

Armadillo-500 Development Board

スタートアップガイド

Version 1.0.1



株式会社アットマークテクノ

<http://www.atmark-techno.com/>

 **Armadillo** 公式サイト

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

目次

| | | |
|----------|-----------------------|----|
| 1. | はじめに | 1 |
| 1.1. | 対象となる読者 | 2 |
| 1.2. | 本書の構成 | 2 |
| 1.3. | 表記について | 2 |
| 1.3.1. | フォント | 2 |
| 1.3.2. | コマンド入力例 | 2 |
| 1.3.3. | アイコン | 3 |
| 1.4. | 謝辞 | 3 |
| 1.5. | 注意事項 | 3 |
| 1.5.1. | 安全に関する注意事項 | 3 |
| 1.5.2. | 取り扱い上の注意事項 | 3 |
| 1.5.3. | ソフトウェアに関する注意事項 | 4 |
| 1.6. | 保障に関して | 4 |
| 1.6.1. | 保障範囲 | 4 |
| 1.6.2. | 保障対象外になる場合 | 4 |
| 1.6.3. | 免責事項 | 4 |
| 1.7. | 商標について | 4 |
| 2. | 作業の前に | 5 |
| 2.1. | 見取り図 | 5 |
| 2.2. | 準備するもの | 6 |
| 2.3. | 接続方法 | 6 |
| 2.4. | ジャンパピンの設定について | 7 |
| 2.5. | シリアル通信ソフトウェアの設定 | 7 |
| 3. | 起動と終了 | 8 |
| 3.1. | 起動 | 8 |
| 3.2. | ログイン | 11 |
| 3.3. | 終了方法 | 12 |
| 4. | 起動モードとブートローダの機能 | 13 |
| 4.1. | 起動モードの選択 | 13 |
| 4.2. | Linuxカーネル起動オプションの設定 | 13 |
| 4.2.1. | コンソールの設定 | 13 |
| 4.2.2. | ルートファイルシステムの設定 | 14 |
| 4.2.2.1. | ルートファイルシステムイメージの場所 | 14 |
| 4.2.2.2. | ルートファイルシステムタイプ | 14 |
| 4.2.2.3. | マウント前ディレイ | 14 |
| 4.2.3. | その他の起動オプション | 15 |
| 4.2.4. | 起動オプションの設定例 | 15 |
| 4.3. | カーネルイメージのロケーション設定 | 16 |
| 5. | コンフィグ領域 - 設定ファイルの保存領域 | 17 |
| 5.1. | コンフィグ領域の読出し | 17 |
| 5.2. | コンフィグ領域の保存 | 17 |
| 5.3. | コンフィグ領域の初期化 | 17 |
| 6. | ネットワーク | 18 |
| 6.1. | ネットワークの設定 | 18 |
| 6.1.1. | 固定IPアドレスに設定する | 18 |
| 6.1.2. | DHCPに設定する | 19 |
| 6.1.3. | DNSサーバを指定する | 19 |
| 6.1.4. | 接続を確認する | 19 |

| | | |
|----------|---------------------|----|
| 6.2. | ファイアーウォール | 19 |
| 6.3. | ネットワークアプリケーション | 20 |
| 6.3.1. | TELNET | 20 |
| 6.3.1.1. | TELNETサーバ | 20 |
| 6.3.1.2. | TELNETクライアント | 20 |
| 6.3.2. | FTP | 20 |
| 6.3.2.1. | FTPサーバ | 20 |
| 6.3.2.2. | FTPクライアント | 20 |
| 6.3.3. | SSH | 21 |
| 6.3.3.1. | SSHサーバ | 21 |
| 6.3.3.2. | SSHクライアント | 21 |
| 6.3.4. | Webサーバ | 21 |
| 6.3.5. | NTPクライアント | 22 |
| 7. | ストレージ | 23 |
| 7.1. | ストレージとして使用可能なデバイス | 23 |
| 7.2. | ストレージの初期化とマウント | 23 |
| 7.2.1. | NANDフラッシュデバイス | 23 |
| 7.2.1.1. | ファイルシステムをJFFS2にする場合 | 23 |
| 7.2.1.2. | マウント | 24 |
| 7.2.2. | その他のデバイス | 24 |
| 7.2.2.1. | ディスクの初期化 | 24 |
| 7.2.2.2. | ファイルシステムの構築 | 25 |
| 7.2.2.3. | マウント | 25 |
| 8. | ビデオ | 26 |
| 8.1. | カーネル起動ログをビデオに出力 | 26 |
| 8.2. | 解像度の変更 | 26 |
| 8.2.1. | VGAに設定する | 26 |
| 8.2.2. | SVGAに設定する | 27 |
| 9. | その他のデバイス | 28 |
| 9.1. | LED | 28 |
| 9.1.1. | ledctrlによる制御 | 28 |
| 9.1.2. | ledctrl使用例 | 29 |
| 9.2. | タクトスイッチ | 29 |
| 9.2.1. | swmgrによるイベント取得 | 29 |
| 9.2.2. | swmgr使用例 | 30 |
| 9.3. | GPIO | 30 |
| 9.3.1. | DirectionをINPUTにする | 31 |
| 9.3.2. | DirectionをOUTPUTにする | 31 |

表目次

| | | |
|-------|---------------------------|----|
| 表 1-1 | 使用しているフォント | 2 |
| 表 1-2 | 表示プロンプトと実行環境の関係 | 2 |
| 表 2-1 | シリアル通信設定 | 7 |
| 表 3-1 | 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 | 11 |
| 表 4-1 | 起動モード | 13 |
| 表 4-2 | コンソール指定に伴う出力先 | 14 |
| 表 4-3 | ルートファイルシステムデバイス | 14 |
| 表 4-4 | カーネルイメージのロケーション | 16 |
| 表 6-1 | 固定IPアドレス設定 | 18 |
| 表 6-2 | telnetでログイン可能なユーザ | 20 |
| 表 6-3 | ftpでログイン可能なユーザ | 20 |
| 表 6-4 | sshでログイン可能なユーザ | 21 |
| 表 7-1 | ストレージデバイス | 23 |
| 表 8-1 | 対応解像度一覧 | 26 |
| 表 9-1 | ledctrl : LED ID | 28 |
| 表 9-2 | ledctrl : 制御ID | 28 |
| 表 9-3 | swmgr : SW ID | 29 |
| 表 9-4 | GPIOファイルノード | 30 |

図目次

| | | |
|-------|----------------------|----|
| 図 2-1 | 見取り図 | 5 |
| 図 2-2 | 接続図 | 6 |
| 図 3-1 | 起動ログ | 8 |
| 図 3-2 | 起動ログ (続き 1) | 9 |
| 図 3-3 | 起動ログ (続き 2) | 10 |
| 図 3-4 | 起動ログ (続き 3) | 11 |
| 図 3-5 | 終了方法 | 12 |
| 図 4-1 | Linuxカーネル起動オプションのクリア | 13 |
| 図 4-2 | コンソールの指定 | 13 |
| 図 4-3 | ルートファイルシステムの指定 | 14 |
| 図 4-4 | 起動オプション設定例 1 | 15 |
| 図 4-5 | 起動オプション設定例 2 | 15 |
| 図 4-6 | 起動オプション設定例 3 | 15 |
| 図 4-7 | カーネルイメージロケーション指定 | 16 |
| 図 5-1 | コンフィグ領域の読出し方法 | 17 |
| 図 5-2 | コンフィグ領域の保存方法 | 17 |
| 図 5-3 | コンフィグ領域の初期化方法 | 17 |
| 図 6-1 | 固定IPアドレス設定 | 18 |
| 図 6-2 | DHCP設定 | 19 |
| 図 6-3 | DNSサーバの設定 | 19 |
| 図 6-4 | 設定を反映させる | 19 |
| 図 6-5 | PING確認 | 19 |
| 図 6-6 | iptables | 19 |
| 図 6-7 | telnet | 20 |
| 図 6-8 | ftp | 20 |
| 図 6-9 | ssh | 21 |

| | |
|---------------------------------|----|
| 図 6-10 Armadillo-500 トップ画面 | 21 |
| 図 6-11 msntp | 22 |
| 図 7-1 MTDパーティション一覧 | 23 |
| 図 7-2 NANDの初期化例 | 23 |
| 図 7-3 NANDのマウント方法 | 24 |
| 図 7-4 ディスク初期化方法 | 24 |
| 図 7-5 ファイルシステムの構築 | 25 |
| 図 7-6 マウント方法 | 25 |
| 図 8-1 コンソール指定 (ビデオ) | 26 |
| 図 8-2 解像度の指定方法 (VGA) | 26 |
| 図 8-3 解像度の指定方法 (SVGA) | 27 |
| 図 9-1 ledctrlコマンド例 | 28 |
| 図 9-2 ledctrl使用例 1 | 29 |
| 図 9-3 ledctrl使用例 2 | 29 |
| 図 9-4 ledctrl使用例 3 | 29 |
| 図 9-5 ledctrl使用例 4 | 29 |
| 図 9-6 swmgrコマンド例 | 29 |
| 図 9-7 swmgr使用例 1 | 30 |
| 図 9-8 swmgr使用例 2 | 30 |
| 図 9-9 GPIO : DirectionをINPUTにする | 31 |
| 図 9-10 GPIO : INPUT時の入力状態を取得する | 31 |
| 図 9-11 DirectionをOUTPUTにする | 31 |
| 図 9-12 OUTPUT時の出力状態を変更する | 31 |

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ベースボードの構成となります。

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 Development Board Software Manual」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。

1.1. 対象となる読者

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

上記以外の方でも、本書を有効に利用していただけたら幸いです。

1.2. 本書の構成

本書では、開発ボード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

1.3. 表記について

1.3.1. フォント

本書では以下のような意味でフォントを使いわけています。

表 1-1 使用しているフォント

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

1.3.2. コマンド入力例

本書に記載されているコマンドの入力例は、表示されているプロンプトによって、それぞれに対応した実行環境を想定して書かれています。「/」の部分はカレントディレクトリによって異なります。各ユーザのホームディレクトリは「~」で表わします。

表 1-2 表示プロンプトと実行環境の関係

| プロンプト | コマンドの実行環境 |
|--------------------|-------------------|
| [PC /]# | 作業用 PC 上の特権ユーザで実行 |
| [PC /]\$ | 作業用 PC 上の一般ユーザで実行 |
| [armadillo500 /]# | 開発ボード上の特権ユーザで実行 |
| [armadillo500 /]\$ | 開発ボード上の一般ユーザで実行 |
| hermit> | 開発ボード上の保守モードで実行 |

1.3.3. アイコン

本書では以下のようにアイコンを使用しています。



注意事項を記載します。



役に立つ情報を記載します。

1.4. 謝辞

開発ボードで使用しているソフトウェアは Free Software / Open Source Software で構成されています。Free Software / Open Source Software は世界中の多くの開発者の成果によってなっています。この場を借りて感謝の意を示したいと思います。

1.5. 注意事項

1.5.1. 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ますようお願いいたします。



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おりますので、その誤作動や故障が直接生命を脅かしたり、身体・財産等に危害を及ぼす恐れのある装置（医療機器・交通機器・燃焼制御・安全装置等）に組み込んで使用したりしないでください。また、半導体部品を使用した製品は、外来ノイズやサージにより誤作動したり故障したりする可能性があります。ご使用になる場合は万一誤作動、故障した場合においても生命・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・ブレーカ等の保護回路の設置、装置の多重化等）に万全を期されますようお願い申し上げます。

1.5.2. 取り扱い上の注意事項

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

- 電源の投入
本製品や周辺回路に電源が入っている状態で汎用入出力コネクタの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ますので、ご使用になるまでは帯電防止対策のされている、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で使用している 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。

1.5.3. ソフトウェアに関する注意事項

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、現状のまま(AS IS)提供されるものであり、特定の目的に適合することや、その信頼性、正確性を保証するものではありません。また、本製品の使用による結果についてもなんら保証するものではありません。

1.6. 保障に関して

1.6.1. 保障範囲

付属品(ソフトウェアを含みます)を使用し、取扱説明書、各注意事項に基づく正常なご使用に限り有効です。万一正常なご使用のもと製品が故障した場合は、初期不良保証期間内であれば新品交換をさせていただきます。

1.6.2. 保障対象外になる場合

次のような場合の故障・損傷は、保証期間内であっても保証対象外になります。

- 取扱説明書に記載されている使用方法、または注意に反したお取り扱いによる場合
- 改造や部品交換に起因する場合。または正規のものではない機器を接続したことによる場合
- お客様のお手元に届いた後の輸送、移動時の落下など、お取り扱いの不備による場合
- 火災、地震、水害、落雷、その他の天災、公害や異常電圧による場合
- ACアダプター、専用ケーブルなどの付属品について、同梱のものを使用していない場合
- 修理依頼の際に購入時の付属品がすべて揃っていない場合

1.6.3. 免責事項

弊社に故意または重大な過失があった場合を除き、製品の使用および、故障、修理によって発生するいかなる損害についても、弊社は一切の責任を負わないものとします。



本製品は購入時の初期不良以外の保証をおこなっておりません。保証期間は商品到着後2週間です。本製品をご購入されましたらお手数でも必ず動作確認をおこなってからご使用ください。本製品に対して注意事項を守らずに発生した故障につきましては保証対象外となります。

1.7. 商標について

Armadilloは(株)アットマークテクノの登録商標です。
その他の記載の会社名、製品名は、それぞれの登録商標または商標です。

2. 作業の前に

2.1. 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。

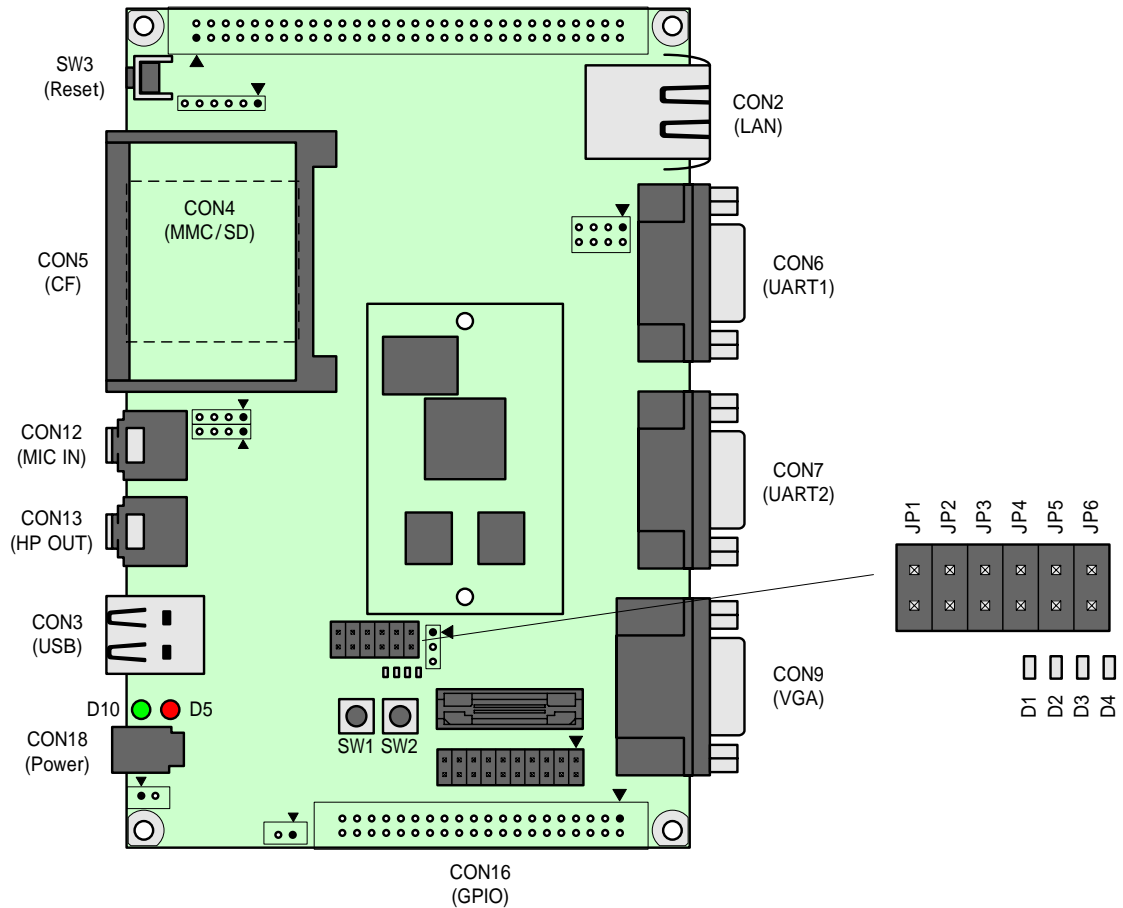


図 2-1 見取り図

2.2. 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ポートを持つ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。
- 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。

2.3. 接続方法

開発ボードの接続例です。

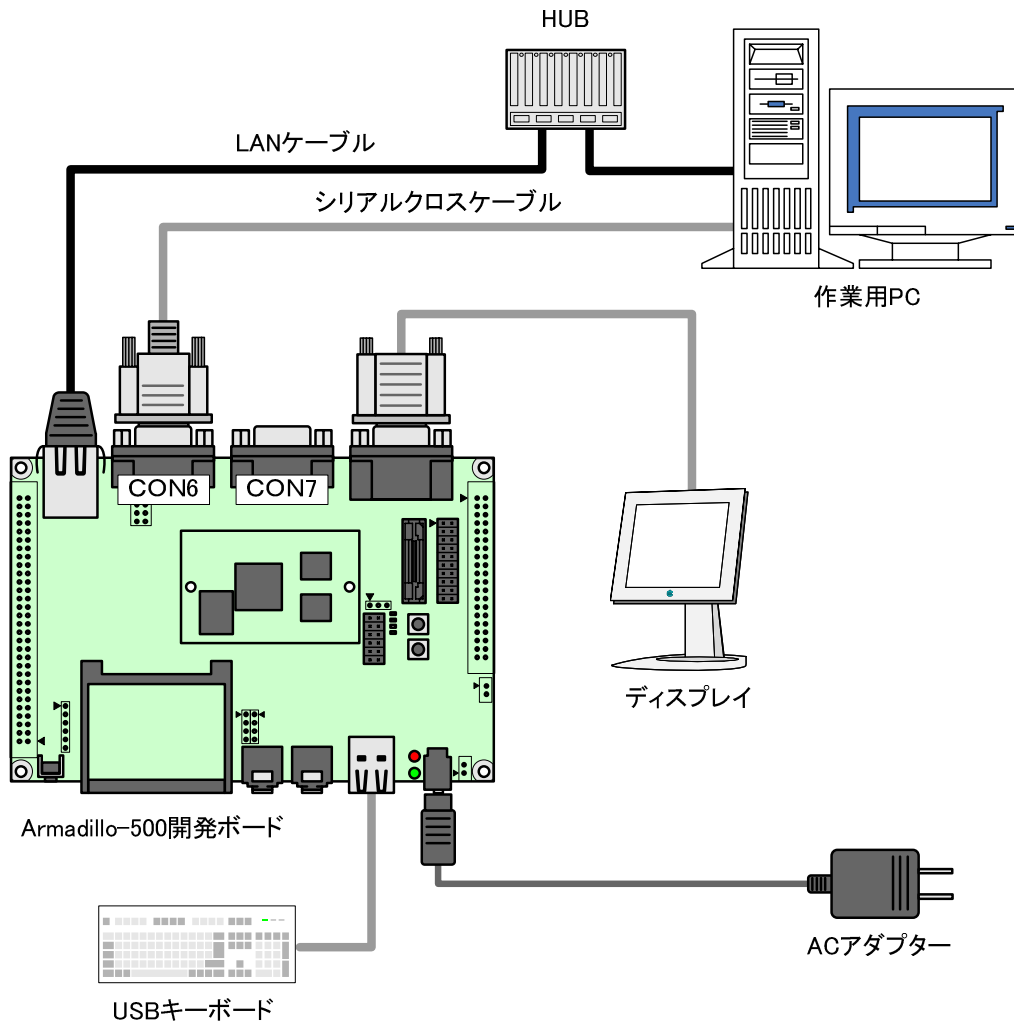


図 2-2 接続図

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

2.4. 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。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。

また、ジャンパピンの位置は図 2-1で確認することができます。

2.5. 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 2-1のように設定してください。

表 2-1 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 |

3. 起動と終了

3.1. 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
mxcmci-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)

```

図 3-1 起動ログ

```
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: ttymxcl at MMIO 0x43f90000 (irq = 45) is a Freescale MXC
mxcintuart.1: ttymxcl 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
```

図 3-2 起動ログ (続き 1)

```
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
```

図 3-3 起動ログ (続き 2)


```

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
Starting inetd: done
Starting sshd: done
Starting tftpd: 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 3-4 起動ログ (続き 3)

3.2. 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 3-1に示すユーザでログインすることができます。

表 3-1 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 | 特権ユーザ |
| guest | guest | 一般ユーザ |

3.3. 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。

```
[root@armadillo500 (ttyMXC0) ~]# halt
[root@armadillo500 (ttyMXC0) ~]#
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 3-5 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にご注意ください。

4. 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。

4.1. 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 4-1 のようになります。

表 4-1 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。 |

4.2. 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に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 4-1 Linux カーネル起動オプションのクリア

4.2.1. 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。

```
hermit> setenv console=ttymxc0  
コンソール指定子
```

図 4-2 コンソールの指定

設定によるログの出力先は、表 4-2 のようになります。

表 4-2 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 |

4.2.2. 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。

| | | | |
|----------------|-----------------------|--------------------|--------------------|
| hermit> setenv | <u>root=/dev/hda1</u> | <u>rootfs=ext2</u> | <u>rootdelay=3</u> |
| | イメージの場所 | ファイルシステムタイプ | マウント前ディレイ |

図 4-3 ルートファイルシステムの指定

4.2.2.1. 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 4-3 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 4-3 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/mtdblock | /dev/mtdblock4 |
| MMC/SD カードディスク | /dev/mmcblk0 | /dev/mmcblk0p1 |

4.2.2.2. 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^[2]。

4.2.2.3. 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。

[1] ブートローダのログは出力されません。

[2] 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。

4.2.3. 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「8. ビデオ」を参照してください。

4.2.4. 起動オプションの設定例

- コンソールを CON7 にする場合

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

図 4-4 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 4-5 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^[1]

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

図 4-6 起動オプション設定例 3

^[1] 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。

4.3. カーネルイメージのロケーション設定

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。

```
hermit> setbootdevice flash
                指定子
```

図 4-7 カーネルイメージロケーション指定

開発ボードでは、表 4-4に示すデバイスを指定できます。

表 4-4 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。

5. コンフィグ領域 - 設定ファイルの保存領域

この章では、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領域について説明します。

5.1. 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」に作成されます。

```
[armadillo500 ~]# flatfsd -r
```

図 5-1 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。

5.2. 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」以下のファイルです。コンフィグ領域のサイズは 128Kバイト^[1]です。

```
[armadillo500 ~]# flatfsd -s
```

図 5-2 コンフィグ領域の保存方法

5.3. 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」以下のファイルがコンフィグ領域に保存され、且つ「/etc/config」にファイルが複製されます。

```
[armadillo500 ~]# flatfsd -w
```

図 5-3 コンフィグ領域の初期化方法

^[1] デフォルトイメージでのサイズです。

6. 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。

6.1. ネットワークの設定

開発ボード上の「/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、「5.2.コンフィグ領域の保存」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。

6.1.1. 固定 IP アドレスに設定する

表 6-1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 6-1のように編集します。

表 6-1 固定 IP アドレス設定

| 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 |

```
[armadillo500 ~]# 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 6-1 固定 IP アドレス設定

6.1.2. 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 6-2のように編集します。

```
[armadillo500 ~]# 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 6-2 DHCP 設定

6.1.3. 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。

```
[armadillo500 ~]# vi /etc/config/resolv.conf
nameserver 192.168.10.1
```

図 6-3 DNS サーバの設定

6.1.4. 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。

```
[armadillo500 ~]# ifdown -a
[armadillo500 ~]# ifup -a
```

図 6-4 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。

```
[armadillo500 ~]# ping 192.168.10.1
```

図 6-5 PING 確認

6.2. ファイアーウォール

開発ボードでは、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 6-6 のようにコマンド実行してください。

```
[armadillo500 ~]# iptables -L
```

図 6-6 iptables

6.3. 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。

6.3.1. TELNET

6.3.1.1. 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 6-2に示すユーザでログインすることができます。

表 6-2 telnet でログイン可能なユーザ

| ユーザ名 | パスワード |
|-------|-------|
| guest | guest |

6.3.1.2. 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。telnetを使用するには、図 6-7のようにコマンドを実行します。

```
[armadillo500 ~]# telnet 192.168.10.1
```

図 6-7 telnet

6.3.2. FTP

6.3.2.1. 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 6-3に示すユーザでログインすることができます。

表 6-3 ftp でログイン可能なユーザ

| ユーザ名 | パスワード |
|------|--------|
| ftp | (none) |

6.3.2.2. 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。ftpを使用するには、図 6-8のようにコマンドを実行してください。

```
[armadillo500 ~]# ftp 192.168.10.1
```

図 6-8 ftp

6.3.3. SSH

6.3.3.1. 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 6-4に示すユーザでログインすることができます。

表 6-4 ssh でログイン可能なユーザ

| ユーザ名 | パスワード |
|-------|-------|
| guest | guest |

6.3.3.2. 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。sshを使用するには、図 6-9のようにコマンドを実行します。

```
[armadillo500 ~]# ssh user@192.168.10.1
```

図 6-9 ssh

6.3.4. Web サーバ

開発ボードでは、Webサーバが動作しています。PCなどのWebブラウザから開発ボードのURL (http://[開発ボードのIPアドレス])^[1]をアクセスすると、「/home/www-data」以下のディレクトリをブラウズすることができます。

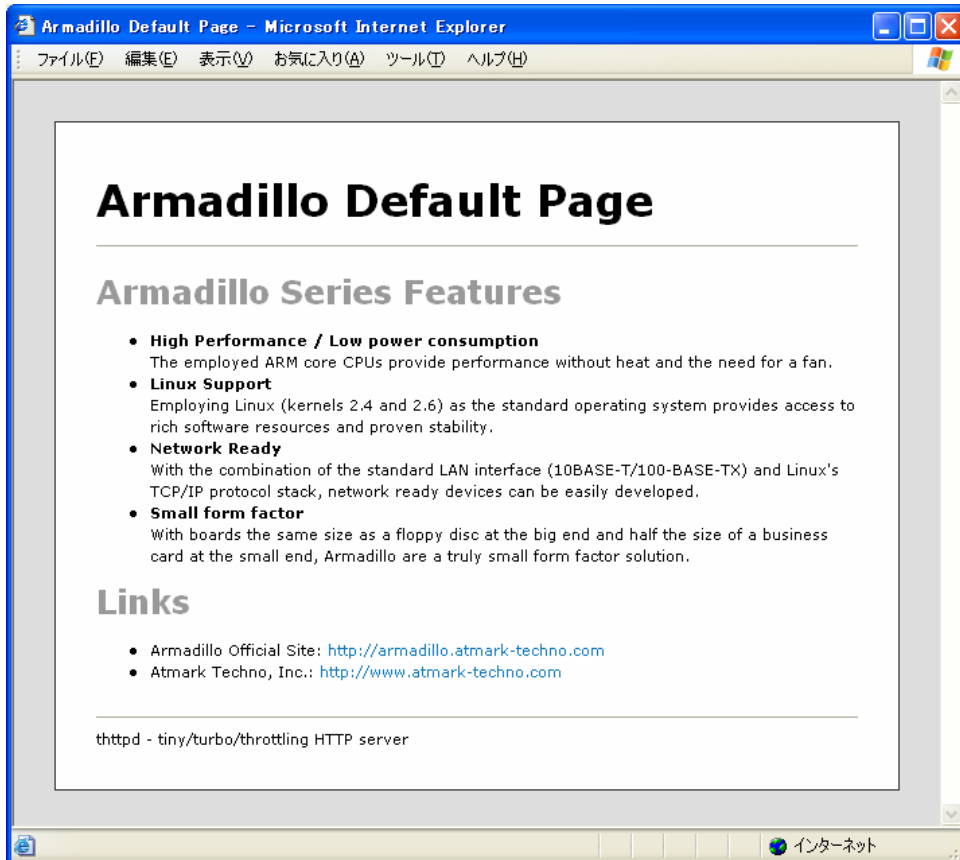


図 6-10 Armadillo-500 トップ画面

[1] 開発ボードの IP アドレスが 192.168.10.10 の場合、<http://192.168.10.10/> となります。

6.3.5. NTP クライアント

開発ボードでは、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 6-11のようにコマンドを実行します。

```
[armadillo500 ~]# 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 6-11 msntp

7.ストレージ

7.1. 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 7-1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 7-1 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 |

7.2. 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。

7.2.1. 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 7-1のようにコマンドを実行すると表示されます。

```
[armadillo500 ~]# 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 7-1 MTD パーティション一覧

7.2.1.1. 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 7-2のようにコマンドを実行してください。

```
[armadillo500 ~]# flash_eraseall -j /dev/flash/nand.free
Erasing 128 Kibyte @ 65c0000 -- 99 % complete. Cleanmarker written at 65c0000.
```

図 7-2 NAND の初期化例

7.2.1.2. マウント

初期化後、マウントするには、図 7-3のようにコマンドを実行します。

```
[armadillo500 ~]# mount -t jffs2 /dev/mtdblock7 /mnt
```

図 7-3 NAND のマウント方法

7.2.2. その他のデバイス

7.2.2.1. 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。

```
[armadillo500 ~]# 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 7-4 ディスク初期化方法

7.2.2.2. 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。

```
[armadillo500 ~]# 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 7-5 ファイルシステムの構築

7.2.2.3. マウント

ファイルシステムを作成後、マウントするには図 7-6のようにコマンドを実行します。

```
[armadillo500 ~]# mount -t ext2 /dev/hda1 /mnt
          ファイルシステム デバイス・パーティション マウントポイント
```

図 7-6 マウント方法

8. 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。

8.1. 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 8-1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 8-1 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。

8.2. 解像度の変更

対応している解像度は、表 8-1に示すとおりです。

表 8-1 対応解像度一覧

| 指定子 | 色深度(bpp) | 解像度 |
|----------|----------|-----------|
| CRT-VGA | 8/16 | 640 × 480 |
| CRT-SVGA | 8/16 | 800 × 600 |

8.2.1. 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 8-2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 8-2 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。

8.2.2. 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 8-3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 8-3 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が変更されてビデオに出力されます。

9. その他のデバイス

9.1. LED

9.1.1. ledctrl による制御

ledctrl では、ボード上の D1 ~ D5 までの LED を制御することができます。

ledctrl は、図 9-1 のように使用します。

```
[armadillo500 ~]# ledctrl led1 on
                        LED ID  制御 ID
```

図 9-1 ledctrl コマンド例

LED ID には、制御対象の LED の ID を指定 ^[1] します。対応する ID は、表 9-1 のとおりです。

表 9-1 ledctrl : LED ID

| 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 | | |

制御 ID には、制御対象の LED に対しての制御方法を指定します。制御 ID は、表 9-2 のとおりです。

表 9-2 ledctrl : 制御 ID

| 制御 ID | 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。 |

[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。

9.1.2. ledctrl 使用例

- 点灯させる

```
[armadillo500 ~]# ledctrl all on
```

図 9-2 ledctrl 使用例 1

- 消灯させる

```
[armadillo500 ~]# ledctrl led1,led2 off
```

図 9-3 ledctrl 使用例 2

- 点滅を開始させる

```
[armadillo500 ~]# ledctrl led3 blink_on 500
```

図 9-4 ledctrl 使用例 3

- 点滅を終了させる

```
[armadillo500 ~]# ledctrl led3 blink_off
```

図 9-5 ledctrl 使用例 4

9.2. タクトスイッチ

9.2.1. 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^[1]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 9-6のように使用します。

```
[armadillo500 ~]# swmgr sw1 1 date
                        SW ID  LOOP  COMMAND
```

図 9-6 swmgr コマンド例

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 9-3のとおりです。

表 9-3 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。
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。

^[1] SW を押してから離れたときにイベントが発生します。

9.2.2. swmgr 使用例

SW1 が押されたときに時刻を表示する

```
[armadillo500 ~]# swmgr sw1 1 date
```

図 9-7 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

```
[armadillo500 ~]# swmgr sw1 0 ledctrl led5 on &
[armadillo500 ~]# 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
[armadillo500 ~]# killall swmgr
```

図 9-8 swmgr 使用例 2

9.3. 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 9-4のとおりです。

表 9-4 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。 |

9.3.1. Direction を INPUT にする

GPIO0 (CON16 2 ピン) を入力にする場合は、図 9-9 のようにコマンドを実行します。

```
[armadillo500 ~]# echo 0 > ¥  
/sys/devices/platform/armadillo5x0_gpio.0/ports/gpio0_dir
```

図 9-9 GPIO : Direction を INPUT にする

現在のGPIO0 の入力状態を取得するには、図 9-10 のようにコマンドを実行します。

```
[armadillo500 ~]# cat ¥  
/sys/devices/platform/armadillo5x0_gpio.0/ports/gpio0_data  
0
```

図 9-10 GPIO : INPUT 時の入力状態を取得する

9.3.2. Direction を OUTPUT にする

GPIO0 (CON16 2 ピン) を出力にする場合は、図 9-11 のようにコマンドを実行します。

```
[armadillo500 ~]# echo 1 > ¥  
/sys/devices/platform/armadillo5x0_gpio.0/ports/gpio0_dir
```

図 9-11 Direction を OUTPUT にする

HIGHレベルを出力するには、図 9-12 のようにコマンドを実行します。

```
[armadillo500 ~]# echo 1 > ¥  
/sys/devices/platform/armadillo5x0_gpio.0/ports/gpio0_data
```

図 9-12 OUTPUT 時の出力状態を変更する

改訂履歴

| Version | 年月日 | 改訂内容 |
|---------|-----------|--------------------------|
| 1.0.0 | 2007.7.27 | ・初版発行 |
| 1.0.1 | 2007.9.14 | ・「1.6.1 保障範囲」の製品の保証方法を修正 |

困ったときは...

Armadillo 開発者サイトでは、本書に記載されていない情報や最新の技術情報や FAQ などが随時更新されています。困ったときは、まず Armadillo 開発者サイトにアクセスしてみてください！

Armadillo開発者サイト

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

開発に関する質問は...

こんなことがしたいけど、実績が知りたい。などの技術的な質問がある方は、Armadillo メールングリストを利用したらどうでしょうか？Armadillo メールングリストには、多くの方々が購読されています。もしかしたら会員メンバーの中に、解決までの道標を提供してくれるかもしれません。

Armadilloメールングリスト

<http://armadillo.atmark-techno.com/maillinglists>

Armadillo-500

Development Board Startup Guide

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 6F

TEL:011-207-6550 FAX:011-207-6570
