

Armadillo-500 開発ボード スタートアップガイド

A5501/A5527/A5567

Version 1.3.2
2011/10/24

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-500 開発ボードスタートアップガイド

株式会社アットマークテクノ

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

製作著作 © 2007-2011 Atmark Techno, Inc.

Version 1.3.2
2011/10/24

目次

1. はじめに	8
1.1. 本書および関連ファイルのバージョンについて	8
1.2. 対象となる読者	8
1.3. 本書の構成	8
1.4. 表記について	9
1.4.1. フォント	9
1.4.2. コマンド入力例	9
1.4.3. アイコン	9
1.5. 謝辞	10
2. 注意事項	11
2.1. 安全に関する注意事項	11
2.2. 取扱い上の注意事項	12
2.3. ソフトウェア使用に関する注意事項	12
2.4. 保証について	12
2.5. 輸出について	12
2.6. 商標について	12
3. 作業の前に	14
3.1. 見取り図	14
3.2. 準備するもの	14
3.3. 接続方法	15
3.4. ジャンパピンの設定について	15
3.5. シリアル通信ソフトウェアの設定	16
4. 起動と終了	17
4.1. 起動	17
4.2. ログイン	20
4.3. 終了方法	21
5. 起動モードとブートローダの機能	22
5.1. 起動モードの選択	22
5.2. Linux カーネル起動オプションの設定	22
5.2.1. コンソールの設定	23
5.2.2. ルートファイルシステムの設定	23
5.2.3. その他の起動オプション	24
5.2.4. 起動オプションの設定例	24
5.3. カーネルイメージのロケーション設定	25
6. コンフィグ領域 – 設定ファイルの保存領域	26
6.1. コンフィグ領域の読出し	26
6.2. コンフィグ領域の保存	26
6.3. コンフィグ領域の初期化	26
7. ネットワーク	28
7.1. ネットワークの設定	28
7.1.1. 固定 IP アドレスに設定する	28
7.1.2. DHCP に設定する	29
7.1.3. DNS サーバを指定する	29
7.1.4. 接続を確認する	29
7.2. ファイアウォール	29
7.3. ネットワークアプリケーション	30
7.3.1. Telnet	30
7.3.2. FTP	30
7.3.3. SSH	31
7.3.4. Web サーバ	31

7.3.5. NTP クライアント	31
8. シリアルインターフェース	32
8.1. シリアルインターフェースの設定	32
8.2. ファイルの送受信	32
8.2.1. ファイルの受信	32
8.2.2. ファイルの送信	32
9. ストレージ	33
9.1. ストレージとして使用可能なデバイス	33
9.2. ストレージの初期化とマウント	33
9.2.1. NAND フラッシュデバイス	33
9.2.2. その他のデバイス	34
10. ビデオ	36
10.1. カーネル起動ログをビデオに出力	36
10.2. 解像度の変更	36
10.2.1. VGA に設定する	36
10.2.2. SVGA に設定する	37
11. サウンド	38
11.1. コーデック仕様	38
11.2. OSS によるサウンド機能の実現	38
11.2.1. サウンドを再生する	38
11.2.2. 録音する	39
11.2.3. 音量を変更する	39
11.2.4. 録音ソースを変更する	40
12. その他のデバイス	41
12.1. LED	41
12.1.1. ledctrl による制御	41
12.1.2. ledctrl 使用例	41
12.2. タクトスイッチ	42
12.2.1. swmgr によるイベント取得	42
12.2.2. swmgr 使用例	43
12.3. GPIO	43
12.3.1. Direction を INPUT にする	43
12.3.2. Direction を OUTPUT にする	44

目次

3.1. 見取り図	14
3.2. 接続図	15
4.1. 起動ログ	17
4.2. 終了方法	21
5.1. Linux カーネル起動オプションのクリア	22
5.2. コンソールの指定	23
5.3. ルートファイルシステムの指定	24
5.4. 起動オプション設定例 1	25
5.5. 起動オプション設定例 2	25
5.6. 起動オプション設定例 3	25
5.7. カーネルイメージロケーション指定	25
6.1. コンフィグ領域の読出し方法	26
6.2. コンフィグ領域の保存方法	26
6.3. コンフィグ領域の初期化方法	27
7.1. 固定 IP アドレス設定	28
7.2. DHCP 設定	29
7.3. DNS サーバの設定	29
7.4. 設定を反映させる	29
7.5. PING 確認	29
7.6. iptables	30
7.7. telnet	30
7.8. ftp	30
7.9. ssh	31
7.10. msntp	31
8.1. 設定の参照	32
8.2. lrz による受信	32
8.3. lsz による送信	32
9.1. MTD パーティション一覧	33
9.2. NAND の初期化例	33
9.3. NAND のマウント方法	34
9.4. ディスク初期化方法	34
9.5. ファイルシステムの構築	35
9.6. マウント方法	35
10.1. コンソール指定 (ビデオ)	36
10.2. 解像度の指定方法 (VGA)	36
10.3. 解像度の指定方法 (SVGA)	37
11.1. サウンドの再生	39
11.2. サウンドの録音	39
11.3. 音量の変更	39
11.4. 録音ソースの指定	40
12.1. ledctrl コマンドフォーマット	41
12.2. ledctrl 使用例 1	42
12.3. ledctrl 使用例 2	42
12.4. ledctrl 使用例 3	42
12.5. ledctrl 使用例 4	42
12.6. swmgr コマンドフォーマット	42
12.7. swmgr 使用例 1	43
12.8. swmgr 使用例 2	43
12.9. GPIO : Direction を INPUT にする	43
12.10. GPIO : INPUT 時の入力状態を取得する	44

12.11. Direction を OUTPUT にする	44
12.12. OUTPUT 時の出力状態を変更する	44

表目次

1.1. 使用しているフォント	9
1.2. 表示プロンプトと実行環境の関係	9
1.3. コマンド入力例での省略表記	9
3.1. JP7 の設定	16
3.2. シリアル通信設定	16
4.1. シリアルコンソールログイン時のユーザ名とパスワード	20
5.1. 起動モード	22
5.2. コンソール指定に伴う出力先	23
5.3. ルートファイルシステムデバイス	24
5.4. カーネルイメージのロケーション	25
7.1. 固定 IP アドレス設定例	28
7.2. telnet でログイン可能なユーザ	30
7.3. ftp でログイン可能なユーザ	30
7.4. ssh でログイン可能なユーザ	31
8.1. 使用可能なシリアルインターフェースとデバイスファイルの対応	32
9.1. ストレージデバイス	33
10.1. 対応解像度一覧	36
11.1. 録音/再生ソース一覧	38
11.2. オーディオコーデックドライバの仕様	38
12.1. ledctrl : ledname	41
12.2. ledcrowl : command	41
12.3. swmgr : SW ID	42
12.4. GPIO ファイルノード	43

1. はじめに

このたびは Armadillo-500 開発セットをお求めいただき、ありがとうございます。

Armadillo-500 は、CPU Core に ARM1136JF-S を搭載した超小型・高性能 CPU モジュールです。情報表示機器やマルチメディア機器などのメインプロセッサとしてご利用頂くことが可能です。

Armadillo-500 開発ボード（以降、開発ボードと表記）は、Armadillo-500 と Armadillo-500 に搭載された機能を効率的に使用することができるように各種コントローラ及び、コネクタを実装した開発用ベースボード（以降、ベースボードと表記）の構成となります。

開発ボードは、標準 OS に Linux を採用していますので、Linux の豊富なソフトウェア資産を利用することができます。また、C などのプログラミング言語を使用し、オリジナルのプログラムを作成して動作させることも可能です。カスタマイズ方法については、「Armadillo-500 開発ボード ソフトウェアマニュアル」等を参照してください。

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

以降、本書では他の Armadillo シリーズにも共通する記述については、製品名を Armadillo と表記します。

1.1. 本書および関連ファイルのバージョンについて

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

1.2. 対象となる読者

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

上記以外の方でも、本書を有効に利用していただけましたら幸いです。

1.3. 本書の構成

本書では、Armadillo の基本的な使用方法について記載されています。

以下に主な項目を挙げます。

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

1.4. 表記について

1.4.1. フォント

本書では以下のような意味でフォントを使いわけています。

表 1.1 使用しているフォント

フォント例	説明
本文中のフォント	本文
[PC ~]\$ ls	プロンプトとユーザ入力文字列
text	編集する文字列や出力される文字列。またはコメント

1.4.2. 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りたっています。この場を借りて感謝の意を表します。

2. 注意事項

2.1. 安全に関する注意事項

本製品を安全にご使用いただくために、特に以下の点にご注意ください。



- ・ ご使用の前に必ず製品マニュアルおよび関連資料をお読みにになり、使用上の注意を守って正しく安全にお使いください。
- ・ マニュアルに記載されていない操作・拡張などを行う場合は、弊社 Web サイトに掲載されている資料やその他技術情報を十分に理解した上で、お客様自身の責任で安全にお使いください。
- ・ 水・湿気・ほこり・油煙等の多い場所に設置しないでください。火災、故障、感電などの原因になる場合があります。
- ・ 本製品に搭載されている部品の一部は、発熱により高温になる場合があります。周囲温度や取扱いによってはやけどの原因となる恐れがあります。本体の電源が入っている間、または電源切断後本体の温度が下がるまでの間は、基板上の電子部品、及びその周辺部分には触れないでください。
- ・ 本製品を使用して、お客様の仕様による機器・システムを開発される場合は、製品マニュアルおよび関連資料、弊社 Web サイトで提供している技術情報のほか、関連するデバイスのデータシート等を熟読し、十分に理解した上で設計・開発を行ってください。また、信頼性および安全性を確保・維持するため、事前に十分な試験を実施してください。
- ・ 本製品は、機能・精度において極めて高い信頼性・安全性が必要とされる用途(医療機器、交通関連機器、燃焼制御、安全装置等)での使用を意図しておりません。これらの設備や機器またはシステム等に使用された場合において、人身事故、火災、損害等が発生した場合、当社はいかなる責任も負いかねます。
- ・ 本製品には、一般電子機器用(OA 機器・通信機器・計測機器・工作機械等)に製造された半導体部品を使用しています。外来ノイズやサージ等により誤作動や故障が発生する可能性があります。万一誤作動または故障などが発生した場合に備え、生命・身体・財産等が侵害されることのないよう、装置としての安全設計(リミットスイッチやヒューズ・ブレーカー等の保護回路の設置、装置の多重化等)に万全を期し、信頼性および安全性維持のための十分な措置を講じた上でお使いください。
- ・ 無線 LAN 機能を搭載した製品は、心臓ペースメーカーや補聴器などの医療機器、火災報知器や自動ドアなどの自動制御器、電子レンジ、高度な電子機器やテレビ・ラジオに近接する場所、移動体識別用の構

内無線局および特定小電力無線局の近くで使用しないでください。製品が発生する電波によりこれらの機器の誤作動を招く恐れがあります。

2.2. 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。

- | | |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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, USB, SD/MMC, コンパクトフラッシュ, オーディオ)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いますので、ご使用になる時までは、帯電防止対策された出荷時のパッケージ等にて保管してください。 |
| 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。 |

2.3. 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 | 本製品に含まれるソフトウェア(付属のドキュメント等も含みます)は、現状有姿(AS IS)にて提供いたします。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。当社は、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、お客様に対しなんら保証も行うものではありません。 |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

2.4. 保証について

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

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

2.5. 輸出について

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

2.6. 商標について

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

- ・ SD、SDHC、microSD、microSDHC、SDIO ロゴは SD-3C、LLC の商標です。



3. 作業の前に

3.1. 見取り図

開発ボードの見取り図です。各インターフェースの配置場所等を確認してください。

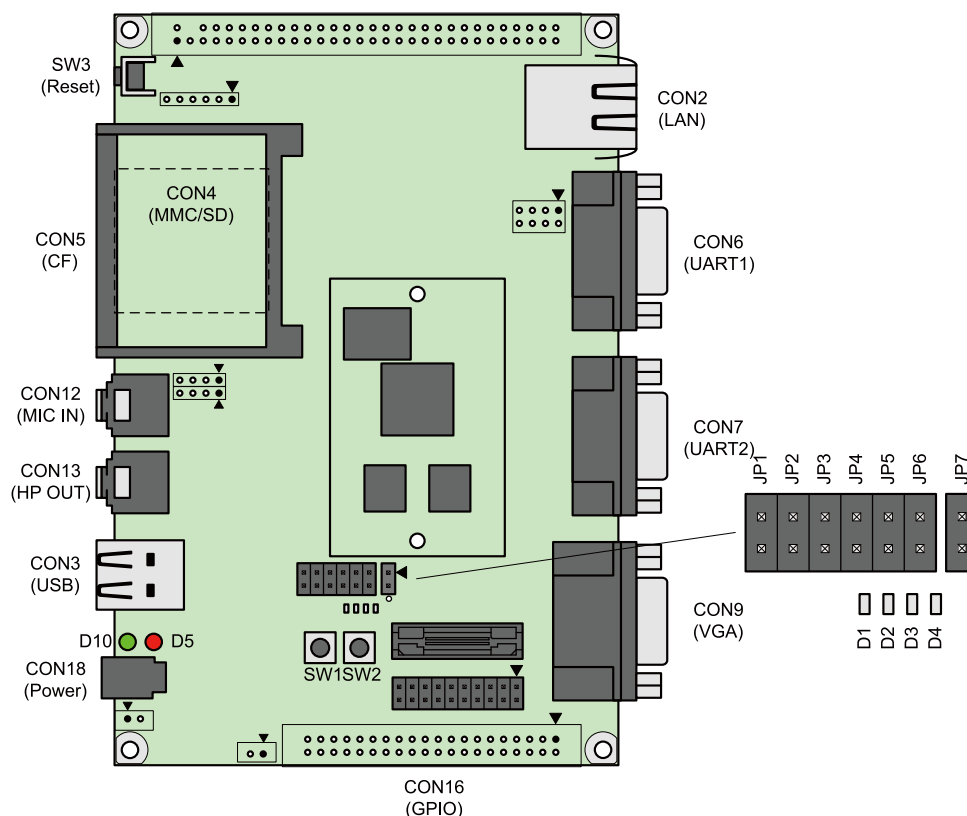


図 3.1 見取り図

3.2. 準備するもの

開発ボードを使用する前に、次のものを準備してください。

作業用 PC とシリアル
クロスケーブル

Linux もしくは Windows が動作し、1 ポート以上のシリアルインターフェースを持つ PC と D-Sub9 ピン（メス - メス）のクロス接続用ケーブルです。作業用 PC にはシリアル通信ソフトウェア^[1]をインストールしてください。（Linux 用のソフトウェアは付属 CD の tool ディレクトリに収録されています。）

ネットワーク環境

ターゲットボードと作業用 PC をネットワーク通信ができるようにしてください。

^[1]Linux では「minicom」、Windows では「Tera Term Pro」などです。

ディスプレイと USB キーボード VGA 入力端子を持つ汎用的なディスプレイと USB ポートに接続するタイプのキーボードです。

3.3. 接続方法

開発ボードの接続例です。

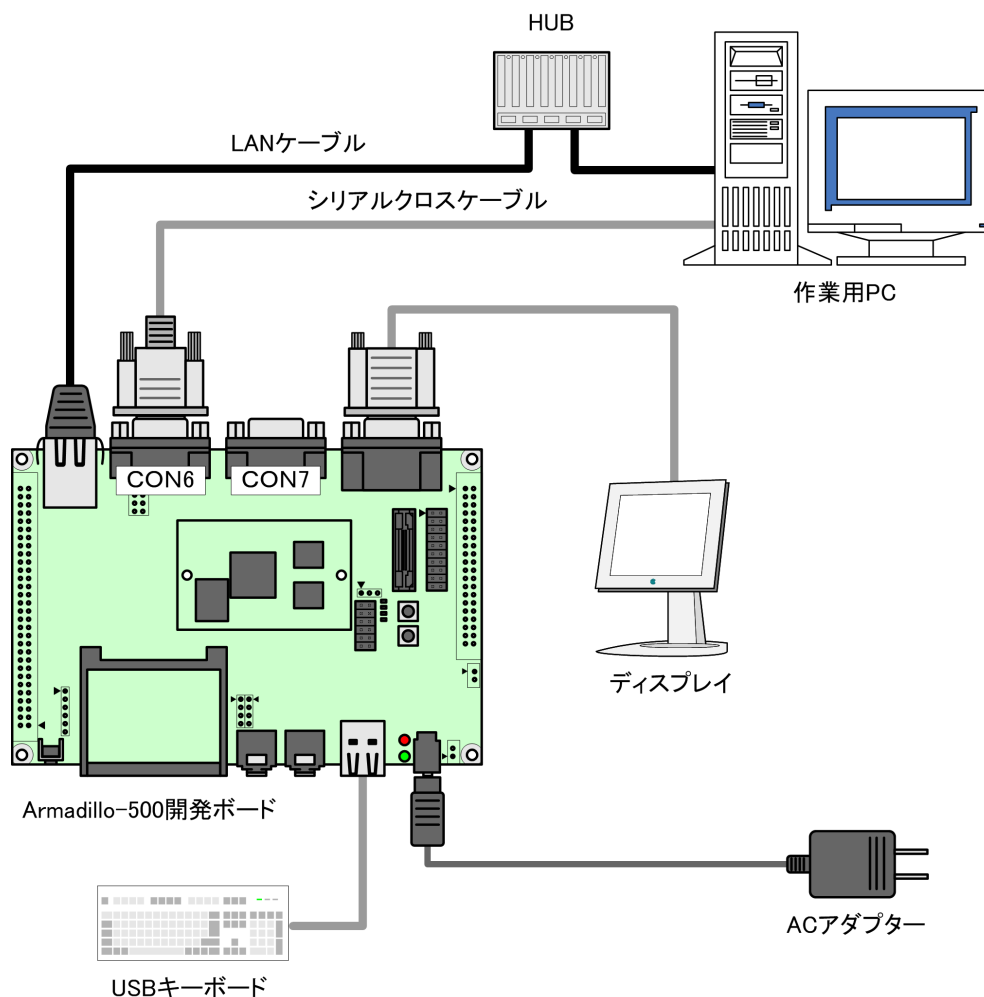




図 3.2 接続図

3.4. ジャンパピンの設定について

本書では JP1 以外は操作することはありません。JP2-JP6 は常にオープンに設定してください。JP7 は表 2-1 のように設定してください。JP1 については、必要に応じて切り替えの指示があります。ここでは、オープンに設定しておきます。



ジャンパのオープン、ショートとは



「オープン」とはジャンパピンにジャンパソケットを接続していない状態です。



「ショート」とはジャンパピンにジャンパソケットを接続している状態です。

また、ジャンパピンの位置は「図 3.1. 見取り図」で確認することができます。

モジュールの型番によって、JP7 を下記のように設定してください。

あやまった設定をすると、正常に動作しないことがあります。

表 3.1 JP7 の設定

CPU モジュール型番	JP7 の状態
A5001-U00-B	ショート
A5001-U00-B 以外	オープン

3.5. シリアル通信ソフトウェアの設定

シリアル通信ソフトウェアを起動し、シリアルの通信設定を、「表 3.2. シリアル通信設定」のように設定してください。



Armadillo-240 では、RS232C レベル変換アダプターを経由させる必要があります。

表 3.2 シリアル通信設定

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

4. 起動と終了

4.1. 起動

開発ボードの電源を投入してください。次のように起動ログがシリアル通信ソフトウェアに表示されます。

```
Hermit-At v1.1.9 (Armadillo-500) compiled at 12:00:00, Jul 27 2007
Uncompressing kernel.....done.
Uncompressing ramdisk.....done.
Linux version 2.6.18-12-at0 (atmark@atde) (gcc version 4.1.2 20061115
(prerelease) (Debian 4.1.1-21)) #1 PREEMPT Thu Jul 26 00:01:12 JST 2007
CPU: Some Random V6 Processor [4107b364] revision 4 (ARMv6TEJ), cr=00c5387f
Machine: Armadillo-500
ATAG_INITRD is deprecated; please update your bootloader.
Memory policy: ECC disabled, Data cache writeback
CPU0: D VIPT write-back cache
CPU0: I cache: 16384 bytes, associativity 4, 32 byte lines, 128 sets
CPU0: D cache: 16384 bytes, associativity 4, 32 byte lines, 128 sets
Built 1 zonelists. Total pages: 16384
Kernel command line: console=ttymxc0
MXC IRQ initialized
PID hash table entries: 512 (order: 9, 2048 bytes)
Actual CLOCK_TICK_RATE is 60000000 Hz
Console: colour dummy device 80x30
Dentry cache hash table entries: 8192 (order: 3, 32768 bytes)
Inode-cache hash table entries: 4096 (order: 2, 16384 bytes)
Memory: 64MB = 64MB total
Memory: 52748KB available (3078K code, 718K data, 120K init)
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
checking if image is initramfs...it isn't (bad gzip magic numbers); looks like
an initrd
Freeing initrd memory: 8192K
NET: Registered protocol family 16
MXC GPIO hardware
system_rev is: 0x11
Clock input source is 26000000
L2 cache: WB
Using SDMA I.API
MXC DMA API initialized
usb: Host 2 registered
usb: OTG HS Host registered
SCSI subsystem initialized
usbcore: registered new driver usbfs
usbcore: registered new driver hub
MXC MMC/SD driver
mxcmmc-0 found
NET: Registered protocol family 2
```

```
IP route cache hash table entries: 512 (order: -1, 2048 bytes)
TCP established hash table entries: 2048 (order: 1, 8192 bytes)
TCP bind hash table entries: 1024 (order: 0, 4096 bytes)
TCP: Hash tables configured (established 2048 bind 1024)
TCP reno registered
Low-Level PM Driver module loaded
NetWinder Floating Point Emulator V0.97 (double precision)
JFFS2 version 2.2. (NAND) (C) 2001-2006 Red Hat, Inc.
Initializing Cryptographic API
io scheduler noop registered
io scheduler anticipatory registered
io scheduler deadline registered
io scheduler cfq registered (default)
Console: switching to colour frame buffer device 80x30
fb0: MXC frame buffer at 640x480x16 (CRT-VGA)
gpio: Armadillo-5x0 GPIO Sample driver [Rev.1.00]
MXC WatchDog Driver 2.0
MXC Watchdog Timer: initial timeout 60 sec
Serial: MXC Internal UART driver
mxcintuart.0: ttyMXC0 at MMIO 0x43f90000 (irq = 45) is a Freescale MXC
mxcintuart.1: ttyMXC1 at MMIO 0x43f94000 (irq = 32) is a Freescale MXC
RAMDISK driver initialized: 16 RAM disks of 16384K size 1024 blocksize
loop: loaded (max 8 devices)
eth0: LAN9118 (rev 1) at 0xc4850000 IRQ 64
eth0: Ethernet addr: 00:11:0c:0a:00:01
Uniform Multi-Platform E-IDE driver Revision: 7.00alpha2
ide: Assuming 50MHz system bus speed for PIO modes; override with idebus=xx
NFTL driver: nftlcore.c $Revision: 1.98 $, nftlmount.c $Revision: 1.41 $
armadillo5x0-nor: Found 1 x16 devices at 0x0 in 16-bit bank
  Intel/Sharp Extended Query Table at 0x010A
  Intel/Sharp Extended Query Table at 0x010A
  Intel/Sharp Extended Query Table at 0x010A
  Intel/Sharp Extended Query Table at 0x010A
  Intel/Sharp Extended Query Table at 0x010A
Using buffer write method
cfi_cmdset_0001: Erase suspend on write enabled
armadillo5x0-nor: use default partitions(4)
Creating 4 MTD partitions on "armadillo5x0-nor":
0x00000000-0x00020000 : "nor.bootloader"
0x00020000-0x00220000 : "nor.kernel"
0x00220000-0x00fe0000 : "nor.userland"
0x00fe0000-0x01000000 : "nor.config"
MXC MTD nand Driver 2.0
NAND device: Manufacturer ID: 0x20, Chip ID: 0xf1 (ST Micro NAND 128MiB 3,3V
8-bit)
Scanning device for bad blocks
Creating 4 MTD partitions on "NAND 128MiB 3,3V 8-bit":
0x00000000-0x00020000 : "nand.ipl"
0x00020000-0x00420000 : "nand.kernel"
0x00420000-0x01a20000 : "nand.userland"
0x01a20000-0x10000000 : "nand.free"
armadillo5x0_pcmcia: PCMCIA driver [Rev.1 (2007/--/--)], (C) 2007 Atmark Techno,
Inc.
usbmon: debugfs is not available
fsl-ehci fsl-ehci.0: Freescale On-Chip EHCI Host Controller
fsl-ehci fsl-ehci.0: new USB bus registered, assigned bus number 1
fsl-ehci fsl-ehci.0: irq 36, io base 0x43f88400
fsl-ehci fsl-ehci.0: USB 2.0 started, EHCI 1.00, driver 10 Dec 2004
```

```
usb usb1: configuration #1 chosen from 1 choice
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 1 port detected
fsl-ehci fsl-ehci.1: Freescale On-Chip EHCI Host Controller
fsl-ehci fsl-ehci.1: new USB bus registered, assigned bus number 2
fsl-ehci fsl-ehci.1: irq 37, io base 0x43f88000
fsl-ehci fsl-ehci.1: USB 2.0 started, EHCI 1.00, driver 10 Dec 2004
usb usb2: configuration #1 chosen from 1 choice
hub 2-0:1.0: USB hub found
hub 2-0:1.0: 1 port detected
Initializing USB Mass Storage driver...
usbcore: registered new driver usb-storage
USB Mass Storage support registered.
usbcore: registered new driver usbhid
drivers/usb/input/hid-core.c: v2.6:USB HID core driver
pegasus: v0.6.13 (2005/11/13), Pegasus/Pegasus II USB Ethernet driver
usbcore: registered new driver pegasus
mice: PS/2 mouse device common for all mice
tactsw: Armadillo-5x0 Tact-SW driver [Rev.1.00]
input: Tact-SW Port1 as /class/input/input0
input: Tact-SW Port2 as /class/input/input1
rtc-s353xxa: S-353XXA Real Time Clock class driver [Rev.1 (2007/--/--)], (C)
2007 Atmark Techno, Inc.
i2c /dev entries driver
i2c_armadillo5x0: Armadillo-5x0 I2C driver [Rev.1 (2007/--/--)], (C) 2007 Atmark
Techno, Inc.
rtc-s353xxa 0-0030: rtc intf: sysfs
rtc-s353xxa 0-0030: rtc intf: proc
rtc-s353xxa 0-0030: rtc intf: dev (253:0)
rtc-s353xxa 0-0030: rtc core: registered S-353XXA as rtc0
tlv320aic: TI TLV320AIC Audio codec driver [Rev.1 (2007/--/--)], (C) 2007 Atmark
Techno, Inc.
IPU Post-filter loading
SSI module loaded successfully
led: Armadillo-5x0 LED driver, (C) 2007 Atmark Techno, Inc.
Registered led device: led1
Registered led device: led2
Registered led device: led3
Registered led device: led4
Registered led device: led5
Advanced Linux Sound Architecture Driver Version 1.0.12rc1 (Thu Jun 22 13:55:50
2006 UTC).
mxc_alsa_i2s: MXC ALSA iis audio driver [Rev.1 (2007/--/--)], (C) 2007 Atmark
Techno, Inc.
usbcore: registered new driver snd-usb-audio
ALSA device list:
#0: mxc i2s audio
IPv4 over IPv4 tunneling driver
ip_tables: (C) 2000-2006 Netfilter Core Team
TCP bic registered
NET: Registered protocol family 1
NET: Registered protocol family 17
NET: Registered protocol family 15
ieee80211: 802.11 data/management/control stack, git-1.1.13
ieee80211: Copyright (C) 2004-2005 Intel Corporation <jketreno@linux.intel.com>
Power Management for Freescale MX31
VFP support v0.3: implementor 41 architecture 1 part 20 variant b rev 2
rtc-s353xxa 0-0030: setting the system clock to 2007-07-26 14:01:52 (1185458512)
```

```

RAMDISK: ext2 filesystem found at block 0
RAMDISK: Loading 8192KiB [1 disk] into ram disk... done.
VFS: Mounted root (ext2 filesystem).
Freeing init memory: 120K
init started: BusyBox v1.00 (2007.07.25-15:02+0000) multi-call binary
Starting fsck for root filesystem.
fsck 1.25 (20-Sep-2001)
ext2fs_check_if_mount: No such file or directory while determining whether
/dev/ram0 is mounted.
/dev/ram0: clean, 719/1024 files, 6572/8192 blocks
Checking root filesystem:           done
Remounting root rw:                 done
Mounting proc:                       done
Mounting usbfs:                      done
Mounting sysfs:                      done
Cleaning up system:                  done
Running local start scripts.
Starting udevd:                       done
Changing file permissions:           done
Configure /home/ftp:                 done
Starting syslogd:                    done
Starting klogd:                       done
Loading /etc/config:                 done
Starting basic firewall:              done
Setting hostname:                     done
Configuring network interfaces: info, udhcpc (v0.9.9-pre) started
eth0: Link down
eth0: Link up, 100Mbps, full-duplex, lpa 0x05E1
debug, Sending select for 172.16.1.93...
info, Lease of 172.16.1.100 obtained, lease time 86400
done
Starting inetd:                       done
Starting sshd:                        done
Starting thttpd:                      done
Mounting ramfs /home/ftp/pub:         done

atmark-dist v1.10.0 (AtmarkTechno/Armadillo-500)
Linux 2.6.18-12-at0 [armv6l arch]

armadillo500 login:

```

図 4.1 起動ログ

4.2. ログイン

起動が完了するとログインプロンプトが表示されます。「表 4.1. シリアルコンソールログイン時のユーザ名とパスワード」に示すユーザでログインすることができます。

表 4.1 シリアルコンソールログイン時のユーザ名とパスワード

ユーザ名	パスワード	権限
root	root	root ユーザ
guest	(なし)	一般ユーザ

4.3. 終了方法

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

```
[armadillo ~]# halt
[armadillo ~]#
System is going down for system reboot now.

Starting local stop scripts.
Exiting Syslogd!
Syncing all filesystems:                done
Unmounting all filesystems:            done
The system is going down NOW !!
Sending SIGTERM to all processes.
The system is halted. Press Reset or turn off power
MXC Watchdog shutdown
System halted.
```

図 4.2 終了方法

リムーバブルディスクを使用していない場合は、電源を切断し終了させることもできます。



リムーバブルディスクにデータを書き込んでいる途中で電源を切断した場合、ファイルシステム、及び、データが破損する恐れがあります。リムーバブルディスクをアンマウントしてから電源を切断するようご注意ください。

5. 起動モードとブートローダの機能

この章では、開発ボードの起動モードと採用しているブートローダ「Hermit-At」の起動設定機能について説明します。

起動モードには、オートブートモード、保守モード等、システム起動時に最初に動作するソフトウェアを選択することができます。

Hermit-At では、Linux カーネルを起動させる時の起動オプションの設定、カーネルイメージのロケーションの設定、クロックの設定等、システム起動時の初期設定を行うことができます。

5.1. 起動モードの選択

起動モードの設定は、JP1 の設定により決定されます。各起動モードは「表 5.1. 起動モード」のようになります。

表 5.1 起動モード

モード	JP1	説明
オートブート	オープン	電源投入後、自動的にカーネルを起動させます。
保守	ショート	各種設定が可能な Hermit コマンドプロンプトが起動します。

5.2. Linux カーネル起動オプションの設定


Linux カーネル起動オプションを変更することで、コンソールや、ルートファイルシステム等の様々な種類の設定を変更することができます。ここでは、開発ボードに関係のある代表的なオプションについて説明します。

また、これらの設定は、Hermit-At の setenv 機能を使用します。setenv で設定されたパラメータはフラッシュメモリに保存され再起動後も設定が反映されます。

設定されたパラメータをクリアするには、clearenv を使用します。

```
hermit> clearenv
```

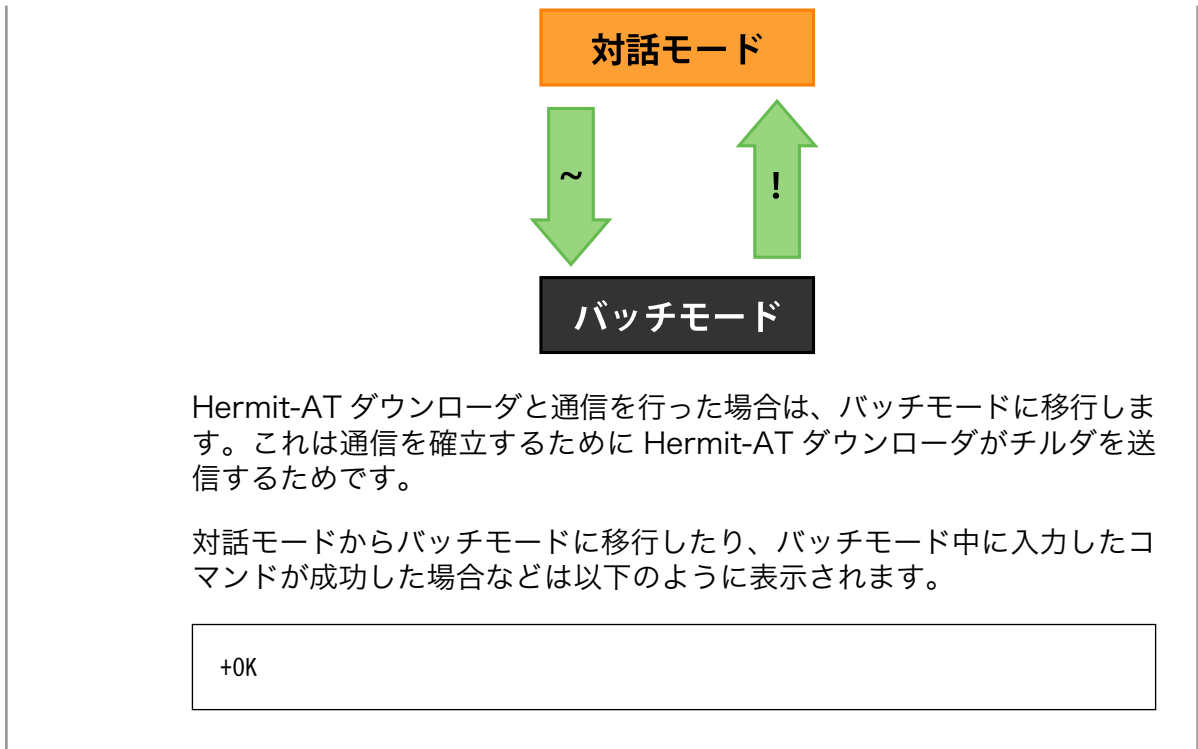
図 5.1 Linux カーネル起動オプションのクリア



Hermit-AT のモード

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

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



5.2.1. コンソールの設定

通常の Linux システムでは、起動ログの出力コンソールを変更する時に指定しますが、開発ボードでは、ブートローダのログ出力先も同時に変更します。

コンソールを指定するには、下記のように console パラメータにコンソール指定子を設定します。

```
hermit> setenv console=ttymxc0
```

図 5.2 コンソールの指定

設定によるログの出力先は、「表 5.2. コンソール指定に伴う出力先」のようになります。

表 5.2 コンソール指定に伴う出力先

コンソール指定子	起動ログ出力先	保守モードプロンプト出力先
ttymxc0	CON6	CON6
ttymxc1	CON7	CON7
null	なし	CON6
その他 (tty1 等)	指定するコンソール ^[1]	CON6

^[1]ブートローダのログは出力されません。

5.2.2. ルートファイルシステムの設定

ルートファイルシステムとしてマウントするファイルシステムイメージの場所や、マウントするファイルシステム等を設定します。各パラメータの意味は以下を参照してください。

root

使用するルートファイルシステムのパーティションを指定します。

rootfs

使用するルートファイルシステムの種類をタイプします。

rootdelay

ルートファイルシステムをマウントする前のディレイ時間を指定します。単位は秒です。

```
hermit> setenv root=/dev/hda1 rootfs=ext2 rootdelay=3
```

図 5.3 ルートファイルシステムの指定

5.2.2.1. ルートファイルシステムイメージの場所

ファイルシステムイメージの場所を設定する場合は、イメージが存在するパーティションを設定します。各デバイスのパーティションノードの例を、「表 5.3. ルートファイルシステムデバイス」に示します。指定がない場合（デフォルト）は、RAM ディスク（/dev/ram0）が指定されます。

表 5.3 ルートファイルシステムデバイス

デバイス名	デバイスノード	先頭パーティションノード
RAM ディスク	/dev/ram	/dev/ram0
コンパクトフラッシュディスク	/dev/hda	/dev/hda1
NAND ディスク	/dev/mentryblock	/dev/mentryblock4
MMC/SD カードディスク	/dev/mmcbk0	/dev/mmcbk0p1

5.2.2.2. ルートファイルシステムタイプ

特異なファイルシステムを使用する場合は、ファイルシステムタイプを指定します。主に JFFS2 が使われている場合となります。指定がない場合は、ext2、ext3、msdos、vfat のいずれかでマウントされます^[1]。

5.2.2.3. マウント前ディレイ

Linux カーネルは、指定するルートファイルシステムが存在するデバイスの認識が完了していなければ、ルートファイルシステムをマウントすることはできません。ドライバのロードタイミングやデバイスに依存する時間等、デバイスの認識時間は様々な要素で変動します。

ここで指定することができるマウント前ディレイは、ルートファイルシステムをマウントする前に、指定秒間待機するオプションとなります。指定がない場合（デフォルト）は、待機なしとなります。

5.2.3. その他の起動オプション

本書で紹介したオプション以外にも様々な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るドキュメント（kernel-source/Documentation/kernel-parameters.txt）等を参照してください。

また、ビデオに関するオプションは「9. ストレージ」を参照してください。

5.2.4. 起動オプションの設定例

- ・ コンソールを CON7 にする場合

^[1]Linux カーネルが標準でサポートするファイルシステムの場合は、特に指定は必要ありません


```
hermit> setenv console=ttymxc1
```

図 5.4 起動オプション設定例 1

- ・ コンソールを CON6、ルートファイルシステムをコンパクトフラッシュのパーティション 1 にする場合

```
hermit> setenv console=ttymxc0 root=/dev/hda1 rootdelay=3
```

図 5.5 起動オプション設定例 2

- ・ コンソールをビデオに出力、ルートファイルシステムを MMC/SD のパーティション 2 にする場合 [2]

```
hermit> setenv console=ttymxc0 video=mxcfb:CRT-SVGA, 16bpp, enable
root=/dev/mmcblk0p2 rootdelay=1
```

図 5.6 起動オプション設定例 3

5.3. カーネルイメージのロケーション設定

setbootdevice コマンドを使用する事で、ブートローダーがロードするカーネルイメージのロケーションを指定することができます。

開発ボードでは、「表 5.4. カーネルイメージのロケーション」に示すデバイスを指定できます。

表 5.4 カーネルイメージのロケーション

指定子	説明
flash	フラッシュメモリの kernel リージョンにあるイメージをロードします。
hda	コンパクトフラッシュの各パーティションの/boot 以下にあるイメージファイル ^[1] を順番に走査し、最初に見つけたイメージをロードします ^[2] 。
hda[1:4]	コンパクトフラッシュの指定するパーティションの/boot 以下にあるイメージファイル ^[2] を走査し、見つけたイメージをロードします ^[2] 。

[1] linux.bin、linux.bin.gz、Image、Image.gz のいずれかとなります。

[2] イメージファイルが見つからない場合は、flash を指定したように動作します。

以下の例では、カーネルイメージのロケーションをフラッシュメモリに設定しています。

```
hermit> setbootdevice flash
```

図 5.7 カーネルイメージロケーション指定

[2] 書面の都合上、折り返して表記していますが通常は 1 行のコマンドとなります。

6. コンフィグ領域 – 設定ファイルの保存領域

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

6.1. コンフィグ領域の読出し

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

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

図 6.1 コンフィグ領域の読出し方法



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

6.2. コンフィグ領域の保存

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

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

図 6.2 コンフィグ領域の保存方法



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

6.3. コンフィグ領域の初期化

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

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

図 6.3 コンフィグ領域の初期化方法

7. ネットワーク

この章では、ネットワークの設定方法やネットワークを使用するアプリケーションの使用方法について説明します。

7.1. ネットワークの設定

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

また、出荷時のネットワーク設定は、DHCP となっています。



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

7.1.1. 固定 IP アドレスに設定する

「表 7.1. 固定 IP アドレス設定例」に示す内容に設定変更するには、vi エディタで/etc/config/interfaces を、「図 7.1. 固定 IP アドレス設定」のように編集します。

表 7.1 固定 IP アドレス設定例

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

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

auto lo eth0
iface lo inet loopback
iface eth0 inet static
  address 192.168.10.10
  netmask 255.255.255.0
  network 192.168.10.0
  broadcast 192.168.10.255
  gateway 192.168.10.1
```

図 7.1 固定 IP アドレス設定

7.1.2. DHCP に設定する

DHCP に設定するには、vi エディタで/etc/config/interfaces を、「[図 7.2. DHCP 設定](#)」のように編集します。

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

auto lo eth0
iface lo inet loopback
iface eth0 inet dhcp
```

図 7.2 DHCP 設定

7.1.3. DNS サーバを指定する

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

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

図 7.3 DNS サーバの設定

7.1.4. 接続を確認する

ここでは、変更した IP 設定で正常に通信が可能か確認します。

まず、設定を反映させます。設定後、コンフィグ領域を保存し再起動した場合は必要ありません。

```
[armadillo ~]# ifdown -a
[armadillo ~]# ifup -a
```

図 7.4 設定を反映させる

同じネットワーク内にある通信機器と PING 通信を行ってみます。

```
[armadillo ~]# ping 192.168.10.1
```

図 7.5 PING 確認

7.2. ファイアーウォール

Armadillo では、簡易ファイアーウォールが動作しています。設定されている内容を参照するには、「[図 7.6. iptables](#)」のようにコマンド実行してください。

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

図 7.6 iptables

7.3. ネットワークアプリケーション

ここでは、出荷時に収録されているソフトウェアのうちネットワークに関するアプリケーションの操作方法を説明します。

7.3.1. Telnet

7.3.1.1. Telnet サーバ

他の PC からネットワーク経由でログインし、リモート操作が可能となります。「表 7.2. telnet でログイン可能なユーザ」に示すユーザでログインすることができます。

表 7.2 telnet でログイン可能なユーザ

ユーザ名	パスワード
guest	(なし)

7.3.1.2. Telnet クライアント

telnet を使用して、他の PC にリモートログインすることができます。「図 7.7. telnet」のようにコマンドを実行します。

```
[armadillo ~]# telnet 192.168.10.1
```

図 7.7 telnet

7.3.2. FTP

7.3.2.1. FTP サーバ

他の PC からネットワーク経由でファイルの転送ができます。「表 7.3. ftp でログイン可能なユーザ」に示すユーザでログインすることが可能です。

表 7.3 ftp でログイン可能なユーザ

ユーザ名	パスワード
ftp	(なし)

7.3.2.2. FTP クライアント

ftp を使用して、他の PC とファイル転送ができます。「図 7.8. ftp」のようにコマンドを実行してください。

```
[armadillo ~]# ftp 192.168.10.1
```

図 7.8 ftp

7.3.3. SSH

7.3.3.1. SSH サーバ

他の PC からネットワーク経由でログインし、安全にリモート操作が可能となります。「表 7.4. ssh でログイン可能なユーザ」に示すユーザでログインすることができます。

表 7.4 ssh でログイン可能なユーザ

ユーザ名	パスワード
guest	(なし)

7.3.3.2. SSH クライアント

ssh を使用して、安全に他の PC へリモートログインすることができます。「図 7.9. ssh」のようにコマンドを実行します。

```
[armadillo ~]# ssh user@192.168.10.1
```

図 7.9 ssh

7.3.4. Web サーバ

Armadillo では、Web サーバが動作しています。PC などの Web ブラウザから Armadillo の URL ([http://\[ArmadilloのIPアドレス\]](http://[ArmadilloのIPアドレス]))^[1]にアクセスすると、「/home/www-data」以下のディレクトリをブラウズすることが出来ます。

7.3.5. NTP クライアント

Armadillo では、SNTP (Simple Network Time Protocol) クライアントが使用できます。SNTP に対応しているタイムサーバから時刻情報を取得することができます。

「図 7.10. msntp」のようにコマンドを実行します。

```
[armadillo ~]# msntp -r 192.168.10.1
The time correction is 17180483.054 +/- 0.075+0.001 seconds
Do you want to correct the time anyway? y
```

図 7.10 msntp

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

8. シリアルインターフェース

開発ボードで使用可能なシリアルインターフェースとデバイスファイルの対応を「表 8.1. 使用可能なシリアルインターフェースとデバイスファイルの対応」に示します。

表 8.1 使用可能なシリアルインターフェースとデバイスファイルの対応

シリアルインターフェース	デバイスファイル
CON6(UART1)	/dev/ttyxc0
CON7(UART2)	/dev/ttyxc1

8.1. シリアルインターフェースの設定

シリアルインターフェースの設定を参照するには、「図 8.1. 設定の参照」のようにコマンドを実行してください。以下の例では、CON6 の設定を参照しています。

```
[armadillo ~]# stty -F /dev/ttyxc0
```

図 8.1 設定の参照

8.2. ファイルの送受信

シリアルインターフェースを使用して、ファイルを送受信することができます。転送プロトコルには、XMODEM、YMODEM および ZMODEM を使用することができます。

8.2.1. ファイルの受信

「図 8.2. lrz による受信」のようにコマンドを実行すると、ZMODEM 受信待ち状態になります。シリアル通信ソフトウェアから ZMODEM 送信をおこなってください。

```
[armadillo ~]# lrz
```

図 8.2 lrz による受信

8.2.2. ファイルの送信

「図 8.3. lsz による送信」のようにコマンドを実行すると、ZMODEM 送信をおこないます。シリアル通信ソフトウェアから ZMODEM 受信をおこなってください。

```
[armadillo ~]# lsz filename
```

filename 送信するファイル

図 8.3 lsz による送信

9. ストレージ

9.1. ストレージとして使用可能なデバイス

開発ボードでは、「表 9.1. ストレージデバイス」に示すデバイスをストレージとして使用することができます。

表 9.1 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
オンボード NAND フラッシュ	/dev/flash/nand.*	/dev/flash/nand.ip1
コンパクトフラッシュ	/dev/hda	/dev/hda1
MMC/SD カード	/dev/mmcblk	/dev/mmcblk0p1
USB メモリ	/dev/sd*	/dev/sda1

9.2. ストレージの初期化とマウント

ストレージの初期化とマウント方法について説明します。

9.2.1. NAND フラッシュデバイス

ここでは、パーティション/dev/flash/nand.free (/dev/mtd7、/dev/mtdblock7) を例にとります。

MTD のパーティション情報は、「図 9.1. MTD パーティション一覧」のようにコマンドを実行すると表示されます。

```
[armadillo ~]# cat /proc/mtd
dev:   size  erasesize  name
mtd0: 00020000 00008000 "nor.bootloader"
mtd1: 00200000 00020000 "nor.kernel"
mtd2: 00dc0000 00020000 "nor.userland"
mtd3: 00020000 00020000 "nor.config"
mtd4: 00020000 00020000 "nand.ip1"
mtd5: 00400000 00020000 "nand.kernel"
mtd6: 01600000 00020000 "nand.userland"
mtd7: 0e5e0000 00020000 "nand.free"
```

図 9.1 MTD パーティション一覧

9.2.1.1. ファイルシステムを JFFS2 にする場合

ファイルシステムを JFFS2 (Journalling Flash File System v2) にする場合は、デバイスの初期化と同時にファイルシステムを構築することができます。「図 9.2. NAND の初期化例」のようにコマンドを実行してください。

```
[armadillo ~]# flash_eraseall --jffs2 /dev/flash/nand.free
Erasing 128 Kibyte @ e5e0000 -- 100 % complete.Cleanmarker written at e5c0000.
```

図 9.2 NAND の初期化例

9.2.1.2. マウント

初期化後、マウントするには、「図 9.3. NAND のマウント方法」のようにコマンドを実行します。

```
[armadillo ~]# mount -t jffs2 /dev/mtdblock7 /mnt
```

図 9.3 NAND のマウント方法

9.2.2. その他のデバイス

9.2.2.1. ディスクの初期化

ここでは、コンパクトフラッシュ（/dev/hda）を例にとり、パーティションを 1 つ作ります。

```
[armadillo ~]# fdisk /dev/hda
Command (m for help): d
Selected partition 1

Command (m for help): d
No partition is defined yet!

Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1011, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-1011, default 1011):
Using default value 1011

Command (m for help): t
Selected partition 1
Hex code (type L to list codes): 83

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

Calling ioctl() to re-read partition table.
hda: hda1
hda: hda1
Syncing disks.
```

図 9.4 ディスク初期化方法

9.2.2.2. ファイルシステムの構築

初期化したディスクのパーティションにファイルシステムを構築します。ここでは、コンパクトフラッシュのパーティション 1（/dev/hda1）に EXT2 を構築します。

```
[armadillo ~]# mke2fs -O none /dev/hda1
mke2fs 1.25 (20-Sep-2001)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
15680 inodes, 62651 blocks
3132 blocks (4%) reserved for the super user
First data block=1
8 block groups
8192 blocks per group, 8192 fragments per group
1960 inodes per group
Superblock backups stored on blocks:
    8193, 16385, 24577, 32769, 40961, 49153, 57345

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

This filesystem will be automatically checked every 32 mounts or
180.00 days, whichever comes first. Use tune2fs -c or -i to override.
```

図 9.5 ファイルシステムの構築

9.2.2.3. マウント

ファイルシステムを作成後、「図 9.6. マウント方法」のようにコマンドを実行する事で、ext2 ファイルシステムを使用して/dev/hda1 を/mnt にマウントします。

```
[armadillo ~]# mount -t ext2 /dev/hda1 /mnt
```

図 9.6 マウント方法

10. ビデオ



開発ボードでは、ビデオ出力する場合は常に JP6 をオープンに設定してください。



現在のソフトウェアでは、デバイスが提供する設定の全てに対応していません。また、Armadillo-500 のビデオ出力は、VESA などの規格化されているタイミングを完全に満しているわけではあません。そのため、許容範囲の狭いディスプレイでは同期ずれが起こる場合があります。

10.1. カーネル起動ログをビデオに出力

ジャンパにより起動モードを保守モードに設定し、再起動します。

「図 10.1. コンソール指定 (ビデオ)」のように `setenv` コマンドでコンソールを変更します。

```
hermit> setenv console=tty1
```

図 10.1 コンソール指定 (ビデオ)

設定した後に通常と同じように Linux を起動するとカーネル起動ログはビデオに出力されます。

10.2. 解像度の変更

対応している解像度は、「表 10.1. 対応解像度一覧」に示すとおりです。

表 10.1 対応解像度一覧

指定子	色深度(bpp)	解像度	水平周波数(kHz)	垂直周波数(Hz)	ドットクロック(MHz)
CRT-VGA	16/24/32	640×480	31.469	59.942	25.176
CRT-SVGA	16/24/32	800×600	35.162	56.342	33.333

10.2.1. VGA に設定する

ジャンパにより起動モードを保守モードに設定し再起動します。

「図 10.2. 解像度の指定方法 (VGA)」のようにコマンドを実行します。

```
hermit> setenv console=tty1 video=mxcfb:CRT-VGA,16bpp,enable
```

図 10.2 解像度の指定方法 (VGA)

設定した後に通常と同じように Linux を起動すると解像度を変更されます。

10.2.2. SVGA に設定する

ジャンパにより起動モードを保守モードに設定し再起動します。

「[図 10.3. 解像度の指定方法 \(SVGA\)](#)」のようにコマンドを実行します。

```
hermit> setenv console=tty1 video=mxcfb:CRT-SVGA,16bpp,enable
```

図 10.3 解像度の指定方法 (SVGA)

設定した後に通常と同じように Linux を起動すると解像度を変更されてビデオに出力されます。

11. サウンド

開発ボードは、オーディオコーデックを搭載しており、録音及び再生をおこなうことができます。

11.1. コーデック仕様

録音及び再生のソースとして「表 11.1. 録音/再生ソース一覧」に示す入出力を使用することができます。

表 11.1 録音/再生ソース一覧

ソース	コネクタ	備考
LINE 入力	CON10	
マイク入力	CON11	デフォルト録音ソース
LINE 出力	CON12	
ヘッドフォン出力	CON13	デフォルト再生ソース

オーディオコーデックドライバは「表 11.2. オーディオコーデックドライバの仕様」に示す値に設定することができます。

表 11.2 オーディオコーデックドライバの仕様

チャンネル数	2 (stereo)
サンプリングフォーマット	Signed 16 bit, little-endian
サンプリング周波数	96000/48000/32000/8000 Hz

11.2. OSS によるサウンド機能の実現

Linux でサウンド機能を実現するには、ALSA^[1]と OSS^[2]の二つの方法があります。OSS は、ALSA の OSS エミュレーション^[3]によって実現されます。OSS エミュレーションには、以下の二つの方法があります。

- ・ カーネルレベル OSS エミュレーション (snd-ocm-oss, snd-mixer-oss, snd-seq-oss を使用)
- ・ alsa-oss パッケージによるユーザ空間 OSS エミュレーション (aoss スクリプトを使用)

開発ボードのデフォルト設定では、カーネルレベル OSS エミュレーションによる OSS 機能を提供していません。

OSS プログラミングについては、「Open Sound System Programmer's Guide」^[4]をご参照ください。

11.2.1. サウンドを再生する

OSS を使用したサウンドの再生には **vplay** コマンドを使用します。

^[1]Advanced Linux Sound Architecture <http://alsa.sourceforge.net>

^[2]Open Sound System <http://developer.opensound.com/>

^[3]<http://alsa.opensrc.org/OssEmulation>

^[4]<http://www.opensound.com/pguide/oss.pdf>

```
[armadillo500 ~]# vplay -r -S -s 8000 -b 16 -t 5 filename
```

- r 再生するファイル形式の指定 -r=raw 形式、-w=wav 形式、-v=voc 形式 (デフォルト)
- S ステレオ (デフォルト モノラル)
- s 8000 サンプリング周波数[samples/sec] 300 以下の場合には 1000 倍される (デフォルト 8000Hz)
- b 16 サンプルサイズ[bits/sample] 8 または 16 を指定可能 (デフォルト 8)
- t 5 再生時間[sec] (デフォルト ファイルサイズ)
- filename 再生するサウンドファイル

図 11.1 サウンドの再生

11.2.2. 録音する

OSS を使用したサウンドの録音には **vrec** コマンドを使用します。

```
[armadillo500 ~]# vrec -r -S -s 8000 -b 16 -t 5 filename
```

- r 再生するファイル形式の指定 -r=raw 形式、-w=wav 形式、-v=voc 形式 (デフォルト)
- S ステレオ (デフォルト モノラル)
- s 8000 サンプリング周波数[samples/sec] 300 以下の場合には 1000 倍される (デフォルト 8000Hz)
- b 16 サンプルサイズ[bits/sample] 8 または 16 を指定可能 (デフォルト 8)
- t 5 録音時間[sec] (デフォルト 無期限)
- filename 録音するサウンドファイル

図 11.2 サウンドの録音

11.2.3. 音量を変更する

OSS を使用して再生/録音時の音量を変更するには **mixer** コマンドを使用します。

```
[armadillo500 ~]# mixer 指定子 音量
```

- 指定子 line(LINE 入力)または pcm2(ヘッドフォン出力)を指定します
LINE 出力とマイク入力の音量は固定です
- 音量 音量を 0-100 の間で指定します

図 11.3 音量の変更

11.2.4. 録音ソースを変更する

OSS を使用して録音ソースを変更するには **mixer** コマンドを使用します。

```
[armadillo500 ~]# mixer +rec 指定子
```

指定子 line(LINE 入力)または mic(マイク入力)を指定します

図 11.4 録音ソースの指定

12. その他のデバイス

12.1. LED

12.1.1. ledctrl による制御

ledctrl では、ボード上の D1 から D5 までの LED を制御することができます。ledctrl は、「図 12.1. ledctrl コマンドフォーマット」のように使用します。

```
[armadillo ~]# ledctrl [ledname] [command]
```

図 12.1 ledctrl コマンドフォーマット

ledname には、制御対象の LED ID を指定^[1]します。対応する ID は、「表 12.1. ledctrl : ledname」のとおりです。

表 12.1 ledctrl : ledname

LED ID	デバイス	色	デフォルトソフトウェアでの使用状況
led1	D1	緑	使用しません
led2	D2	緑	使用しません
led3	D3	緑	使用しません
led4	D4	緑	使用しません
led5	D5	赤	ソフトウェアの起動ステータスを表示します。
all	D1 ~ D5	-	-

command には、制御対象の LED に対しての制御方法を指定します。command 一覧は、「表 12.2. ledcrown : command」のとおりです。

表 12.2 ledcrown : command

command	動作
on	LED を点灯させます。
off	LED を消灯させます。
blink_on [interval_ms]	LED の点滅を開始します。点滅間隔は、interval_ms で指定します。デフォルトでは 200ms の点滅間隔となります。 また、点滅を開始する 1 つ前の状態を保存し、blink_off 時に状態を元に戻すことができます。
blink_off	LED の点滅を終了します。
status	現在の状態を表示します。表示する項目は、brightness、delay_on、delay_off となります。

12.1.2. ledctrl 使用例

- ・ 点灯させる

^[1] LED を複数指定する場合は、LED ID を',' (カンマ) で区切って指定してください。

```
[armadillo ~]# ledctrl all on
```

図 12.2 ledctrl 使用例 1

- ・ 消灯させる

```
[armadillo ~]# ledctrl led1, led2 off
```

図 12.3 ledctrl 使用例 2

- ・ 点滅を開始させる

```
[armadillo ~]# ledctrl led3 blink_on 500
```

図 12.4 ledctrl 使用例 3

- ・ 点滅を終了させる

```
[armadillo ~]# ledctrl led3 blink_off
```

図 12.5 ledctrl 使用例 4

12.2. タクトスイッチ

12.2.1. swmgr によるイベント取得

swmgr では、ボード上の SW1 及び SW2 の押下イベント^[2]を取得し、指定する動作を実行することができます。

swmgr は、「図 12.6. swmgr コマンドフォーマット」のように使用します。

```
[armadillo ~]# swmgr [SW ID] [LOOP] [COMMAND]
```

図 12.6 swmgr コマンドフォーマット

SW ID には、イベントを取得したい SW の ID を指定します。対応する ID は、「表 12.3. swmgr : SW ID」のとおりです。

表 12.3 swmgr : SW ID

SW ID	デバイス
sw1	SW1
sw2	SW2

LOOP には、イベントを待つ回数を指定します。0 を指定すると無制限になります。

^[2]SW を押してから離れたときにイベントが発生します。

COMMAND には、イベントが発生した時に実行する動作を記述します。

12.2.2. swmgr 使用例

SW1 が押されたときに時刻を表示する

```
[armadillo ~]# swmgr sw1 1 date
```

図 12.7 swmgr 使用例 1

SW1 が押されたときに D5 を点灯、SW2 が押されたときには D5 を消灯する

```
[armadillo ~]# swmgr sw1 0 ledctrl led5 on &
[armadillo ~]# swmgr sw2 0 ledctrl led5 off &
```

※終了する場合は次のコマンドを実行します

```
[armadillo ~]# killall swmgr
```

図 12.8 swmgr 使用例 2

12.3. GPIO

開発ボードの GPIO (CON16) は、/sys/devices/platform/armadillo5x0_gpio.0/ports の下にあるファイルを読み書きすることにより、簡易的に制御することができます。

制御できる項目は、「表 12.4. GPIO ファイルノード」のとおりです。

表 12.4 GPIO ファイルノード

ファイル名	説明
gpio*_dir	GPIO の Direction を変更することができます。0 を書き込むと INPUT に、1 を書き込むことにより OUTPUT になります。それ以外は未定義となります。 読み出した場合は、現在の状態を取得できます。
gpio*_data	Direction が INPUT の場合は、GPIO の現在の入力レベルを読み出すことができます。書き込みは無視されます。Direction が OUTPUT の場合は、GPIO の出力レベルを設定できます。読み出した場合は、現在の出力レベルを取得できます。 読み書きする値は、0 で LOW レベルを表し、1 で HIGH レベルを表します。 それ以外は未定義となります。

12.3.1. Direction を INPUT にする

GPIO0 (CON16 2 ピン) を入力にする場合は、「図 12.9. GPIO : Direction を INPUT にする」のようにコマンドを実行します。

```
[armadillo ~]# echo 0 > /sys/devices/platform/armadillo5x0_gpio.0/ports/gpio0_dir
```

図 12.9 GPIO : Direction を INPUT にする

現在の GPIO0 の入力状態を取得するには、「図 12.10. GPIO : INPUT 時の入力状態を取得する」のようにコマンドを実行します。

```
[armadillo ~]# cat /sys/devices/platform/armadillo5x0_gpio.0/ports/gpio0_data  
0
```

図 12.10 GPIO : INPUT 時の入力状態を取得する

12.3.2. Direction を OUTPUT にする

GPIO0 (CON16 2 ピン) を出力にする場合は、「図 12.11. Direction を OUTPUT にする」のようにコマンドを実行します。

```
[armadillo ~]# echo 1 > /sys/devices/platform/armadillo5x0_gpio.0/ports/gpio0_dir
```

図 12.11 Direction を OUTPUT にする

HIGH レベルを出力するには、「図 12.12. OUTPUT 時の出力状態を変更する」のようにコマンドを実行します。

```
[armadillo ~]# echo 1 > /sys/devices/platform/armadillo5x0_gpio.0/ports/gpio0_data
```

図 12.12 OUTPUT 時の出力状態を変更する

改訂履歴

バージョン	年月日	改訂内容
1.0.0	2007/7/27	・ 初版発行
1.0.1	2007/9/14	・ 「1.6. 保証に関する注意事項」の製品の保証方法を修正
1.0.2	2008/3/27	・ 「図 3.1. 見取り図」に JP7 の情報を追加 ・ 「3.4. ジャンパピンの設定について」に JP7 の情報を追加
1.0.3	2008/10/02	・ タイトルを英語表記からカタカナ表記に ・ 「表 3.1. JP7 の設定」JP7 に関する注意事項を追記
1.0.4	2008/12/29	・ 「Development board」を「開発ボード」という表記に統一 ・ 「6. コンフィグ領域 – 設定ファイルの保存領域」を修正 ・ 「11. サウンド」について追加 ・ 「図 5.3. ルートファイルシステムの指定」「10. ビデオ」誤記修正
1.1.0	2009/03/18	・ 「1. はじめに」、「3. 作業の前に」、「4. 起動と終了」、「7. ネットワーク」構成変更 ・ 誤記、表記ゆれ修正
1.1.1	2009/07/17	・ 「5. 起動モードとブートローダの機能」の表記を変更 ・ 「9.2.2.3. マウント」をより正確な情報に修正 ・ 「12.1.1. ledctrl による制御」の記述を簡潔にした ・ 本文のレイアウト統一 ・ 表記ゆれ修正 ・ 「図 7.4. 設定を反映させる」のコマンドオプションの誤記を修正
1.1.2	2009/07/29	・ 製品保証に関する記載を http://www.atmark-techno.com/support/warranty-policy に移動(2009/08/03 適用)
1.2.0	2010/01/29	・ 表のレイアウト統一 ・ guest ユーザパスワードの誤記を修正 ・ ftp ユーザパスワードの表記ゆれを修正 ・ 「表 10.1. 対応解像度一覧」色深度の誤記を修正 ・ A5567 に対応 ・ 「2.2. 取扱い上の注意事項」誤記修正
1.3.0	2010/12/24	・ 「表 10.1. 対応解像度一覧」に水平/垂直周波数およびドットクロックを追記 ・ 「8. シリアルインターフェース」を追記 ・ 「図 9.2. NAND の初期化例」誤記修正 ・ 注意事項を「2. 注意事項」に移動 ・ ジャンパ設定の説明を追記
1.3.1	2011/03/25	・ 「5.2. Linux カーネル起動オプションの設定」に Hermit-AT のモードについて説明を追記 ・ 会社住所変更
1.3.2	2011/10/21	・ Web サイトの名称を変更

Armadillo-500 開発ボードスタートアップガイド
Version 1.3.2
2011/10/24

株式会社アットマークテクノ

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