1) を利用しています。Linux では、フレームバッファおよび DRM (Direct Rendering Manager) アクセスをサポートしています。

Armadillo-EVA 1500 LCD 拡張ボードでは、バックライトの輝度調整および、タッチスクリーンに対応しています。Linux では、それぞれ PWM 制御用バックライトドライバ (pwm\_bl)、ST1232 タッチスクリーンドライバを利用しています。

Armadillo-EVA 1500 の標準状態では、LCD 利用することができません。LCD を利用するには、カーネルをコンフィギュレーションし、LCD の有効化<sup>[7]</sup>をおこないます。

### 機能

対応可能 LCD: TFT カラー液晶 (最大解像度<sup>[8]</sup>: 1920 x 1080 ピクセル)  
カラーフォーマット<sup>[9]</sup>: RGB565, ARGB1555, XRGB1555, XRGB8888, ARGB8888, YUYV, UYUV, NV12, NV16, NV21  
ダブルバッファ, トリプルバッファサポート  
256 段階バックライト輝度調整  
マルチタッチサポート (ST1232)

### デバイスファイル

フレームバッファデバイス: /dev/fb0  
DRM デバイス: /dev/dri/card0  
タッチスクリーンデバイス: /dev/input/event0<sup>[10]</sup>  
バックライト輝度調整用 sysfs ファイル: /sys/class/backlight/pwm-backlight/brightness

### 関連するソースコード

drivers/gpu/drm/rcar-du/  
drivers/video/backlight/backlight.c  
drivers/video/backlight/pwm\_bl.c  
drivers/misc/rcar-pwm.c  
drivers/input/touchscreen/st1232.c

<sup>[7]</sup>LCD を有効化すると、デジタル HD 出力は利用できなくなります。

<sup>[8]</sup>ドットクロックが 100MHz を超えるビデオモードには対応できません

<sup>[9]</sup>YUV フォーマットは DRM アクセスのみ対応しています

<sup>[10]</sup>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、番号が異なる可能性があります

## カーネルコンフィギュレーション

```

Device Drivers --->
  Misc devices --->
    <*> R-Car PWM driver <CONFIG_RCAR_PWM>
  Input device support --->
    -* Generic input layer (needed for keyboard, mouse, ...) <CONFIG_INPUT>
    <*> Event interface <CONFIG_INPUT_EVDEV>
      *** Input Device Drivers ***
    [*] Touchscreens ---> <CONFIG_INPUT_TOUCHSCREEN>
      <*> Sitronix ST1232 touchscreen controllers <CONFIG_TOUCHSCREEN_ST1232>
      [ ] Enable single touch event for ST1232 <CONFIG_TOUCHSCREEN_ST1232_SINGLETOUCH>
  Graphics support --->
    <*> Direct Rendering Manager (XFree86 4.1.0 and higher DRI support) ---> <CONFIG_DRM>
    [*] Enable FBDev CRTC select <CONFIG_DRM_FBDEV_CRTC>
      (1) FBDev CRTC number <CONFIG_DRM_FBDEV_CRTC_NUM>
    <*> DRM Support for R-Car Display Unit <CONFIG_DRM_RCAR_DU>
    [*] R-Car DU LCD Support <CONFIG_DRM_RCAR_LCD>
      (800) Horizontal screen resolution
      (480) Vertical screen resolution
    [*] Backlight & LCD device support ---> <CONFIG_BACKLIGHT_LCD_SUPPORT>
      <*> Lowlevel Backlight controls <CONFIG_BACKLIGHT_CLASS_DEVICE>
      <*> Generic PWM based Backlight Driver <CONFIG_BACKLIGHT_PWM>
  
```

### 7.4.24. LVDS

Armadillo-EVA 1500 の LVDS は、RZ/G1M の DU0(Display Unit 0) を利用しています。

Linux では、フレームバッファおよび DRM(Direct Rendering Manager) アクセスをサポートしています。デフォルトの解像度は、1024 x 768 ピクセルです。

#### 機能

出力信号: TIA/EIA-644 標準(差動ペアがデータ 4 組、クロック 1 組)  
 最大ドットクロック: 148.5 MHz

#### デバイスファイル

フレームバッファデバイス: /dev/fb0  
 DRM デバイス: /dev/dri/card0

#### 関連するソースコード

drivers/gpu/drm/rcar-du/

## カーネルコンフィギュレーション

```
Device Drivers --->
  Graphics support --->
    <*> Direct Rendering Manager (XFree86 4.1.0 and higher DRI support) --->
      <CONFIG_DRM>
    [*] Enable FBDev CRTC select                                <CONFIG_DRM_FBDEV_CRTC>
    (1)  FBDev CRTC number                                    <CONFIG_DRM_FBDEV_CRTC_NUM>
    <*> DRM Support for R-Car Display Unit                      <CONFIG_DRM_RCAR_DU>
    [*]  R-Car DU LVDS Encoder Support                        <CONFIG_DRM_RCAR_LVDS>
```

# 8. ブートローダー仕様

本章では、ブートローダーの起動シーケンスや機能について説明します。

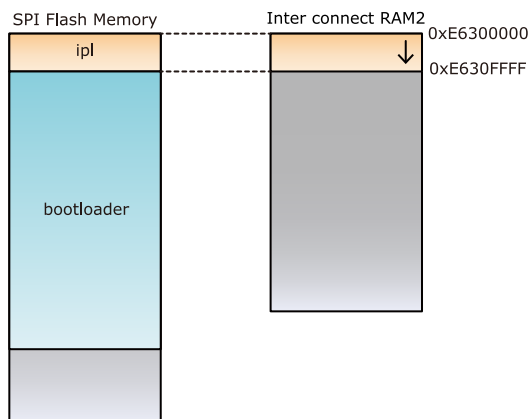
Armadillo-EVA 1500 のブートローダーは、IPL(1st ブートローダー)と Hermit-At(2nd ブートローダー)から構成されます。工場出荷状態では、各ブートローダーイメージは SPI フラッシュメモリに配置されています。

IPL は Hermit-At を起動する機能のみを持ちます。Hermit-At は Linux カーネルの起動や、Linux カーネル起動オプションを設定する機能などを持ちます。

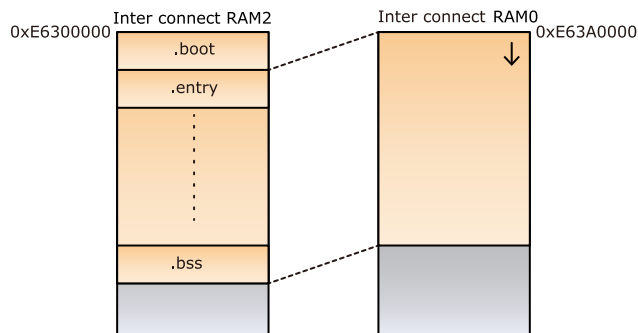
## 8.1. 起動シーケンス

ここでは、工場出荷状態の Armadillo-EVA 1500 で Hermit-At が起動するまでの流れについて説明します。

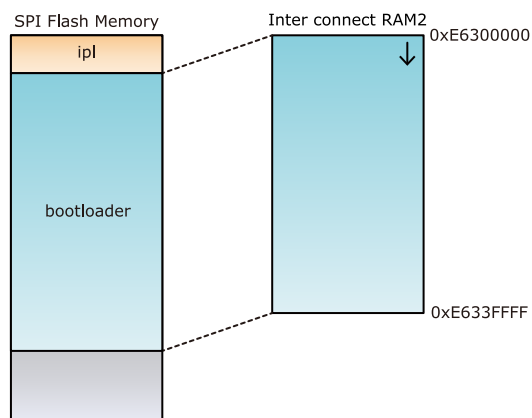
1. Armadillo-EVA 1500 へ電源を投入すると、プロセッサ(RZ/G1M)の機能により SPI フラッシュメモリの ipl パーティション(16kByte)が Inter Connect RAM2 にコピーされ、先頭にジャンプします。



2. スタックポインタなどの最低限の初期化をおこなった後、IPL 自身が Hermit-At の起動処理部を Inter Connect RAM0 にリロケートします。これは、Inter Connect RAM2 全体を bootloader パーティションのコピー領域として開放するためです。



- IPL は SPI フラッシュメモリの bootloader パーティション(256kByte) を Inter Connect RAM2 にコピーして先頭にジャンプします。



SW1.7、SW1.8 を OFF に設定した場合は IPL を起動することができません。これは、プロセッサが SPI フラッシュメモリ から Inter Connect RAM2 にコピーするサイズが 4kByte になるためです。IPL のプログラムサイズは約 12kByte です。

## 8.2. 2nd ブートローダーイメージの選択

IPL が持つ機能によって、2nd ブートローダーイメージを読み出すデバイスを選択することができます。

電源投入時の SW2.7 の設定により SPI フラッシュメモリと SD カードのどちらを使用するかを選択することができます。SW2.7 が ON であれば SPI フラッシュメモリの bootloader パーティションに書き込まれているブートローダーが起動し、OFF であれば SD カードの第 1 パーティションのブートローダー(/sdboot.bin)が起動します。

また SW2.7 が OFF の場合は SW9 の状態により、2 つある SD インターフェースのどちらを使用するかを選択することができます。SW9 を押下した状態で電源を投入した場合は CON4 が、押下しない状態で電源を投入した場合は CON6 が選択されます。

表 8.1 スイッチと 2nd ブートローダーの対応

SW2.7	SW9	2nd ブートローダー
ON	-	SPI フラッシュメモリの bootloader パーティション
OFF	押下する	CON4 に接続した SD カードの第 1 パーティションの/sdboot.bin
	押下しない	CON6 に接続した SD カードの第 1 パーティションの/sdboot.bin

## 8.3. Hermit-At の起動モード

Hermit-At が起動すると SW2.8 の設定により 2 つある起動モードのどちらかに遷移します。

表 8.2 Hermit-At 起動モード

起動モード	SW2.8	説明
保守モード	ON	各種設定が可能な Hermit-At コマンドプロンプトが起動します。
OS 自動起動モード	OFF	電源投入後、自動的に Linux カーネルを起動させます。

## 8.4. Hermit-At の機能

Hermit-At の保守モードでは、Linux カーネルの起動オプションの設定や SPI フラッシュメモリの書き換えなどを行うことができます。

保守モードで利用できるコマンドを、「表 8.3. Hermit-At の保守モードコマンド一覧」に示します。

表 8.3 Hermit-At の保守モードコマンド一覧

コマンド	説明
tftpd erase program download	SPI フラッシュメモリを書き換える場合に使用します
memmap	SPI フラッシュメモリのメモリマップを表示します
setbootdevice setenv clearenv	OS の起動設定を変更する場合に使用します
boot tftpboot	OS を起動する場合に使用します
frob	簡易的にメモリアクセスする場合に使用します
md5sum	SPI フラッシュメモリ空間の MD5 チェックサム値を表示する場合に使用します
info	ハードウェアの情報を表示します
version	ブートローダーのバージョンを表示します

各コマンドのヘルプを表示するには「図 8.1. hermit コマンドのヘルプを表示」のようにします。

```
hermit> help [コマンド]
```

図 8.1 hermit コマンドのヘルプを表示



tftpd および tftpboot は、TFTP プロトコルを使用して TFTP サーバーからイメージファイルをダウンロードします。デフォルトのデータブロックサイズが 512Byte であるため、イメージファイルの最大サイズがブロック番号の桁溢れが発生しない 33554431Byte(32MByte - 1Byte)に制限されます。これよりもサイズの大きいイメージファイルをダウンロードする場合は、"--blksize"オプションを利用してデータブロックサイズを増やす必要があります。

"--blksize"オプションには、IP フラグメンテーションが起きないデータブロックサイズを指定する必要があります。

### 8.4.1. コンソールの指定方法

ブートローダーおよび Linux カーネルのコンソールを指定するには、後述する Linux カーネル起動オプションを設定する場合の setenv コマンドで行います。Linux カーネル起動オプションの console パラメータは、ブートローダーのコンソールにも影響する仕組みとなっています。

コンソール指定子とそれに対応するログ表示先/保守モードプロンプト出力先を「表 8.4. コンソール指定子とログ出力先」に示します。

表 8.4 コンソール指定子とログ出力先

コンソール指定子	OS 自動起動モード時のログ出力先	保守モードプロンプト出力先 <sup>[a]</sup>
ttySC3	CON14	CON14
ttySC2	CON16	CON16
none	なし	CON14
その他(tty1 等)	指定するコンソール <sup>[b]</sup>	CON14

<sup>[a]</sup>ブートローダーの再起動後に反映されます

<sup>[b]</sup>ブートローダーのログは出力されません

### 8.4.2. Linux カーネルイメージの指定方法

Hermit-At が OS を起動させる際、eMMC と SD カードのどちらに保存されている Linux カーネルイメージを使用するかを指定することができます。

Linux カーネルイメージを指定するには、"setbootdevice"コマンドを使用します。「表 8.5. Linux カーネルイメージ指定子」に示す指定子を設定することができます。

表 8.5 Linux カーネルイメージ指定子

指定子	Linux カーネルイメージの配置場所
mmcblk0p1 <sup>[a]</sup>	eMMC のパーティション 1 に保存されている Linux カーネルイメージ
mmcblk1p2 <sup>[a]</sup>	CON6 に接続された SD カードのパーティション 2 に保存されている Linux カーネルイメージ
mmcblk2p2 <sup>[a]</sup>	CON4 に接続された SD カードのパーティション 2 に保存されている Linux カーネルイメージ
tftp	TFTP サーバー上に保存されている Linux カーネルイメージ

<sup>[a]</sup>"pN"(N は'1'から'4')はパーティションを示し、任意のパーティションを指定可能です

Linux カーネルイメージ指定子として"tftp"を指定する場合は、Linux カーネルイメージ指定子の他に TFTP サーバーの IP アドレスなどを指定する必要があります。次の例では、Armadillo の IP アドレスを 192.168.10.10 に設定し、IP アドレスが 192.168.10.1 の TFTP サーバー上にある linux.bin を Linux カーネルイメージとして指定します。

```
hermit> setbootdevice tftp 192.168.10.10 192.168.10.1 --kernel=linux.bin
```

図 8.2 TFTP サーバー上の Linux カーネルイメージを指定する

### 8.4.3. Linux カーネル起動オプションの指定方法

Linux カーネルには様々な起動オプションがあります。詳しくは、Linux の解説書や、Linux カーネルのソースコードに含まれているドキュメント(Documentation/kernel-parameters.txt)を参照してください。

ここでは Armadillo-EVA 1500 で使用することができる、代表的な起動オプションを「表 8.6. Linux カーネルの起動オプションの一例」に紹介します。

表 8.6 Linux カーネルの起動オプションの一例

オプション指定子	説明
console=	<p>起動ログなどが出力されるインisialコンソールを指定します。 次の例では、コンソールに ttySC3(CON14)を、ボーレートに 115200 を指定しています。</p> <pre>console=ttySC2, 115200</pre>



オプション指定子	説明
root=	<p>ルートファイルシステムが構築されているデバイスを指定します。                      デバイスには Linux カーネルが認識した場合のデバイスを指定します。                      次の例では、デバイスに eMMC の第 2 パーティションを指定しています。</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <pre>root=/dev/mmcblk0p2</pre> </div>
rootwait	"root="で指定したデバイスが利用可能になるまでルートファイルシステムのマウントを遅らせます。
noinitrd	initrd を利用しないことを明示します。
mem	Linux カーネルが利用可能なメモリの量を指定します <sup>[a]</sup> 。

<sup>[a]</sup>メモリマップについては「表 3.5. Linux 起動時の DDR3-SDRAM メモリマップ」を参照してください。

## 9. ビルド手順

本章では、ソースコードから工場出荷イメージと同じイメージを作成する手順について説明します。

使用するソースコードは、評価セット付属の DVD に収録されています。最新版のソースコードは、Armadillo サイトからダウンロードすることができます。新機能の追加や不具合の修正などが行われているため、DVD に収録されているものよりも新しいバージョンがリリースされているかを確認して、最新バージョンのソースコードを利用することを推奨します。

Armadillo サイト - Armadillo-EVA 1500 ドキュメント・ダウンロード

<http://armadillo.atmark-techno.com/armadillo-eva-1500/downloads>

### 9.1. Linux カーネルをビルドする

ここでは、Linux カーネルのソースコードから Linux カーネルイメージと DTB(Device Tree Blob)を作成する手順について説明します。

#### 手順 9.1 Linux カーネルをビルド

##### 1. ソースコードの準備

ソースコードを準備します。Linux カーネルのソースコードアーカイブを準備し、展開します。展開後、`linux-3.4-ae1500-[version]`ディレクトリに移動します。

```
[ATDE ~]$ ls
linux-3.4-ae1500-[version].tar.gz
[ATDE ~]$ tar xzf linux-3.4-ae1500-[version].tar.gz
[ATDE ~]$ ls
linux-3.4-ae1500-[version] linux-3.4-ae1500-[version].tar.gz
[ATDE ~]$ cd linux-3.4-ae1500-[version]
```

##### 2. デフォルトコンフィギュレーションの適用

Armadillo-EVA 1500 のデフォルトコンフィギュレーションを適用します。

```
[ATDE ~/linux-3.4-ae1500-[version]]$ make ARCH=arm armadilloeva1500_defconfig
```

##### 3. ビルド

カーネルをビルドします。Linux カーネルイメージは `arch/arm/boot/`以下に作成されます。

```
[ATDE ~/linux-3.4-ae1500-[version]]$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf-
[ATDE ~/linux-3.4-ae1500-[version]]$ ls arch/arm/boot/Image
arch/arm/boot/Image
```

## 4. DTB の作成

DTS(Device Tree Source)から、DTB(Device Tree Blob)を作成します。

```
[ATDE ~/linux-3.4-ae1500-[version]]$ ./scripts/dtc/dtc -o r8a7791-armadilloeva1500.dtb
-I dts -O dtb -R 8 -p 0x3000 arch/arm/boot/dts/r8a7791-armadilloeva1500.dts
[ATDE ~/linux-3.4-ae1500-[version]]$ gzip -c r8a7791-armadilloeva1500.dtb > r8a7791-
armadilloeva1500.dtb.gz
[ATDE ~/linux-3.4-ae1500-[version]]$ ls r8a7791-armadilloeva1500.dtb.gz
r8a7791-armadilloeva1500.dtb.gz
```

↵

↵

## 9.2. ブートローダーをビルドする

ここでは、IPL(1st ブートローダー)と Hermit-At(2nd ブートローダー)のブートローダーイメージを作成する手順について説明します。IPL をビルドする場合は「9.2.1. IPL をビルドする」を、Hermit-At をビルドする場合は「9.2.2. Hermit-At をビルドする」を参照してください。

### 9.2.1. IPL をビルドする

IPL ブートローダーイメージ作成する手順について説明します。

#### 手順 9.2 IPL をビルド

##### 1. ソースコードの準備

IPL のソースコードアーカイブを準備し、展開します。展開後、ipl-ae1500-[version]ディレクトリに移動します。

```
[ATDE ~]$ ls
ipl-ae1500-[version].tar.gz
[ATDE ~]$ tar xzf ipl-ae1500-[version].tar.gz
[ATDE ~]$ ls
ipl-ae1500-[version] ipl-ae1500-[version].tar.gz
[ATDE ~]$ cd ipl-ae1500-[version]
```

##### 2. ビルド

IPL をビルドします。IPL ブートローダーイメージはビルドディレクトリ直下に作成されます。

```
[ATDE ~/ipl-ae1500-[version]]$ make
[ATDE ~/ipl-ae1500-[version]]$ ls *.bin
ipl-ae1500-[version].bin
```

### 9.2.2. Hermit-At をビルドする

Hermit-At ブートローダーイメージ作成する手順について説明します。

## 手順 9.3 Hermit-At をビルド

### 1. ソースコードの準備

Hermit-At のソースコードアーカイブを準備し、展開します。展開後、hermit-at-ae1500-[*version*]ディレクトリに移動します。

```
[ATDE ~]$ ls
hermit-at-ae1500-[version].tar.gz
[ATDE ~]$ tar xzf hermit-at-ae1500-[version].tar.gz
[ATDE ~]$ ls
hermit-at-ae1500-[version] hermit-at-ae1500-[version].tar.gz
[ATDE ~]$ cd hermit-at-ae1500-[version]
```

### 2. デフォルトコンフィギュレーションの適用

Armadillo-EVA 1500 のデフォルトコンフィギュレーションを適用します。

```
[ATDE ~/hermit-at-ae1500-[version]]$ make armadilloeva1500_defconfig
```

### 3. ビルド

Hermit-At をビルドします。Hermit-At ブートローダーイメージは src/target/armadillo15x0/以下に作成されます。

```
[ATDE ~/hermit-at-ae1500-[version]]$ make
[ATDE ~/hermit-at-ae1500-[version]]$ ls src/target/armadillo15x0/*.bin
src/target/armadillo15x0/loader-armadilloeva1500-[version].bin
```

## 9.3. ビルドしたイメージのインストール

ここでは、「9.1. Linux カーネルをビルドする」および「9.2. ブートローダーをビルドする」で作成したイメージのインストール方法を説明します。各イメージは、配置できる場所が決められています。各イメージの配置可能場所の一覧を次に示します

表 9.1 イメージの配置可能場所一覧

イメージ	SPI フラッシュメモリ	eMMC	SD カード
IPL	o <sup>[a]</sup>	x	x
Hermit-At	o	x	o
Linux カーネル	o	o	o
FDT	o	x	x

<sup>[a]</sup>オンボード SPI フラッシュメモリへ書き込むことはできません。ビルドした IPL イメージは、拡張インターフェース(CON2)に接続された外部 SPI フラッシュメモリからブートする場合に使用することを想定しています。

以降では、イメージを SPI フラッシュメモリおよび eMMC へインストールする方法について説明します。SD カードにイメージを配置する方法に関しては、「10. SD ブートの活用」を参照してください。

### 9.3.1. SPI フラッシュメモリへの書き込み方法

SPI フラッシュメモリへの書き込み方法について説明します。SPI フラッシュメモリの書き換え方法には、大きく分けて次に示す 2 種類の方法があります。

表 9.2 フラッシュメモリの書き換え方法

方法	特徴
ダウンローダーを使用する	<ul style="list-style-type: none"> <li>・ イメージファイルをシリアルで転送するため書き換えが低速</li> <li>・ Armadillo をネットワークに接続しなくても良い</li> </ul>
TFTP を使用する	<ul style="list-style-type: none"> <li>・ イメージファイルをネットワークで転送するため書き換えが高速</li> <li>・ Armadillo がネットワークに接続されている必要がある</li> </ul>

SPI フラッシュメモリを書き換えるためには、Hermit-At が起動している必要があります。SPI フラッシュメモリに書き込まれている Hermit-At が起動しない状態になってしまった場合は、「10. SD ブートの活用」を参照して SD カードからソフトウェアを起動させてください。

### 9.3.1.1. フラッシュメモリのパーティションについて

フラッシュメモリの書き換えは、パーティション毎に行います。パーティションは"リージョン"とも呼ばれます。

各パーティションのサイズはフラッシュメモリ内には保存されていません。IPL、Hermit-At と Linux カーネルそれぞれが同じパーティションテーブルを保持することにより、一意的に扱うことができるようになっています。

各パーティションは、書き込みを制限することが可能です。書き込みを制限する理由は、誤動作や予期せぬトラブルにより、フラッシュメモリ上のデータが不意に破壊または消去されることを防ぐためです。

読み込みは、常時可能です。読み込みに制限を付けることはできません。

各パーティションのデフォルト状態での書き込み制限の有無と、対応するイメージファイル名を次に示します。


表 9.3 パーティションのデフォルト状態での書き込み制限の有無と対応するイメージファイル名

パーティション	書き込み制限	イメージファイル名	備考
ipl	あり	ipl-ae1500-[version].bin	IPL ブートローダーイメージを配置するパーティションです。
bootloader	あり	loader-armadilloeva1500-[version].bin	Hermit-At ブートローダーイメージを配置するパーティションです。
fdt	なし	r8a7791-armadilloeva1500-[version].dtb.gz	DTB を配置するパーティションです。
firmware	あり	squashfs-ae1500-firmware-[version].img	有償ミドルウェアなどのファームウェアを配置するパーティションです。



firmware パーティションのファームウェアは、次に示すバージョンのソフトウェアで使用可能です。

ソフトウェア	対応バージョン
Hermit-At	v4.1.0 以降(Hermit-At ブートローダーイメージ loader-armadilloeva1500-v4.1.0.bin 以降)
Linux カーネル	v3.4-ae1500-at2 以降(Linux カーネルイメージ linux-ae1500-v1.01.bin 以降)
Debian GNU/Linux	v20141022 以降(Debian GNU/Linux ルートファイルシステムアーカイブ debian-wheezy-armhf_ae1500_20141022.tar.gz 以降)
ファームウェア	v1.00 以降(ファームウェアイメージ squashfs-ae1500-firmware-v1.00.img 以降)



工場出荷状態でフラッシュメモリに書き込まれているイメージファイルは、最新版ではない可能性があります。最新版の Hermit-At ブートローダー、DTB は Armadillo サイトから、ファームウェアイメージファイルはユーザーズサイトからダウンロード可能です。最新版のイメージファイルに書き換えてからのご使用を推奨します。

ダウンローダーでは、書き込みが制限されているパーティションを"ロック(locked)されている"と呼びます。このパーティションを強制的に書き換える場合は、"--force-locked"というオプションを付けます。他のオプションについては、「9.3.1.2. ダウンローダーを使用して SPI フラッシュメモリを書き換える」を参照してください。

Linux が動いている場合、パーティションの書き込み制限をコマンドで外すことが可能です。Sysfs の MTD クラスディレクトリ以下にある"ro"というファイルに 0 を書き込むことで制限を外すことが可能です。逆に 1 を書き込めば、パーティションへの書き込みを制限する事が可能です。

MTD クラスディレクトリとパーティションの対応は次の通りです。

**表 9.4 パーティションと MTD クラスディレクトリの対応**

パーティション	MTD クラスディレクトリ
ipl	/sys/class/mtd/mtd0
bootloader	/sys/class/mtd/mtd1
fdt	/sys/class/mtd/mtd2
firmware	/sys/class/mtd/mtd3

以降の説明では、任意のパーティションを示す MTD クラスディレクトリを"/sys/class/mtd/[MTD]"のように表記します。

書き込み制限を外すには、ro ファイルに 0 を書き込みます。

```
[armadillo ~]# echo 0 > /sys/class/mtd/[MTD]/ro
```

**図 9.1 書き込み制限を外す**

書き込みを制限するには、ro ファイルに 1 を書き込みます。

```
[armadillo ~]# echo 1 > /sys/class/mtd/[MTD]/ro
```

**図 9.2 書き込みを制限する**

### 9.3.1.2. ダウンローダーを使用して SPI フラッシュメモリを書き換える

Armadillo をネットワークに接続できない場合は、ダウンローダー(hermit)を使用して SPI フラッシュメモリを書き換える必要があります。hermit は ATDE に標準でインストールされています。

hermit は Armadillo で動作している Hermit-At と協調動作を行い SPI フラッシュメモリを書き換えることができます。hermit と Armadillo で動作している Hermit-At 間の通信には、シリアル<sup>[1]</sup>が使用されます。

<sup>[1]</sup>通信速度(ボーレート)は、115200bps です

hermit のヘルプは次のとおりです。

```
[ATDE ~]# hermit
Usage: hermit [options] command [command options]
Available commands: download, erase, help, go, map, terminal, upload, md5sum
Armadillo-J command: firmupdate
Multiple commands may be given.
General options (defaults) [environment]:
  -e, --ethernet
  -i, --input-file <path>
  --netif <ifname> (eth0) [HERMIT_NETIF]
  --memory-map <path>
  --port <dev> (/dev/ttyS0) [HERMIT_PORT]
  -o, --output-file <path>
  --remote-mac <MAC address>
  -v, --verbose
  -V, --version
Download/Erase options:
  -a, --address <addr>
  -b, --baudrate <baudrate>
  --force-locked
  -r, --region <region name>
Memory map options:
  --anonymous-regions
Md5sum options:
  -a, --address <addr>
  -r, --region <region name>
  -s, --size <size>
```

### 図 9.3 hermit コマンドのヘルプ

ここでは、bootloader パーティションを書き換える手順について説明します。

#### 手順 9.4 ダウンローダーを使用して書き換える

1. Hermit-At ブートローダーが保守モードで起動するように設定します。設定方法については、「8.3. Hermit-At の起動モード」を参照してください。
2. Armadillo が保守モードで起動したことを確認するために、ATDE で minicom を起動しておきます。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。

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

3. Armadillo に電源を投入します。Hermit-At ブートローダーが保守モードで起動すると、次のように保守モードのプロンプトが表示されます。

```
hermit>
```

4. minicom を終了させシリアルポート(/dev/ttyUSB0)を開放します。

- bootloader パーティションと書き込むイメージファイル(loader-armadilloeva1500-[version].bin)を指定して hermit コマンドを実行します。bootloader パーティションを更新する場合は、必ず"--force-locked"オプションを指定する必要があります。

```
[ATDE ~]$ hermit download --input-file loader-armadilloeva1500-[version].bin --region
bootloader --force-locked --port /dev/ttyUSB0
serial: completed 0x0000a92c (43308) bytes.
```



書き込みが制限されているパーティションを書き換える場合、"--force-locked"オプションを指定する必要があります。

- ATDE のプロンプトが表示されると SPI フラッシュメモリの書き換えは完了です。次回起動時から書き換えた Hermit-At ブートローダーイメージで起動します。

```
[ATDE ~]$
```

### 9.3.1.3. TFTP を使用して SPI フラッシュメモリを書き換える

Hermit-At ブートローダーの tftpd 機能を使用することで、ネットワーク経由で、SPI フラッシュメモリを書き換えることができます。

tftpd 機能は、所属するネットワークにある TFTP サーバーが公開しているファイルをダウンロードして、自分自身の SPI フラッシュメモリを書き換えることができる機能です。



ATDE5 では、標準で TFTP サーバー (atftpd) が動作しています。/var/lib/tftpboot/ ディレクトリにファイルを置くことで、TFTP によるアクセスが可能になります。

tftpd 機能を使用するには、Hermit-At を保守モードで起動してください。

作業用 PC のシリアル通信ソフトウェアを使用して、コマンドを入力します。「図 9.4. tftpd コマンド例」は、Armadillo の IP アドレスを 192.168.10.10 に設定し、IP アドレスが 192.168.10.1 の TFTP サーバー上にある、r8a7791-armadilloeva1500.dtb.gz を fdt パーティションに書き込む例です。

```
hermit> tftpd 192.168.10.10 192.168.10.1 --fdt=r8a7791-armadilloeva1500.dtb.gz
```

図 9.4 tftpd コマンド例

書き込み対象となるパーティションを指定するオプションと、パーティションの対応を次に示します。



表 9.5 パーティションとオプションの対応

パーティション	オプション
bootloader	--bootloader
fdt	--fdt



tftpdは、TFTP プロトコルを使用して TFTP サーバーからイメージファイルをダウンロードします。デフォルトのデータブロックサイズが 512Byte であるため、イメージファイルの最大サイズがブロック番号の桁溢れが発生しない 33554431Byte(32MByte - 1Byte)に制限されます。これよりもサイズの大きいイメージファイルをダウンロードする場合は、"--blksize"オプションを利用してデータブロックサイズを増やす必要があります。

"--blksize"オプションには、IP フラグメンテーションが起きないデータブロックサイズを指定する必要があります。

### 9.3.2. eMMC への書き込み方法

カーネルイメージを、eMMC にインストールします。事前に「10. SD ブートの活用」を参照して、Debian GNU/Linux を SD ブートしている必要があります。

「9.1. Linux カーネルをビルドする」で作成したカーネルイメージ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
Image.bin
[armadillo ~]# mount /dev/mmcblk0p1 /mnt ❶
[armadillo ~]# rm -f /mnt/boot/*.bin ❷
[armadillo ~]# cp Image.bin /mnt/boot/Image.bin ❸
[armadillo ~]# umount /mnt
```

- ❶ eMMC パーティション 1 をマウントします。
- ❷ eMMC パーティション 1 の /boot/以下のカーネルイメージファイルを削除します。
- ❸ eMMC パーティション 1 の /boot/以下にカーネルイメージファイルをコピーします。使用できるファイル名に関しては「表 10.7. ブートローダーが Linux カーネルを検出可能な条件」を参照してください。

図 9.5 Debian GNU/Linux システムへのカーネルイメージのインストール

# 10. 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 2GByte 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 7(コードネーム wheezy)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。



SD ブートを行った場合でも、ブートローダーの設定(保守モードの `setenv/setboodevice` コマンドで設定する項目)については SPI フラッシュメモリに保存されます。

SD カードに対する作業は、ATDE で行います。そのため、ATDE に SD カードを接続する必要があります。詳しくは「4.2.2. 取り外し可能デバイスの使用」を参照してください。

ATDE に SD カードを接続すると、自動的に `/media/` ディレクトリにマウントされます。本章に記載されている手順を実行するためには、次のように SD カードをアンマウントしておく必要があります。

```
[ATDE ~]$ mount
(省略)
/dev/sdb1 on /media/52E6-5897 type ext2
(rw,nosuid,nodev,relatime,uid=1000,gid=1000,fmask=0022,dmask=0077,codepage=cp437,ioccharset=utf8,sh
ortname=mixed,showexec=utf8,flush,errors=remount-ro,uhelper=udisks)
[ATDE ~]$ sudo umount /dev/sdb1
```

図 10.1 自動マウントされた SD カードのアンマウント

本章で使用するブートローダーイメージファイルなどは、評価セット付属の DVD に収録されています。最新版のファイルは、「Armadillo サイト」でダウンロードすることができます。新機能の追加や不具合の修正などが行われているため、DVD に収録されているものよりも新しいバージョンがリリースされているかを確認して、最新バージョンのソースコードを利用することを推奨します。

**Armadillo サイト - Armadillo-EVA 1500 ドキュメント・ダウンロード**

<http://armadillo.atmark-techno.com/armadillo-eva-1500/downloads>

## 10.1. ブートディスクの作成

ATDE でブートディスクを作成します。ブートディスクの作成に使用するファイルを次に示します。

表 10.1 ブートディスクの作成に使用するファイル

ファイル	ファイル名
SD ブート用ブートローダーイメージ	loader-armadilloeva1500- <i>[version]</i> .bin

SD カードにブートローダーイメージを配置する際、「表 10.2. ブートディスクの制約」に示す制約があります。本章に示す手順を実行した場合は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。

表 10.2 ブートディスクの制約

項目	制約
パーティション番号	1
パーティションのシステムタイプ	0x83(Linux)
ファイルシステム	ext2
ブートローダーイメージファイル名	sdboot.bin
ブートローダーイメージファイルの配置場所	ルートディレクトリ直下

「表 10.3. ブートディスクの構成例」に示すブートディスクを作成する手順を、「手順 10.1. ブートディスクの作成例」に示します。

表 10.3 ブートディスクの構成例

パーティション番号	パーティションサイズ	ファイルシステム	説明
1	128MByte	ext2	ブートローダーイメージを配置します。
2	残り全て	ext3	ルートファイルシステムを構築するために ext3 ファイルシステムを構築しておきます。

手順 10.1 ブートディスクの作成例

1. SD ブート用のブートローダーイメージファイルを取得します。

```
[ATDE ~]$ ls
loader-armadilloeva1500-[version].bin
```

2. 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。

```
[ATDE ~]$ sudo fdisk /dev/sdb ❶

Command (m for help): o ❷
Building a new DOS disklabel with disk identifier 0x8cb9edcc.
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)

Command (m for help): n ❸
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended

Select (default p): ❹
Using default response p

Partition number (1-4, default 1): ❺
Using default value 1

First sector (2048-3862527, default 2048): ❻
Using default value 2048

Last sector, +sectors or +size{K,M,G} (2048-3862527, default 3862527): +128M ❼
```

```

Command (m for help): n ❸
Partition type:
   p   primary (1 primary, 0 extended, 3 free)
   e   extended
Select (default p): ❹
Using default response p
Partition number (1-4, default 2): ❺
Using default value 2
First sector (264192-3862527, default 264192): ❻
Using default value 264192
Last sector, +sectors or +size{K,M,G} (264192-3862527, default 3862527): ❼
Using default value 3862527

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

Calling ioctl() to re-read partition table.

WARNING: If you have created or modified any DOS 6.x
partitions, please see the fdisk manual page for additional
information.
Syncing disks.
[ATDE ~]$

```

- ❶ 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。
- ❷ 新しく空の DOS パーティションテーブルを作成します。
- ❸ 新しくパーティションを追加します。
- ❹ パーティション種別にはデフォルト値(p: プライマリ)を指定するので、そのまま改行を入力してください。
- ❺ パーティション番号にはデフォルト値(1)を指定するので、そのまま改行を入力してください。
- ❻ 開始セクタにはデフォルト値(使用可能なセクタの先頭)を使用するので、そのまま改行を入力してください。
- ❼ 最終シリンダは、128MByte 分を指定します。
- ❽ 新しくパーティションを追加します。
- ❹ パーティション種別にはデフォルト値(p: プライマリ)を指定するので、そのまま改行を入力してください。
- ❺ パーティション番号にはデフォルト値(2)を指定するので、そのまま改行を入力してください。
- ❻ 開始セクタにはデフォルト値(第 1 パーティションの最終セクタの次のセクタ)を使用するので、そのまま改行を入力してください。
- ❼ 最終セクタにはデフォルト値(末尾セクタ)を使用するので、そのまま改行を入力してください。
- ❿ 変更を SD カードに書き込みます。

- パーティションリストを表示し、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。

```
[ATDE ~]$ sudo fdisk -l /dev/sdb

Disk /dev/sdb: 1977 MB, 1977614336 bytes
61 heads, 62 sectors/track, 1021 cylinders, total 3862528 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x8cb9edcc

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048        264191    131072    83  Linux
/dev/sdb2          264192        3862527   1799168    83  Linux
```

- 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。

```
[ATDE ~]$ sudo mkfs.ext2 /dev/sdb1 ❶
sudo mkfs.ext2 /dev/sdb1
mke2fs 1.42.5 (29-Jul-2012)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
32768 inodes, 131072 blocks
6553 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67371008
16 block groups
8192 blocks per group, 8192 fragments per group
2048 inodes per group
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

[ATDE ~]$ sudo mkfs.ext3 -L rootfs /dev/sdb2 ❷
mke2fs 1.42.5 (29-Jul-2012)
Filesystem label=rootfs
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
112448 inodes, 449792 blocks
22489 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=461373440
14 block groups
32768 blocks per group, 32768 fragments per group
8032 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912
```


```
Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

[ATDE ~]$
```

- ❶ 第 1 パーティションに ext2 ファイルシステムを構築します。
  - ❷ 第 2 パーティションに ext3 ファイルシステムを構築します。ボリュームラベルには "rootfs" を設定します。
5. SD ブート用のブートローダーイメージファイルを第 1 パーティションに配置します。

```
[ATDE ~]$ ls
loader-armadilloeva1500-[version].bin
[ATDE ~]$ mkdir sd ❶
[ATDE ~]$ sudo mount -t ext2 /dev/sdb1 sd ❷
[ATDE ~]$ sudo cp loader-armadilloeva1500-[version].bin sd/sdboot.bin ❸
[ATDE ~]$ sudo umount sd ❹
[ATDE ~]$ rmdir sd ❺
```

- ❶ SD カードをマウントするための sd/ディレクトリを作成します。
- ❷ 第 1 パーティションを sd/ディレクトリにマウントします。
- ❸ sd/ディレクトリにブートローダーイメージをコピーします。ファイル名は "sdboot.bin" にリネームする必要があります。
- ❹ sd/ディレクトリにマウントした第 1 パーティションをアンマウントします。
- ❺ sd/ディレクトリを削除します。



アンマウントが完了する前に SD カードを作業用 PC から取り外すと、SD カードのデータが破損する場合があります。

## 10.2. ルートファイルシステムの構築

「10.1. ブートディスクの作成」で作成したブートディスクにルートファイルシステムを構築します。ルートファイルシステムの構築に使用するファイルを次に示します。

表 10.4 ルートファイルシステムの構築に使用するファイル

Linux ディストリビューション	ファイル名	ファイルの説明
Debian GNU/Linux	debian-wheezy-ae1500-[version].tar.gz	ARM(armhf)アーキテクチャ用 Debian GNU/Linux 7(コードネーム「wheezy」)のルートファイルシステムアーカイブ

Debian GNU/Linux ルートファイルシステムアーカイブから、ルートファイルシステムを構築する手順を次に示します。

## 手順 10.2 Debian GNU/Linux ルートファイルシステムアーカイブからルートファイルシステムを構築する

1. Debian GNU/Linux ルートファイルシステムアーカイブを準備しておきます。

```
[ATDE ~]$ ls
debian-wheezy-ae1500-[version].tar.gz
```

2. ルートファイルシステムをブートディスクの第 2 パーティションに構築します。

```
[ATDE ~]$ mkdir sd ❶
[ATDE ~]$ sudo mount -t ext3 /dev/sdb2 sd ❷
[ATDE ~]$ sudo tar zxf debian-wheezy-ae1500-[version].tar.gz -C sd ❸
[ATDE ~]$ sudo umount sd ❹
[ATDE ~]$ rmdir sd ❺
```

- ❶ SD カードをマウントするための sd/ディレクトリを作成します。
- ❷ 第 2 パーティションを sd/ディレクトリにマウントします。
- ❸ ルートファイルシステムアーカイブを sd/ディレクトリに展開します。
- ❹ sd/ディレクトリにマウントしたブートディスクの第 2 パーティションをアンマウントします。
- ❺ sd/ディレクトリを削除します。



アンマウントが完了する前に SD カードを作業用 PC から取り外すと、SD カードのデータが破損する場合があります。

3. 標準イメージの Debian GNU/Linux 7 ルートファイルシステムは、eMMC パーティション 1 をルートファイルシステムとして使用するように設定されています。ルートファイルシステムの配置先を変更する場合は、fstab の設定を修正する必要があります。fstab の設定を、「表 10.5. ディスクデバイスの割り当て」を参考に修正します。この例では、SD スロット 1 (CON6) に接続した SD カードのパーティション 2 をルートファイルシステムとして使用するように、"/dev/mmcblk0p1"から"/dev/mmcblk1p2"に書き換えています。

```
[ATDE ~]$ sudo mount -t ext3 /dev/sdb2 /mnt
[ATDE ~]$ sudo vi /mnt/etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount> <type> <options> <dump> <pass>
/dev/mmcblk1p2 / ext3 errors=remount-ro 0 1
```

```
/dev/mtdblock3 /opt/firmware squashfs exec,ro 0 0
[ATDE ~]$ sudo umount /mnt
```

図 10.2 fstab の編集

表 10.5 ディスクデバイスの割り当て

インターフェース	ディスクデバイス
eMMC	/dev/mmcblk0
SD スロット 1(CON6)	/dev/mmcblk1
SD スロット 2(CON4)	/dev/mmcblk2

### 10.3. Linux カーネルイメージの配置

「10.2. ルートファイルシステムの構築」で作成したルートファイルシステムに Linux カーネルイメージを配置します。Linux カーネルイメージの配置に使用するファイルを次に示します。

表 10.6 ブートディスクの作成に使用するファイル

ファイル	ファイル名
Linux カーネルイメージ	linux-ae1500-[version].bin

SD カードに Linux カーネルイメージを配置する際は、次の条件を満たすようにしてください。この条件から外れた場合、ブートローダーが Linux カーネルイメージを検出することができなくなる場合があります。

表 10.7 ブートローダーが Linux カーネルを検出可能な条件

項目	条件
ファイルシステム	ext2 または ext3
圧縮形式	gzip 形式 または 非圧縮
Linux カーネルイメージファイル名(gzip 形式)	Image.gz, linux.gz, Image.bin.gz, linux.bin.gz のいずれか
Linux カーネルイメージファイル名(非圧縮)	Image, linux, Image.bin, linux.bin のいずれか
Linux カーネルイメージファイルの配置場所	/boot/ディレクトリ直下

Linux カーネルイメージをルートファイルシステムに配置する手順を次に示します。

#### 手順 10.3 Linux カーネルイメージの配置例

- Linux カーネルイメージを準備しておきます。

```
[ATDE ~]$ ls
linux-ae1500-[version].bin
```

- Linux カーネルイメージをブートディスクの第 2 パーティションに配置します。

```
[ATDE ~]$ mkdir sd ①
[ATDE ~]$ sudo mount -t ext3 /dev/sdb2 sd ②
[ATDE ~]$ sudo mkdir -p sd/boot ③
[ATDE ~]$ sudo cp linux-ae1500-[version].bin sd/boot/Image.bin ④
[ATDE ~]$ sudo umount sd ⑤
[ATDE ~]$ rmdir sd ⑥
```



- ❶ SD カードをマウントするための sd/ディレクトリを作成します。
- ❷ 第 2 パーティションを sd/ディレクトリにマウントします。
- ❸ Linux カーネルイメージを配置するための boot/ディレクトリを作成します。
- ❹ Linux カーネルイメージを sd/boot/ディレクトリにコピーします。
- ❺ sd/ディレクトリにマウントしたブートディスクの第 2 パーティションをアンマウントします。
- ❻ sd/ディレクトリを削除します。



アンマウントが完了する前に SD カードを作業用 PC から取り外すと、SD カードのデータが破損する場合があります。

## 10.4. SD ブートの実行

「10.1. ブートディスクの作成」で作成したブートディスクから起動する方法を説明します。

Armadillo に電源を投入する前に次の準備を行います。



SD ブートを行う場合は、SD インターフェース(CON6)もしくは SD インターフェース(CON4)の、どちらか一方にだけ SD カードを接続してください。Linux がルートファイルシステムのマウントを完了した後は、二枚目の SD カードを接続する事ができます。

1. CON6 または CON4 にブートディスクを接続します。
2. Hermit-At が保守モードとなるように、機能選択スイッチの Hermit-At 起動モード設定(SW2.8)を ON に設定します。
3. SD カードの Hermit-At を起動するように、機能選択スイッチの Hermit-At 起動デバイス設定(SW2.7)を OFF に設定します。
4. CON4 からブートする場合は、機能選択スイッチの SD/WLAN 設定(SW2.5)を ON の状態で、SD ブートデバイス選択スイッチ(SW9)を押下しながら電源を投入します。

「10.2. ルートファイルシステムの構築」で構築したルートファイルシステムで起動する場合は、「図 10.3. ルートファイルシステムの起動設定」のように setenv コマンドで Linux カーネル起動オプションを設定します。rootfs オプションにて指定する値は、「表 10.5. ディスクデバイスの割り当て」を参考に設定してください。setenv コマンドの詳細については「8.4. Hermit-At の機能」を参照してください。

```
hermit> setenv console=ttySC3,115200 mem=768M noinitrd rootwait root=/dev/mmcblk1p2
hermit> setenv
1: console=ttySC3,115200
2: noinitrd
3: rootwait
4: root=/dev/mmcblk1p2
5: mem=768M
```

図 10.3 ルートファイルシステムの起動設定



マルチメディアパッケージドライバを無効にした Linux カーネルを利用する場合は、mem=1024M を指定することができます。



Linux カーネル起動オプションを工場出荷状態に戻すには、次のようにコマンドを実行します。

```
hermit> setenv console=ttySC3,115200 mem=768M noinitrd rootwait root=/dev/mmcblk0p1
```



「10.3. Linux カーネルイメージの配置」で配置した Linux カーネルイメージで起動する場合は、保守モードで「図 10.4. Linux カーネルの起動設定」のように setbootdevice コマンドで Linux カーネルイメージを指定します。bootdevice の指定は、「表 10.5. ディスクデバイスの割り当て」を参照してください。setbootdevice コマンドの詳細については「8.4.2. Linux カーネルイメージの指定方法」を参照してください。

```
hermit> setbootdevice mmcblk1p2
hermit> setbootdevice
bootdevice: mmcblk1p2
```

図 10.4 Linux カーネルの起動設定



起動デバイス設定を工場出荷状態(eMMC から起動)に戻すには、次のようにコマンドを実行します。

```
hermit> setbootdevice mmcblk0p1
```

# 11. リカバリ手順

本章では、Armadillo-EVA 1500 に搭載された内蔵ストレージ(eMMC および SPI フラッシュメモリ)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 2GByte 以上の SD カードが必要です。

## 11.1. パーティション構成

Armadillo-EVA 1500 の内蔵ストレージのパーティション構成を示します。eMMC のパーティションについては「11.1.1. eMMC のパーティション構成」を、SPI フラッシュメモリのパーティションについては「11.1.2. SPI フラッシュメモリのパーティション構成」を参照してください。

### 11.1.1. eMMC のパーティション構成

工場出荷状態の eMMC のパーティション構成を次に示します。

表 11.1 eMMC 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/mmcblk0p1	約 5GByte	83 (Linux)
/dev/mmcblk0p2	2GByte	83 (Linux)

JEDEC 規格の eMMC 4.3 以降では、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-EVA 1500 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 11.2 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	4MByte
/dev/mmcblk0boot1	4MByte

「表 11.1. eMMC のパーティション構成」および「表 11.2. eMMC のブートパーティション」に示す通り、eMMC を 4 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 11.3. eMMC 各領域の用途」に示します。Armadillo-EVA 1500 のプロセッサである RZ/G1M は eMMC からの起動をサポートしないため、ブートパーティションは使用していません。

表 11.3 eMMC 各領域の用途

パーティション	用途
パーティション 1 (/dev/mmcblk0p1)	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 2 (/dev/mmcblk0p2)	使用していません。ストレージ用途としてなど、自由に使用することができます。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)	使用していません。
ブートパーティション 2 (/dev/mmcblk0boot1)	使用していません。

## 11.1.2. SPI フラッシュメモリのパーティション構成

SPI フラッシュメモリは eMMC とは異なり、自身でパーティションテーブルを保持していません。SPI フラッシュメモリを使用するソフトウェア (IPL、Hermit-At および Linux カーネル) がそれぞれが同じパーティションテーブルを保持することにより、一意的に扱うことができるようになっています。

工場出荷状態の SPI フラッシュメモリのパーティション構成を次に示します。

表 11.4 SPI フラッシュメモリのパーティション構成

パーティション名	アドレス	サイズ
ipl	0x00000000   0x0000FFFF	64kByte
bootloader	0x00010000   0x0008FFFF	512kByte
fdt	0x00090000   0x000CFFFF	256kByte
firmware	0x000D0000   0x007FFFFFFF	7.1875MByte

「表 11.4. SPI フラッシュメモリのパーティション構成」に示す通り、SPI フラッシュメモリを 4 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次に示します。

表 11.5 SPI フラッシュメモリ 各領域の用途

パーティション名	用途
ipl	IPL ブートローダーイメージがインストールされています。
bootloader	Hermit-At ブートローダーイメージがインストールされています。
fdt	Device Tree Blob がインストールされています。Linux カーネルに渡すための Armadillo-EVA 1500 のデバイス情報を持っています。
firmware	有償ミドルウェアなどのファームウェアがインストールされています。

## 11.2. 内蔵ストレージ全体をリカバリする

eMMC と SPI フラッシュメモリをリカバリする方法を示します。

### 11.2.1. リカバリディスクの作成

作業用 PC で、リカバリを行うための 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。Linux が起動している Armadillo-EVA 1500 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。

### 11.2.1.1. リカバリディスクの作成に必要なファイルの取得

「表 11.6. 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。

表 11.6 リカバリディスクの作成に必要なファイル

名称	ファイル名	説明
リカバリシステム本体	recovery-system_ae1500_[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
リカバリシステム用イメージアーカイブ	recovery-image_ae1500_[version].tar.gz	内蔵ストレージ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
Firmware パーティション用イメージファイル	squashfs-ae1500-firmware-[version].img	SPI フラッシュメモリの firmware パーティションに書き込むためのイメージファイルです

リカバリシステム本体とリカバリシステム用イメージアーカイブは、Armadillo サイトまたは評価セット付属の DVD-ROM から取得可能です。Firmware パーティション用イメージファイルは、アットマークテクノ ユーザーズサイトから取得可能です。

Armadillo サイト - Armadillo-EVA 1500 ドキュメント・ダウンロード

<http://armadillo.atmark-techno.com/armadillo-eva-1500/downloads>

アットマークテクノ ユーザーズサイト

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



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」を行う必要があります。

### 11.2.1.2. 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 11.1. 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
Command (m for help): n ❷
Partition type:
  p   primary (0 primary, 0 extended, 4 free)
  e   extended
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-4009983, default 2048): ❸
```

```

Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-4009983, default 4009983): +1G ④

Command (m for help): n ⑤
Partition type:
   p   primary (1 primary, 0 extended, 3 free)
   e   extended
Select (default p): p
Partition number (1-4, default 2): 2
First sector (2099200-4009983, default 2099200): ⑥
Using default value 2099200
Last sector, +sectors or +size{K,M,G} (2099200-4009983, default 4009983): ⑦
Using default value 4009983

Command (m for help): w ⑧
The partition table has been altered!

Calling ioctl() to re-read partition table.
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 新しくプライマリパーティション 1 を作成します。
- ③ 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。
- ④ 最終シリンダは 1GByte 分を指定します。
- ⑤ 新しくプライマリパーティション 2 を作成します。
- ⑥ 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。
- ⑦ 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。
- ⑧ 変更を SD カードに書き込みます。

### 図 11.1 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 11.2. 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: 2053 MB, 2053111808 bytes
64 heads, 62 sectors/track, 1010 cylinders, total 4009984 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

   Device Boot      Start         End      Blocks   Id  System
    
```

/dev/sdb1	2048	2099199	1048576	83	Linux
/dev/sdb2	2099200	4009983	955392	83	Linux

図 11.2 パーティション確認手順

### 11.2.1.3. ファイルシステムの構築

「11.2.1.2. 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 11.3. 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。

```
[PC ~]# mkfs.ext2 /dev/sdb1 ❶
mke2fs 1.42.5 (29-Jul-2012)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
65536 inodes, 262144 blocks
13107 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=268435456
8 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

[PC ~]# mkfs.ext3 /dev/sdb2 ❷
mke2fs 1.42.5 (29-Jul-2012)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
59776 inodes, 238848 blocks
11942 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=247463936
8 block groups
32768 blocks per group, 32768 fragments per group
7472 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
[PC ~]#
```

- ① プライマリパーティション 1 に ext2 ファイルシステムを構築します。
- ② 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 11.3 ファイルシステム作成手順

#### 11.2.1.4. リカバリイメージの展開

「11.2.1.1. リカバリディスクの作成に必要なファイルの取得」で取得したリカバリシステム用イメージアーカイブとリカバリシステム本体を、「11.2.1.3. 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 11.4. ファイル展開手順」のようにコマンドを実行します。次のコマンド例では、各ファイルがホーム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_ae1500_[version].tar.gz  recovery-system_ae1500_[version].tar.gz
[PC ~]# mount -t ext2 /dev/sdb1 /mnt
[PC ~]# cd /mnt
[PC /mnt]# tar zxf ~/recovery-image_ae1500_[version].tar.gz
[PC /mnt]# cd
[PC ~]# umount /mnt

[PC ~]# mount -t ext3 /dev/sdb2 /mnt
[PC ~]# cd /mnt
[PC /mnt]# tar zxf ~/recovery-system_ae1500_[version].tar.gz
[PC /mnt]# cd
[PC ~]# umount /mnt
[PC ~]#
```

### 図 11.4 ファイル展開手順

#### 11.2.1.5. Firmware パーティション用イメージファイルの配置

「11.2.1.1. 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 Firmware パーティション用イメージファイルを、「11.2.1.4. リカバリイメージの展開」でリカバリイメージを展開した SD カードのパーティション 1 に配置し、設定ファイル (recovery.conf) 更新用のシェルスクリプト (update\_firmware.sh) を実行します。

作業用 PC に SD カードを接続したまま、「図 11.5. Firmware イメージファイルの配置」のようにコマンドを実行します。次のコマンド例では、Firmware パーティション用イメージファイルがホームディレクトリ以下にあることを想定しています。

```
[PC ~]# ls
squashfs-ae1500-firmware-[version].img
[PC ~]# mount -t ext2 /dev/sdb1 /mnt
[PC ~]# cd /mnt
[PC /mnt]# cp ~/squashfs-ae1500-firmware-[version].img .
[PC /mnt]# ./update_conf_firmware.sh squashfs-ae1500-firmware-[version].img
[PC /mnt]# cd
[PC ~]# umount /mnt
[PC ~]#
```

### 図 11.5 Firmware イメージファイルの配置



## 11.2.2. リカバリを実行する



以下に示す手順を実行すると、Armadillo-EVA 1500 に搭載された内蔵ストレージ(eMMC および SPI フラッシュメモリ)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、「11.3. eMMC の Debian GNU/Linux ルートファイルシステムをリカバリする」または、「11.4. SPI フラッシュメモリの特定パーティションをリカバリする」を参照してください。

「11.2.1. 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-EVA 1500 に電源を投入する前に次に示す準備を行います。

- 機能選択スイッチの SW1.7、SW2.7 および SW2.8 を OFF に、その他を全て ON に設定します。



- SD インターフェース(CON6)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 11.7 リカバリ進捗と LED の対応

LED4	LED5	LED6	LED7	リカバリ進捗
点滅	消灯	消灯	消灯	eMMC をフォーマット中
点滅	点灯	点灯	点灯	SPI フラッシュメモリ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 1 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、機能選択スイッチの Hermit-At 起動デバイス設定(SW2.7)を ON に設定し、SD カードを抜いた後電源を投入してください。

## 11.3. eMMC の Debian GNU/Linux ルートファイルシステムをリカバリする

eMMC のパーティション 1 にインストールされている Debian GNU/Linux ルートファイルシステムをリカバリする方法を紹介します。事前に「10. SD ブートの活用」を参照して、Armadillo-EVA 1500 が SD ブートしている必要があります。

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

「表 11.8. 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。これらファイルは Armadillo サイト (<http://armadillo.atmark-techno.com>)または、評価セット付属の DVD-ROM から取得可能です。

表 11.8 Debian GNU/Linux のリカバリに必要なファイル

ファイル	説明
debian-wheezy-armhf_ae1500_[version].tar.gz	Debian GNU/Linux 7 のルートファイルシステムアーカイブ
linux-ae1500-[version].bin	Linux カーネルイメージファイル

### 11.3.2. Debian GNU/Linux のリカバリを実行する

「図 11.6. 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。「11.3.1. 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ホームディレクトリ以下にあることを想定しています。

```
[armadillo ~]# ls
debian-wheezy-armhf_ae1500_[version].tar.gz  linux-ae1500-[version].bin
[armadillo ~]# mkfs.ext3 /dev/mmcblk0p1 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/mmcblk0p1 /mnt ❷
[armadillo ~]# tar xzf ~/debian-wheezy-armhf_ae1500_[version].tar.gz -C /mnt ❸
[armadillo ~]# cp ~/linux-ae1500-[version].bin /mnt/boot/Image.bin ❹
[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 11.6 Debian GNU/Linux のリカバリ手順

## 11.4. SPI フラッシュメモリの特定パーティションをリカバリする

SPI フラッシュメモリの特定パーティションのみをリカバリする方法を紹介します。事前に「10. SD ブートの活用」を参照して、Armadillo-EVA 1500 がSD ブートしている必要があります。

### 11.4.1. SPI フラッシュメモリのリカバリに必要なファイルの取得

SPI フラッシュメモリのリカバリに必要なファイルを「表 11.9. SPI フラッシュメモリのリカバリに必要なファイル」に示します。「11.1.2. SPI フラッシュメモリのパーティション構成」を参照して、リカバリするパーティション用のファイルを取得してください。

表 11.9 SPI フラッシュメモリのリカバリに必要なファイル

パーティション名	ファイル	説明
bootloader	loader-armadilloeva1500- <i>[version]</i> .bin	Hemrit-At ブートローダーイメージファイル
fdt	r8a7791-armadilloeva1500- <i>[version]</i> .dtb.gz	Device Tree Blob
firmware	squashfs-ae1500-firmware- <i>[version]</i> .img	Firmware パーティション用イメージファイル

Hemrit-At ブートローダーイメージファイルと Device Tree Blob は、Armadillo サイトまたは評価セット付属の DVD-ROM から取得可能です。Firmware パーティション用イメージファイルは、アットマークテクノ ユーザーズサイトから取得可能です。

Armadillo サイト - Armadillo-EVA 1500 ドキュメント・ダウンロード

<http://armadillo.atmark-techno.com/armadillo-eva-1500/downloads>

アットマークテクノ ユーザーズサイト

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



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」を行う必要があります。

### 11.4.2. ブートローダーのリカバリを実行する

SPI フラッシュのいくつかのパーティションは、デフォルト状態で書き込みが制限されています。このパーティションへの書き込みを行う場合は、事前に書き込み制限を外す必要があります。

表 11.10 パーティションのデフォルト状態での書き込み制限の有無と対応する MTD ノード

パーティション	書き込み制限	MTD クラスディレクトリ	MTD ブロックデバイスファイル
bootloader	あり	/sys/class/mtd/mtd1	/dev/mtdblock1
fdt	なし	/sys/class/mtd/mtd2	/dev/mtdblock2
firmware	あり	/sys/class/mtd/mtd3	/dev/mtdblock3

次の例では、「11.4.1. SPI フラッシュメモリのリカバリに必要なファイルの取得」で取得した Hemrit-At ブートローダーイメージファイルをカレントディレクトリに以下に配置し、bootloader パーティションをリカバリしています。

```
[armadillo ~]# ls
loader-armadilloeva1500-[version].bin
[armadillo ~]# echo 0 > /sys/class/mtd/mtd1/ro ①
[armadillo ~]# cat loader-armadilloeva1500-[version].bin > /dev/mtdblock1 ②
[armadillo ~]# sync
[armadillo ~]#
```

- ① MTD クラスディレクトリ以下にある ro に 0 を書き込み、書き込み制限を外します。書き込み制限の無い fdt パーティションをリカバリする場合は実行する必要はありません。
- ② MTD ブロックデバイスファイルに該当パーティション用のファイルを書き込みます。

図 11.7 SPI フラッシュメモリのリカバリ手順

## 12. JTAG ICE を利用する

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

### 12.1. 準備

JTAG ICE のケーブルを、JTAG インターフェース(CON1)に接続します。信号配列などの JTAG インターフェースについての詳細は、「15.2. CON1 JTAG インターフェース」を参照してください。

### 12.2. 機能選択スイッチの設定

機能選択スイッチの JTAG 設定を Coresight デバックポート(SW1.5 を OFF、SW1.6 を ON)に設定します。



機能選択スイッチの JTAG 設定を Coresight デバックポートに設定した場合は、CPU がシングルコアで動作します。CPU がシングルコアで動作している場合は、Linux カーネルの起動ログに次のように表示されます。

```
Unable to boot CPU1 when MD21 is set
CPU1: failed to boot: -524
```

### 12.3. 接続確認

「12.1. 準備」に従って設定されている場合に、CPU は次のように見えます。

項目	値
デバイス ID	0x4BA00477
コマンド長	4

### 12.4. 各種デバッガへの対応について

お使いのデバッガが Armadillo-EVA 1500 に対応しているか等の情報につきましては、各デバッガメーカーにお問い合わせください。

# 13. ハードウェア仕様

本章では、Armadillo-EVA 1500 のハードウェア仕様について説明します。

## 13.1. プロセッサ

ルネサスエレクトロニクス製アプリケーションプロセッサ「RZ/G1M」を搭載しています。

### 13.1.1. 特長

- ・ ARM Cortex-A15 デュアルコア 1.5GHz
- ・ L1 命令キャッシュ: 32KByte、L1 データキャッシュ: 32KByte、L2 キャッシュ: 1MByte
- ・ DDR3(L)-SDRAM(DDR3-1600)対応メモリコントローラ、データバス幅: 32bit x 2
- ・ PowerVR SGX544MP2(3D)
- ・ VCP3(H.264/AVC、MPEG-2/4、VC-1 等)
- ・ USB 3.0、USB 2.0 インターフェース
- ・ PCI Express インターフェース
- ・ シリアル ATA インターフェース
- ・ SD ホストインターフェース
- ・ その他各種周辺機能
- ・ 電源電圧: 3.3V、1.8V、1.5/1.35V、1.03V

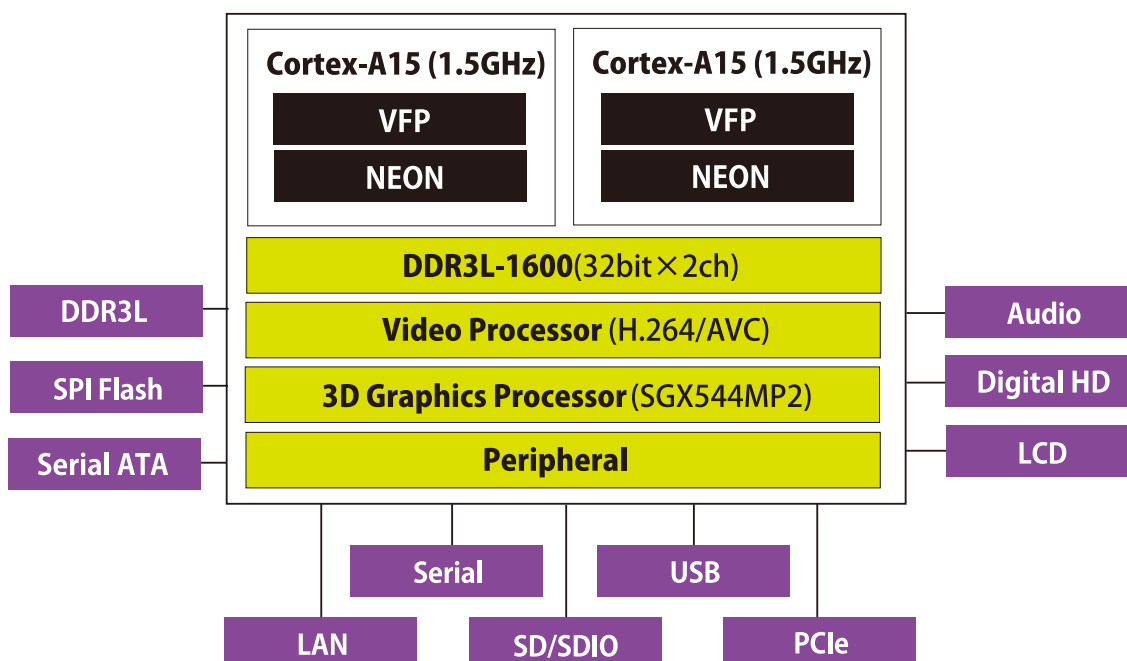


図 13.1 RZ/G1M 概要

### 13.1.2. モード設定

RZ/G1M の動作モードはパワーオンリセット解除時の MD ピンの状態により設定されます。

### 13.1.2.1. 動作モード

MD ピンで設定できる動作モードについて説明します<sup>[1]</sup>。動作モードの詳細については、「R8A77430HA01BG User's Manual Hardware」をご確認ください。

MD0 は自走モード、ステップアップモードの設定をします。

表 13.1 MD0 - 自走モード/ステップアップモードの設定

MD0	設定
0	自走モード
1	ステップアップモード

MD1、MD2、MD3 はブートデバイスの設定をします。

表 13.2 MD1、MD2、MD3 - ブートデバイスの設定

MD3	MD2	MD1	設定
0	0	0	エリア 0 Boot(CON2 に接続した外部メモリから起動)
0	1	0	QSPI Boot(48.75MHz/16KByte transfer)
1	0	0	QSPI Boot(39MHz/16KByte transfer)
1	1	0	QSPI Boot(39MHz/4KByte transfer)

MD4 は CS0 空間サイズを 64MByte 空間とするか 128MByte 空間とするかの設定をします。

表 13.3 MD4 - CS0 空間サイズ設定

MD4	設定
0	64MByte
1	128MByte

MD5 は予約されています。High レベルに固定する必要があります。

MD6、MD7 は Master boot processor の設定をします。

表 13.4 MD6、MD7 - Master boot processor の設定

MD7	MD6	設定
0	0	Cortex-A15 Boot

MD8 は CS0 空間のデータバス幅の設定をします。

表 13.5 MD8 - CS0 空間データバス幅の設定

MD8	設定
0	8bit バス
1	16bit バス

MD9 は EXTAL ピン、XTAL ピンに接続する源発振の設定をします。

表 13.6 MD9 - 水晶発振器/水晶発振子の設定

MD9	設定
0	EXTAL ピンに水晶発振器を接続
1	EXTAL/XTAL ピンに水晶振動子を接続

[1]"0": Low レベル入力、"1": High レベル入力、"-": 不定(ただし"0"か"1"どちらかに設定する必要あり)

MD10、MD11、MD12、MD20、MD21、MDT0、MDT1 は JTAG(CON1)でデバッグする機能を設定します。

**表 13.7 MD10、MD11、MD12、MD20、MD21、MDT0、MDT1 - JTAG の機能設定**

MD12	MD10	MD21	MD20	MD11	MDT1	MDT0	JTAG(CON1)
0	0	0	0	-	-	-	バウンダリスキャン
0	0	1	0	0	-	-	Coresight デバッグポート

MD13、MD14 は周波数モードの設定をします。

**表 13.8 MD13、MD14 - 周波数モードの設定**

MD14	MD13	設定
0	0	15MHz
0	1	20MHz
1	0	26MHz
1	1	30MHz

MD19 は DDR3-SDRAM のバスクロックの設定をします。

**表 13.9 MD19 - DDR3-SDRAM バスクロックの設定**

MD19	設定
0	DDR3-1600 mode
1	DDR3-1333 mode

MD22、MD27、MD28 は DDR3-SDRAM のインターフェース設定をします。

**表 13.10 MD22、MD27、MD28 - DDR3-SDRAM のインターフェース設定**

MD28	MD27	MD22	設定
0	1	1	DDR64bit x 1ch
1	1	1	DDR32bit x 2ch

MD23 は SATA0、USB3.0 のどちらを使用するかの設定をします。

**表 13.11 MD23 - SATA0/USB3.0 の設定**

MD23	設定
0	SATA0
1	USB3.0

MD24 は SATA1、PCIEC のどちらを使用するかの設定をします。

**表 13.12 MD24 - SATA1/PCIe の設定**

MD24	設定
0	SATA1
1	PCIEC

### 13.1.2.2. Armadillo-EVA 1500 の MD ピンの初期設定

Armadillo-EVA 1500 の MD ピンの出荷時の設定を「表 13.13. MD ピンの出荷時の設定」に示します。



表 13.13 MD ピンの出荷時の設定

MD ピン	出荷時の設定	設定方法	設定内容
MD0	0	SW1	自走モード
MD1	0	抵抗	QSPI(48.75MHz/16KByte transfer)
MD2	1	SW1	
MD3	0	SW1	
MD4	0	抵抗	CS0 空間サイズ 64MByte
MD5	1	抵抗	Reserved
MD6	0	抵抗	Cortex-A15 boot
MD7	0	抵抗	
MD8	1	抵抗	CS0 空間データバス幅 16bit
MD9	0	抵抗	水晶発振器
MD10	0	抵抗	バウンダリスキャン
MD11	0	SW1	
MD12	0	抵抗	
MD20	0	抵抗	
MD21	0	SW1	
MDT0	0	抵抗	
MDT1	1	抵抗	
MD13	1	抵抗	入力周波数 20MHz
MD14	0	抵抗	
MD19	0	抵抗	DDR3-1600 mode
MD22	1	抵抗	DDR32bit x 2ch
MD27	1	抵抗	
MD28	1	抵抗	
MD23	1	抵抗	USB 3.0
MD24	0	SW1	SATA1

## 13.2. RAM

4Gbit、16bit 幅の DDR3L-SDRAM(DDR3-1600)を 4 個搭載しています。RZ/G1M とは 32bit x 2 チャンネルモードで接続しています。

## 13.3. eMMC

8GByte の SAMSUNG 製 eMMC を搭載しています。eMMC は RZ/G1M の MMC(Multi Media Card Interface)と接続されています。

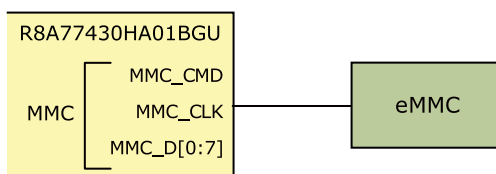


図 13.2 eMMC 周辺の構成

## 13.4. SPI フラッシュメモリ

8MByte の Micron 製 SPI フラッシュメモリを搭載しています。SPI フラッシュメモリはマルチプレクサを経由して RZ/G1M の QSPI(Quad Serial Peripheral Interface)と接続されています。QSPI は拡

張インターフェース(CON2)にも接続されており、SPI フラッシュメモリを使用する場合は、SW2.6 を ON に設定します。

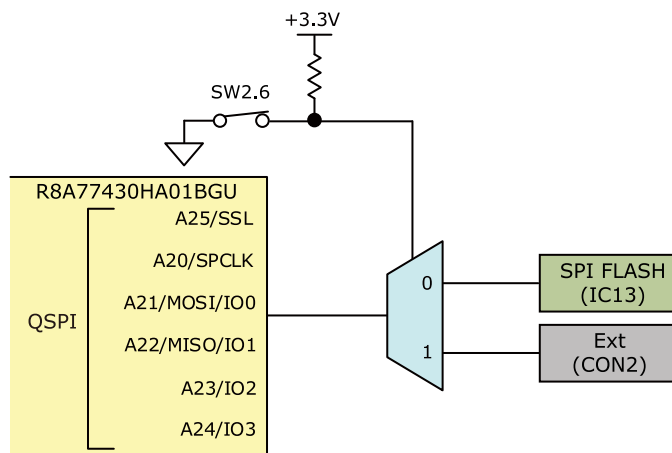


図 13.3 SPI フラッシュメモリ周辺の構成

### 13.5. LAN(Ethernet)

10BASE-T/100BASE-TX に対応した LAN インターフェース(CON7)を搭載しています。カテゴリ 5 以上の LAN ケーブルを接続することができます。信号線は Microchip Technology 製 PHY(LAN8720AI-CP)を経由して RZ/G1M の Ether(Ethernet MAC Controller)に接続されています。

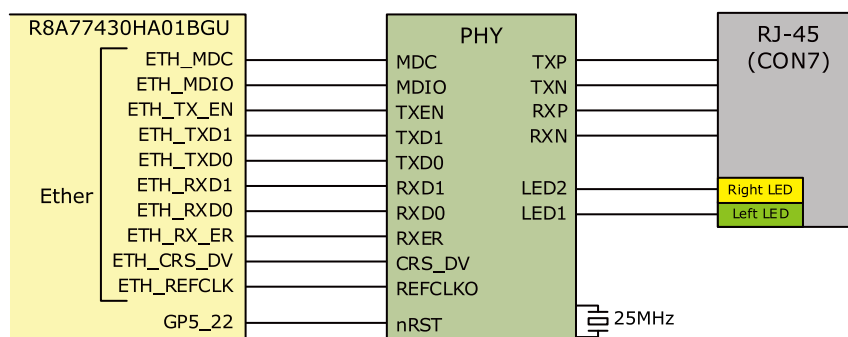


図 13.4 LAN 周辺の構成

表 13.14 LAN コネクタ LED

名称(色)	状態	説明
LINK_ACTIVE_LED(緑色)	消灯	リンクが確立されていない
	点灯	リンクが確立されている
	点滅	リンクが確立されておりキャリアを検出した状態
SPEED_LED(黄色)	消灯	10BASE-T
	点灯	100BASE-TX

### 13.6. 無線 LAN

Armadillo-WLAN(AWL13)用コネクタ(CON5)を搭載しています。信号線はマルチプレクサを経由して RZ/G1M の SDHI2(SD Card Host Interface)に接続されています。SDHI2 は SD インターフェー

ス(CON4)、拡張インターフェース(CON3)にも接続されており、CON5 を使用する場合は、SW2.5 を OFF に設定します。

CON5 に供給される電源は、GP6\_26 により制御しており、High で電源供給、Low で電源切断されます。信号線の電源は GP6\_27 により 3.3V/1.8V の切替をしており、CON5 を使用する場合は、3.3V(High)に設定する必要があります。

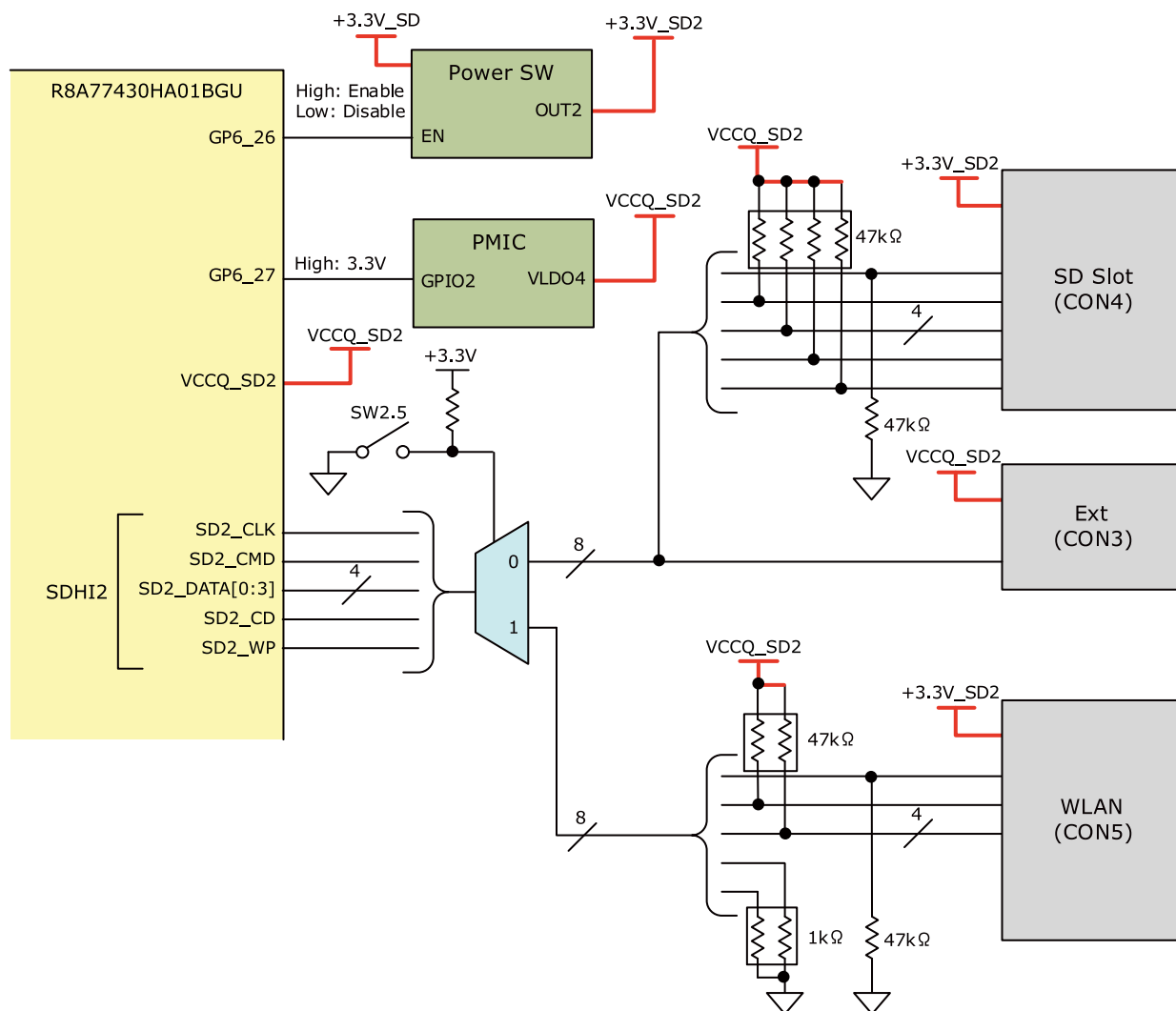


図 13.5 WLAN インターフェース(CON5)周辺の構成

## 13.7. シリアル(UART)

非同期(調歩同期)シリアルインターフェース(CON16)を搭載しています。信号線は Texas Instruments 製 RS232C レベル変換 IC(SN65C3232EPW)とマルチプレクサを経由して RZ/G1M の SCIFB0(Serial Communications Interface with FIFO B)に接続されています。SCIFB0 は拡張インターフェース(CON15)にも接続されており、CON16 を使用する場合は SW2.3 を ON に設定します。

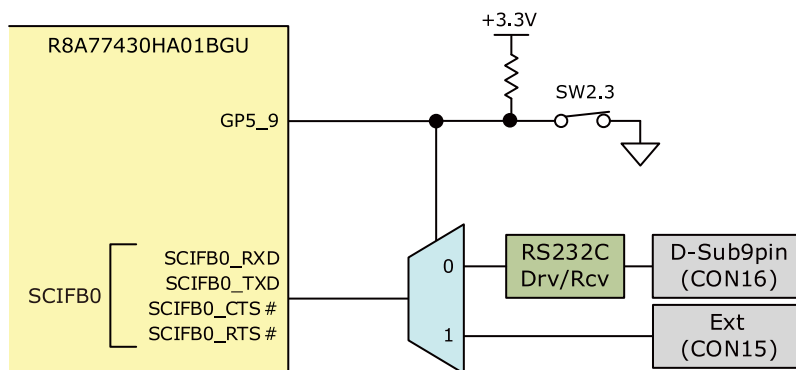


図 13.6 シリアル(CON16)周辺の構成

## 13.8. USB シリアル

非同期(調歩同期)の USB シリアルインターフェース(CON14)を搭載しています。信号線は FTDI 製 USB シリアル変換 IC(FT232RL-REEL)、バッファ IC を経由して RZ/G1M の SCIFB1(Serial Communications Interface with FIFO B)に接続されています。

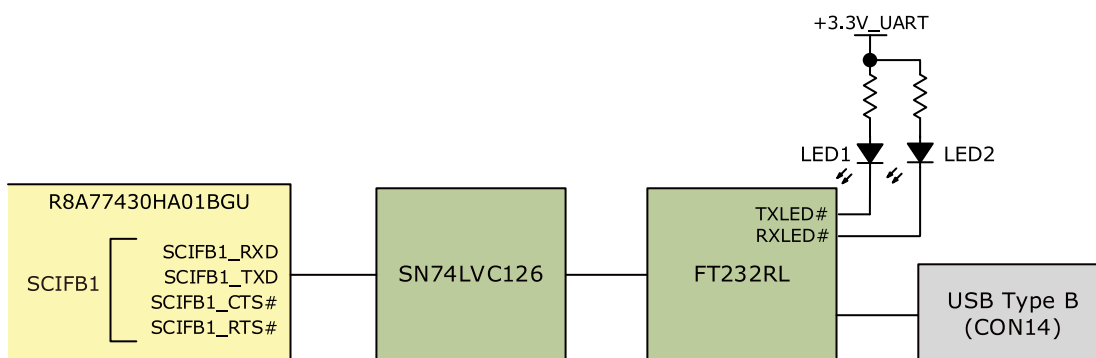


図 13.7 USB シリアル(CON14)周辺の構成

表 13.15 シリアル LED

部品番号(色)	状態	説明
LED1(黄)	点滅	データ送信中
LED2(黄)	点滅	データ受信中

## 13.9. SD

SD インターフェースを 2 つ(CON4、CON6)搭載しています。

### 13.9.1. CON4

CON4 は UHS-I(SDR50、最大クロック周波数: 97.5MHz)に対応した SD インターフェースです。信号線はマルチプレクサを経由して RZ/G1M の SDHI2(SD Card Host Interface)に接続されています。SDHI2 は WLAN インターフェース(CON5)にも接続されており、CON4 を使用する場合は、SW2.5 を

ON に設定します。CON4 の信号線は拡張インターフェース(CON3)にも接続されており、同時に使用することはできません。

SD スロットに供給される電源は、GP6\_26 により制御しており、High で電源供給、Low で電源切断されます。信号線の電源は GP6\_27 により 3.3V/1.8V の切替をしており、High で 3.3V、Low で 1.8V が供給されます。

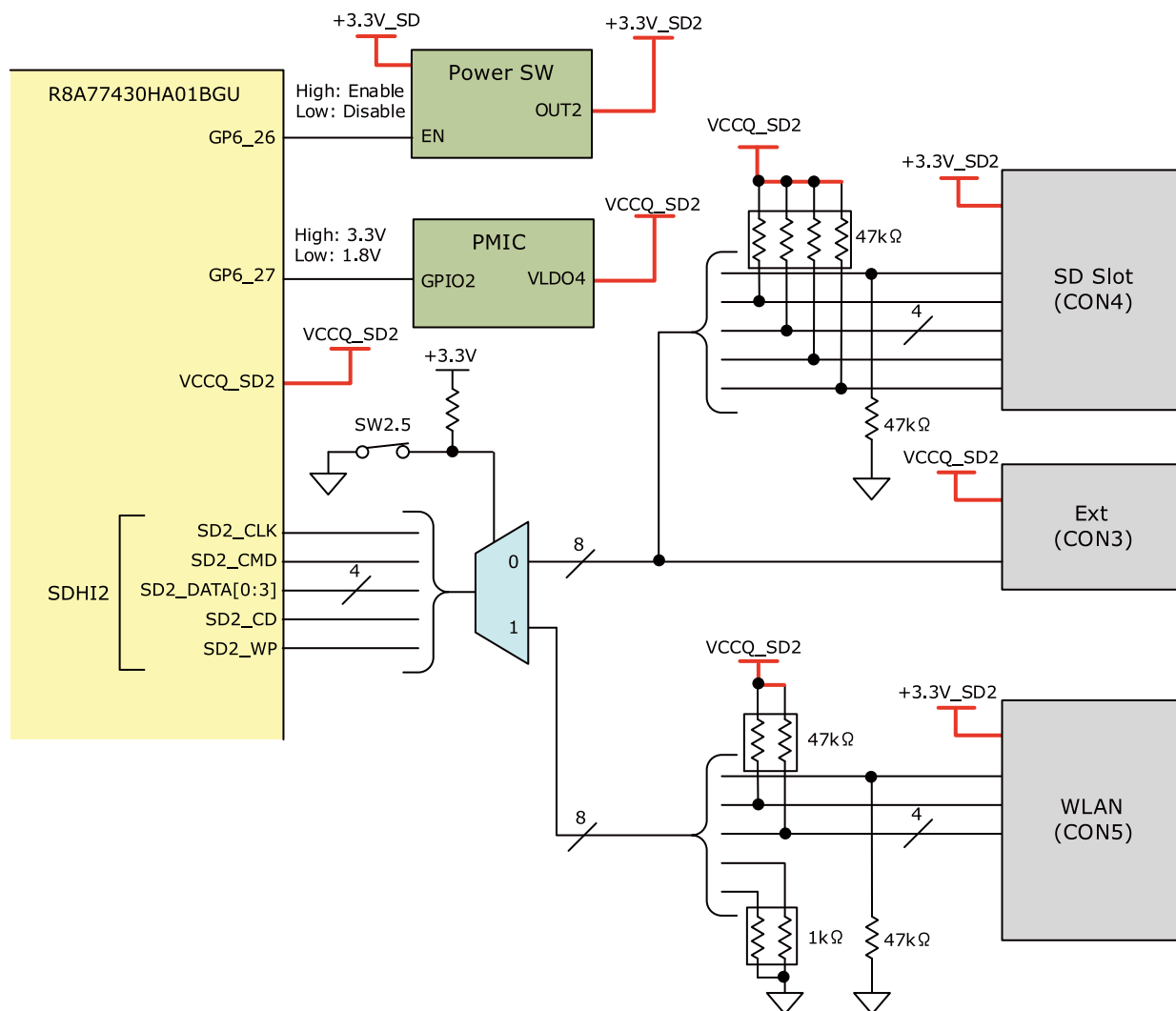


図 13.8 SD スロット(CON4)周辺の構成

### 13.9.2. CON6

CON6 は UHS-I(SDR104、最大クロック周波数: 156MHz)に対応した SD インターフェースです。信号線は RZ/G1M の SDHI0(SD Card Host Interface)に接続されています。SD スロットに供給される電源は、GP6\_24 により制御しており、High で電源供給、Low で電源切断されます。信号線の電源は GP6\_25 により 3.3V/1.8V の切替をしており、High で 3.3V、Low で 1.8V が供給されます。

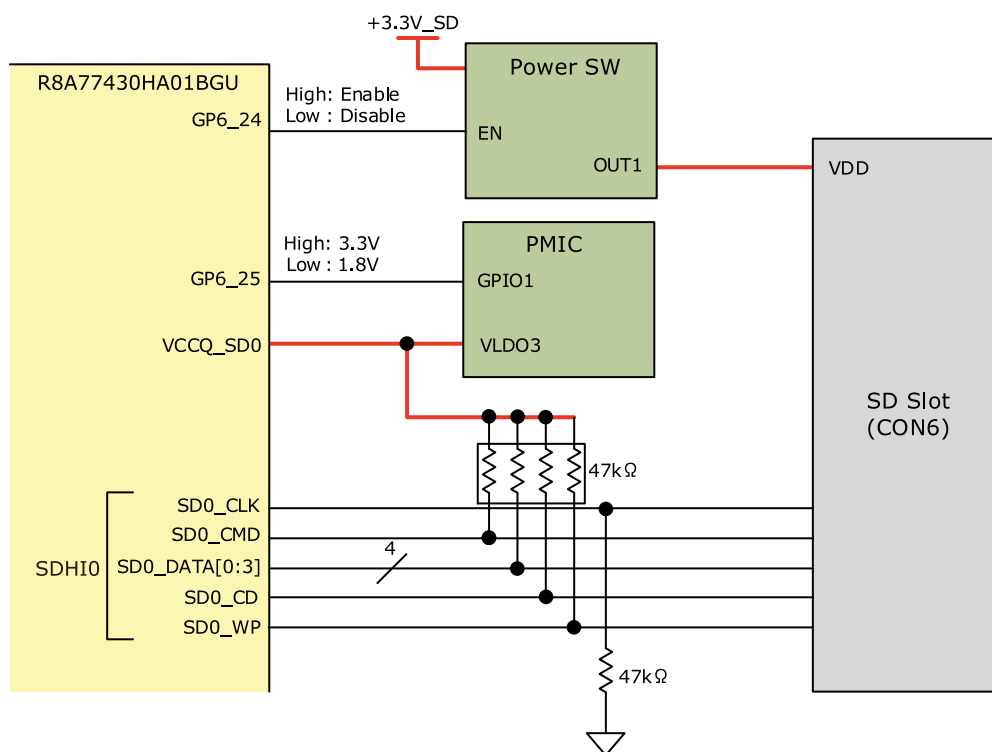


図 13.9 SD スロット (CON6) 周辺の構成

## 13.10. USB 2.0

USB 2.0 ホストインターフェース (CON8) と USB 2.0 ファンクションインターフェース (CON9) を搭載しています。信号線はマルチプレクサを経由して RZ/G1M の USB0 (USB 2.0 Controller) に接続されています。USB0 の接続先は、SW2.4 により切替可能で、ON で CON8、OFF で CON9 に接続されます。CON8 の電源は USB0\_PWEN 信号で制御をしており、High で電源供給、Low で電源切断されます。

### データ転送モード

- High Speed (480Mbps)
- Full Speed (12Mbps)
- Low Speed (1.5Mbps)

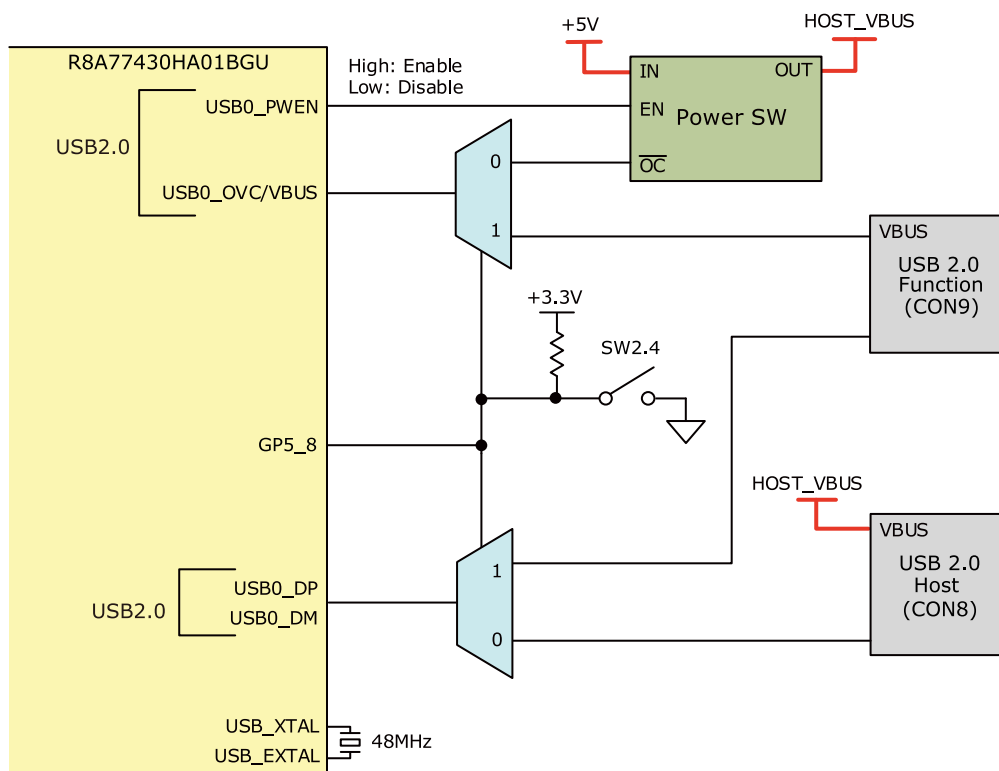


図 13.10 USB 2.0 周辺の構成

## 13.11. USB 3.0

USB 3.0 ホストインターフェース(CON10)を搭載しています。信号線は RZ/G1M の USB3(USB 3.0 Controller)、USB1(USB 2.0 Controller)に接続されています。USB1 の信号線は miniPCle インターフェース(CON12)にも接続されており、CON10 を使用する場合は、SW1.1 を ON に設定します。VBUS に供給する電源は USB1\_PWEN 信号で制御しており、High で電源供給、Low で電源切断されます。

### データ転送モード

- ・ Super Speed(5Gbps)
- ・ High Speed(480Mbps)
- ・ Full Speed(12Mbps)
- ・ Low Speed(1.5Mbps)

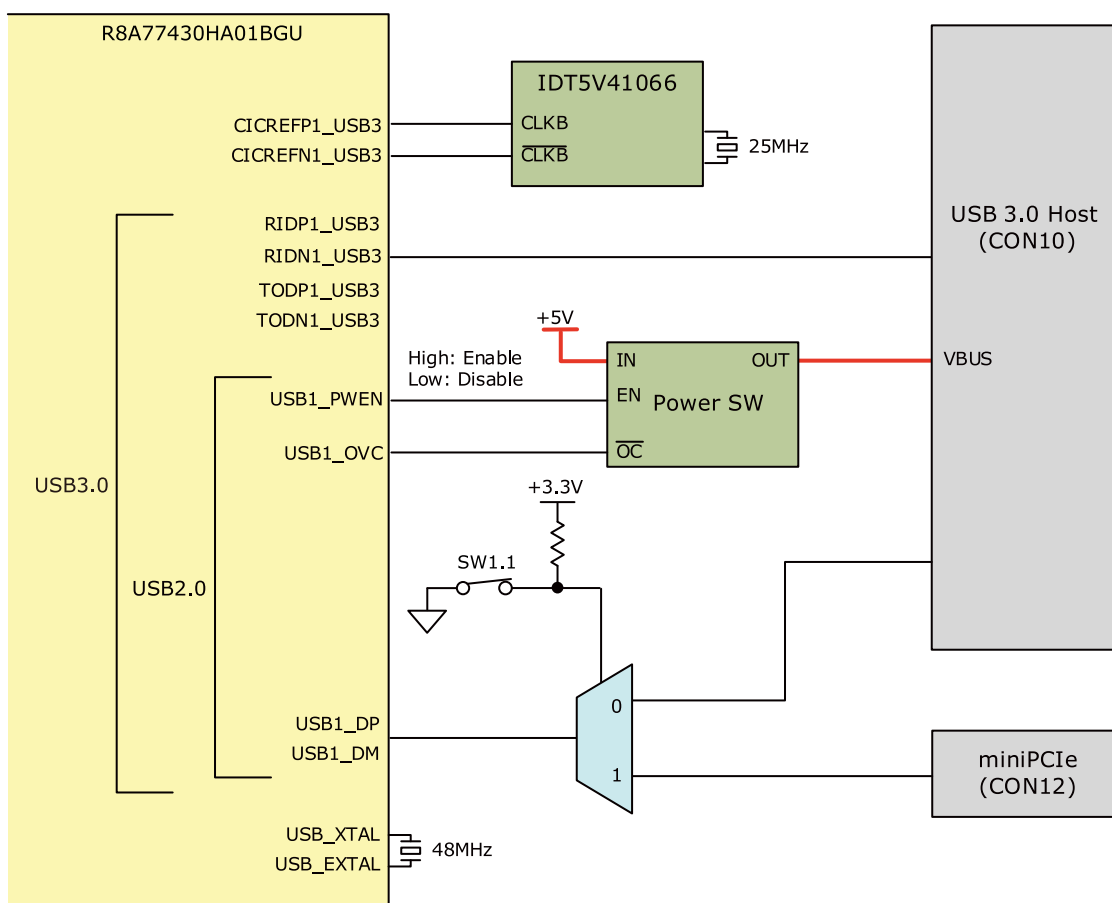


図 13.11 USB 3.0 周辺の構成

## 13.12. PCI Express

miniPCIe インターフェース(CON12)を搭載しています。信号線はマルチプレクサを経由してRZ/G1MのPCIEC(PCI-Express Controller)、USB1(USB 2.0 Controller)に接続されています。PCIECとUSB1の信号線はそれぞれSATAインターフェース(CON11)、USB 3.0 ホストインターフェース(CON10)にも接続されており、CON12を使用する場合、SW1.1、SW1.2をOFFに設定します。また、MDピンを設定する必要があり、SW1.3をOFFに設定します。RZ/G1M内蔵のPCIECはPCI Express Base Specification Rev2.0(5.0GT/s、2.5GT/s)に対応しています。



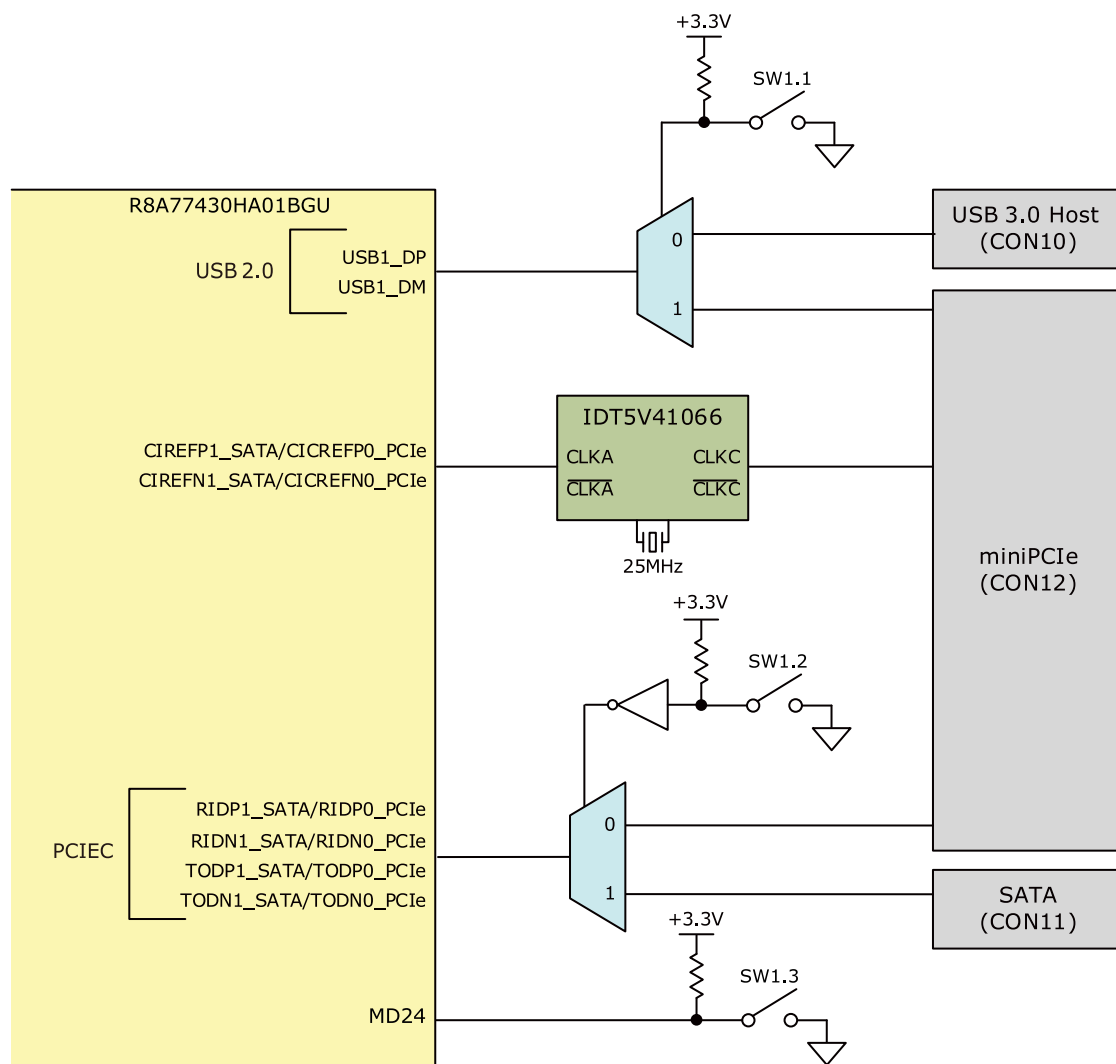



図 13.12 PCI Express 周辺の構成

 差動配線長差を抑えるため、D+/D-のスイッチ機能を使い、TODNO\_PCIe と TODPO\_PCIe を入れ替えて接続しています。

### 13.13. シリアル ATA

シリアル ATA インターフェース(CON11)を搭載しています。信号線はマルチプレクサを経由して、RZ/G1M の Serial-ATA インターフェースに接続されています。miniPCIe インターフェース(CON12)にも共通の信号線が接続されており、CON11 を使用する場合は、SW1.2 を OFF に設定します。また、MD ピンを設定する必要があり、SW1.3 を ON に設定します。RZ/G1M 内蔵の Serial-ATA インターフェースは、SATA Revision 3.1 に準拠し、1.5Gbps および 3.0Gbps の転送速度に対応しています。SATA デバイスに電源を供給する場合、CON13 を使用することが可能です。

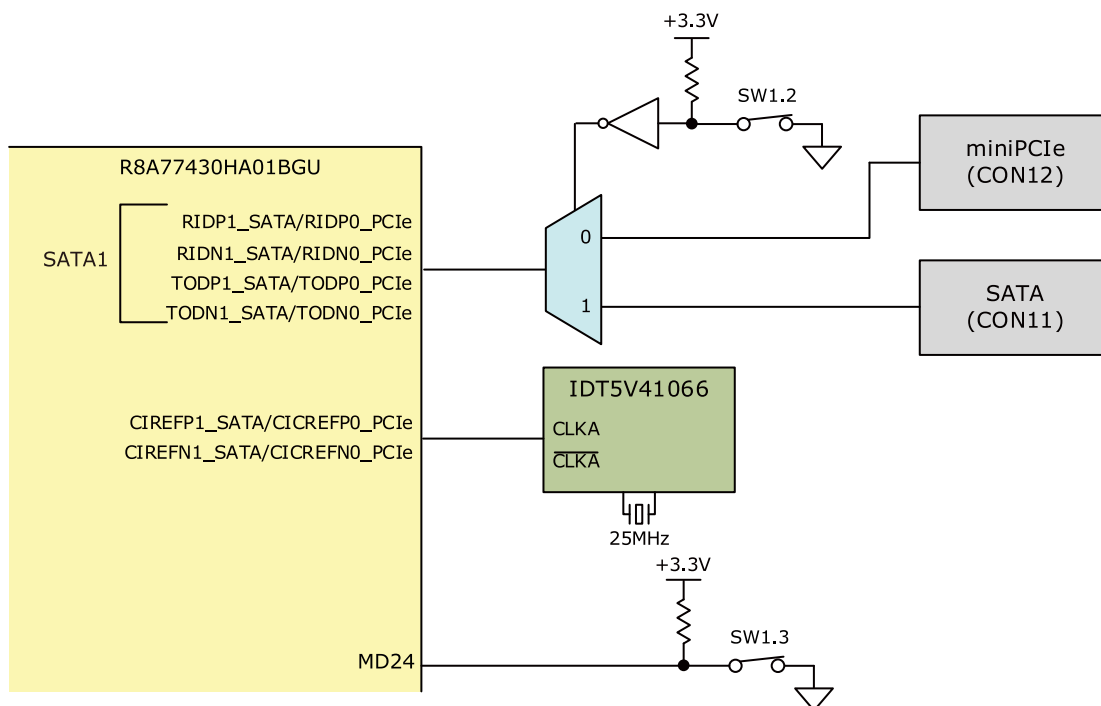


図 13.13 シリアル ATA 周辺の構成

### 13.14. コンポジットビデオ入力

コンポジットビデオ入力用コネクタ(CON26)を搭載しています。信号線はビデオコーデック(ADV7180/ Analog Devices)、マルチプレクサを経由してRZ/G1MのVIN2(Video Input Module)に接続されています。VIN2は拡張インターフェース(CON30)にも接続されており、CON26で使用する場合、SW2.1をONに設定します。

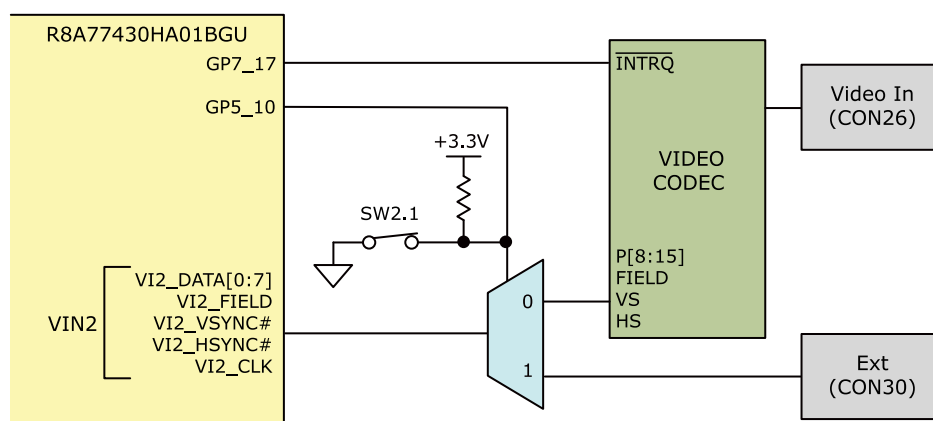


図 13.14 コンポジットビデオ入力周辺の構成

### 13.15. LCD

Ampire 製タッチパネル LCD(AM-800480L1TMQW-T00H7)用のコネクタ(CON23)を搭載しています。信号線はRZ/G1MのDU1(Digital RGB)に接続されています。

HDMI トランスミッター、拡張インターフェース(CON24)にも共通の信号線が接続されており、同時に使用することはできません。

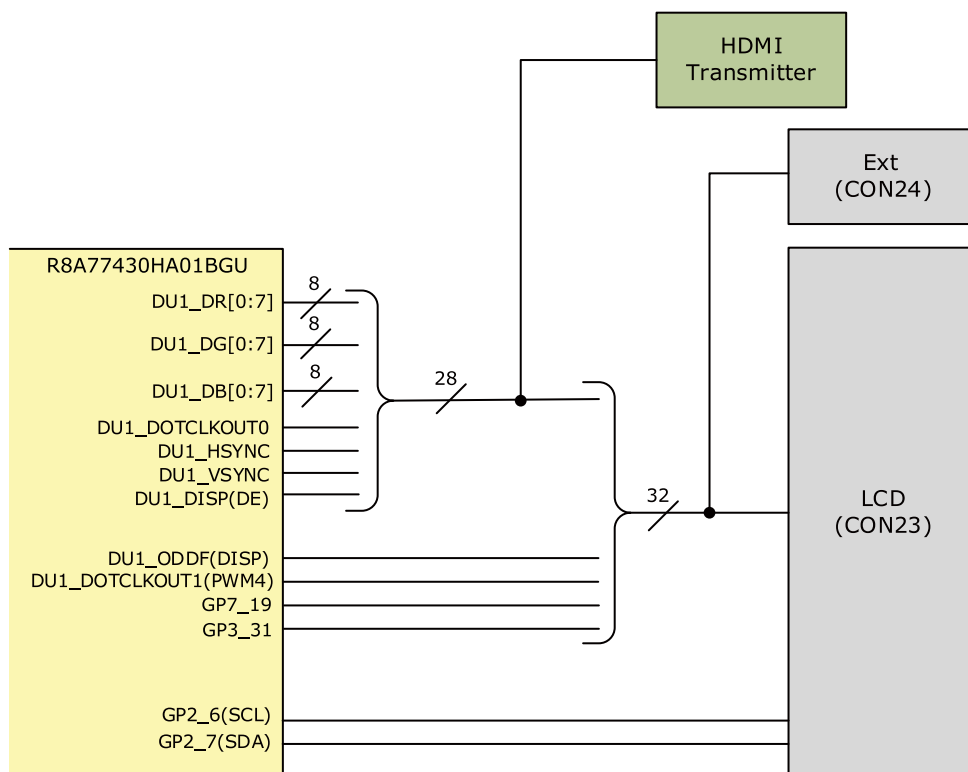


図 13.15 LCD 周辺の構成



CON23 に接続可能なタッチパネル LCD と取り付け用のネジ類をひとまとめにしたオプションセットを販売しています。詳細については、「17.1. Armadillo-EVA 1500 LCD オプションセット」をご確認ください。

## 13.16. LVDS

LVDS インターフェース(CON25)を搭載しています。信号線は RZ/G1M の DU0(LVDS インターフェース)に接続されています。ドットクロック用として、DU1\_DOTCLKIN に 74.25MHz の発振器を、DU0\_DOTCLKIN に 2.5 x 2.0mm パッケージ用のパターンを設けています。

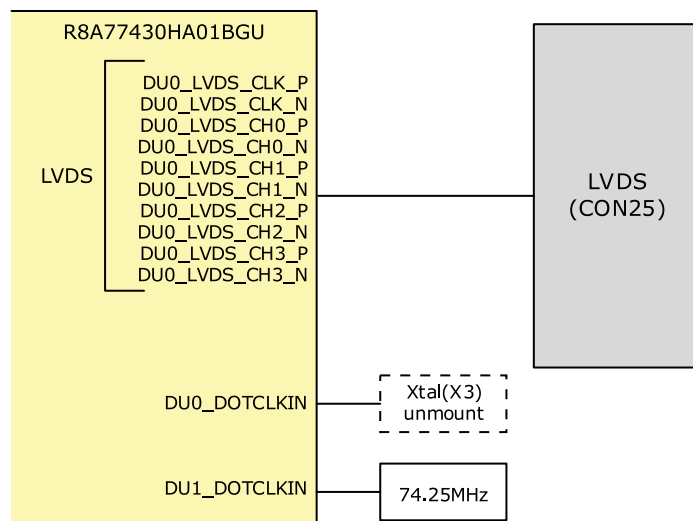


図 13.16 LVDS 周辺の構成

## 13.17. デジタル HD 出力

デジタル HD 出力インターフェース(CON17)を搭載しています。信号線は Analog Devices 製の HDMI トランスミッター(ADV7511W)を経由して RZ/G1M の DU1(Digital RGB)、SSI0、1、2、9(Serial Sound Interface)に接続されています。LCD インターフェース(CON23)、拡張インターフェース(CON24)にも共通の信号線が接続されており、同時に使用することはできません。

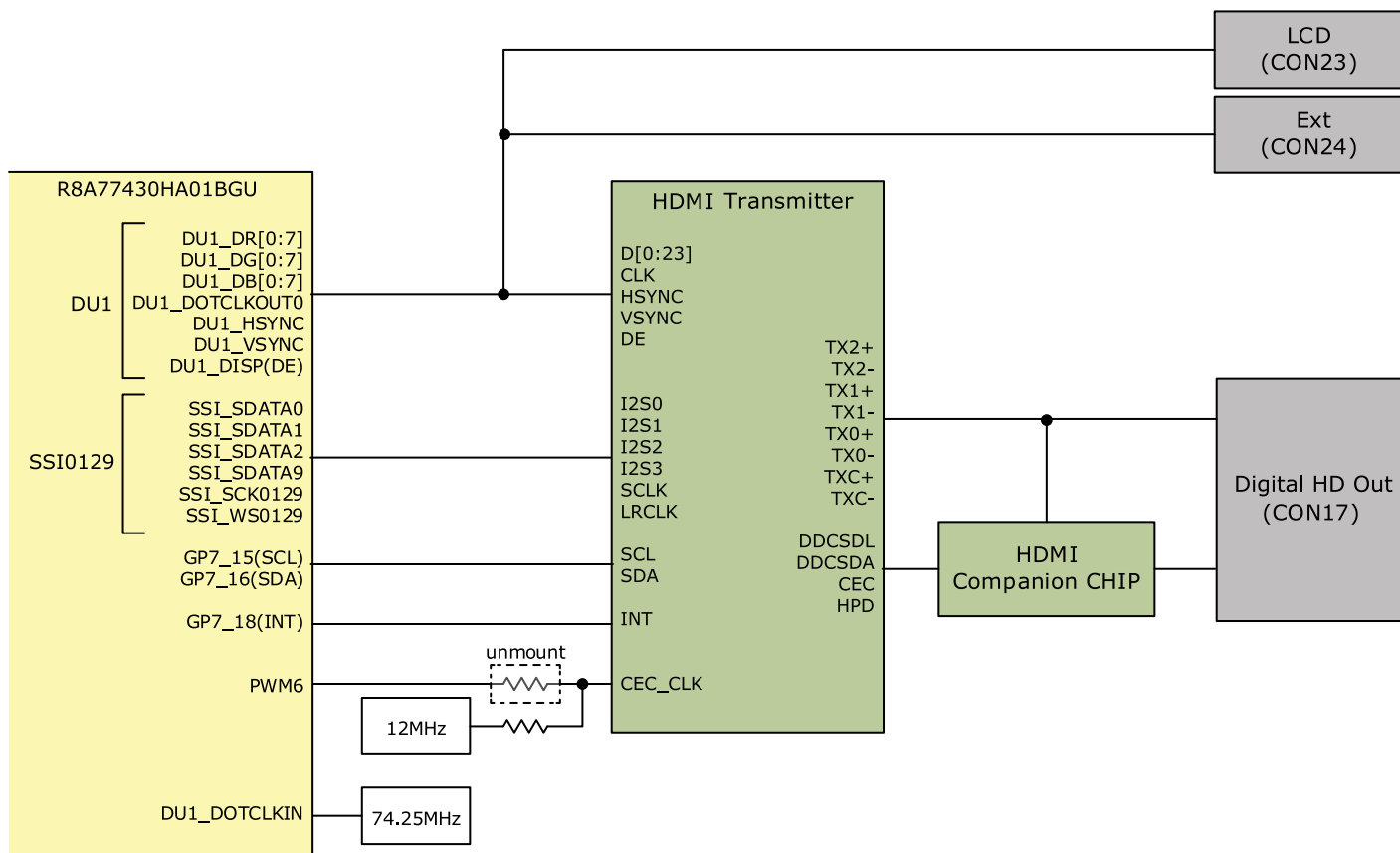


図 13.17 Digital HD 周辺の構成

## 13.18. オーディオ

ヘッドホン用ミニジャック(CON18)、マイク/ライン入力用ミニジャック(CON21)を搭載しています。信号線はマルチプレクサ、オーディオコーデック(CS42L52-CNZ/Cirrus Logic)を経由してRZ/G1MのSSI3、4(Serial Sound Interface Unit)に接続されています。CON22の8ピンにマルチプレクサのセレクトピンが接続されており、Highレベル(+3.3V)入力でCON22、Open(基板上にて10kΩ抵抗でGNDに接続されています)でオーディオコーデックに切替可能です。

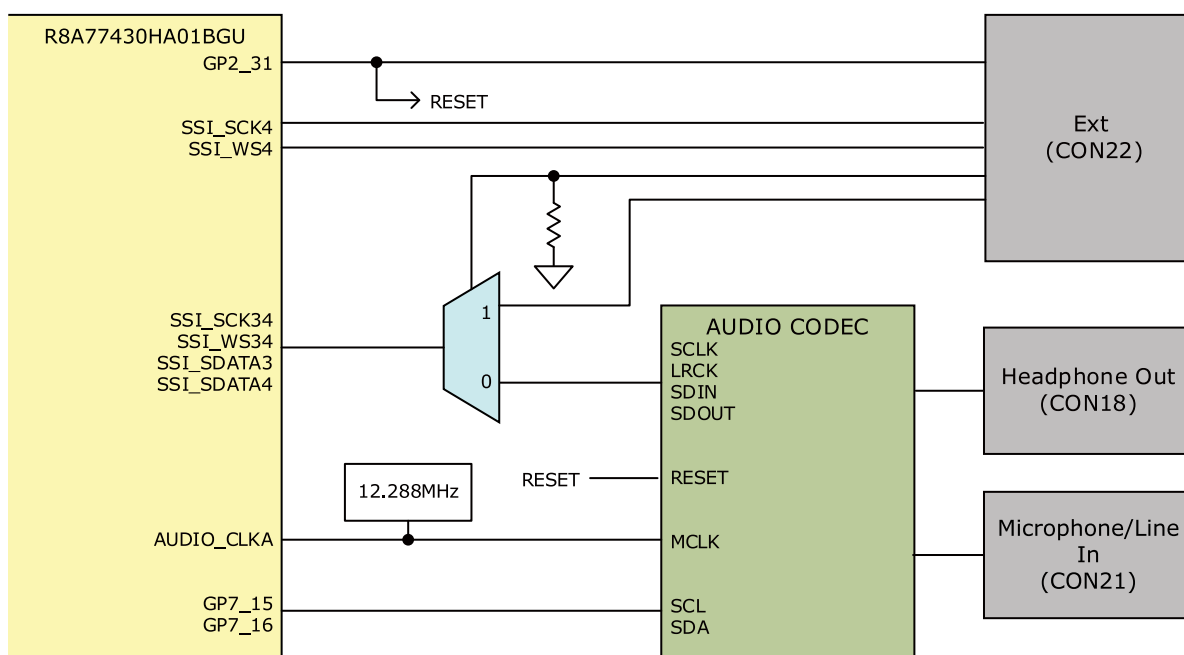


図 13.18 AUDIO CODEC 周辺の構成

バイアスジャンパ(JP2、JP3)をショートすることにより、バイアス電圧を供給することが可能です。ライン入力、バイアスを必要としないマイク入力を使用する場合は、ジャンパは必要ありません。

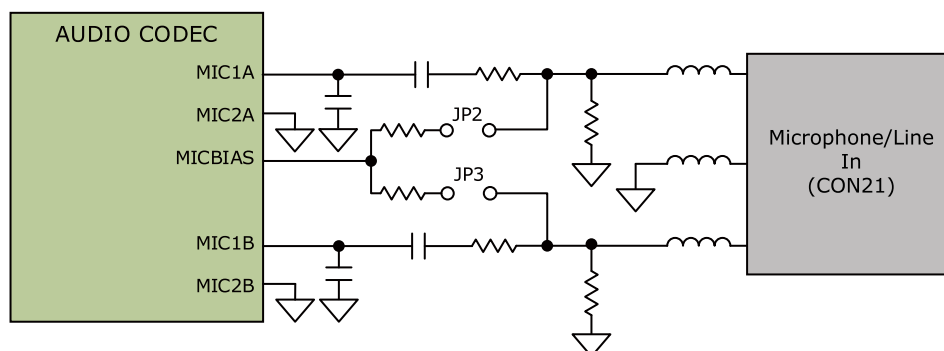


図 13.19 JP2、JP3 周辺回路

## 13.19. 拡張インターフェース

拡張インターフェースを 6 つ(CON2、CON3、CON15、CON22、CON24、CON30)搭載しています。拡張インターフェースには複数の機能が割り当てられた信号を配線しています。

### 13.19.1. CON2

CON2 の信号線は RZ/G1M の LBSC(Local Bus State controller)等に接続されており、SRAM インターフェース、パラレル ATA インターフェースに対応したバスを拡張することが可能です。LBSC の上位アドレス(A20~A25)のピンは SPI フラッシュメモリにも接続されており、CON2 で使用する場合は SW2.6 を OFF に設定します。

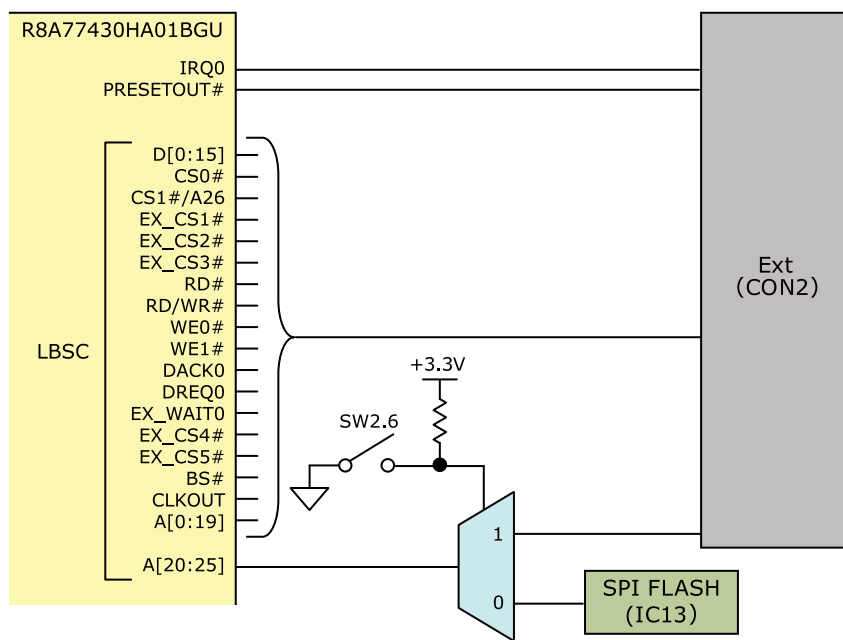


図 13.20 CON2 周辺の構成

### 13.19.2. CON3

CON3 の信号線はマルチプレクサを経由して RZ/G1M の SDHI2(SD Card Host Interface)に接続されています。CON3 を使用する場合は、SW2.5 を ON に設定します。CON3 の信号線は SD インターフェース(CON4)にも接続されており、同時に使用することはできません。

CON3 に供給される電源および信号線の電源は GP6\_27 により 3.3V/1.8V の切替をしており、High で 3.3V、Low で 1.8V が供給されます。

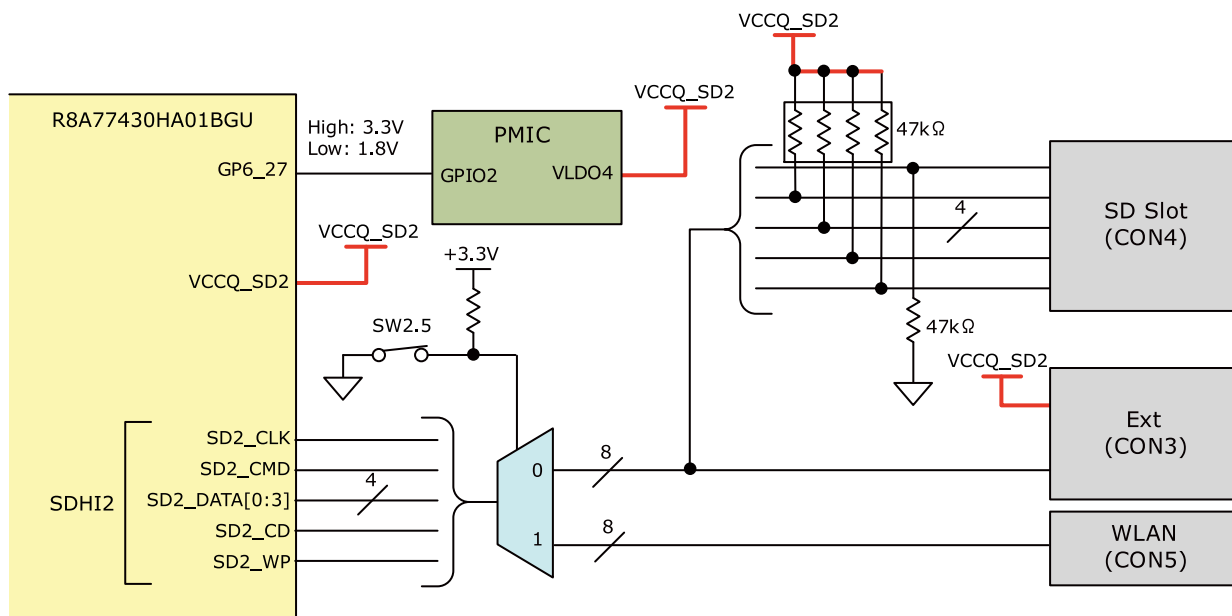


図 13.21 CON3 周辺の構成

### 13.19.3. CON15

CON15 の信号線は RZ/G1M の CAN1 (Controller Area Network Interface) とマルチプレクサを経由して RZ/G1M の SCIFB0 (Serial Communications Interface with FIFO B) に接続されています。CON15 を使用する場合は、SW2.3 を OFF に設定します。

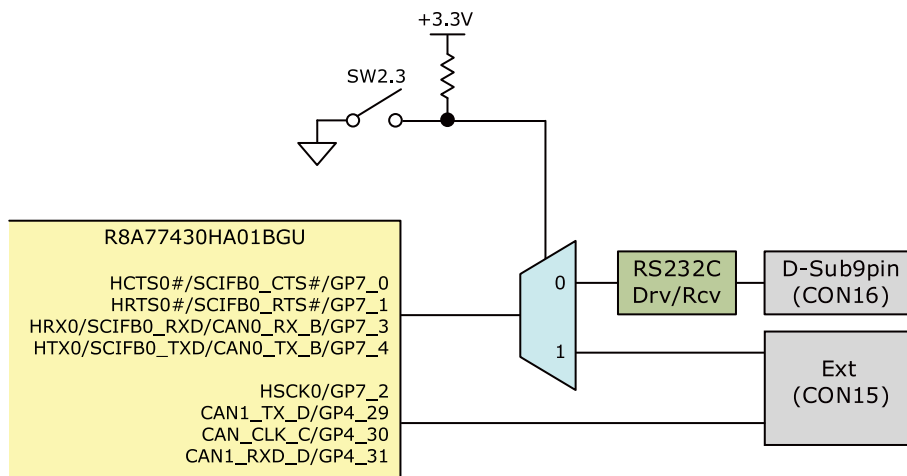


図 13.22 CON15 周辺の構成

### 13.19.4. CON22

CON22 の信号線は RZ/G1M の SSI3、4 (Serial Sound Interface) に接続されており、音声の入出力インターフェース等を拡張可能です。一部の信号線はマルチプレクサを経由して RZ/G1M に接続されており、CON22 で使用する場合は、CON22 の 8 ピンに High レベル (+3.3V) 入力します。



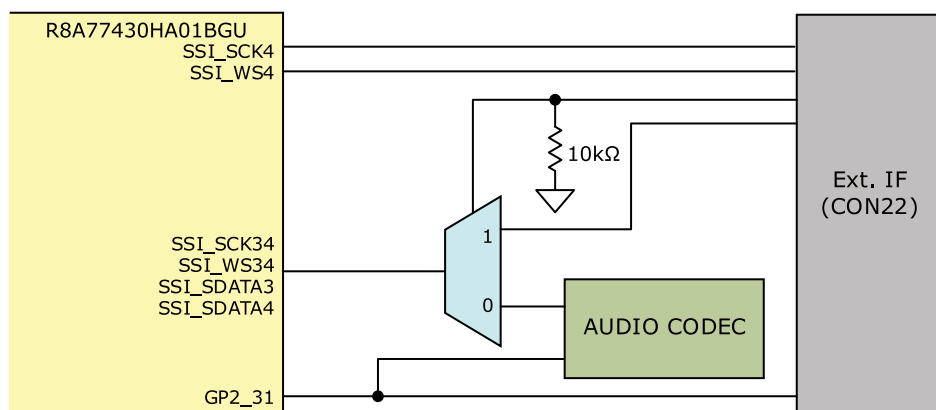


図 13.23 CON22 周辺の構成

### 13.19.5. CON24

CON24 の信号線は RZ/G1M の DU1 (Digital RGB) に接続されており、LCD 等を接続することが可能です。

HDMI トランスミッター、LCD インターフェース (CON23) にも共通の信号線が接続されており、同時に使用することはできません。

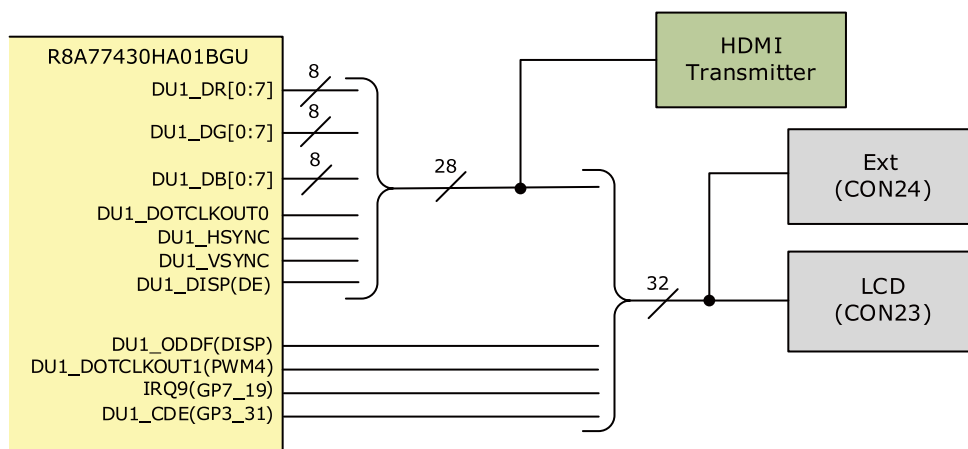


図 13.24 CON24 周辺の構成

### 13.19.6. CON30

CON30 の信号線は主に RZ/G1M の VINO (Video Input Module) に接続されており、カメラ等を接続することが可能です。一部の信号線はマルチプレクサを経由して接続されており、CON30 で使用する場合、SW2.1 を OFF に設定します。

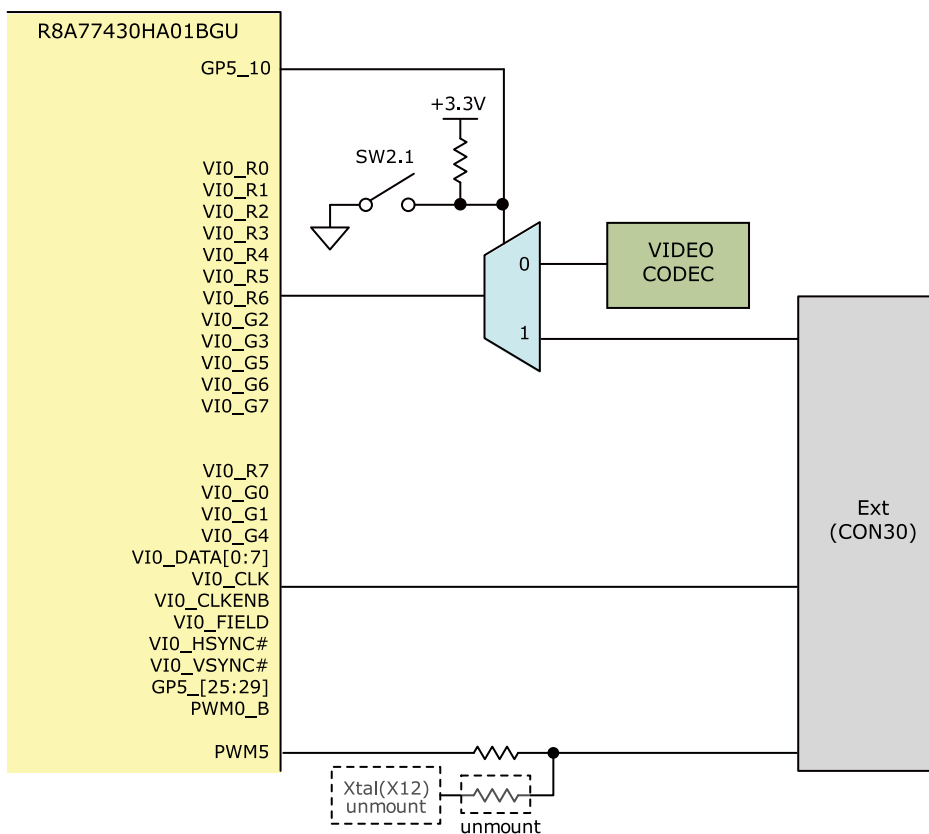


図 13.25 CON30 周辺の構成

## 13.20. I2C

I2C6、GPIO\_I2C0、GPIO\_I2C1、GPIO\_I2C2 バスに接続されているデバイス、インターフェースは次のとおりです。

表 13.16 I2C バスの接続

I2C バス	接続
I2C6	PMIC(DA9063)
	PMIC(DA9210)
GPIO_I2C0	PMIC(DA9063)
	RTC(S-35390A)
GPIO_I2C1	EEPROM(M24C01)
	HDMI トランスミッター(ADV7511)
	オーディオコーデック(CS42L52)
GPIO_I2C2	ビデオデコーダ(ADV7180)
	miniPCIe インターフェース(CON12)
	LCD インターフェース(CON23)

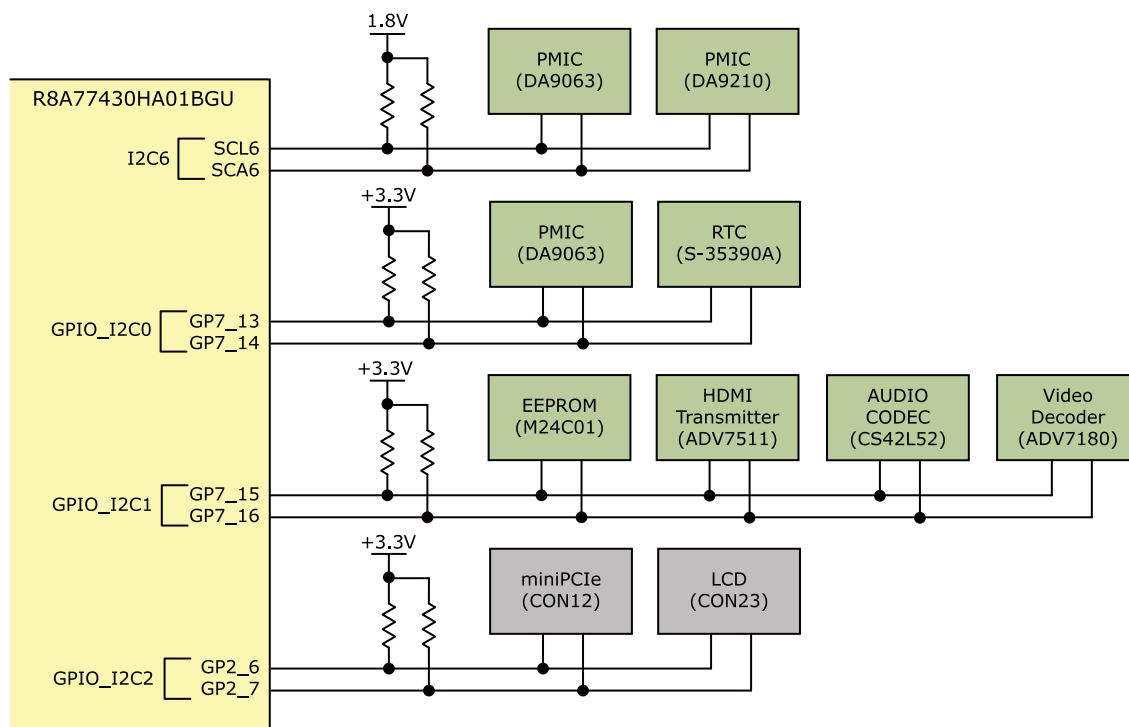


図 13.26 I2C 周辺の構成

## 13.21. クロック

次の水晶発振器、水晶振動子を使用しています。

表 13.17 水晶発振器、水晶振動子

部品番号	周波数	接続先	備考
X1	14.7456MHz	RZ/G1M(SCIF_CLK)	
X2	74.25MHz	RZ/G1M(DU1_DOTCLKIN)	
X3	-	RZ/G1M(DU0_DOTCLKIN)	未実装
X4	20MHz	RZ/G1M(XTAL、EXTAL)	未実装、X5 と排他で使用
X5	20MHz	RZ/G1M(EXTAL)	
X6	48MHz	RZ/G1M(USB_XTAL、USB_EXTAL)	
X7	25MHz	Ethernet PHY	
X8	25MHz	miniPCIe/SATA/USB3.0 用クロックジェネレータ	
X9	12MHz	HDMI トランスミッター	
X10	12.288MHz	オーディオコーデック、RZ/G1M(AUDIO_CLKA)	
X11	28.6363MHz	ビデオデコーダ	
X12	-	拡張インターフェース(CON30)	未実装、PWM5 と排他で使用
X13	32.768kHz	リアルタイムクロック	
X14	32.768kHz	PMIC	

## 13.22. スイッチ

機能選択用のディップスイッチ、ユーザーが自由に使用できるタクトスイッチ、電源を ON/OFF するためのスライドスイッチ、リセット用のタクトスイッチ、SD ブートデバイス選択用のタクトスイッチを搭載しています。

### 13.22.1. 機能選択スイッチ

機能選択用に 8 接点のディップスイッチを 2 つ(SW1、SW2)搭載しています。RZ/G1M の MD ピン、マルチプレクサのセレクトピン等に接続されており、有効にする機能の選択をします。

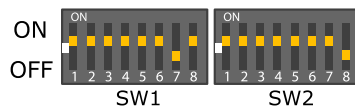


図 13.27 機能選択スイッチの形状

表 13.18 機能選択スイッチの機能

部品番号	No.	機能
SW1	1	USB 3.0、SATA/miniPCle 設定
	2	
	3	
	4	自走モード/ステップアップモード設定
	5	JTAG 設定
	6	
	7	IPL 起動デバイス設定
	8	
SW2	1	ビデオ入力設定
	2	Reserved
	3	シリアル(UART)設定
	4	USB 2.0 設定
	5	SD/WLAN 設定
	6	SPI 設定
	7	Hermit-At 起動デバイス設定
	8	Hermit-At 起動モード設定

#### 13.22.1.1. USB 3.0、SATA/miniPCle の設定

SW1.1、SW1.2、SW1.3 は RZ/G1M の MD24 ピン、マルチプレクサのセレクトピンに接続されており、USB 3.0、SATA、miniPCle インターフェースの有効/無効を設定します。

表 13.19 SW1.1、SW1.2、SW1.3 - USB 3.0、SATA/miniPCle の設定

SW1			機能		
1	2	3	USB 3.0	SATA	miniPCle
ON			有効		無効
OFF			無効		有効
その他			設定禁止		

#### 13.22.1.2. 自走モード/ステップアップモード設定

SW1.4 は RZ/G1M の MD0 に接続されており、RZ/G1M が自走モードで動作するかステップアップモードで動作するかを設定します。

表 13.20 SW1.4 - 自走モード/ステップアップモード設定

SW1	機能
4	
ON	自走モード
OFF	ステップアップモード

### 13.22.1.3. JTAG 設定

SW1.5、SW1.6 は RZ/G1M の MD11、MD21 ピンに接続されており、デバッグ機能を設定します。

表 13.21 SW1.5、SW1.6 - JTAG 設定

SW1		機能
5	6	
ON	ON	バウンダリスキャン
ON	OFF	
OFF	ON	Coresight デバッグポート
OFF	OFF	設定禁止

### 13.22.1.4. IPL 起動デバイス設定

SW1.7、SW1.8 は RZ/G1M の MD2、MD3 ピンに接続されており、IPL の起動デバイスを設定します。

表 13.22 SW1.7、SW1.8 - IPL 起動デバイス設定

SW1		機能
7	8	
ON	ON	エリア 0(CS0 に割り当てられたデバイス)から起動
OFF	ON	SPI フラッシュメモリから起動(48.75MHz/16KByte transfer)
ON	OFF	SPI フラッシュメモリから起動(39MHz/16KByte transfer)
OFF	OFF	SPI フラッシュメモリから起動(39MHz/4KByte transfer) <sup>[a]</sup>

[a]IPL は 4KByte 以上のサイズですので、起動できません。

### 13.22.1.5. ビデオ入力設定

SW2.1 は VIN2(Video Input Module)の接続先を切替するマルチプレクサのセレクトピンに接続されています。

表 13.23 SW2.1 - ビデオ入力設定

SW2	機能
1	
ON	ビデオデコーダに接続 コンポジットビデオ入力インターフェース(CON26)有効
OFF	拡張インターフェース(CON30)に接続

### 13.22.1.6. シリアル(UART)設定

SW2.3 は SCIFB0(Serial Communications Interface with FIFO B)の接続先を切替するマルチプレクサのセレクトピンに接続されています。

表 13.24 SW2.3 - シリアル(UART)設定

SW2	機能
3	
ON	RS232C レベル変換 IC に接続 シリアルインターフェース(CON16)有効
OFF	拡張インターフェース(CON15)に接続

### 13.22.1.7. USB 2.0 設定

SW2.4 は USB0(USB 2.0 Controller)の接続先を切替するマルチプレクサのセレクトピンに接続されています。

表 13.25 SW2.4 - USB 2.0 設定

SW2	機能
4	
ON	USB 2.0 ホストインターフェース(CON8)有効
OFF	USB 2.0 ファンクションインターフェース(CON9)有効

### 13.22.1.8. SD/WLAN 設定

SW2.5 は SDHI2(SD Card Host Interface)の接続先を切替するマルチプレクサのセレクトピンに接続されています。

表 13.26 SW2.5 - SD/WLAN 設定

SW2	機能
5	
ON	SD インターフェース(CON4)有効
OFF	WLAN インターフェース(CON5)有効

### 13.22.1.9. SPI 設定

SW2.6 は QSPI(Quad Serial Peripheral Interface)の接続先を切替するマルチプレクサのセレクトピンに接続されています。

表 13.27 SW2.6 - SPI 設定

SW2	機能
6	
ON	SPI フラッシュメモリ有効
OFF	拡張インターフェース(CON2)の 57~62 ピン有効

### 13.22.1.10. Hermit-At 起動デバイス設定

SW2.7 は RZ/G1M の GP2\_3 に接続されています。Hermit-At を起動させるデバイスを設定します。

表 13.28 SW2.7 - Hermit-At 起動デバイス設定

SW2	機能
7	
ON	SPI フラッシュメモリから起動
OFF	SD カードから起動

### 13.22.1.11. Hermit-At 起動モード設定

SW2.8 は RZ/G1M の GP7\_7 に接続されています。Hermit-At 起動後、保守モードと OS 自動起動モードのどちらに遷移させるかを設定します。

表 13.29 SW2.8 - Hermit-At 起動デバイス設定

SW2	機能
8	
ON	保守モード
OFF	OS 自動起動モード

### 13.22.2. ユーザースイッチ

ユーザー側で自由に利用できるタクトスイッチを 4 つ(SW3、SW4、SW5、SW6)搭載しています。信号線は RZ/G1M の GP5\_0、GP5\_1、GP5\_2、GP5\_3 に接続されています。スイッチが押されている時は Low レベル、押されていない時は High レベルが各ピンに入力されます。

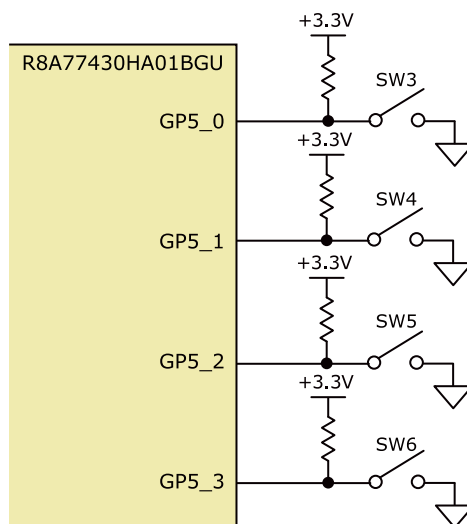


図 13.28 ユーザースイッチ周辺の構成

### 13.22.3. SD ブートデバイス選択スイッチ

SD ブートデバイス選択用のタクトスイッチ(SW9)を搭載しています。信号線は RZ/G1M の GP5\_31 に接続されてます。スイッチが押されている時は Low レベル、押されていない時は High レベルが入力されます。

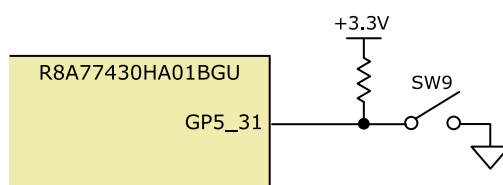


図 13.29 SD ブートデバイス選択スイッチ周辺の構成

### 13.22.4. 電源スイッチ

2 接点のスライドスイッチ(SW7)を搭載しています。+12/+5V に変換するスイッチングレギュレータのイネーブルピンと+12V\_SATA 電源の ON/OFF 制御をする eFuse のイネーブルピンに接続されています。ON で電源供給、OFF で電源切断されます。

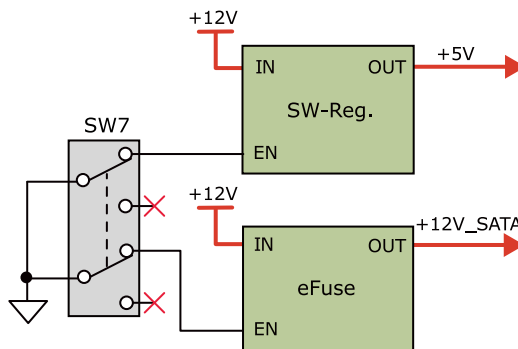


図 13.30 電源スイッチ周辺の構成

### 13.22.5. リセットスイッチ

リセット用のタクトスイッチ(SW8)を搭載しています。ON でリセット状態、OFF でリセット解除となります。

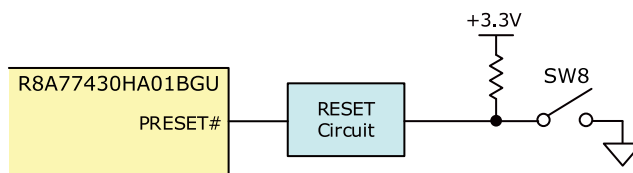


図 13.31 リセットスイッチ周辺の構成

## 13.23. LED

シリアル LED、ユーザー LED、電源 LED、警告 LED を搭載しています。

### 13.23.1. シリアル LED

USB シリアルインターフェース(CON14)の送受信の状態を示す面実装の黄色 LED を 2 つ(LED1、LED2)搭載しています。詳細については「13.8. USB シリアル」をご参照ください。

### 13.23.2. ユーザー LED

ユーザー側で自由に利用できる面実装の黄色 LED を 4 つ(LED4、LED5、LED6、LED7)搭載しています。信号線は RZ/G1M の GP5\_4、GP5\_5、GP5\_6、GP5\_7 に接続されています。High レベル出力で点灯、Low レベル出力で消灯します。



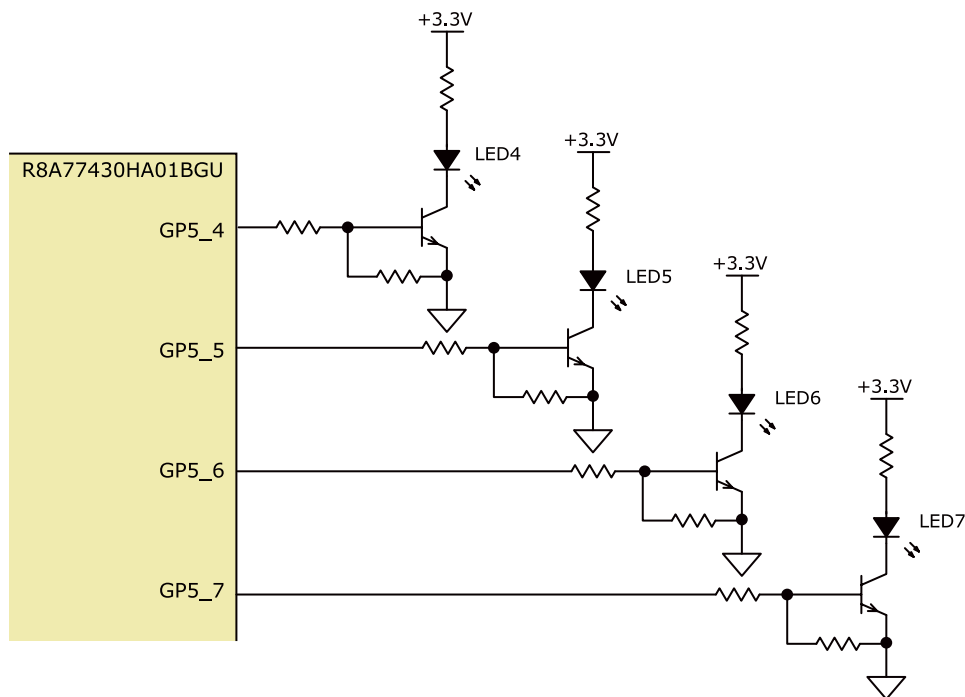


図 13.32 ユーザー LED 周辺の構成

### 13.23.3. 電源 LED

Armadillo-EVA 1500 に電源が供給されていることを示す、面実装の緑色 LED(LED8)を搭載しています。電源 LED は+3.3V が供給されると点灯します。

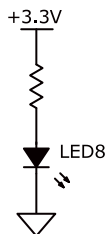


図 13.33 電源 LED 周辺の構成

### 13.23.4. 警告 LED

警告用に面実装の赤色 LED(LED9)を搭載しています。RZ/G1M の GP5\_4 に接続されており、High レベル出力で点灯、Low レベル出力で消灯します。

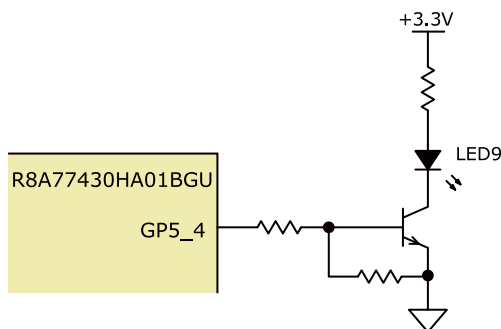


図 13.34 警告 LED 周辺の構成

## 13.24. JTAG

ARM JTAG デバッガを接続することができる JTAG インターフェース(CON1)を搭載しています。信号線は RZ/G1M の JTAG ピンに接続されています。JTAG を使用する場合、SW1.5 を OFF、SW1.6 を ON に、通常使用時は SW1.5 を ON に設定します。

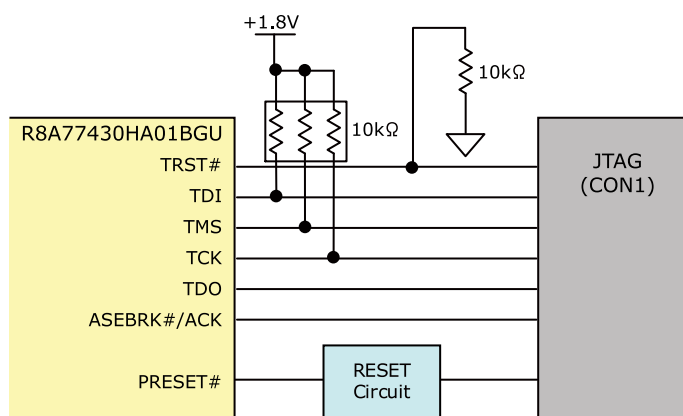


図 13.35 JTAG 周辺の構成

## 13.25. リアルタイムクロック

セイコーインスツル製リアルタイムクロック(S-35390A)を搭載しています。リアルタイムクロックは積層セラミックコンデンサにより、電源切断後も数分間動作することが可能です。長時間電源が切断されても時刻データを保持させたい場合は、RTC 外部バックアップインターフェース(CON31)に別途バッテリー-(CR2032 等)を接続することができます。

リアルタイムクロックの主な仕様は次のとおりです。

表 13.30 リアルタイムクロック仕様

バックアップ	300 秒(Typ.)、60 秒(Min.) RTC 外部バックアップインターフェース(CON31)経由で外部バッテリーを接続可能
電源電圧	DC2.0~3.5V

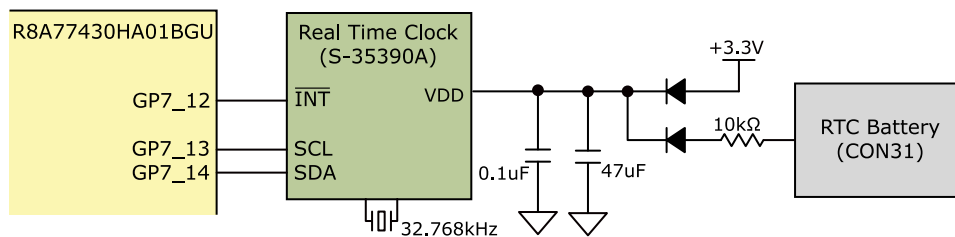


図 13.36 リアルタイムクロック周辺の構成

## 13.26. 電源

### 13.26.1. 電源回路の構成

Armadillo-EVA 1500 の電源回路の構成は次のとおりです。CON32 もしくは CON33 からの入力電圧を電源 IC で各電圧に変換し、内部回路および各インターフェースに供給しています。デバイスの電流容量の制限を超えないように、外部機器の接続、供給電源の設計を行ってください。

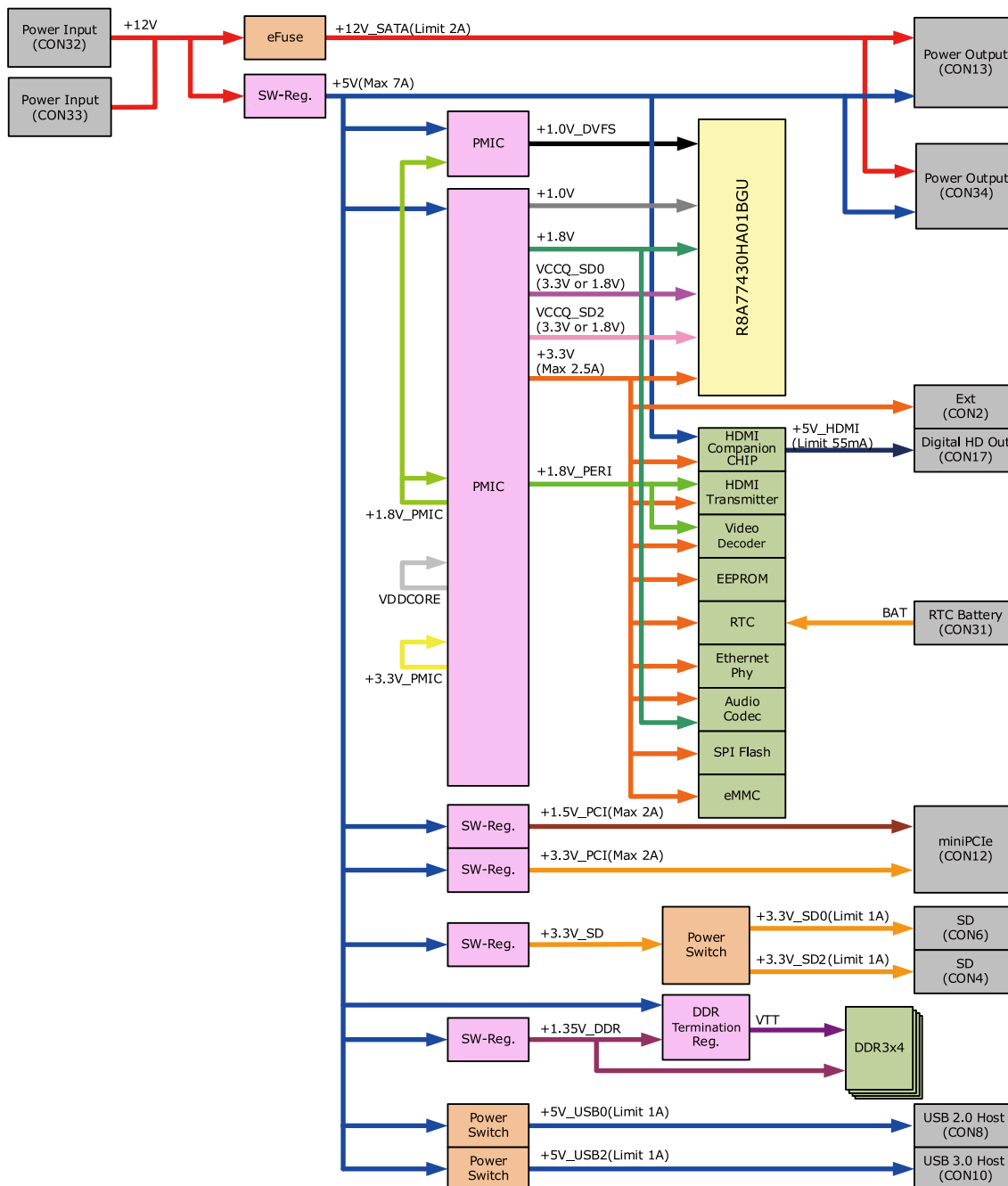


図 13.37 電源回路の構成

### 13.26.2. 電源シーケンス

電源シーケンスは次のとおりです。

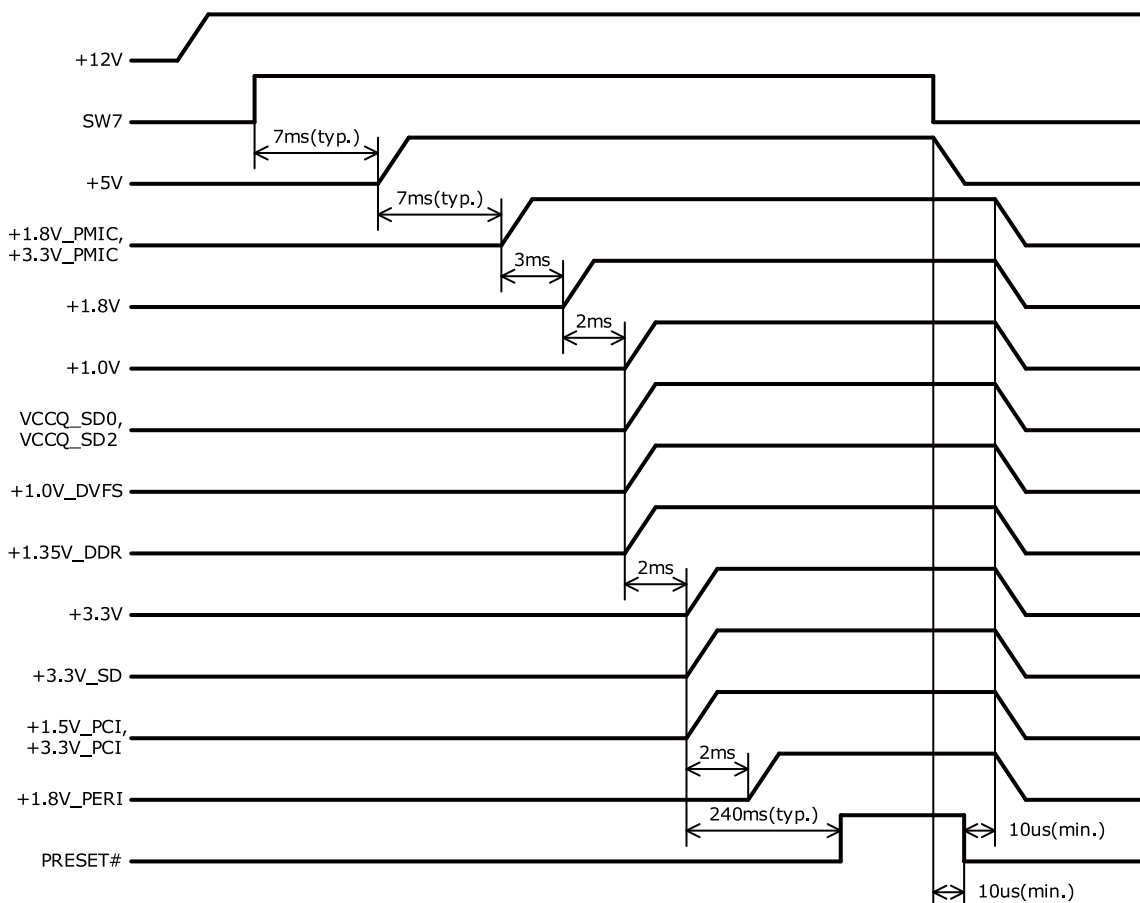


図 13.38 電源シーケンス

## 13.27. リセット

リセット回路では+3.3V、+5V を監視しています。電源投入時には+3.3V が確定すると、パワーオンリセット信号が解除されます。リセットスイッチ(SW8)、JTAG インターフェース(CON1)の 15 ピン、拡張インターフェース(CON3)の 13 ピンからもパワーオンリセット信号を発生させることが可能です。外部からリセットする場合、確実にリセットさせるため、200µs 以上の Low 期間を設定してください。パワーオンリセット信号は、eMMC、ビデオデコーダ、拡張インターフェース(CON2)に接続されています。

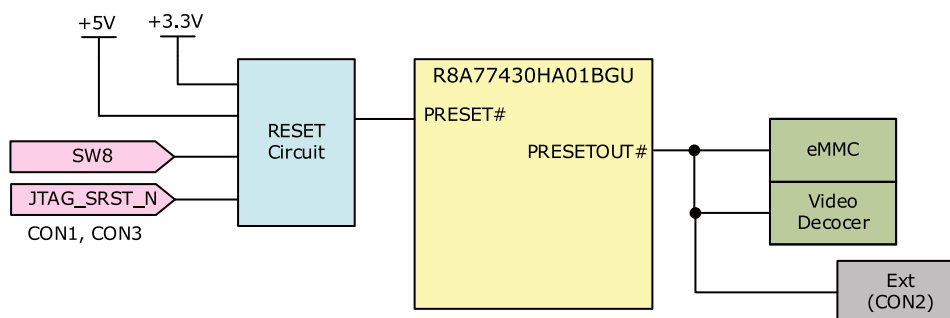


図 13.39 リセット回路の構成

# 14. 電氣的仕様

## 14.1. 絶対最大定格

表 14.1 絶対最大定格

項目	記号	Min	Max	単位	備考
電源電圧	+12V	-0.3	13.8	V	
動作温度範囲	Topr	10	40	°C	ただし結露なきこと



絶対最大定格はあらゆる使用条件、または試験条件であっても瞬時たりとも越えてはならない値です。上記の値に対して余裕をもってご使用ください。

## 14.2. 推奨動作条件

表 14.2 推奨動作条件

項目	記号	Min	Typ	Max	単位	備考
電源電圧	+12V	11.4	12	12.6	V	
使用周囲温度	Ta	10	25	40	°C	ただし結露なきこと



入出力インターフェースの電氣的仕様につきましては、各部品のデータシートをご確認ください。<sup>[1]</sup>

<sup>[1]</sup>部品表は「アットマークテクノ ユーザーズサイト」で公開しています。

# 15. インターフェース仕様

## 15.1. インターフェースレイアウト

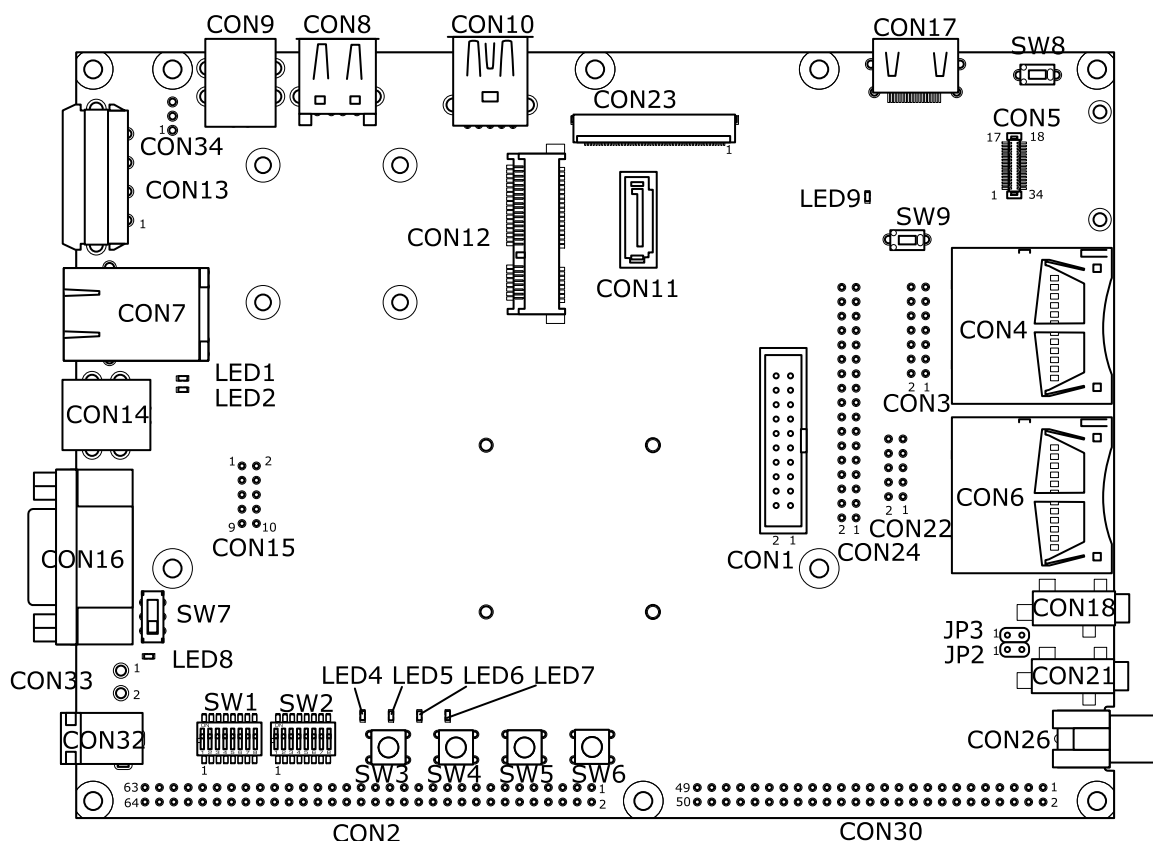


図 15.1 インターフェースレイアウト(A面)

表 15.1 搭載コネクタ、スイッチ型番一覧(A面)<sup>[a]</sup>

部品番号	インターフェース名	型番	メーカー
CON1	JTAG インターフェース	XG4C-2031	OMRON
CON2	拡張インターフェース	XG4C-6431	OMRON
CON3	拡張インターフェース	XG4C-1431	OMRON
CON4	SD インターフェース	SCDA9A0400	ALPS ELECTRIC
CON5	WLAN インターフェース	AXK6F34347YG-E	Panasonic
CON6	SD インターフェース	SCDA9A0400	ALPS ELECTRIC
CON7	LAN インターフェース	08B0-1X1T-36-F	Bel Fuse
CON8	USB 2.0 ホストインターフェース	UBA-4R-D14T-4D	J.S.T. Mfg.
CON9	USB 2.0 ファンクションインターフェース	UBB-4R-D14T-4D	J.S.T. Mfg.
CON10	USB 3.0 ホストインターフェース	CMS1600-010020	HOSIDEN
CON11	SATA インターフェース	67800-8005	MOLEX
CON12	miniPCIe インターフェース	MM60-52B1-E1-R650	JAE
CON13	電源出力インターフェース	LC-04A	J.S.T. Mfg.
CON14	USB シリアルインターフェース	UBB-4R-D14T-4D	J.S.T. Mfg.
CON15	拡張インターフェース	A1-10PA-2.54DSA(71)	HIROSE ELECTRIC

部品番号	インターフェース名	型番	メーカー
CON16	シリアルインターフェース	DEL09PBTK1YS-F	GTK
CON17	デジタル HD 出カインターフェース	CSS5019-0711FSZ	SMK
CON18	ヘッドホンインターフェース	SJ-3524-SMT-TR	CUI
CON21	マイク/ライン入カインターフェース	SJ-3524-SMT-TR	CUI
CON22	拡張インターフェース	A1-10PA-2.54DSA(71)	HIROSE ELECTRIC
CON23	LCD インターフェース	XF2M-5015-1A	OMRON
CON24	拡張インターフェース	XG4C-3431	OMRON
CON26	コンポジットビデオ入カインターフェース	MR-100E Y	MARUSHIN
CON30	拡張インターフェース	XG4C-5031	OMRON
CON32	電源入カインターフェース	PJ-102AH	CUI
CON33	電源入カインターフェース	B2PS-VH(LF)(SN)	J.S.T. Mfg.
CON34	電源出カインターフェース	22-04-1031	MOLEX
JP2	バイアスジャンパ	A2-2PA-2.54DSA(71)	HIROSE ELECTRIC
JP3			
SW1	機能選択スイッチ	218-8LPST	CTS
SW2			
SW3	ユーザースイッチ	SKHHAMA010	ALPS ELECTRIC
SW4			
SW5			
SW6			
SW7	電源スイッチ	JS202011CQN	C & K Components
SW8	リセットスイッチ	SKHLACA010	ALPS ELECTRIC
SW9	SD ブートデバイス選択スイッチ		
LED1	シリアル LED	SML-310YTT86	ROHM
LED2			
LED4	ユーザー LED		
LED5			
LED6			
LED7			
LED8	電源 LED		
LED9	警告 LED	SML-310VTT86	ROHM

<sup>[a]</sup>色のついたセルの部品は実装しておらず、実装例を記載しています。



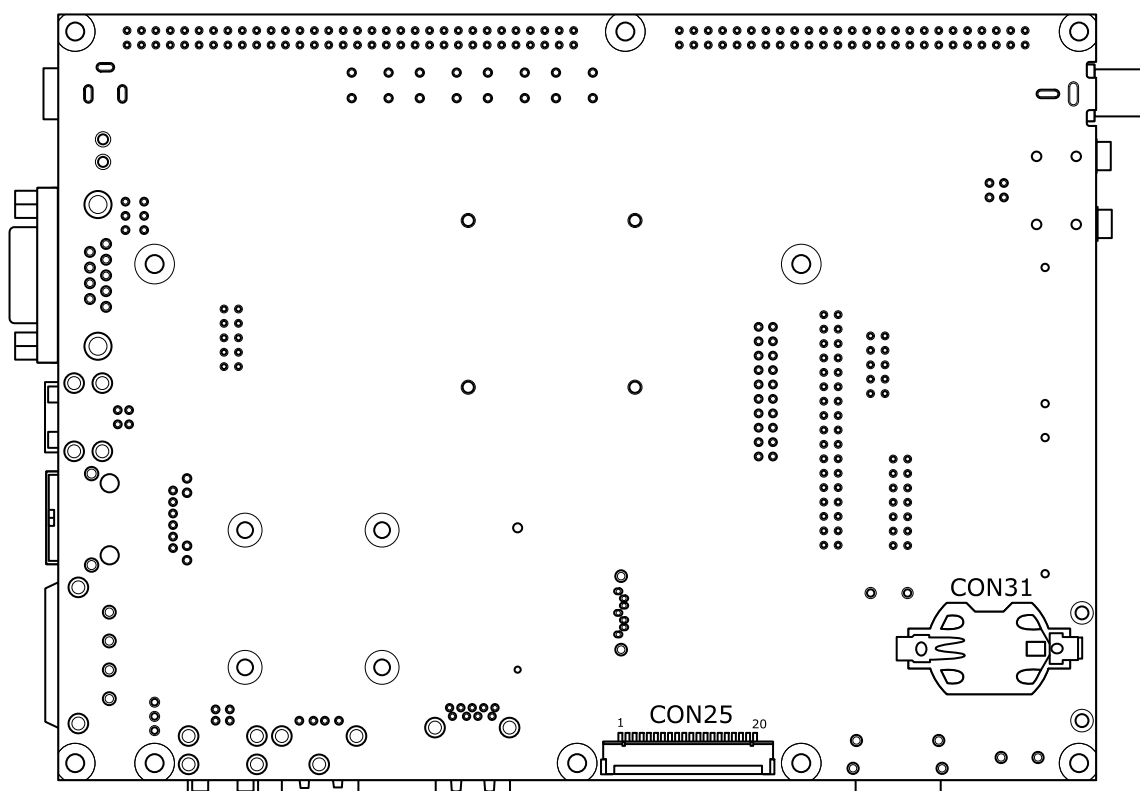


図 15.2 インターフェースレイアウト(B面)

表 15.2 搭載コネクタ、スイッチ型番一覧(B面)

部品番号	インターフェース名	型番	メーカー
CON25	LVDS インターフェース	DF14A-20P-1.25H(55)	HIROSE ELECTRIC
CON31	RTC 外部バックアップインターフェース	SMTU2032-LF	RENATA

## 15.2. CON1 JTAG インターフェース

表 15.3 CON1 信号配列

ピン番号	ピン名	I/O	説明
1	VTref	Power	電源(+1.8V)
2	TVDD	Power	電源(+1.8V)
3	TRST#	In	テストリセット、RZ/G1M の TRST#ピンに接続、GND に 10kΩ プルダウン
4	GND	Power	電源(GND)
5	TDI	In	テストデータ入力、RZ/G1M の TDI ピンに接続、+1.8V で 10kΩ プルアップ
6	GND	Power	電源(GND)
7	TMS	In	テストモード選択、RZ/G1M の TMS ピンに接続、+1.8V で 10kΩ プルアップ
8	GND	Power	電源(GND)
9	TCK	In	テストクロック、RZ/G1M の TCK ピンに接続、+1.8V で 10kΩ プルアップ
10	GND	Power	電源(GND)
11	RTCK	-	未接続
12	GND	Power	電源(GND)
13	TDO	Out	テストデータ出力、RZ/G1M の TDO ピンに接続
14	GND	Power	電源(GND)

ピン番号	ピン名	I/O	説明
15	SRST#	In	リセット、リセット IC に接続、+1.8V で 1kΩ プルアップ (Low: リセット状態、High: リセット解除)
16	GND	Power	電源(GND)
17	DBGRRQ	In	デバッグリクエスト、RZ/G1M の ASEBRK#/ACK ピンに接続
18	GND	Power	電源(GND)
19	DBGACK	-	未接続
20	GND	Power	電源(GND)

## 15.3. CON2 拡張インターフェース

表 15.4 CON2 信号配列

ピン番号	ピン名	I/O	説明
1	+3.3V	Power	電源(+3.3V)
2	GND	Power	電源(GND)
3	D0	In/Out	拡張入出力、RZ/G1M の D0 ピンに接続
4	D1	In/Out	拡張入出力、RZ/G1M の D1 ピンに接続
5	D2	In/Out	拡張入出力、RZ/G1M の D2 ピンに接続
6	D3	In/Out	拡張入出力、RZ/G1M の D3 ピンに接続
7	D4	In/Out	拡張入出力、RZ/G1M の D4 ピンに接続
8	D5	In/Out	拡張入出力、RZ/G1M の D5 ピンに接続
9	D6	In/Out	拡張入出力、RZ/G1M の D6 ピンに接続
10	D7	In/Out	拡張入出力、RZ/G1M の D7 ピンに接続
11	D8	In/Out	拡張入出力、RZ/G1M の D8 ピンに接続
12	D9	In/Out	拡張入出力、RZ/G1M の D9 ピンに接続
13	D10	In/Out	拡張入出力、RZ/G1M の D10 ピンに接続
14	D11	In/Out	拡張入出力、RZ/G1M の D11 ピンに接続
15	D12	In/Out	拡張入出力、RZ/G1M の D12 ピンに接続
16	D13	In/Out	拡張入出力、RZ/G1M の D13 ピンに接続
17	D14	In/Out	拡張入出力、RZ/G1M の D14 ピンに接続
18	D15	In/Out	拡張入出力、RZ/G1M の D15 ピンに接続
19	EX_CS1#	Out	拡張入出力、RZ/G1M の EX_CS1#ピンに接続
20	CS0#	Out	拡張入出力、RZ/G1M の CS0#ピンに接続
21	CS1#/A26	Out	拡張入出力、RZ/G1M の CS1#/A26 ピンに接続
22	EX_CS2#	Out	拡張入出力、RZ/G1M の EX_CS2#ピンに接続
23	EX_CS3#/MD9	In/Out	拡張入出力、RZ/G1M の EX_CS3#/MD9 ピンに接続、GND に 100kΩ プルダウン
24	PRESETOUT#	Out	リセット信号、RZ/G1M の PRESETOUT#ピンに接続、GND に 10kΩ プルダウン (Low: リセット状態、High: リセット解除)
25	RD#/MD12	In	拡張入出力、RZ/G1M の RD#/MD12 ピンに接続、GND に 100kΩ プルダウン
26	RD/WR#	In/Out	拡張入出力、RZ/G1M の RD/WR#ピンに接続
27	WE1#/MD4	In/Out	拡張入出力、RZ/G1M の WE1#/MD4 ピンに接続、GND に 100kΩ プルダウン
28	WE0#/MD6	In/Out	拡張入出力、RZ/G1M の WE0#/MD6 ピンに接続、GND に 100kΩ プルダウン
29	DACK0/MD7	In/Out	拡張入出力 0、RZ/G1M の DACK0/MD7 ピンに接続、GND に 100kΩ プルダウン
30	DREQ0	Out	拡張入出力、RZ/G1M の DREQ_0 ピンに接続
31	EX_WAIT0	Out	拡張入出力、RZ/G1M の EX_WAIT0 ピンに接続
32	EX_CS4#	Out	拡張入出力、RZ/G1M の EX_CS4#ピンに接続
33	EX_CS5#/MD8	In/Out	拡張入出力、RZ/G1M の EX_CS5#/MD8 ピンに接続、+3.3V で 100kΩ プルアップ
34	BS#/MD10	In/Out	拡張入出力、RZ/G1M の BS#/MD10 ピンに接続、GND に 100kΩ プルダウン
35	CLKOUT	Out	拡張入出力、RZ/G1M の CLKOUT ピンに接続
36	IRQ0	In	拡張入出力、RZ/G1M の IRQ0 ピンに接続

ピン番号	ピン名	I/O	説明
37	A0	Out	拡張入出力、RZ/G1M の A0 ピンに接続
38	A1/MD28	In/Out	拡張入出力、RZ/G1M の A1/MD28 ピンに接続、+3.3V で 100kΩ プルアップ
39	A2/MD23	In/Out	拡張入出力、RZ/G1M の A2/MD23 ピンに接続、+3.3V で 100kΩ プルアップ
40	A3/MD13	In/Out	拡張入出力、RZ/G1M の A3/MD13 ピンに接続、+3.3V で 100kΩ プルアップ
41	A4/MD24	In/Out	拡張入出力、RZ/G1M の A4/MD24 ピン、SW1 に接続
42	A5	Out	拡張入出力、RZ/G1M の A5 ピンに接続
43	A6	Out	拡張入出力、RZ/G1M の A6 ピンに接続
44	A7/MD27	In/Out	拡張入出力、RZ/G1M の A7/MD27 ピンに接続、+3.3V で 100kΩ プルアップ
45	A8	Out	拡張入出力、RZ/G1M の A8 ピンに接続
46	A9	Out	拡張入出力、RZ/G1M の A9 ピンに接続
47	A10/MD22	In/Out	拡張入出力、RZ/G1M の A10/MD22 ピンに接続、+3.3V で 100kΩ プルアップ
48	A11	Out	拡張入出力、RZ/G1M の A11 ピンに接続
49	A12	Out	拡張入出力、RZ/G1M の A12 ピンに接続
50	A13/MD21	In/Out	拡張入出力、RZ/G1M の A13/MD21 ピン、SW1 に接続
51	A14/MD19	In/Out	拡張入出力、RZ/G1M の A14/MD19 ピンに接続、GND に 100kΩ プルダウン
52	A15/MD20	In/Out	拡張入出力、RZ/G1M の A15/MD20 ピンに接続、GND に 100kΩ プルダウン
53	A16	Out	拡張入出力、RZ/G1M の A16 ピンに接続
54	A17	Out	拡張入出力、RZ/G1M の A17 ピンに接続
55	A18	Out	拡張入出力、RZ/G1M の A18 ピンに接続
56	A19/MD14	In/Out	拡張入出力、RZ/G1M の A19/MD14 ピンに接続、GND に 100kΩ プルダウン
57	A20	Out	拡張入出力、RZ/G1M の A20 ピンに接続
58	A21	Out	拡張入出力、RZ/G1M の A21 ピンに接続
59	A22	Out	拡張入出力、RZ/G1M の A22 ピンに接続
60	A23	Out	拡張入出力、RZ/G1M の A23 ピンに接続
61	A24	Out	拡張入出力、RZ/G1M の A24 ピンに接続
62	A25	Out	拡張入出力、RZ/G1M の A25 ピンに接続
63	+3.3V	Power	電源(+3.3V)
64	GND	Power	電源(GND)

## 15.4. CON3 拡張インターフェース

表 15.5 CON3 信号配列

ピン番号	ピン名	I/O	説明
1	SD2_DATA0	In/Out	拡張入出力、マルチプレクサを経由して RZ/G1M の SD2_DATA0 ピンに接続
2	SD2_CD	In	拡張入出力、マルチプレクサを経由して RZ/G1M の SD2_CD ピンに接続
3	SD2_CMD	In/Out	拡張入出力、マルチプレクサを経由して RZ/G1M の SD2_CMD ピンに接続
4	SD2_WP	In	拡張入出力、マルチプレクサを経由して RZ/G1M の SD2_WP ピンに接続
5	SD2_CLK	Out	拡張入出力、マルチプレクサを経由して RZ/G1M の SD2_CLK ピンに接続
6	GND	Power	電源(GND)
7	SD2_DATA3	In/Out	拡張入出力、マルチプレクサを経由して RZ/G1M の SD2_DATA3 ピンに接続
8	VCCQ_SD2	Power	電源(VCCQ_SD2)
9	SD2_DATA1	In/Out	拡張入出力、マルチプレクサを経由して RZ/G1M の SD2_DATA1 ピンに接続
10	GND	Power	電源(GND)
11	SD2_DATA2	In/Out	拡張入出力、マルチプレクサを経由して RZ/G1M の SD2_DATA2 ピンに接続
12	GND	Power	電源(GND)
13	SRST#	In	リセット、リセット IC に接続、+1.8V で 1kΩ プルアップ (Low: リセット状態、High: リセット解除)
14	GND	Power	電源(GND)

## 15.5. CON4 SD インターフェース

表 15.6 CON4 信号配列

ピン番号	ピン名	I/O	説明
1	DAT3	In/Out	データバス(bit3)、マルチプレクサを経由して RZ/G1M の SD2_DATA3 ピンに接続
2	CMD	In/Out	SD コマンド/レスポンス、マルチプレクサを経由して RZ/G1M の SD2_CMD ピンに接続
3	VSS1	Power	電源(GND)
4	VDD	Power	電源(+3.3V_SD2)
5	CLK	Out	SD クロック、マルチプレクサを経由して RZ/G1M の SD2_CLK ピンに接続
6	VSS2	Power	電源(GND)
7	DAT0	In/Out	データバス(bit0)、マルチプレクサを経由して RZ/G1M の SD2_DATA0 ピンに接続
8	DAT1	In/Out	データバス(bit1)、マルチプレクサを経由して RZ/G1M の SD2_DATA1 ピンに接続
9	DAT2	In/Out	データバス(bit2)、マルチプレクサを経由して RZ/G1M の SD2_DATA2 ピンに接続

表 15.7 CON4 カード検出、ライトプロテクト

項目	説明
カード検出	マルチプレクサを経由して RZ/G1M の SD2_CD ピンに接続 (Low:カード挿入、High:カード未挿入)
ライトプロテクト検出	マルチプレクサを経由して RZ/G1M の SD2_WP ピンに接続 (Low:書き込み可能、High:書き込み不可能)

## 15.6. CON5 WLAN インターフェース

表 15.8 CON5 信号配列

ピン番号	ピン名	I/O	説明
1	SDDATA1	In/Out	SDIO データ(bit1)、マルチプレクサを経由して RZ/G1M の SD2_DATA1 ピンに接続、VCCQ_SD2 に 47kΩ プルアップ
2	SDDATA0	In/Out	SDIO データ(bit0)、マルチプレクサを経由して RZ/G1M の SD2_DATA0 ピンに接続、VCCQ_SD2 に 47kΩ プルアップ
3	GND	Power	電源(GND)
4	GND	Power	電源(GND)
5	USB_DM	-	TP5 に接続
6	USB_DP	-	TP6 に接続
7	SDCLK	Out	SDIO クロック、マルチプレクサを経由して RZ/G1M の SD2_CLK ピンに接続、GND に 47kΩ プルダウン
8	VCC	Power	電源(+3.3V_SD2)
9	NC	-	未接続
10	SDCMD	In/Out	SDIO コマンド、マルチプレクサを経由して RZ/G1M の SD2_CMD ピンに接続、VCCQ_SD2 に 47kΩ プルアップ
11	SDDATA3	In/Out	SDIO データ(bit3)、マルチプレクサを経由して RZ/G1M の SD2_DATA3 ピンに接続、VCCQ_SD2 に 47kΩ プルアップ
12	SDDATA2	In/Out	SDIO データ(bit2)、マルチプレクサを経由して RZ/G1M の SD2_DATA2 ピンに接続、VCCQ_SD2 に 47kΩ プルアップ
13	UART_RXD	-	未接続
14	UART_TXD	-	未接続
15	BOOT_SEL1	Out	起動モード設定、SDIO モードに設定
16	BOOT_SELO	Out	
17	HOST_SEL	Out	
18	FLASH_RXD	-	未接続
19	FLASH_CSB	-	未接続
20	FLASH_CLK	-	未接続
21	FLASH_TXD	Out	GND に 47kΩ プルダウン
22	FLASH_SEL	-	未接続

ピン番号	ピン名	I/O	説明
23	GPIO0	-	TP7 に接続
24	GPIO1	-	TP8 に接続
25	M_ANA	-	未接続
26	GPIO2	-	TP9 に接続
27	GPIO6	-	TP11 に接続
28	HRST	Out	+3.3V に接続
29	PRST	-	TP10 に接続
30	TMS	-	未接続
31	TCK	-	未接続
32	TDI	-	未接続
33	TDO	-	未接続
34	TRSTB	-	未接続

## 15.7. CON6 SD インターフェース

表 15.9 CON6 信号配列

ピン番号	ピン名	I/O	説明
1	DAT3	In/Out	データバス(bit3)、RZ/G1M の SD0_DATA3 ピンに接続
2	CMD	In/Out	SD コマンド/レスポンス、RZ/G1M の SD0_CMD ピンに接続
3	VSS1	Power	電源(GND)
4	VDD	Power	電源(+3.3V_SD0)
5	CLK	Out	SD クロック、RZ/G1M の SD0_CLK ピンに接続
6	VSS2	Power	電源(GND)
7	DAT0	In/Out	データバス(bit0)、RZ/G1M の SD0_DATA0 ピンに接続
8	DAT1	In/Out	データバス(bit1)、RZ/G1M の SD0_DATA1 ピンに接続
9	DAT2	In/Out	データバス(bit2)、RZ/G1M の SD0_DATA2 ピンに接続

表 15.10 CON6 カード検出、ライトプロテクト

項目	説明
カード検出	マルチプレクサを経由して RZ/G1M の SD0_CD ピンに接続 (Low:カード挿入、High:カード未挿入)
ライトプロテクト検出	マルチプレクサを経由して RZ/G1M の SD0_WP ピンに接続 (Low:書き込み可能、High:書き込み不可能)

## 15.8. CON7 LAN インターフェース

表 15.11 CON7 信号配列

ピン番号	ピン名	I/O	説明
1	TX+	In/Out	送信出力 差動ペア(+)
2	TX-	In/Out	送信出力 差動ペア(-)
3	RX+	In/Out	受信入力 差動ペア(+)
4	-	-	CON7 5 ピンと接続後に 75Ω 終端
5	-	-	CON7 4 ピンと接続後に 75Ω 終端
6	RX-	In/Out	受信入力 差動ペア(-)
7	-	-	CON7 8 ピンと接続後に 75Ω 終端
8	-	-	CON7 7 ピンと接続後に 75Ω 終端

## 15.9. CON8 USB 2.0 ホストインターフェース

表 15.12 CON8 信号配列

ピン番号	ピン名	I/O	説明
1	VBUS	Power	USB 電源(+5V)
2	D-	In/Out	USB マイナス側信号、マルチプレクサを経由して RZ/G1M の USB0_DM ピンに接続
3	D+	In/Out	USB プラス側信号、マルチプレクサを経由して RZ/G1M の USB0_DP ピンに接続
4	GND	Power	電源(GND)

## 15.10. CON9 USB 2.0 ファンクションインターフェース

表 15.13 CON9 信号配列

ピン番号	ピン名	I/O	説明
1	VBUS	Power	USB 電源
2	D-	In/Out	USB マイナス側信号、マルチプレクサを経由して RZ/G1M の USB0_DM ピンに接続
3	D+	In/Out	USB プラス側信号、マルチプレクサを経由して RZ/G1M の USB0_DP ピンに接続
4	GND	Power	電源(GND)

## 15.11. CON10 USB 3.0 ホストインターフェース

表 15.14 CON10 信号配列

ピン番号	ピン名	I/O	説明
1	VBUS	Power	USB 電源(+5V)
2	D-	In/Out	USB マイナス側信号、マルチプレクサを経由して RZ/G1M の USB1_DM ピンに接続
3	D+	In/Out	USB プラス側信号、マルチプレクサを経由して RZ/G1M の USB1_DP ピンに接続
4	GND	Power	電源(GND)
5	StdA_SSRX-	In/Out	USB プラス側信号、RZ/G1M の RIDN1_USB3 ピンに接続
6	StdA_SSRX+	In/Out	USB プラス側信号、RZ/G1M の RIDP1_USB3 ピンに接続
7	GND_DRAIN	Power	電源(GND)
8	StdA_SSTX-	In/Out	USB プラス側信号、RZ/G1M の TODN1_USB3 ピンに接続
9	StdA_SSTX+	In/Out	USB プラス側信号、RZ/G1M の TODP1_USB3 ピンに接続

## 15.12. CON11 SATA インターフェース

表 15.15 CON11 信号配列

ピン番号	ピン名	I/O	説明
1	GND	Power	電源(GND)
2	TXP	Out	送信出力 差動ペア(+), マルチプレクサを経由して RZ/G1M の TODP1_SATA ピンに接続
3	TXN	Out	送信出力 差動ペア(-), マルチプレクサを経由して RZ/G1M の TODN1_SATA ピンに接続
4	GND	Power	電源(GND)
5	RXN	In	受信入力 差動ペア(+), マルチプレクサを経由して RZ/G1M の RIDN1_SATA ピンに接続
6	RXP	In	受信入力 差動ペア(-), マルチプレクサを経由して RZ/G1M の RIDP1_SATA ピンに接続
7	GND	Power	電源(GND)

## 15.13. CON12 miniPCle インターフェース

表 15.16 CON12 信号配列

ピン番号	ピン名	I/O	説明
1	WAKE#	In	リンク再アクティブ信号、RZ/G1M の SSI_WS9(GP2_26)ピンに接続
2	+3.3V	Power	電源(+3.3V_PCI)
3	COEX1	-	未接続
4	GND	Power	電源(GND)
5	COEX2	-	未接続
6	+1.5V	Power	電源(+1.5V_PCI)
7	CLKREQ#	-	未接続
8	UIM_PWR	-	未接続
9	GND	Power	電源(GND)
10	UIM_DATA	-	未接続
11	REFCLK-	In/Out	リファレンスクロック 差動ペア(-)、クロックジェネレータに接続
12	UIM_CLK	-	未接続
13	REFCLK+	In/Out	リファレンスクロック 差動ペア(+)、クロックジェネレータに接続
14	UIM_RESET	-	未接続
15	GND	Power	電源(GND)
16	UIM_VPP	-	未接続
17	Reserved	-	未接続
18	GND	Power	電源(GND)
19	Reserved	-	未接続
20	W_DISABLE#	Out	無線動作の有効/無効信号、RZ/G1M の SSI_SCK9(GP2_25)ピンに接続、+3.3V で 1kΩ プルアップ(Low: 無線無効、High: 無線有効)
21	GND(CD)	Power	電源(GND)
22	PERST#	Out	基本リセット、RZ/G1M の AUDIO_CLKC(GP2_30)ピンに接続、+3.3V で 1kΩ プルアップ
23	PERn0	In	レーン 0 レシーバ差動ペア(-)、RZ/G1M の RIDN1_SATA(RIDN0_PClE)ピンに接続
24	+3.3V	Power	電源(+3.3V_PCI)
25	PERp0	In	レーン 0 レシーバ差動ペア(+)、RZ/G1M の RIDP1_SATA(RIDP0_PClE)ピンに接続
26	GND	Power	電源(GND)
27	GND	Power	電源(GND)
28	+1.5V	Power	電源(+1.5V_PCI)
29	GND	Power	電源(GND)
30	SMB_CLK	In/Out	SMBus クロック、RZ/G1M の SSI_SCK2(GP2_6)ピンに接続、+3.3V に 1kΩ プルアップ
31	PETn0	Out	レーン 0 トランスミッタ差動ペア(+), RZ/G1M の TODP1_SATA(TODP0_PClE)ピンに接続
32	SMB_DATA	In/Out	SMBus データ、RZ/G1M の SSI_WS2(GP2_7)ピンに接続、+3.3V に 1kΩ プルアップ
33	PETp0	Out	レーン 0 トランスミッタ差動ペア(-)、RZ/G1M の TODN1_SATA(TODN0_PClE)ピンに接続
34	GND	Power	電源(GND)
35	GND	Power	電源(GND)
36	USB_D-	In/Out	USB マイナス側信号、マルチプレクサを経由して RZ/G1M の USB1_DM ピンに接続
37	GND	Power	電源(GND)
38	USB_D+	In/Out	USB プラス側信号、マルチプレクサを経由して RZ/G1M の USB1_DP ピンに接続
39	+3.3V	Power	電源(+3.3V_PCI)
40	GND	Power	電源(GND)
41	+3.3V	Power	電源(+3.3V_PCI)
42	LED_WWAN#	-	未接続

ピン番号	ピン名	I/O	説明
43	GND	Power	電源(GND)
44	LED_WLAN#	-	未接続
45	Reserved	-	未接続
46	LED_WPAN#	-	未接続
47	Reserved	-	未接続
48	+1.5V	Power	電源(+1.5V_PCI)
49	Reserved	-	未接続
50	GND	Power	電源(GND)
51	Reserved	-	未接続
52	+3.3V	Power	電源(+3.3V_PCI)

## 15.14. CON13 電源出力インターフェース

表 15.17 CON13 信号配列

ピン番号	ピン名	I/O	説明
1	+12V_SATA	Power	+12V 電源出力(+12V_SATA)
2	GND	Power	電源(GND)
3	GND	Power	電源(GND)
4	+5V	Power	+5V 電源出力(+5V)

## 15.15. CON14 USB シリアルインターフェース

表 15.18 CON14 信号配列

ピン番号	ピン名	I/O	説明
1	VBUS	Power	USB 電源
2	D-	In/Out	USB マイナス側信号
3	D+	In/Out	USB プラス側信号
4	GND	Power	電源(GND)

## 15.16. CON15 拡張インターフェース

表 15.19 CON15 信号配列

ピン番号	ピン名	I/O	説明
1	+3.3V	Power	電源(+3.3V)
2	GND	Power	電源(GND)
3	HCTS0#	In/Out	拡張入出力、マルチプレクサを経由して RZ/G1M の HCTS0#ピンに接続
4	HRTS0#	In/Out	拡張入出力、マルチプレクサを経由して RZ/G1M の HRTS0#ピンに接続
5	HCKO	In/Out	拡張入出力、RZ/G1M の HCKO ピンに接続
6	HRX0	In	拡張入出力、マルチプレクサを経由して RZ/G1M の HRX0 ピンに接続
7	HTX0	Out	拡張入出力、マルチプレクサを経由して RZ/G1M の HTX0 ピンに接続
8	CAN1_TX_D	Out	拡張入出力、マルチプレクサを経由して RZ/G1M の CAN1_TX_D/MDT1 ピンに接続、GND に 100kΩ プルダウン
9	CAN_CLK_C	In	拡張入出力、マルチプレクサを経由して RZ/G1M の CAN_CLK_C/MDT0 ピンに接続、GND に 100kΩ プルダウン
10	CAN1_RXD_D	In	拡張入出力、RZ/G1M の CAN1_RXD_D ピンに接続



## 15.17. CON16 シリアルインターフェース

表 15.20 CON16 信号配列

ピン番号	ピン名	I/O	説明
1	NC	-	未接続
2	RXD	In	UART 受信データ、レベル変換 IC とマルチプレクサを經由して RZ/G1M の HRX0(SCIFB0_RXD)ピンに接続
3	TXD	Out	UART 送信データ、レベル変換 IC とマルチプレクサを經由して RZ/G1M の HTX0(SCIFB0_TXD)ピンに接続
4	NC	-	未接続
5	GND	Power	電源(GND)
6	NC	-	未接続
7	RTS	Out	送信要求、レベル変換 IC とマルチプレクサを經由して RZ/G1M の HRTS0#(SCIFB0_RTS#)ピンに接続
8	CTS	In	送信可能、レベル変換 IC とマルチプレクサを經由して RZ/G1M の HCTS0#(SCIFB0_CTS#)ピンに接続
9	NC	-	未接続

## 15.18. CON17 デジタル HD 出カインターフェース

表 15.21 CON17 信号配列

ピン番号	ピン名	I/O	説明
1	TX2+	Out	TMDS データ 2(+)
2	TX2_Shield	-	TMDS データ 2 シールド
3	TX2-	Out	TMDS データ 2(-)
4	TX1+	Out	TMDS データ 1(+)
5	TX1_Shield	-	TMDS データ 1 シールド
6	TX1-	Out	TMDS データ 1(-)
7	TX0+	Out	TMDS データ 0(+)
8	TX0_Shield	-	TMDS データ 0 シールド
9	TX0-	Out	TMDS データ 0(-)
10	TXC+	Out	TMDS クロック(+)
11	TXC_Shield	-	TMDS クロックシールド
12	TXC-	Out	TMDS クロック(-)
13	CEC	In/Out	CEC 信号
14	Utility	-	未接続
15	SCL	In/Out	DDC クロック
16	SDA	In/Out	DDC データ
17	GND	Power	電源(GND)
18	+5V	Power	電源(+5V_HDMI)
19	HPD	In	ホットプラグ検出

## 15.19. CON18 ヘッドホンインターフェース

表 15.22 CON18 信号配列

ピン	ピン名	I/O	説明
TIP	HP_OUT_L	Out	ヘッドホン出力(左チャンネル)
RING	HP_OUT_R	Out	ヘッドホン出力(右チャンネル)
SLEEVE	GND	Power	電源(GND)

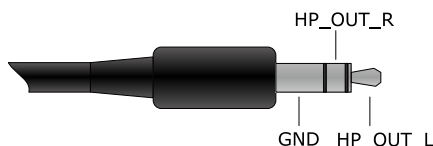


図 15.3 ミニプラグ(ヘッドホン)のピンアサイン

## 15.20. CON21 マイク/ライン入力インターフェース

表 15.23 CON21 信号配列

ピン	ピン名	I/O	機能
TIP	MIC_IN	In	マイク入力
RING	NC	-	未接続
SLEEVE	GND	Power	電源(GND)

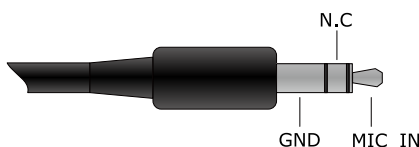


図 15.4 ミニプラグ(マイク)のピンアサイン

## 15.21. CON22 拡張インターフェース

表 15.24 CON22 信号配列

ピン番号	ピン名	I/O	説明
1	SSI_SCK34	In/Out	拡張入出力、マルチプレクサを経由して RZ/G1M の SSI_SCK34 ピンに接続
2	SSI_WS34	In/Out	拡張入出力、マルチプレクサを経由して RZ/G1M の SSI_WS34 ピンに接続
3	SSI_SDATA3	In/Out	拡張入出力、マルチプレクサを経由して RZ/G1M の SSI_SDATA3 ピンに接続
4	SSI_SDATA4	In/Out	拡張入出力、マルチプレクサを経由して RZ/G1M の SSI_SDATA4 ピンに接続
5	SSI_SCK4	In/Out	拡張入出力、RZ/G1M の SSI_SCK4 ピンに接続、GND に 100kΩ プルダウン
6	SSI_WS4	In/Out	拡張入出力、RZ/G1M の SSI_WS4 ピンに接続、GND に 100kΩ プルダウン
7	AUDIO_RESET#	In	オーディオコーデック用リセット信号、マルチプレクサを経由して RZ/G1M の AUDIO_CLKOUT/MD5(GP2_31)ピンに接続、GND に 100kΩ プルダウン
8	AUDIO_EN#	In	SSI3、4 の信号をオーディオコーデックと拡張インターフェース(CON22)のどちらに接続するかを設定、マルチプレクサのセレクトピンに接続、GND に 10kΩ プルダウン (Low: オーディオコーデック、High: 拡張インターフェース(CON22))
9	+3.3V	Power	電源(+3.3V)
10	GND	Power	電源(GND)

## 15.22. CON23 LCD インターフェース

表 15.25 CON23 信号配列

ピン番号	ピン名	I/O	説明
1	+5V	Power	電源(+5V)
2	+5V	Power	電源(+5V)
3	+5V	Power	電源(+5V)
4	NC	-	未接続
5	+3.3V	Power	電源(+3.3V)

ピン番号	ピン名	I/O	説明
6	+3.3V	Power	電源(+3.3V)
7	NC	-	未接続
8	GND	Power	電源(GND)
9	GND	Power	電源(GND)
10	TP_RST	Out	リセット信号、マルチプレクサを経由して RZ/G1M の DU1_CDE/MD0(GP3_31)ピンに接続、GND に 100kΩ プルダウン
11	TP_INT	In	割り込み信号、RZ/G1M の IRQ9(GP7_19)ピンに接続
12	GND	Power	電源(GND)
13	SCL	In/Out	I2C クロック、RZ/G1M の SSI_SCK2(GP2_6)ピンに接続、+3.3V に 1kΩ プルアップ
14	SDA	In/Out	I2C データ、RZ/G1M の SSI_WS2(GP2_7)ピンに接続、+3.3V に 1kΩ プルアップ
15	GND	Power	電源(GND)
16	LED_CONT	Out	LED_CONT 信号、RZ/G1M の DU1_DOTCLKOUT1 ピンに接続
17	GND	Power	電源(GND)
18	DISP	Out	DISP 信号、RZ/G1M の DU1_EXODDF/DU1_ODDF/DISP/CDE ピンに接続
19	DE	Out	DE 信号、マルチプレクサを経由して RZ/G1M の DU1_DISP/MD1 ピンに接続、GND に 100kΩ プルダウン
20	VSYNC	Out	VSYNC 信号、マルチプレクサを経由して RZ/G1M の DU1_EXVSYNC/DU1_VSYNC/MD2 ピンに接続、GND に 100kΩ プルダウン
21	HSYNC	Out	HSYNC 信号、マルチプレクサを経由して RZ/G1M の DU1_EXHSYNC/DU1_HSYNC/MD3 ピンに接続、GND に 100kΩ プルダウン
22	DCLK	Out	DCLK 信号、RZ/G1M の DU1_DOTCLKOUT0 ピンに接続
23	GND	Power	電源(GND)
24	B7	Out	LCD データ(bit7)、RZ/G1M の DU1_DB7 ピンに接続
25	B6	Out	LCD データ(bit6)、RZ/G1M の DU1_DB6 ピンに接続
26	B5	Out	LCD データ(bit5)、RZ/G1M の DU1_DB5 ピンに接続、GND に 100kΩ プルダウン
27	B4	Out	LCD データ(bit4)、RZ/G1M の DU1_DB4 ピンに接続
28	B3	Out	LCD データ(bit3)、RZ/G1M の DU1_DB3 ピンに接続
29	B2	Out	LCD データ(bit2)、RZ/G1M の DU1_DB2 ピンに接続
30	B1	Out	LCD データ(bit1)、RZ/G1M の DU1_DB1 ピンに接続
31	B0	Out	LCD データ(bit0)、RZ/G1M の DU1_DB0 ピンに接続
32	GND	Power	電源(GND)
33	G7	Out	LCD データ(bit15)、RZ/G1M の DU1_DG7 ピンに接続
34	G6	Out	LCD データ(bit14)、RZ/G1M の DU1_DG6 ピンに接続
35	G5	Out	LCD データ(bit13)、RZ/G1M の DU1_DG5 ピンに接続
36	G4	Out	LCD データ(bit12)、RZ/G1M の DU1_DG4 ピンに接続
37	G3	Out	LCD データ(bit11)、RZ/G1M の DU1_DG3 ピンに接続
38	G2	Out	LCD データ(bit10)、RZ/G1M の DU1_DG2 ピンに接続
39	G1	Out	LCD データ(bit9)、RZ/G1M の DU1_DG1 ピンに接続
40	G0	Out	LCD データ(bit8)、RZ/G1M の DU1_DG0 ピンに接続
41	GND	Power	電源(GND)
42	R7	Out	LCD データ(bit23)、RZ/G1M の DU1_DR7 ピンに接続
43	R6	Out	LCD データ(bit22)、RZ/G1M の DU1_DR6 ピンに接続
44	R5	Out	LCD データ(bit21)、RZ/G1M の DU1_DR5 ピンに接続
45	R4	Out	LCD データ(bit20)、RZ/G1M の DU1_DR4 ピンに接続
46	R3	Out	LCD データ(bit19)、RZ/G1M の DU1_DR3 ピンに接続
47	R2	Out	LCD データ(bit18)、RZ/G1M の DU1_DR2 ピンに接続
48	R1	Out	LCD データ(bit17)、RZ/G1M の DU1_DR1 ピンに接続
49	R0	Out	LCD データ(bit16)、RZ/G1M の DU1_DR0 ピンに接続
50	GND	Power	電源(GND)

## 15.23. CON24 拡張インターフェース

表 15.26 CON24 信号配列

ピン番号	ピン名	I/O	説明
1	DU1_DR0	Out	拡張入出力、RZ/G1M の DU1_DR0 ピンに接続
2	DU1_DR1	Out	拡張入出力、RZ/G1M の DU1_DR1 ピンに接続
3	DU1_DR2	Out	拡張入出力、RZ/G1M の DU1_DR2 ピンに接続
4	DU1_DR3	Out	拡張入出力、RZ/G1M の DU1_DR3 ピンに接続
5	DU1_DR4	Out	拡張入出力、RZ/G1M の DU1_DR4 ピンに接続
6	DU1_DR5	Out	拡張入出力、RZ/G1M の DU1_DR5 ピンに接続
7	DU1_DR6	Out	拡張入出力、RZ/G1M の DU1_DR6 ピンに接続
8	DU1_DR7	Out	拡張入出力、RZ/G1M の DU1_DR7 ピンに接続
9	DU1_DG0	Out	拡張入出力、RZ/G1M の DU1_DG0 ピンに接続
10	DU1_DG1	Out	拡張入出力、RZ/G1M の DU1_DG1 ピンに接続
11	DU1_DG2	Out	拡張入出力、RZ/G1M の DU1_DG2 ピンに接続
12	DU1_DG3	Out	拡張入出力、RZ/G1M の DU1_DG3 ピンに接続
13	DU1_DG4	Out	拡張入出力、RZ/G1M の DU1_DG4 ピンに接続
14	DU1_DG5	Out	拡張入出力、RZ/G1M の DU1_DG5 ピンに接続
15	DU1_DG6	Out	拡張入出力、RZ/G1M の DU1_DG6 ピンに接続
16	DU1_DG7	Out	拡張入出力、RZ/G1M の DU1_DG7 ピンに接続
17	DU1_DB0	Out	拡張入出力、RZ/G1M の DU1_DB0 ピンに接続
18	DU1_DB1	Out	拡張入出力、RZ/G1M の DU1_DB1 ピンに接続
19	DU1_DB2	Out	拡張入出力、RZ/G1M の DU1_DB2 ピンに接続
20	DU1_DB3	Out	拡張入出力、RZ/G1M の DU1_DB3 ピンに接続
21	DU1_DB4	Out	拡張入出力、RZ/G1M の DU1_DB4 ピンに接続
22	DU1_DB5	Out	拡張入出力、RZ/G1M の DU1_DB5 ピンに接続、GND に 100kΩ プルダウン
23	DU1_DB6	Out	拡張入出力、RZ/G1M の DU1_DB6 ピンに接続
24	DU1_DB7	Out	拡張入出力、RZ/G1M の DU1_DB7 ピンに接続
25	TP_INT	In	拡張入出力、RZ/G1M の IRQ9 ピンに接続
26	DU1_DOTCLKOUT0	Out	拡張入出力、RZ/G1M の DU1_DOTCLKOUT0 ピンに接続
27	DU1_DOTCLKOUT1	Out	拡張入出力、RZ/G1M の DU1_DOTCLKOUT1 ピンに接続
28	DU1_HSYNC	Out	拡張入出力、マルチプレクサを経由して RZ/G1M の DU1_EXHSYNC/ DU1_HSYNC/MD3 ピンに接続、GND に 100kΩ プルダウン
29	DU1_VSYNC	Out	拡張入出力、マルチプレクサを経由して RZ/G1M の DU1_EXVSYNC/ DU1_VSYNC/MD2 ピンに接続、GND に 100kΩ プルダウン
30	DU1_ODDF	Out	拡張入出力、RZ/G1M の DU1_EXODDF/DU1_ODDF/DISP/CDE ピンに接続
31	DU1_DISP	Out	拡張入出力、マルチプレクサを経由して RZ/G1M の DU1_DISP/MD1 ピンに接続、GND に 100kΩ プルダウン
32	TP_RST	Out	拡張入出力、マルチプレクサを経由して RZ/G1M の DU1_CDE/MD0 ピンに接続、GND に 100kΩ プルダウン
33	+3.3V	Power	電源(+3.3V)
34	GND	Power	電源(GND)

## 15.24. CON25 LVDS インターフェース

表 15.27 CON25 信号配列

ピン番号	ピン名	I/O	説明
1	VDD	Power	電源(+3.3V)
2	VDD	Power	電源(+3.3V)
3	GND	Power	電源(GND)
4	GND	Power	電源(GND)

ピン番号	ピン名	I/O	説明
5	Tx0-	Out	LVDS CH0(-)、RZ/G1M の DU0_LVDS_CH0_N ピンに接続
6	Tx0+	Out	LVDS CH0(+), RZ/G1M の DU0_LVDS_CH0_P ピンに接続
7	GND	Power	電源(GND)
8	Tx1-	Out	LVDS CH1(-)、RZ/G1M の DU0_LVDS_CH1_N ピンに接続
9	Tx1+	Out	LVDS CH1(+), RZ/G1M の DU0_LVDS_CH1_P ピンに接続
10	GND	Power	電源(GND)
11	Tx2-	Out	LVDS CH2(-)、RZ/G1M の DU0_LVDS_CH2_N ピンに接続
12	Tx2+	Out	LVDS CH2(+), RZ/G1M の DU0_LVDS_CH2_P ピンに接続
13	GND	Power	電源(GND)
14	TCLK-	Out	LVDS クロック(-)、RZ/G1M の DU0_LVDS_CLK_N ピンに接続
15	TCLK+	Out	LVDS クロック(+), RZ/G1M の DU0_LVDS_CLK_P ピンに接続
16	GND	Power	電源(GND)
17	Tx3-	Out	LVDS CH3(-)、RZ/G1M の DU0_LVDS_CH3_N ピンに接続
18	Tx3+	Out	LVDS CH3(+), RZ/G1M の DU0_LVDS_CH3_P ピンに接続
19	Reserved	-	GND に 0Ω 抵抗で接続
20	Reserved	-	+3.3V に 0Ω 抵抗で接続

## 15.25. CON26 コンポジットビデオ入力インターフェース

表 15.28 CON26 信号配列

ピン番号	ピン名	I/O	説明
1	VIDEO_IN	In	コンポジットビデオ入力、ビデオデコーダ(ADV7180)の 35(AIN1)ピンに接続
2	GND	Power	電源(GND)

## 15.26. CON30 拡張インターフェース

表 15.29 CON30 信号配列

ピン番号	ピン名	I/O	説明
1	+5V	Power	電源(+5V)
2	+5V	Power	電源(+5V)
3	+3.3V	Power	電源(+3.3V)
4	+3.3V	Power	電源(+3.3V)
5	GND	Power	電源(GND)
6	GND	Power	電源(GND)
7	VIO_R0	In	拡張入出力、RZ/G1M の VIO_R0 ピンに接続
8	VIO_R1	In	拡張入出力、RZ/G1M の VIO_R1 ピンに接続
9	VIO_R2	In	拡張入出力、RZ/G1M の VIO_R2 ピンに接続
10	VIO_R3	In	拡張入出力、RZ/G1M の VIO_R3 ピンに接続
11	VIO_R4	In	拡張入出力、RZ/G1M の VIO_R4 ピンに接続、GND に 100kΩ プルダウン
12	VIO_R5	In	拡張入出力、RZ/G1M の VIO_R5 ピンに接続、GND に 100kΩ プルダウン
13	VIO_R6	In	拡張入出力、RZ/G1M の VIO_R6 ピンに接続、GND に 100kΩ プルダウン
14	VIO_R7	In	拡張入出力、RZ/G1M の VIO_R7 ピンに接続、GND に 100kΩ プルダウン
15	VIO_G0	In	拡張入出力、RZ/G1M の VIO_G0 ピンに接続
16	VIO_G1	In	拡張入出力、RZ/G1M の VIO_G1 ピンに接続
17	VIO_G2	In	拡張入出力、RZ/G1M の VIO_G2 ピンに接続
18	VIO_G3	In	拡張入出力、RZ/G1M の VIO_G3 ピンに接続
19	VIO_G4	In	拡張入出力、RZ/G1M の VIO_RG ピンに接続
20	VIO_G5	In	拡張入出力、RZ/G1M の VIO_G5 ピンに接続
21	VIO_G6	In	拡張入出力、RZ/G1M の VIO_G6 ピンに接続
22	VIO_G7	In	拡張入出力、RZ/G1M の VIO_G7 ピンに接続
23	VIO_DATA0	In	拡張入出力、RZ/G1M の VIO_DATA0 ピンに接続
24	VIO_DATA1	In	拡張入出力、RZ/G1M の VIO_DATA1 ピンに接続

ピン番号	ピン名	I/O	説明
25	VIO_DATA2	In	拡張入出力、RZ/G1M の VIO_DATA2 ピンに接続
26	VIO_DATA3	In	拡張入出力、RZ/G1M の VIO_DATA3 ピンに接続
27	VIO_DATA4	In	拡張入出力、RZ/G1M の VIO_DATA4 ピンに接続
28	VIO_DATA5	In	拡張入出力、RZ/G1M の VIO_DATA5 ピンに接続
29	VIO_DATA6	In	拡張入出力、RZ/G1M の VIO_DATA6 ピンに接続
30	VIO_DATA7	In	拡張入出力、RZ/G1M の VIO_DATA7 ピンに接続
31	GND	Power	電源(GND)
32	GND	Power	電源(GND)
33	VIO_CLK	In	拡張入出力、RZ/G1M の VIO_CLK ピンに接続
34	VIO_CLKENB	In	拡張入出力、RZ/G1M の VIO_CLKENB ピンに接続
35	VIO_FIELD	In	拡張入出力、RZ/G1M の VIO_FIELD ピンに接続
36	VIO_HSYNC#	In	拡張入出力、RZ/G1M の VIO_HSYNC#ピンに接続
37	VIO_VSYNC#	In	拡張入出力、RZ/G1M の VIO_VSYNC#ピンに接続
38	CAM_MCLK	Out	拡張入出力、RZ/G1M の TX4_C(PWM5)ピンに接続
39	GP5_25	In/Out	拡張入出力、RZ/G1M の SCIFB2_TXD_D(GP5_25)ピンに接続
40	GP5_26	In/Out	拡張入出力、RZ/G1M の SCIFB2_RXD_D(GP5_26)ピンに接続
41	GP5_27	In/Out	拡張入出力、RZ/G1M の SCIFB2_SCK_D(GP5_27)ピンに接続
42	GP5_28	In/Out	拡張入出力、RZ/G1M の MSIOF0_SS2_C(GP5_28)ピンに接続
43	GP5_29	In/Out	拡張入出力、RZ/G1M の MSIOF0_SS2_C(GP5_29)ピンに接続
44	PWM0_B	Out	拡張入出力、RZ/G1M の PWM0_B ピンに接続
45	NC	-	未接続
46	NC	-	未接続
47	NC	-	未接続
48	NC	-	未接続
49	+3.3V	Power	電源(+3.3V)
50	GND	Power	電源(GND)

## 15.27. CON31 RTC 外部バックアップインターフェース

表 15.30 CON31 信号配列

ピン番号	ピン名	I/O	機能
1	BAT	Power	リアルタイムクロックの外部バックアップ用電源入力
2	GND	Power	電源(GND)

## 15.28. CON32 電源入力インターフェース

表 15.31 CON32 信号配列

ピン番号	ピン名	I/O	機能
1	+12V	Power	+12V 電源入力(+12V)
2	GND	Power	電源(GND)
3	GND	Power	電源(GND)



図 15.5 AC アダプターの極性マーク



CON32 を使用する場合、同時に CON33 から電源供給しないでください。故障の原因となる可能性があります。

## 15.29. CON33 電源入力インターフェース

表 15.32 CON33 信号配列

ピン番号	ピン名	I/O	機能
1	GND	Power	電源(GND)
2	+12V	Power	+12V 電源入力(+12V)



CON33 を使用する場合、同時に CON32 から電源供給しないでください。故障の原因となる可能性があります。

## 15.30. CON34 電源出力インターフェース

表 15.33 CON34 信号配列

ピン番号	ピン名	I/O	機能
1	+12V_SATA	Power	+12V 電源出力(+12V_SATA)
2	GND	Power	電源(GND)
3	+5V	Power	+5V 電源出力(+5V)

## 15.31. JP2、JP3 バイアスジャンパ

表 15.34 バイアスジャンパの接続

部品番号	説明
JP2	ショートさせるとマイク入力にバイアス電圧供給
JP3	

## 15.32. SW1、SW2 機能選択スイッチ

表 15.35 機能選択スイッチ(SW1)の接続

スイッチ番号	信号名	説明
1	USB/PCI_SEL	RZ/G1M の USB1 の接続先を切替するマルチプレクサのセレクトピンに接続
2	SATA/PCI_SEL	RZ/G1M の SATA1/PCIe0 の接続先を切替するマルチプレクサのセレクトピンに接続
3	MD24	RZ/G1M の A4/MD24 ピンに接続
4	MD0	マルチプレクサを経由して RZ/G1M の DU1_CDE/MD0 ピンに接続
5	MD21	RZ/G1M の A13/MD21 ピンに接続
6	MD11	マルチプレクサを経由して RZ/G1M の DU1_DB5/MD11 ピンに接続
7	MD2	マルチプレクサを経由して RZ/G1M の DU1_EXVSYNC/DU1_VSYNC/MD2 ピンに接続
8	MD3	マルチプレクサを経由して RZ/G1M の DU1_EXHSYNC/DU1_HSYNC/MD3 ピンに接続

表 15.36 機能選択スイッチ(SW2)の接続

スイッチ番号	信号名	説明
1	VIDEO_EN_N	RZ/G1M の VIN0 の接続先を切替するマルチプレクサのセレクトピンに接続
2	Reserved	-
3	RS232C_EN_N	RZ/G1M の SCIFB0 の接続先を切替するマルチプレクサのセレクトピンに接続
4	USB_HOST/FUNC_SEL	RZ/G1M の USB0 の接続先を切替するマルチプレクサのセレクトピンに接続
5	SD/AWLAN_SEL	RZ/G1M の SDHI2 の接続先を切替するマルチプレクサのセレクトピンに接続

スイッチ番号	信号名	説明
6	SPI_EN_N	RZ/G1M の A[20:25]の接続先を切替するマルチプレクサのセレクトピンに接続
7	SD_BOOT_EN_N	RZ/G1M の SSI_SCK1(GP2_3)ピンに接続
8	HERMIT_EN_N	RZ/G1M の HSCK1(GP7_7)ピンに接続

### 15.33. SW3～SW6 ユーザースイッチ

表 15.37 ユーザースイッチの接続

部品番号	説明
SW3	RZ/G1M の V11_HSYNC#(GP5_0)ピンに接続 (ON: Low、OFF: High)
SW4	RZ/G1M の V11_VSYNC#(GP5_1)ピンに接続 (ON: Low、OFF: High)
SW5	RZ/G1M の V11_CLKENB(GP5_2)ピンに接続 (ON: Low、OFF: High)
SW6	RZ/G1M の V11_FIELD(GP5_3)ピンに接続 (ON: Low、OFF: High)

### 15.34. SW7 電源スイッチ

表 15.38 電源スイッチの接続

部品番号	説明
SW7	スイッチングレギュレータのイネーブルピン、eFuse のイネーブルピンに接続 (ON: 電源供給、OFF: 電源切断)

### 15.35. SW8 リセットスイッチ

表 15.39 リセットスイッチの接続

部品番号	説明
SW8	リセット IC 等を経由して RZ/G1M の PRESET#ピンに接続 (ON: リセット状態、OFF: リセット解除)

### 15.36. SW9 SD ブートデバイス選択スイッチ

表 15.40 SD ブートデバイス選択スイッチの接続

部品番号	説明
SW9	RZ/G1M の HSCK0_C(GP5_31)ピンに接続 (ON: Low、OFF: High)

### 15.37. LED1、LED2 シリアル LED

表 15.41 シリアル LED の接続

部品番号	説明
LED1	USB シリアル変換 IC の CBUS0(TXLED#)ピンに接続 (点滅: データ送信中)
LED2	USB シリアル変換 IC の CBUS1(RXLED#)ピンに接続 (点滅: データ受信)



## 15.38. LED4~LED7 ユーザー LED

表 15.42 ユーザー LED の接続

部品番号	説明
LED4	RZ/G1M の V11_CLK(GP5_4)に接続 (Low: 消灯、High: 点灯)
LED5	RZ/G1M の V11_DATA0(GP5_5)に接続 (Low: 消灯、High: 点灯)
LED6	RZ/G1M の V11_DATA1(GP5_6)に接続 (Low: 消灯、High: 点灯)
LED7	RZ/G1M の V11_DATA2(GP5_7)に接続 (Low: 消灯、High: 点灯)

## 15.39. LED8 電源 LED

表 15.43 電源 LED の接続

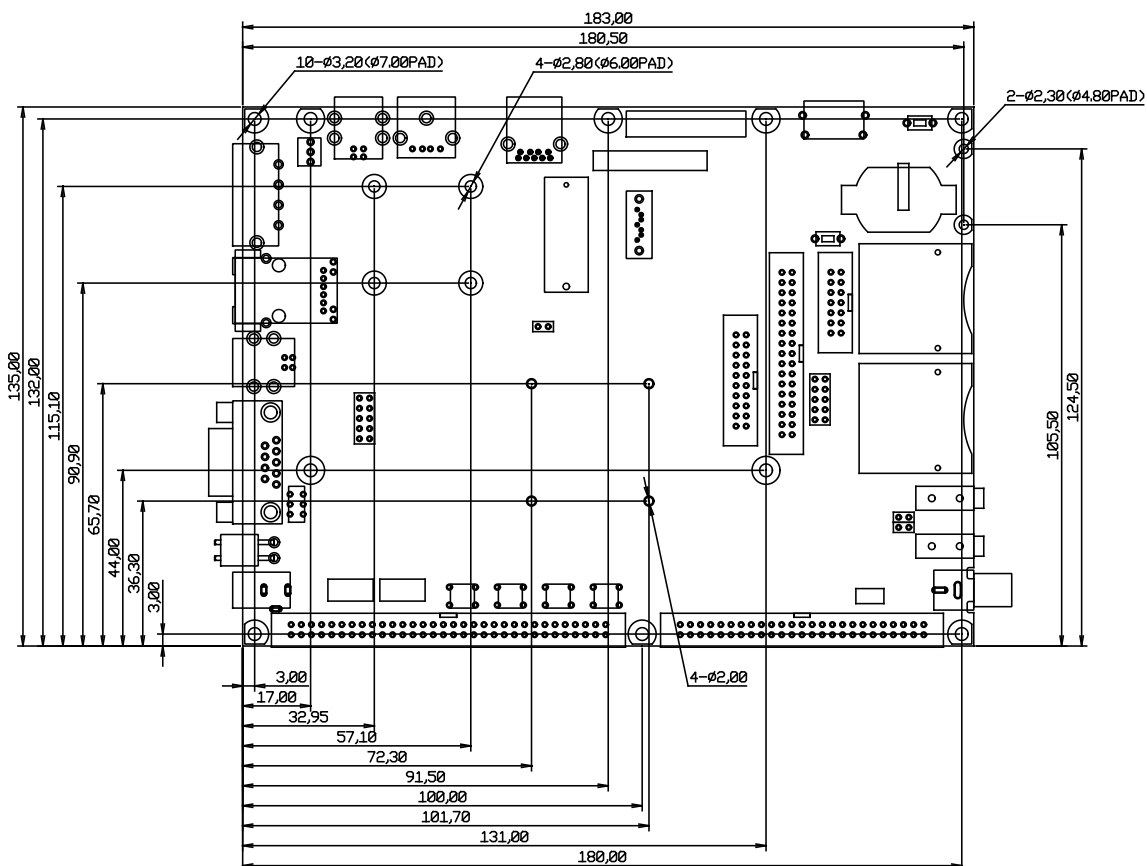
部品番号	説明
LED8	+3.3V が供給されると点灯

## 15.40. LED9 警告 LED

表 15.44 警告 LED の接続

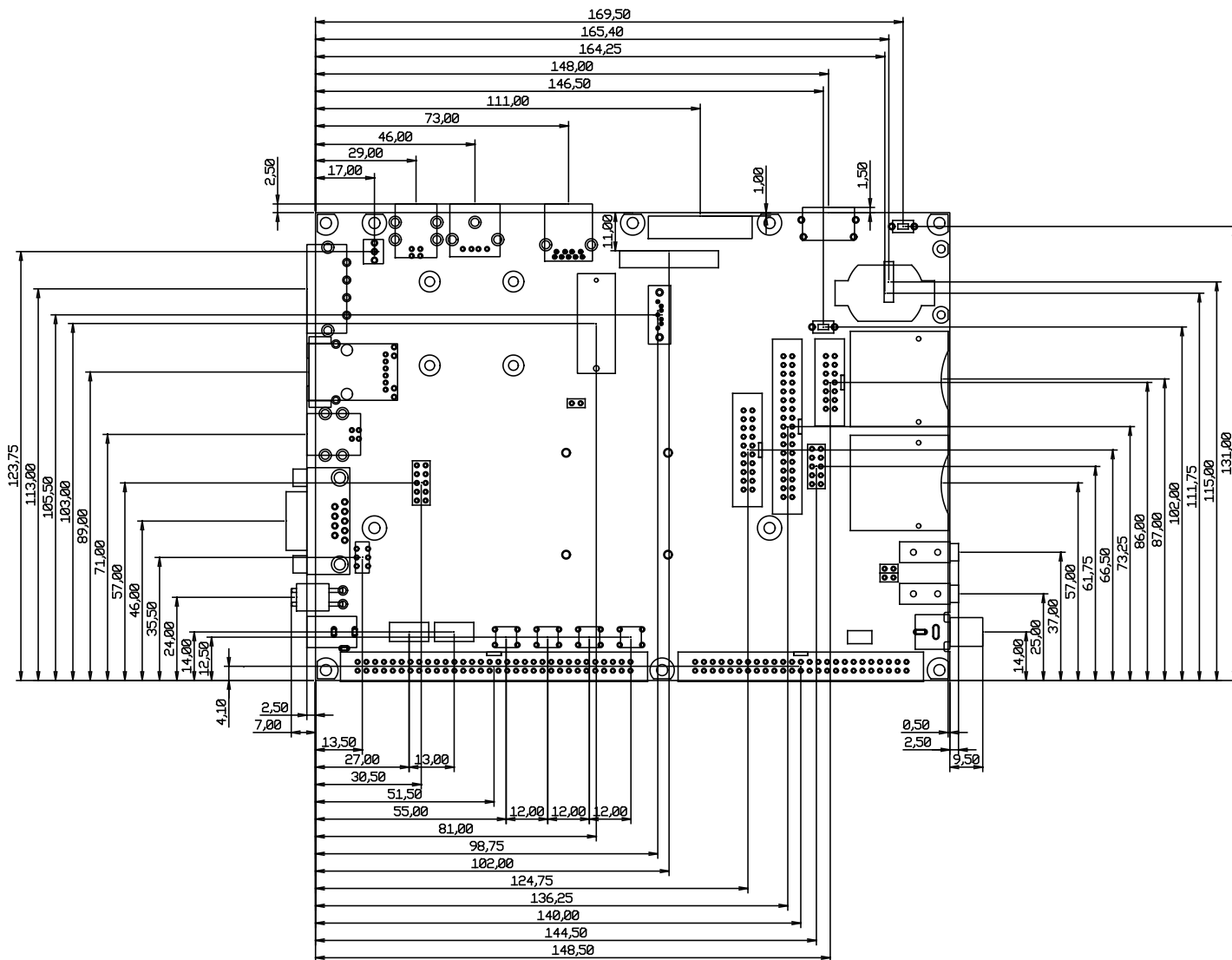
部品番号	説明
LED9	RZ/G1M の SSI_WS1(GP2_4)に接続 (Low: 消灯、High: 点灯)

# 16. 基板形状図



[Unit : mm]

図 16.1 基板形状および固定穴寸法



[Unit : mm]

図 16.2 コネクタ中心寸法

# 17. オプション品

本章では、Armadillo-EVA 1500 関連のオプション品について説明します。

表 17.1 Armadillo-EVA 1500 関連のオプション品

名称	型番
Armadillo-EVA 1500 LCD オプションセット	SA-AE15-LCD50EXT-L00
Armadillo-WLAN モジュール(AWL13)	AWL13-U00Z

## 17.1. Armadillo-EVA 1500 LCD オプションセット

### 17.1.1. 概要

Armadillo-EVA 1500 LCD オプションセットは、Armadillo-EVA 1500 の LCD インターフェース (CON23) を評価するための、タッチパネル LCD、LCD 拡張ボード、ネジ、ケーブルをひとまとめにしています。

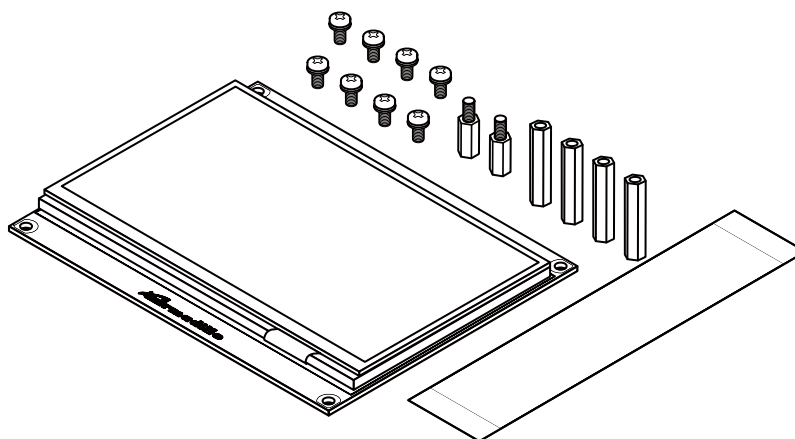


図 17.1 Armadillo-EVA 1500 LCD オプションセット

タッチパネル LCD の主な仕様は次のとおりです。

表 17.2 タッチパネル LCD の仕様

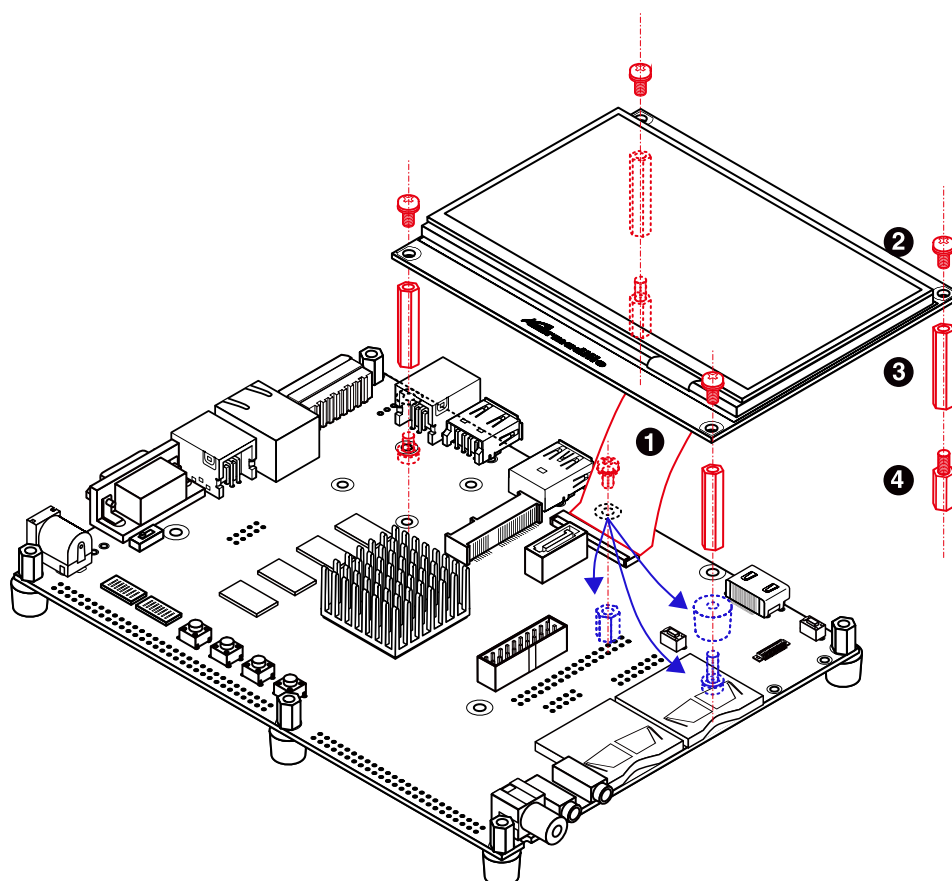
型番	AM-800480L1TMQW-TN0H
メーカー名	AMPIRE
タイプ	TFT
サイズ	5 インチ
解像度	WVGA(800×480 ピクセル)
バックライト	LED
色数	True Color(1677 万色、24bit)
タッチパネル	静電容量方式、マルチタッチ対応



LCD 拡張ボードの回路図/部品表は「アットマークテクノ ユーザーズサイト」で「購入者向けの限定公開データ」としてダウンロード可能です。

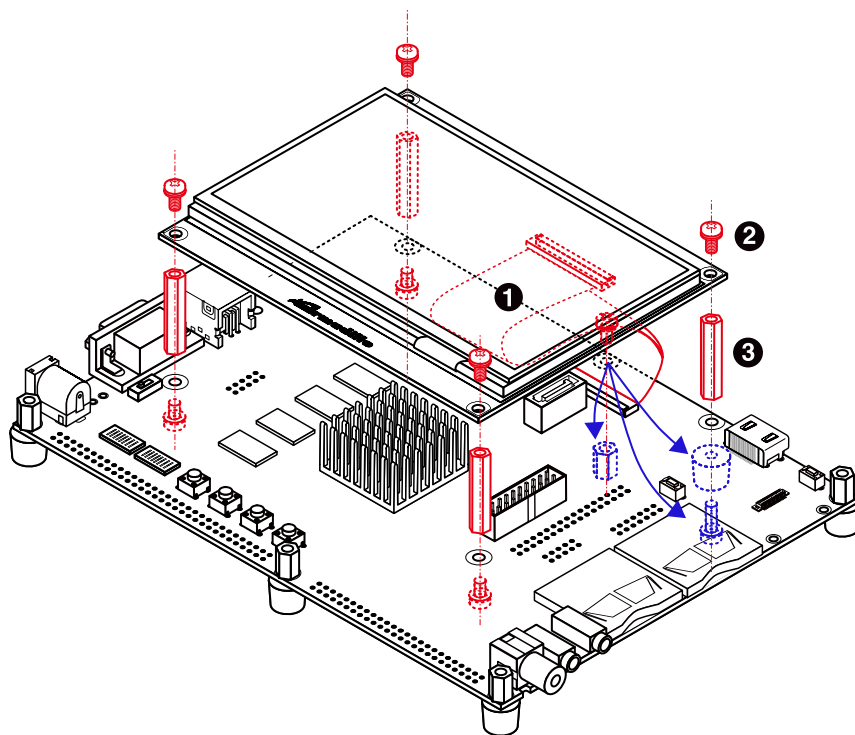
### 17.1.2. 組み立て

オシロスコープやテスターで信号の確認をしやすいように、Armadillo-EVA 1500 の上に LCD が被さらないように組み立てる方法と、コンパクトに組み立てる方法の 2 通りが可能です。「図 17.2. Armadillo-EVA 1500 と LCD の組み立て(信号確認)」「図 17.3. Armadillo-EVA 1500 と LCD の組み立て(コンパクト)」を参考に組み立ててください。



- ❶ フレキシブルフラットケーブル
- ❷ なべ小ねじ(M3、L=5mm)×6
- ❸ 金属スペーサ(M3、L=24mm、両メネジ六角)×4
- ❹ 金属スペーサ(M3、L=10mm、オネジメネジ六角)×2

図 17.2 Armadillo-EVA 1500 と LCD の組み立て(信号確認)



- ❶ フレキシブルフラットケーブル
- ❷ なべ小ねじ(M3、L=5mm)×8
- ❸ 金属スペーサ(M3、L=24mm、両メネジ六角)×4

図 17.3 Armadillo-EVA 1500 と LCD の組み立て(コンパクト)



フレキシブルフラットケーブルを取り付けづらい為、ゴム足を移動させています。基板がたわみ、断線等の原因となるため、ゴム足が固定されていたねじ穴には、必ず図のようにスペーサを取り付けてください。

## 17.2. Armadillo-WLAN モジュール(AWL13)

### 17.2.1. 概要

AWL13 は、ローム社製無線 LAN デバイス「BP3580」を搭載した小型無線 LAN モジュールです。IEEE 802.11b/g/n 規格対応の無線通信機能と IEEE 802.11i 規格対応のセキュアエンジン機能をハードウェアとして実装しています。

Armadillo-EVA 1500 の WLAN インターフェース(CON5)に SDIO モードで接続することができます。

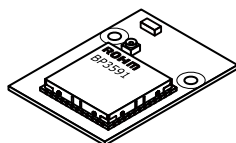


図 17.4 Armadillo-WLAN モジュール(AWL13)

AWL13 の主な仕様は次のとおりです。AWL13 の詳細につきましては、Armadillo-WLAN(AWL13) 製品情報 [http://armadillo.atmark-techno.com/armadillo-wlan/awl13] をご確認ください。

表 17.3 AWL13 の仕様

デバイス	ローム BP3580 (BU1805GU チップセット搭載)	
無線 LAN 規格	IEEE 802.11b、IEEE 802.11g、IEEE 802.11n、IEEE 802.11i	
送受信周波数	2400MHz~2483.5MHz(ch1~ch13)	
周波数偏差	±25ppm 以下	
送信電力 <sup>[a]</sup>	IEEE 802.11b	15dBm±2dBm
	IEEE 802.11g	13dBm±2dBm
	IEEE 802.11n	12dBm±2dBm
通信速度 <sup>[b]</sup>	IEEE 802.11b	1, 2, 5.5, 11Mbps
	IEEE 802.11g	6, 9, 12, 18, 24, 36, 48, 54Mbps
	IEEE 802.11n	(800ns GI <sup>[c]</sup> ): 6.5, 13, 19.5, 26, 39, 52, 58.5, 65Mbps (400ns GI <sup>[c]</sup> ): 7.2, 14.5, 21.7, 28.9, 43.3, 57.8, 65, 72.2Mbps
受信感度 <sup>[a]</sup>	IEEE 802.11b	-94dBm @1Mbps、-89dBm @11Mbps
	IEEE 802.11g	-92dBm @6Mbps、-73dBm @54Mbps
	IEEE 802.11n	-90dBm @6.5Mbps、-68dBm @72.2Mbps
アクセス方式	インフラストラクチャモード(AP <sup>[d]</sup> 、STA <sup>[e]</sup> 、アドホックモード)	
セキュリティ機能	WEP(64bit、128bit)、WPA-PSK(TKIP、AES)、WPA2-PSK(TKIP、AES)	
無線 LAN 機能	WPS: プッシュボタン方式、PIN 方式(Enrollee のみ)、オートレート(STA <sup>[g]</sup> のみ)、自動チャンネル選択(STA <sup>[g]</sup> のみ)	
ホストインターフェース	UART(~921600bps)	
	SDIO Ver.2.00(High Speed、Normal Speed 対応)	
	USB2.0(High Speed、Full Speed 対応)	
インターフェースコネクタ	モジュール側	AXK5F34347YG(パナソニック電工)
	ホスト側	AXK6F34347YG(パナソニック電工)
電源電圧	DC3.3V±0.2V	
消費電流(Typ.)	送信時	300mA
		340mA(USB 使用)
	受信時	200mA
		240mA(USB 使用)
スリープ時	500μA	
動作環境	温度	-40°C~85°C
	湿度	85%以下(結露なきこと)
基板サイズ	24.0 x 33.1mm	

<sup>[a]</sup>送信電力、受信感度はアンテナ給電点(BP3580 アンテナ端子)での性能値となります。

<sup>[b]</sup>規格による理論値上の速度であり、ご利用の環境や接続機器などにより実際のデータ速度は異なります。

<sup>[c]</sup>ガードインターバル

<sup>[d]</sup>AP=アクセスポイント

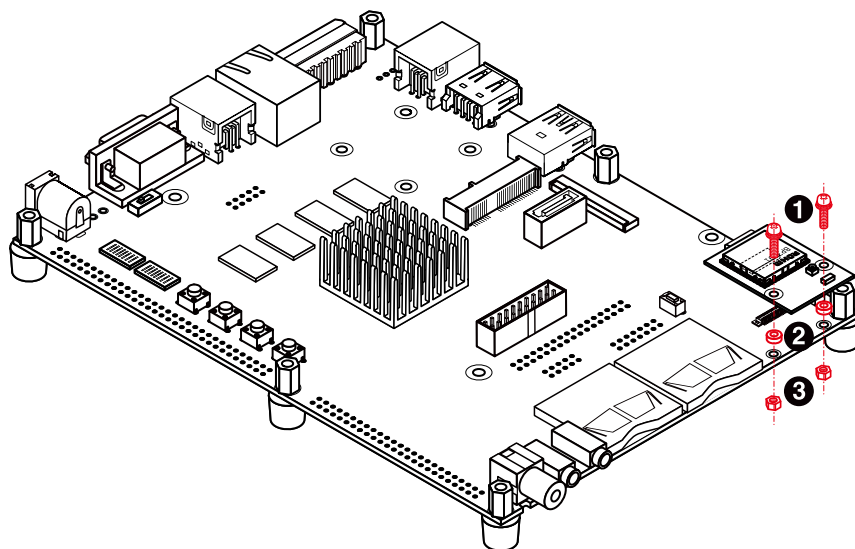
<sup>[e]</sup>ファームウェア v4.3.2 以降で有効です。対応するデバイスドライバは v3.0.2 以降です。

<sup>[f]</sup>STA の接続可能局数は 5 台です。

<sup>[g]</sup>STA=ステーション

## 17.2.2. 組み立て

「図 17.5. AWL13 の組み立て」を参考に組み立ててください。



- ❶ なべ小ねじ スプリングワッシャー、小径平ワッシャー付(M2、L=8mm)×2
- ❷ 金属スペーサ(M2、L=1.5mm、直径=4mm)×2
- ❸ ナット(M2、L=1.6mm、平径=4mm)×2

図 17.5 AWL13 の組み立て



# 18. Howto

本章では、Armadillo-EVA 1500 の Linux カーネルイメージをカスタマイズする方法や、オプション品を使用する方法について説明します。

## 18.1. Linux カーネルイメージをカスタマイズする

コンフィギュレーションを変更して、Linux カーネルイメージをカスタマイズする方法を説明します。

### 手順 18.1 Linux カーネルイメージのカスタマイズ

#### 1. ソースコードの準備

Linux カーネルのソースコードアーカイブを準備し展開します。展開後、linux-3.4-ae1500 ディレクトリに移動します。

```
[ATDE ~]$ ls
linux-3.4-ae1500-at.tar.gz
[ATDE ~]$ tar zxf linux-3.4-ae1500-at.tar.gz
[ATDE ~]$ ls
linux-3.4-ae1500-at linux-3.4-ae1500-at.tar.gz
[ATDE ~]$ cd linux-3.4-ae1500-at
[ATDE ~/linux-3.4-ae1500-at]$
```

#### 2. デフォルトコンフィギュレーションの適用

Armadillo-EVA 1500 のデフォルトコンフィギュレーションを適用します。

```
[ATDE ~/linux-3.4-ae1500-at]$ make ARCH=arm armadilloeva1500_defconfig
```

#### 3. コンフィギュレーションの開始

Linux カーネルコンフィギュレーションを開始します。

```
[ATDE ~/linux-3.4-ae1500-at]$ make ARCH=arm menuconfig
```

#### 4. カーネルコンフィギュレーションの変更

Linux カーネルコンフィギュレーションメニューが表示されます。カーネルコンフィギュレーションを変更後、「Exit」を選択して「Do you wish to save your new kernel configuration? <ESC><ESC> to continue.」で「Yes」とし、カーネルコンフィギュレーションを確定します。

```

----- Linux/arm 3.4.81-ae1500-at1 Kernel Configuration -----
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

-----
[*] Patch physical to virtual translations at runtime
    General setup --->
[*] Enable loadable module support --->
[*] Enable the block layer --->
    System Type --->
    Bus support --->
    Kernel Features --->
    Boot options --->
    CPU Power Management --->
    Floating point emulation --->

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

```



Linux Kernel Configuration メニューで"/"キーを押下すると、カーネルコンフィギュレーションの検索を行うことができます。カーネルコンフィギュレーションのシンボル名(の一部)を入力して"Ok"を選択すると、部分一致するシンボル名を持つカーネルコンフィギュレーションの情報が一覧されます。

## 5. ビルド

コンフィギュレーションの確定後にビルドを行います。

```
[ATDE ~/linux-3.4-ae1500-at]$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf-
```

## 6. イメージファイルの生成確認

ビルドが終了すると、カスタマイズされたイメージファイル(arch/arm/boot/Image)が作成されています。

```
[ATDE ~/linux-3.4-ae1500-at]$ ls arch/arm/boot/Image
arch/arm/boot/Image
```

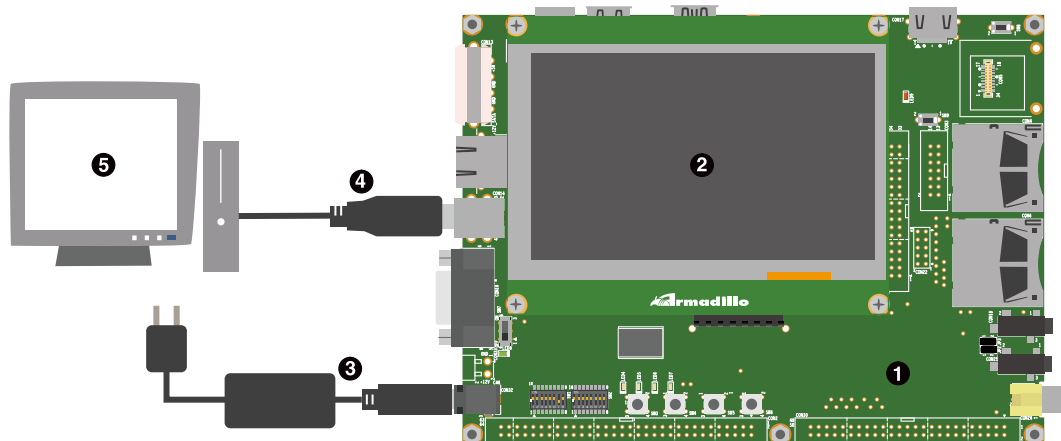
# 18.2. Armadillo-EVA 1500 LCD 拡張ボードを使用する

オプション品の Armadillo-EVA 1500 LCD オプションセットに含まれる Armadillo-EVA 1500 LCD 拡張ボードを使用する方法について説明します。以降の説明では、Armadillo-EVA 1500 LCD 拡張ボードを LCD 拡張ボードと表記します。

LCD 拡張ボードを使用するためには、コンフィギュレーションを変更した Linux カーネルイメージファイルを作成する必要があります。

## 18.2.1. 接続方法

Armadillo-EVA 1500 と周辺装置の接続例を次に示します。



- ❶ Armadillo-EVA 1500
- ❷ Armadillo-EVA 1500 LCD 拡張ボード
- ❸ AC アダプタ(12V/5A)<sup>[1]</sup>
- ❹ USB 2.0 ケーブル(A-B タイプ)<sup>[1]</sup>
- ❺ 作業用 PC

## 18.2.2. ビルド手順

カーネルコンフィギュレーションを変更して、LCD 拡張ボードに対応した Linux カーネルイメージファイルを作成します。

### 18.2.2.1. LCD 拡張ボード対応イメージのビルド

カーネルコンフィギュレーションを変更して、LCD 拡張ボードを使用可能にします。「18.1. Linux カーネルイメージをカスタマイズする」を参照して次のコンフィギュレーションを有効化します。

```

Device Drivers --->
  Misc devices --->
    <*> R-Car PWM driver <CONFIG_RCAR_PWM>
  Input device support --->
    -*> Generic input layer (needed for keyboard, mouse, ...) <CONFIG_INPUT>
    <*> Event interface <CONFIG_INPUT_EVDEV>
    *** Input Device Drivers ***
    [*] Touchscreens ---> <CONFIG_INPUT_TOUCHSCREEN>
      <*> Sitronix ST1232 touchscreen controllers <CONFIG_TOUCHSCREEN_ST1232>
      [ ] Enable single touch event for ST1232 <CONFIG_TOUCHSCREEN_ST1232_SINGLETOUCH>
  Graphics support --->
    <*> Direct Rendering Manager (XFree86 4.1.0 and higher DRI support) ---> <CONFIG_DRM>
    
```

<sup>[1]</sup>Armadillo-EVA 1500 評価セットに同梱されているもの

[*] Enable FBDev CRTC select	<CONFIG_DRM_FBDEV_CRTC>
(1) FBDev CRTC number	<CONFIG_DRM_FBDEV_CRTC_NUM>
<*> DRM Support for R-Car Display Unit	<CONFIG_DRM_RCAR_DU>
[*] R-Car DU LCD Support	<CONFIG_DRM_RCAR_LCD>
(800) Horizontal screen resolution	
(480) Vertical screen resolution	
[*] Backlight & LCD device support --->	<CONFIG_BACKLIGHT_LCD_SUPPORT>
<*> Lowlevel Backlight controls	<CONFIG_BACKLIGHT_CLASS_DEVICE>
<*> Generic PWM based Backlight Driver	<CONFIG_BACKLIGHT_PWM>

カーネルコンフィギュレーションの確定後、ビルドしてイメージファイルを作成します。

### 18.2.3. インストール


「9.3.2. eMMC への書き込み方法」を参照して、「18.2.2. ビルド手順」で作成した Linux カーネルイメージを、eMMC にインストールします。

### 18.2.4. 動作確認方法

#### 18.2.4.1. LCD への画像表示

Armadillo-EVA 1500 に電源を投入すると、次のように LCD に Linux カーネルの起動ロゴが表示されます。



 表示されるペンギンの数は CPU の数を現しています。機能選択スイッチの JTAG 設定を Coresight デバックポート (SW1.5 を OFF、SW1.6 を ON) に設定した場合は CPU がシングルコアで動作するため、ペンギンが 1 匹のみ表示されます。

#### 18.2.4.2. バックライトの輝度調整

LCD 拡張ボード 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 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 18.1 輝度設定に使用する sysfs ファイル

ファイル	説明
brightness	0(最高輝度) ~ max_brightness(消灯)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。

**brightness に書きこむ数値の最大値を取得する**

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

**最高輝度で点灯させる**

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

**18.2.4.3. タッチスクリーンのイベント確認**

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 (EV_SYN)
  Event type 1 (EV_KEY)
  Event type 3 (EV_ABS)
    Event code 48 (ABS_MT_TOUCH_MAJOR)
      Value      0
      Min        0
      Max       255
    Event code 53 (ABS_MT_POSITION_X)
      Value      0
      Min        0
      Max       799
    Event code 54 (ABS_MT_POSITION_Y)
      Value      0
      Min        0
      Max       479
Properties:
Testing ... (interrupt to exit)
Event: time 946694115.241012, type 3 (EV_ABS), code 48 (ABS_MT_TOUCH_MAJOR), value 42 ❶
Event: time 946694115.241016, type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X), value 511 ❷
Event: time 946694115.241019, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y), value 259 ❸
```

```
Event: time 946694115.241022, ++++++ SYN_MT_REPORT ++++++
Event: time 946694115.241025, ----- SYN_REPORT -----
:
[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 18.1 タッチスクリーンのイベント確認

## 18.3. Armadillo-WLAN(AWL13)を使用する

オプション品の Armadillo-WLAN(AWL13)を使用する方法について記載します。以降の説明では、Armadillo-WLAN(AWL13)を AWL13 と表記します。

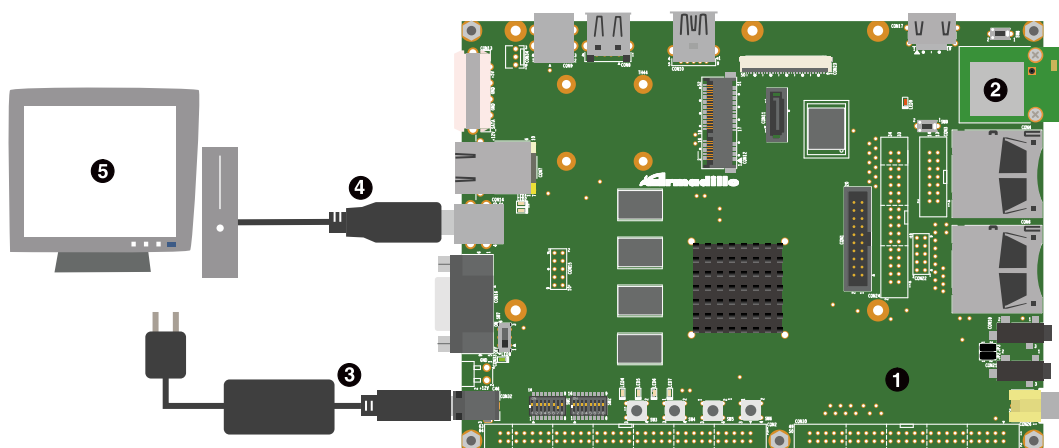
AWL13 を使用するためには、Armadillo-EVA 1500 に電源を投入する前に機能選択スイッチの SD/WLAN 設定(SW2.5)を OFF に設定する必要があります。



WLAN インターフェース(CON5)と SD インターフェース(CON4)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。

### 18.3.1. 接続方法

Armadillo-EVA 1500 と周辺装置の接続例を次に示します。



- ① Armadillo-EVA 1500
- ② Armadillo-WLAN(AWL13)
- ③ AC アダプタ(12V/5A)<sup>[2]</sup>
- ④ USB 2.0 ケーブル(A-B タイプ)<sup>[2]</sup>
- ⑤ 作業用 PC

### 18.3.2. ビルド手順

AWL13 用 Linux デバイスドライバをビルドします。AWL13 用 Linux デバイスドライバのビルドには、ビルド済みの Linux カーネルソースコードが必要です。「9.1. Linux カーネルをビルドする」を参照して Linux カーネルをビルドしてください。

使用するソースコードは、評価セット付属の DVD に収録されています。最新版のソースコードは、Armadillo サイトからダウンロードすることができます。ビルド手順を次に示します。

#### 手順 18.2 AWL13 Linux デバイスドライバをビルド

##### 1. ソースコードの準備

AWL13 デバイスドライバのソースコードアーカイブを準備し展開します。次の例では、Linux カーネルと同じディレクトリに展開しています。展開後、awl13 ディレクトリに移動します。

```
[ATDE ~]$ ls
awl13-[version].tar.gz  linux-3.4-ae1500-at[version]  linux-3.4-ae1500-
at[version].tar.gz
[ATDE ~]$ tar zxf awl13-[version].tar.gz
[ATDE ~]$ ls
awl13-[version]  awl13-[version].tar.gz  linux-3.4-ae1500-at[version]  linux-3.4-ae1500-
at[version].tar.gz
[ATDE ~]$ cd awl13-[version]
```

<sup>[2]</sup>Armadillo-EVA 1500 評価セットに同梱されているもの

## 2. ビルド

AWL13 デバイスドライバをビルドします。

```
[ATDE ~/awl13-[version]]$ AWL13_DEVICE=SDIO make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- KERNELDIR=./linux-3.4-ae1500-at[version]/
```



## 3. イメージファイルの生成確認

ビルドが終了すると、AWL13 デバイスドライバ(src/awl13\_sdio.ko)が作成されています。

```
[ATDE ~/awl13-[version]]$ ls src/awl13_sdio.ko
src/awl13_sdio.ko
```

### 18.3.3. AWL13 デバイスドライバのインストール

AWL13 デバイスドライバをインストールします。次のようにコマンドを実行します。

```
[armadillo ~]$ ls
awl13_sdio.ko
[armadillo ~]$ insmod awl13_sdio.ko
awl13: RX Transmission mode SDINT HT
mmc2: registerd "awl13" device as awlan0
awl13: Version 3.0.2 Load.
```

### 18.3.4. ファームウェアのロード

AWL13 ファームウェアをロードします。事前にファームウェアのロードに使用する Wireless Tools をインストールしておきます。

```
[armadillo ~]# apt-get install wireless-tools
```

AWL13 ファームウェア(SDIO 用 STA モード対応)をロードします。次のようにコマンドを実行します。

```
[armadillo ~]$ ls
fwimage[version] STA_SDIO.bin
[armadillo ~]$ cat fwimage[version] STA_SDIO.bin > /sys/module/awl13_sdio/awlan0/firmware
[armadillo ~]$ iwpriv awlan0 fwload
[armadillo ~]$ iwpriv awlan0 fwsetup
[armadillo ~]$ cat /sys/module/awl13_sdio/awlan0/firmware_version
[version]
```

### 18.3.5. 無線設定

インフラストラクチャ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。確実に無線通信を行いたい場合は、「図 6.2. 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、無線 LAN 通信が不安定になる場合があります。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」を参照してください。

### 18.3.5.1. 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。

「18.3.3. AWL13 デバイスドライバのインストール」および「18.3.4. ファームウェアのロード」が完了している場合は、Linux の無線 LAN インターフェース awlan0 を通して AWL13 の無線 LAN 機能が使用できます。

以降の設定は、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「18.3.5.1.1. インフラストラクチャモード: WPA-PSK/WPA2-PSK」を、WEP の場合は「18.3.5.1.2. インフラストラクチャモード: WEP」を参照してください。

#### 18.3.5.1.1. 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、設定パラメーターを「表 18.2. インフラストラクチャモード: WPA-PSK/WPA2-PSK パラメーター例」に、設定手順を「図 18.2. 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 18.2 インフラストラクチャモード: WPA-PSK/WPA2-PSK パラメーター例

項目	設定値
ESSID	myessid
パスフレーズ	my passphrase
暗号化方式	WPA2-PSK(AES)
IP アドレス	192.168.0.1

```
[armadillo ~]# iwconfig awlan0 essid myessid ❶  
[armadillo ~]# iwconfig awlan0 mode Managed ❷  
[armadillo ~]# iwpriv awlan0 set_psk mypassphrase ❸  
[armadillo ~]# iwpriv awlan0 set_cryptmode WPA2-AES ❹  
[armadillo ~]# ifconfig awlan0 192.168.0.1 up ❺
```

- ❶ SSID に myessid を設定します。
- ❷ 通信モードをインフラストラクチャモードに設定します。
- ❸ パスフレーズに mypassphrase を設定します。
- ❹ 暗号化方式に WPA2-PSK(AES) を設定します。
- ❺ 無線 LAN インターフェースを有効化します。IP アドレスは 192.168.0.1 を設定します。

### 図 18.2 インフラストラクチャモード: WPA2-PSK(AES)設定手順

設定確認手順を「図 18.3. インフラストラクチャモード: WPA2-PSK(AES)設定確認手順」に示します。  
`iwconfig` コマンドの結果で、**Access Point:**に BSSID(アクセスポイントの MAC アドレス)が表示されていれば、アクセスポイントへの接続は完了です。

```
[armadillo ~]# iwconfig awlan0 ❶  
awlan0 IEEE 802.11bgn ESSID:"myessid"  
Mode:Managed Frequency:2.412 GHz Access Point: XX:XX:XX:XX:XX:XX  
Bit Rate=65 Mb/s  
Encryption key:off  
Power Management:off  
Link Signal level=-37 dBm  
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0  
Tx excessive retries:0 Invalid misc:0 Missed beacon:0  
[armadillo ~]# ifconfig awlan0 ❷  
awlan0 Link encap:Ethernet HWaddr 00:1d:12:xx:xx:xx  
inet addr:192.168.0.1 Bcast:192.168.0.255 Mask:255.255.255.0  
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
RX packets:47 errors:0 dropped:0 overruns:0 frame:0  
TX packets:8 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:1000  
RX bytes:6760 (6.6 KiB) TX bytes:1196 (1.1 KiB)
```

- ❶ 無線 LAN 固有の状態を表示します。
- ❷ 無線 LAN インターフェースの状態を表示します。

### 図 18.3 インフラストラクチャモード: WPA2-PSK(AES)設定確認手順

#### 18.3.5.1.2. 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、設定パラメーターを「表 18.3. インフラストラクチャモード: WEP パラメーター例」に、設定手順を「図 18.4. インフラストラクチャモード: WEP 設定手順」に示します。

表 18.3 インフラストラクチャモード: WEP パラメーター例

項目	設定値
ESSID	myssid
WEP キー(WEP-64 ビット)	1234567890
IP アドレス	192.168.0.1

```
[armadillo ~]# iwconfig awlan0 essid myssid ❶
[armadillo ~]# iwconfig awlan0 mode Managed ❷
[armadillo ~]# iwconfig awlan0 enc 1234567890 ❸
[armadillo ~]# ifconfig awlan0 192.168.0.1 up ❹
```

- ❶ SSID に myssid を設定します。
- ❷ 通信モードをインフラストラクチャモードに設定します。
- ❸ WEP キーに 1234567890 を設定します。暗号化方式は、キー長に応じて適切に設定されます。
- ❹ 無線 LAN インターフェースを有効化します。IP アドレスは 192.168.0.1 を設定します。

図 18.4 インフラストラクチャモード: WEP 設定手順

設定確認手順を「図 18.5. インフラストラクチャモード: WEP 設定確認手順」に示します。iwconfig コマンドの結果で、Access Point:に BSSID(アクセスポイントの MAC アドレス)が表示されていれば、アクセスポイントへの接続は完了です。

```
[armadillo ~]# iwconfig awlan0 ❶
awlan0 IEEE 802.11bgn ESSID:"myssid"
Mode:Managed Frequency:2.412 GHz Access Point: XX:XX:XX:XX:XX:XX
Bit Rate=65 Mb/s
Encryption key:1234-5678-90
Power Management:off
Link Signal level=-37 dBm
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0

[armadillo ~]# ifconfig awlan0 ❷
awlan0 Link encap:Ethernet HWaddr 00:1d:12:xx:xx:xx
inet addr:192.168.0.1 Bcast:192.168.0.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:47 errors:0 dropped:0 overruns:0 frame:0
TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:6760 (6.6 KiB) TX bytes:1196 (1.1 KiB)
```

- ❶ 無線 LAN 固有の状態を表示します。
- ❷ 無線 LAN インターフェースの状態を表示します。

図 18.5 インフラストラクチャモード: WEP 設定確認手順

### 18.3.6. 動作確認方法

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。次の例では、通信機器が「192.168.0.10」という IP アドレスを持っていると想定しています。

```
[armadillo ~]# ping 192.168.0.10
```



eth0 または usb0 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 awlan0 が使用されない場合があります。確実に awlan0 を使用させる場合は、awlan0 以外のネットワークインターフェースを無効化してください。



awlan0 を使用した無線通信を行った場合、Linux カーネルが次のような警告メッセージを出力する場合があります。

```
sh_mobile_sdhi sh_mobile_sdhi.1: timeout waiting for SD bus idle
```

これは、SDHI レジスタへのアクセスが正しいシーケンスとならない場合があるためです。警告メッセージが出力された場合でもデータ化けなどの通信不具合は確認されていません。

## 18.4. USB ガジェットを使用する

USB ガジェットを使用する方法について説明します。USB デバイス機能を提供する USB ガジェットドライバには様々な種類のものが用意されていますが、ここでは例としてイーサネットガジェットドライバ<sup>[3]</sup>を有効化した Linux カーネルイメージファイルを作成し、Armadillo-EVA 1500 を USB デバイスとして使用する方法を紹介します。

USB ガジェットを使用するためには、Armadillo-EVA 1500 に電源を投入する前に機能選択スイッチの USB 2.0 設定(SW2.4)を OFF に設定する必要があります。

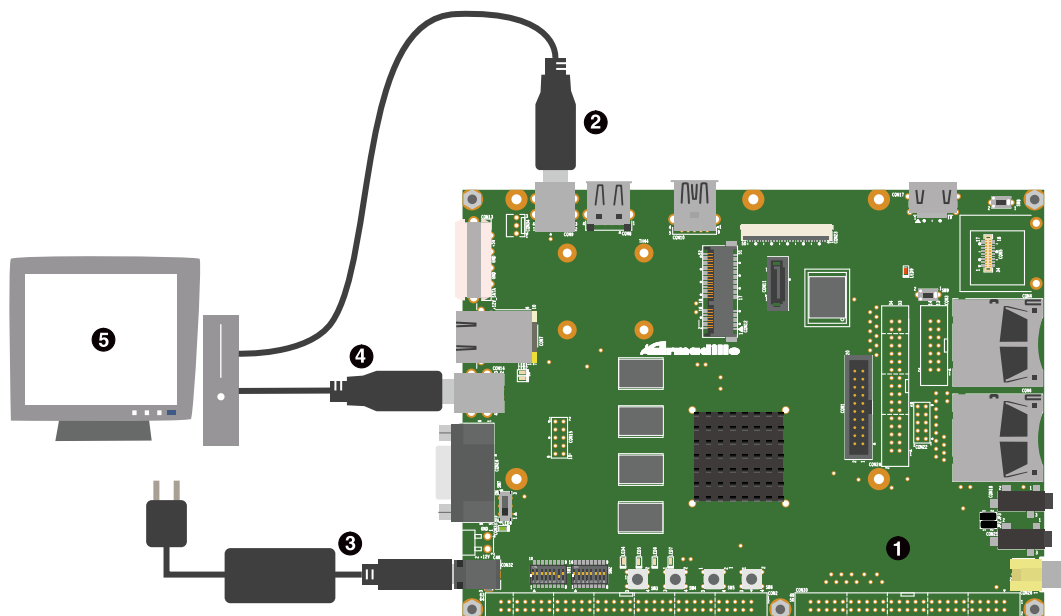


USB 2.0 ファンクションインターフェース(CON9)と USB 2.0 ホストインターフェース(CON8)は排他になっており、どちらか一方しか使用することができません。

### 18.4.1. 接続方法

Armadillo-EVA 1500 と周辺装置の接続例を次に示します。

<sup>[3]</sup>イーサネット機能を提供する USB ガジェットドライバ



- ① Armadillo-EVA 1500
- ② USB2.0 ケーブル(A-B タイプ)
- ③ AC アダプタ(12V/5A)<sup>[4]</sup>
- ④ USB2.0 ケーブル(A-B タイプ)<sup>[4]</sup>
- ⑤ 作業用 PC

## 18.4.2. ビルド手順

カーネルコンフィギュレーションを変更して、イーサネットガジェットに対応した Linux カーネルイメージファイルを作成します。

### 18.4.2.1. イーサネットガジェット対応イメージのビルド

カーネルコンフィギュレーションを変更して、イーサネットガジェットを使用可能にします。「18.1. Linux カーネルイメージをカスタマイズする」を参照して次のようにコンフィギュレーションを変更します。

```

Device Drivers --->
[*] USB support --->                                <CONFIG_USB_SUPPORT>
    *** USB Miscellaneous drivers ***
    <*> USB Gadget Support --->                        <CONFIG_USB_GADGET>
        <*> USB Peripheral Controller (Renesas R8A66597 USB Peripheral Controller) --->
            (X) Renesas R8A66597 USB Peripheral Controller    <CONFIG_USB_R8A66597>
            [*] on-chip USBHS type with extended bulk endpoints support
                <CONFIG_USB_R8A66597_TYPE_BULK_PIPES_12>
        <*> USB Gadget Drivers (Ethernet Gadget (with CDC Ethernet support))
            (X) Ethernet Gadget (with CDC Ethernet support)    <CONFIG_USB_ETH>
            ( ) Serial Gadget (with CDC ACM and CDC OBEX support) <CONFIG_USB_G_SERIAL>
                :
    
```

<sup>[4]</sup>Armadillo-EVA 1500 評価セットに同梱されているもの

```
*** OTG and related infrastructure ***
<*> GPIO based peripheral-only VBUS sensing 'transceiver' <CONFIG_USB_GPIO_VBUS>
```

カーネルコンフィギュレーションの確定後、make コマンドを実行してイメージファイルを作成します。

### 18.4.3. インストール

「9.3.2. eMMC への書き込み方法」を参照して、「18.2.2. ビルド手順」で作成した Linux カーネルイメージを、eMMC にインストールします。

### 18.4.4. 動作確認方法

ATDE5 と Armadillo-EVA 1500 で、イーサネットガジェットを使用した PING 通信を行います。

事前に「4.2.2. 取り外し可能デバイスの使用」を参照して、ATDE5 に Armadillo-EVA 1500 を接続する必要があります。Armadillo-EVA 1500 のデバイス名は「Netchip RNDIS/Ethernet Gadget」と表示されます。

接続が完了すると、ATDE5 では自動的にネットワークインターフェース"usb0"が有効化されます。有効化が完了し、リンクローカルアドレス<sup>[5]</sup>が割り当てられると、"ifconfig"コマンドの出力結果は次のように表示されます。

```
[ATDE ~]$ LANG=C sudo ifconfig
(省略)
usb0      Link encap:Ethernet HWaddr a2:e4:92:6f:c6:54 ①
          inet6 addr: fe80::a0e4:92ff:fe6f:c654/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

usb0:avahi Link encap:Ethernet HWaddr a2:e4:92:6f:c6:54 ②
          inet addr:169.254.9.18 Bcast:169.254.255.255 Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

- ① イーサネットガジェットを認識すると"usb0"が表示されます
- ② リンクローカルアドレスが割り当てられると"usb0:avahi"が表示されます

Armadillo-EVA 1500 では、手動でネットワークインターフェース"usb0"を有効化する必要があります。まずリンクローカルアドレスを割り当てるための avahi-autoipd パッケージをインストールします。

```
[armadillo ~]# ifup eth0
[armadillo ~]# apt-get update
[armadillo ~]# apt-get install avahi-autoipd
```

avahi-autoipd を実行し、ネットワークインターフェース"usb0"を有効化します。有効化が完了し、リンクローカルアドレスが割り当てられると、"ifconfig"コマンドの出力結果は次のように表示されます。

<sup>[5]</sup>IPv4LL という機構によって割り当てられる特定のアドレス範囲(169.254.0.1~169.254.255.254)の IP アドレス。

```
[armadillo ~]# avahi-autoipd usb0 &
[armadillo ~]# ifconfig
(省略)
usb0      Link encap:Ethernet HWaddr b2:d7:79:24:c7:83
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:58 errors:0 dropped:27 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8443 (8.2 KiB) TX bytes:588 (588.0 B)

usb0:avahi Link encap:Ethernet HWaddr b2:d7:79:24:c7:83
          inet addr:169.254.8.225 Bcast:169.254.255.255 Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

Armadillo-EVA 1500 から ATDE5 に、イーサネットガジェットを使用した PING 通信を行う例を次に示します。

```
[armadillo ~]# ping 169.254.9.18
```



eth0 または awlan0 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 usb0 が使用されない場合があります。確実に usb0 を使用させる場合は、usb0 以外のネットワークインターフェースを無効化してください。

# 19. 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/>

## 19.1. 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<sup>[1]</sup>」をダウンロードすることができるようになります。

Armadillo-EVA 1500 購入製品登録

<https://users.atmark-techno.com/armadillo-eva-1500/register>

Armadillo-EVA 1500 の購入製品登録を行うには、Armadillo-EVA 1500 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-EVA 1500 から正規認証ファイル(board-info.txt)を取り出す手順は次のとおりです。

### 19.1.1. 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 USB メモリなどのストレージを使用し取り出して下さい。

1. ATDE で minicom を立ち上げて、Armadillo-EVA 1500 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。

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

<sup>[1]</sup>Armadillo-EVA 1500 の回路図データや、ミドルウェアパッケージなど



```
Armadillo-EVA 1500-0 login: root
Password:
[armadillo ~]#
```

2. "get-board-info-a1500"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。

```
[armadillo ~]# get-board-info-ae1500
[armadillo ~]# ls
board-info.txt
[armadillo ~]#
```

3. ストレージに正規認証ファイルをコピーし取り出します。下記の手順は USB メモリを使用した場合の手順です。ストレージの使用方法については、「6.6. ストレージ」を参照してください。

```
[armadillo ~]# mount -t vfat /dev/sda1 /mnt
[armadillo ~]# mv board-info.txt /mnt
[armadillo ~]# umount /mnt
```

取り出した正規認証ファイルを「Armadillo-EVA 1500 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。

## 改訂履歴

バージョン	年月日	改訂内容
1.0.0	2014/07/24	・ 初版発行
1.0.1	2014/09/26	・ 「表 15.19. CON15 信号配列」の I/O に対する誤記修正 ・ 誤記・表記ゆれの修正
1.1.0	2014/10/30	・ マルチメディアパッケージに対応 ・ 表記ゆれの修正
1.1.1	2015/2/17	・ 「2.8. 輸出について」の記載内容を一部変更 ・ 「8.1. 起動シーケンス」で利用している内蔵 RAM の名称を「Inter connect RAM2」に修正 ・ 「13.1.2. モード設定」の MD5 ピンの機能を「予約」に修正 ・ 「13.1.1. 特長」から RZ/G1M に存在しない「ルネサスグラフィックプロセッサ(2D)」の記載を削除
1.1.2	2015/7/27	・ プロセッサの名称を、型番表記からブランド名表記「RZ/G1M」に変更

Armadillo-EVA 1500 製品マニュアル  
Version 1.1.2  
2015/07/27

---

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

---