

FutureNet MA-E120 Series

ユーザーズガイド

Version 1.0.1



はじめに

このたびは本装置をご購入いただきまして、誠にありがとうございます。

本書には、本装置を安全に使用していただくための重要な情報が記載されています。ご使用前に本書をよくお読みになり、正しくお使いいただけますようお願い致します。

特に、本書に記載されている「安全にお使いいただくために」をよく読み、理解されたうえで本装置をご使用ください。

また、本書は本装置の使用し中、いつでも参照できるように大切に保管してください。

■ご注意

本書の内容の一部または全部を無断で転用、転載しないようお願いいたします。

(2) 本書の内容および製品仕様、外観は、改良のため予告なく変更することがあります。

(3) 本装置の仕様は日本国内向けとなっておりますので、海外ではご利用できません。

This equipment is designed for use in Japan only and cannot be used in any other country.

(4) 本書の作成にあたっては万全を期しておりますが、本書の内容の誤りや省略に対して、また本書の適用の結果生じた間接損害を含め、いかなる損害についても責任を負いかねますのでご了承ください。

(5) 製品の保証に関する規定については製品添付の製品保証書をご覧ください。

(6) 本製品にて提供されるファームウェアおよび本製品用として弊社より提供される更新用ファームウェアを、本製品に組み込んで使用する以外の方法で使用することは一切許可しておりません。

■セキュリティの確保について

パスワードを設定しない、もしくはデフォルト・パスワードを使用する場合、ネットワーク上のだれからでも本装置の設定をおこなうことができます。

セキュリティの面からは非常に危険なため、ユニークなパスワードを設定することを強く推奨します。

■最新情報の入手について

当社では、製品に関する最新の情報(最新のファームウェア、マニュアルなど)を下記ホームページでご案内しています。

ぜひご利用下さい。

センチュリー・システムズ(株)
FutureNet サポートデスク
<http://www.centurysys.co.jp/support/>

また、本書について万一ご不審な点や誤り、記載漏れなど、お気づきの点がございましたら、下記までご連絡ください。

センチュリー・システムズ(株)
FutureNet サポートデスク
support@centurysys.co.jp

■商標について

「FutureNet」はセンチュリー・システムズ株式会社の登録商標です。
その他の商品名、会社名は、各社の商標または登録商標です。

目次

■ ご注意	2
■ セキュリティの確保について	2
■ 最新情報の入手について	3
■ 商標について	3
1. FutureNet MA-E120 の概要.....	7
1.1. FUTURENET MA-E120 の特徴	8
1.2. FUTURENET MA-E120 応用例	10
1.3. 外観	11
1.4. CONFIG の説明	14
1.5. LED の説明	15
2. MA-E120 の基本的な操作.....	17
2.1. ソフトウェアの概要	17
2.2. システムへのログイン.....	17
2.3. 起動と停止の方法.....	18
2.3.1. 起動の方法	18
2.3.2. ネットワークで接続する.....	18
2.3.3. コンソールから接続する.....	18
2.3.4. JFFS2 ルート起動	19
2.3.5. NFS ルート起動.....	19
2.3.6. コマンドによる起動方法の変更	19
2.3.7. 停止方法.....	19
2.3.8. 初期化	20
2.4. 基本的な使い方	20
2.4.1. ssh ログイン	20
2.4.2. シェル環境	20
2.4.3. ウェブユーザインターフェース.....	20
2.5. ソフトウェアのバージョンと更新.....	21
2.5.1. ソフトウェアのバージョン	21
2.5.2. ファームウェアのアップデート	21
3. ネットワーク機能の設定.....	25
3.1. ネットワーク機能の概要	25
3.2. IP アドレスの設定	25
3.3. ルーティングの設定.....	26
3.3.1. デフォルトゲートウェイ	26

3.3.2.	スタティックルーティング	26
3.3.3.	IP フォワーディング	26
3.4.	IP マスカレードの設定	27
3.5.	パケットフィルタの設定	27
4.	インターフェースの仕様.....	28
4.1.	インターフェース概要.....	28
4.2.	USB HOST	28
4.2.1.	USB serial converter	28
4.2.2.	USB Flash Memory(USB Mass Storage Class 対応).....	28
4.3.	LED	29
4.3.1.	/dev/led インターフェース.....	29
4.4.	INIT ボタン	30
5.	NFS ルートを利用したセルフ開発環境	31
5.1.	NFS ルートの概要	31
5.2.	NFS ルートの設定	31
5.3.	アプリケーションのビルド方法.....	33
6.	VMware を利用したクロス開発環境.....	34
6.1.	VMWARE 環境の概要.....	34
6.2.	クロス開発環境の開発フロー.....	34
6.3.	クロス開発環境の構成	35
6.4.	VMWARE 仮想マシンイメージのインストール	36
6.5.	NFS ルートファイルシステムの展開	37
6.6.	VMWARE 仮想マシンイメージの LINUX 設定	37
6.7.	カーネルイメージのビルド	40
6.8.	ルートファイルシステムイメージのビルド	41
6.9.	ファームウェアイメージのビルド	45
7.	SD ブートシステムの構築	48
7.1.	SD ブートシステムの概要.....	48
7.2.	SD カードの初期化	48
7.3.	ルートファイルシステムの構築.....	51
7.4.	SD カードからの起動	52
8.	仕様一覧.....	53
8.1.	MA-E120 の仕様	53
8.2.	オープン・ソース・ソフトウェアのライセンスについて	55

1. FutureNet MA-E120 の概要

FutureNet MA-E120 は米 Freescale 社の高性能ネットワーキングプロセッサ「**MPC8313E**」を搭載した Linux プラットフォーム製品です。2つのギガビットイーサネットインタフェースとモバイル通信用インタフェースを備え、ルータ機能をベースとした特定サービス専用ルータ、アプリケーション・ゲートウェイ、ネットワーク監視ボックス、遠隔監視システムの拠点側サーバなどとして利用できます。

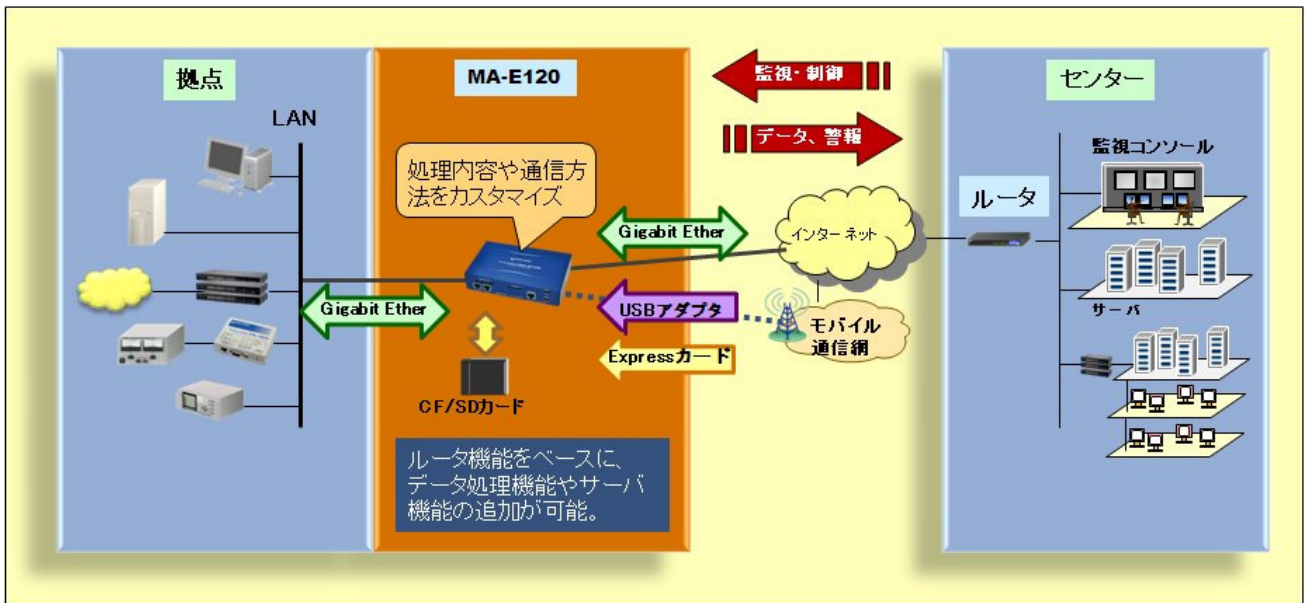
FutureNet MA-E120 は、モバイル通信用インタフェースとして USB ポートと ExpressCard スロットを備えます。また、アプリケーションやデータの保存用として SD カード、CF カードを利用できます。

OS には Linux カーネル 2.6 をプリインストールしており、オープンソースで提供されている Linux の各種アプリケーションが利用可能です。また、標準添付の開発環境 (SDK) を使って Linux アプリケーションの移植や新規開発、ファームウェアのカスタマイズ等が柔軟におこなえます。

1.1. FutureNet MA-E120 の特徴

■ 強力なインターフェース

FutureNet MA-E120 は2つのギガビットイーサネットインターフェース(RJ-45)を搭載しています。また、暗号化処理専用のハードウェアを搭載しています(*)。USB 2.0は2ポート搭載しており、USBメモリやデータ通信端末などを接続できます。USBファイルシステム(USBメモリ)からの起動も可能です。ExpressCardスロットにはデータ通信用ExpressCard(USB接続タイプ)が接続できます。また、プログラムやデータの保存用、もしくはシステム起動用として、SDメモリーカード(外部スロット)、CFメモリーカード(筐体内に内蔵)が利用できます。



【FutureNet MA-E120 の利用イメージ】

*暗号化処理ハードウェア用ライブラリは今後のバージョンアップで提供予定です。

■ 強力なネットワーク機能

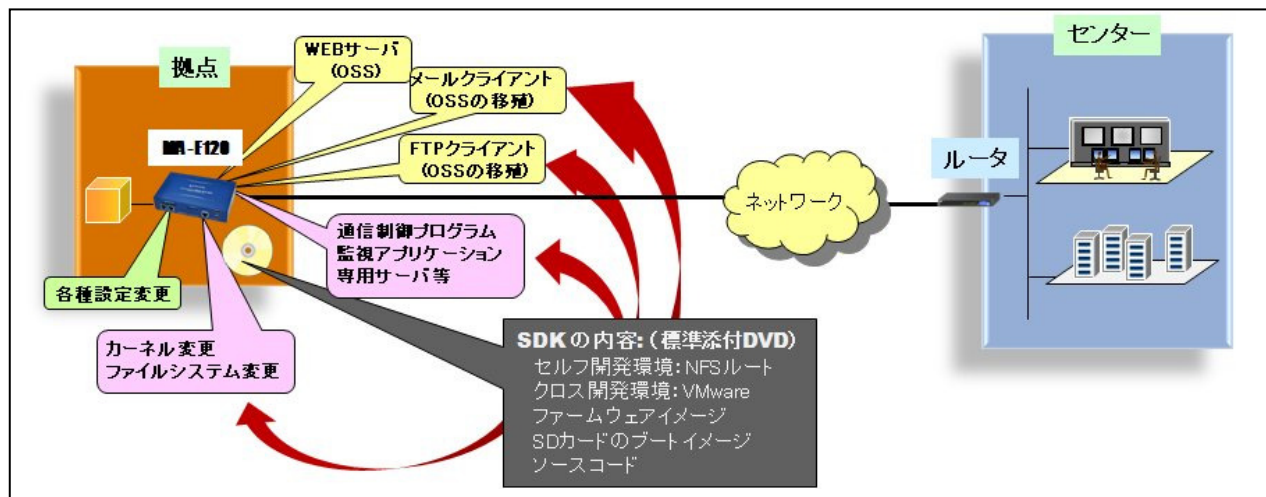
FutureNet MA-E120 はギガビットイーサネットによる有線のネットワーク接続に加え、モバイルデータ通信端末接続インターフェースを備えており NTT ドコモ、KDDI、イーモバイル、ソフトバンク、日本通信(予定)、等から提供される USB 型や ExpressCard 型のモバイルデータ通信端末を接続できます。NAT/NAPT、パケットフィルタ、SSHによる暗号化など Linux の優れたネットワーク機能を自由に利用できるため、サーバアプリケーションと組み合わせると、特定用途向けの専用サーバ兼リモートルータを短期間で実現できます。

■ 低消費電力、高性能、高信頼性、耐環境性

FutureNet MA-E120 は省電力 CPU や電源回路の最適化により、最大約 5.5W という低消費電力を実現しています。ヒートシンクも必要とせず、ファンレスで動作すると共に高信頼性を確保し、24 時間 365 日の常時稼働と 0°C ~ 50°C (AC アダプタを除く)での動作保証を実現しています。

■ 開発の容易さ、開発工数の短縮

FutureNet MA-E120にはLinux 2.6をプリインストールしており、初期状態でLinuxを起動し各種アプリケーションを実行できます。また、セルフコンパイルに必要なソフトウェアをネットワーク上の他のLinux機から提供するためのNFSルート開発環境と、Windows機でLinuxのクロス開発環境を作るためのVMware用OSイメージをSDKで提供します。これらの開発環境を使って、本装置用のアプリケーションの開発や、起動用USB/SDメモ리카ードの作成、独自のファームウェアの作成等が可能です(*)。



【FutureNet MA-E120 SDKによるソフトウェアのカスタマイズ】

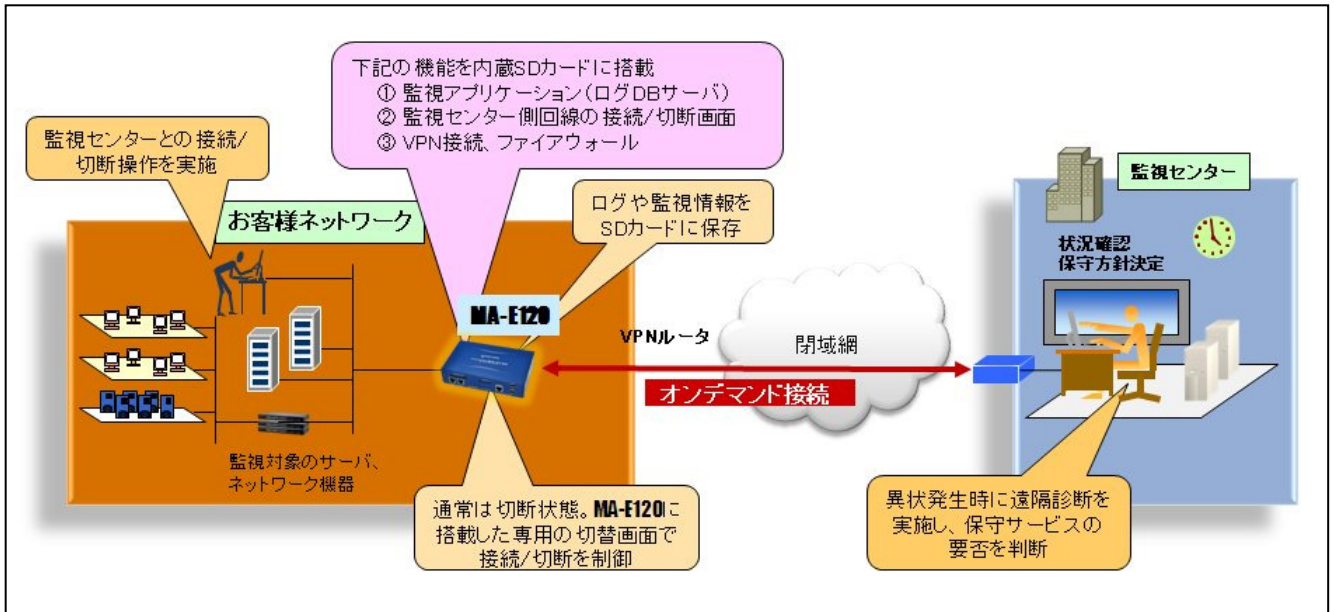
なお、センチュリー・システムズでは **FutureNet MA-E120** 上でのアプリケーション開発、ソフトウェアの移植、ドライバソフトの開発、インタフェース基板の開発、および OEM に向けた製品化なども承ります。

* 本製品で利用できるすべてのソフトウェアがクロス開発環境でビルドできることを保証するものではありません。

1.2. FutureNet MA-E120 応用例

■ ネットワーク監視ボックス

FutureNet MA-E120 を使って IP ネットワーク経由でリモート拠点のネットワークを監視、制御できます。



【FutureNet MA-E120 による遠隔監視システム】

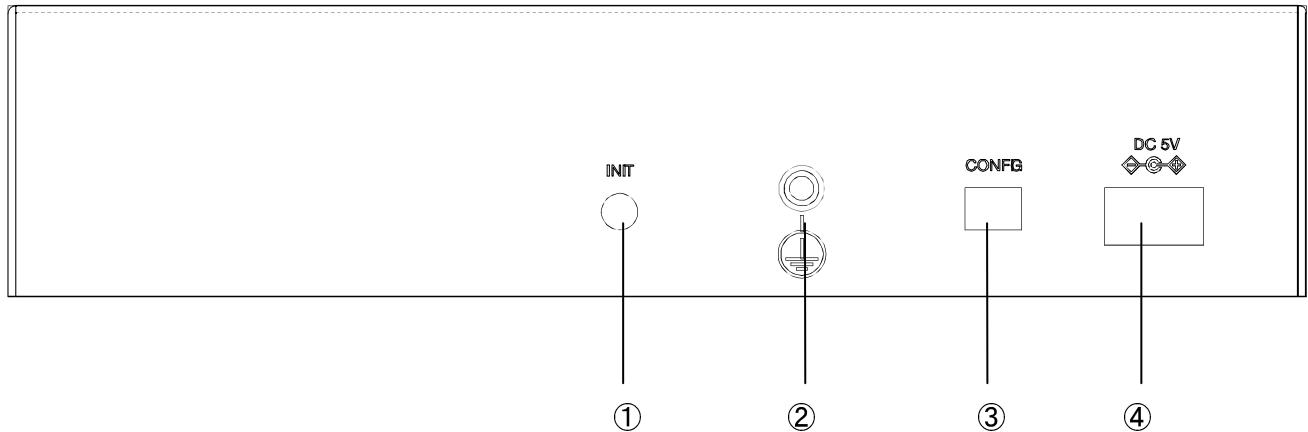
FutureNet MA-E120 では単にデータを送受信するだけでなく、受信したデータの変換やデータ処理ロジックの実装が可能です。さらに外部メモリを利用してデータを蓄積した上での処理も可能なため、センター側サーバと本装置とで柔軟に通信量や機能範囲を調整できます。

* 本製品で利用できるすべてのソフトウェアがクロス開発環境でビルドできることを保証するものではありません。

1.3. 外観

本装置本体各部の名称は以下のとおりです。

<背面図>



① INIT スイッチ

システム・シャットダウン用のスイッチです。停止時または設定データの初期化を実行する際に使用します。

② FG(アース) 端子

保安用接続端子です。必ずアース線を接続してください。

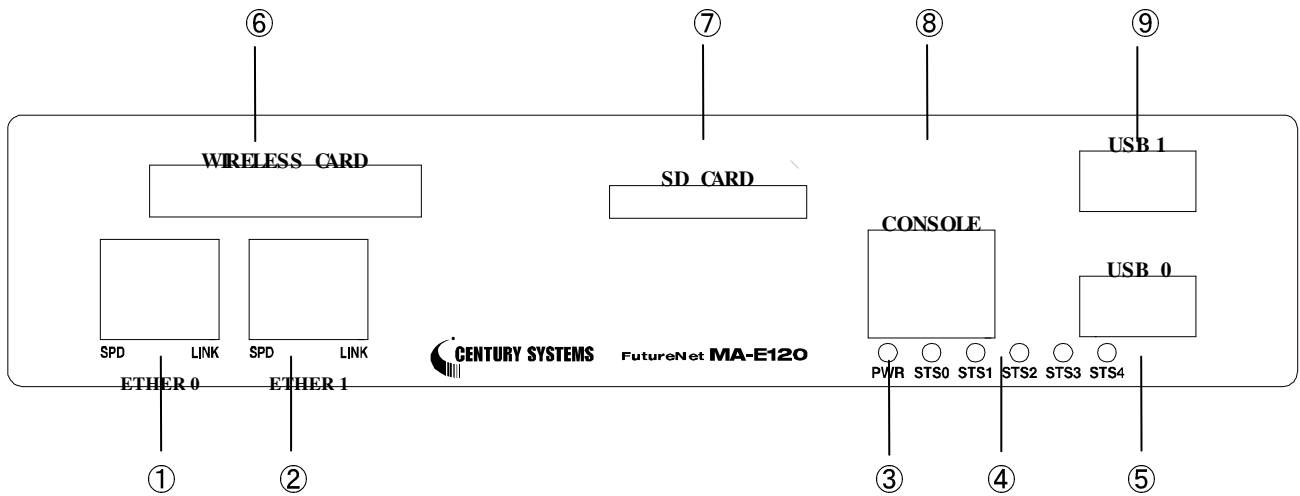
③ CONFIG

本装置の動作モードを指定するディップ・スイッチです。ファームウェアの更新や起動モードの切り替えに使用します。設定方法は 1.44 節をご参照下さい。

④ DC 5V 電源コネクタ

製品付属の AC アダプタを接続します。

<正面図>



①・② Ethernet ポート

10BASE-T/100BASE-TX 対応で、Ether0, Ether1 の2ポートが使用可能です。Auto-MDI/MDIXにも対応しています。

LED は各 Ethernet ポートの状態を表示します。

- ・Link/Active: LAN ケーブルが正常接続時に緑色に点灯し、フレーム送受信時に点滅します。
- ・Speed: 10Base-T で接続時は消灯、100BASE-TX でリンクした場合に黄色に点灯します。

③ POWER LED

本装置の電源状態を表示します。

④ STATUS LED

本装置の動作状態等を表示します。設定方法は 1.55 節をご参照ください。

⑤・⑨ USB

USB2.0(ホスト)対応の USB0,USB1 の 2 ポートが使用可能です

⑥ WIRELESS CARD

ExpressCard スロットです。対応するカードが使用可能です。

⑦ SD CARD

SD/SDHC のメモ리카ードが使用可能です。著作権保護機能には対応していません。

⑧ CONSOLE

コンソール・アダプタを使用することにより、PCなどのターミナル上からLinuxのコンソールとして使用できます。

1.4. CONFIG の説明

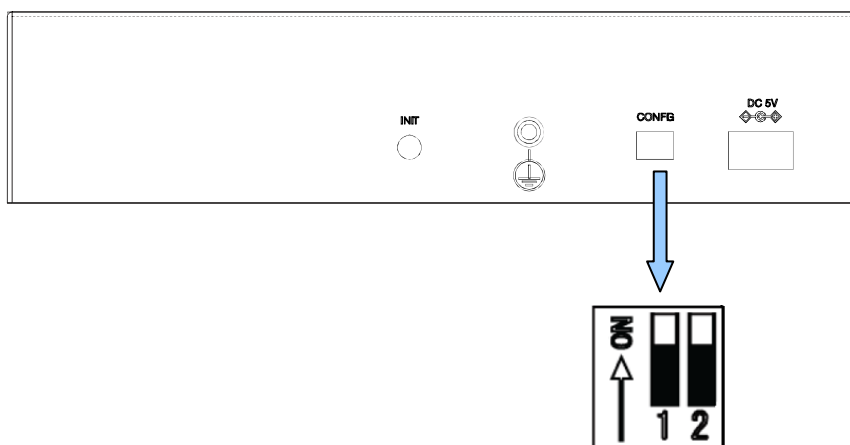
本装置の CONFIG、INIT の操作により、ファームウェアの更新や起動モードを変更することができます。

CONFIG はスイッチを上方向にすると ON, 下方向にすると OFF になります。

また CONFIG スイッチは向かって左から順に 1,2 の番号が割り当てられています。

INIT スイッチを ON にする場合、押しながら起動してください。

<背面図>



CONFIG、INIT によって切り替わる動作モードは、以下のとおりです。

ディップ・スイッチ		INIT スイッチ	動作モード
1	2		
ON	ON	D. C.	TFTP download (ファームウェア更新)
ON	OFF	D. C.	u-boot
OFF	ON	OFF	Linux boot (bootp & NFS) 'root=/dev/nfs ip=dhcp'
OFF	ON	ON	Linux boot (USB, sda1) 'root=/dev/sda1
OFF	OFF	D. C.	Linux boot (JFFS2) 'root=/dev/mtdblock4' (デフォルト)

(注) u-boot 環境変数により起動オプションを変更した場合は、上記ディップ・スイッチは反映されません。

1.5. LED の説明

1.5.1.1. MA-E120 状態表示 LED

本装置では、**MA-E120** の状態を表示する STATUS 0 1 2 3 4 の 5 つの LED があります。STS0、1 は 2 色 (red, green)、STS2~4 は 1 色 (green) になります。LED の表示によって動作状態の確認を行うことができます。各 LED はユーザアプリケーションによってコントロールすることもできます。ここでは説明のため、各 LED を次のように表現します。

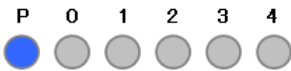


1.5.1.2. 起動時の LED パターン

本装置では、起動時の状態を LED で確認することができます。

・ 電源投入

P: 点灯 (青)



・ Linux 起動中

P: 点灯 (青)、0: 点灯 (赤)、1: 点灯 (緑)



・ Linux 起動完了

P: 点灯 (青)、0: 点灯 (緑)、1: 点灯 (緑)

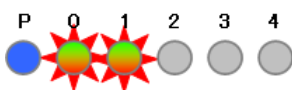


1.5.1.3. その他の LED パターン

本装置では、起動時以外にも LED で動作状態を確認することができます。

・ WebUI からファームウェア・アップグレード実行時

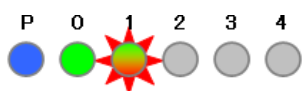
P: 点灯 (青)、0: 点滅 (赤・緑)、1: 点滅 (赤・緑)



ファームアップグレード完了後、0、1 が消灯し再起動します。

稼働中に Init ボタンを2秒押し続け shutdown を実行した時

P: 点灯(青)、0: 点灯(緑)、1: 点滅(赤・緑)

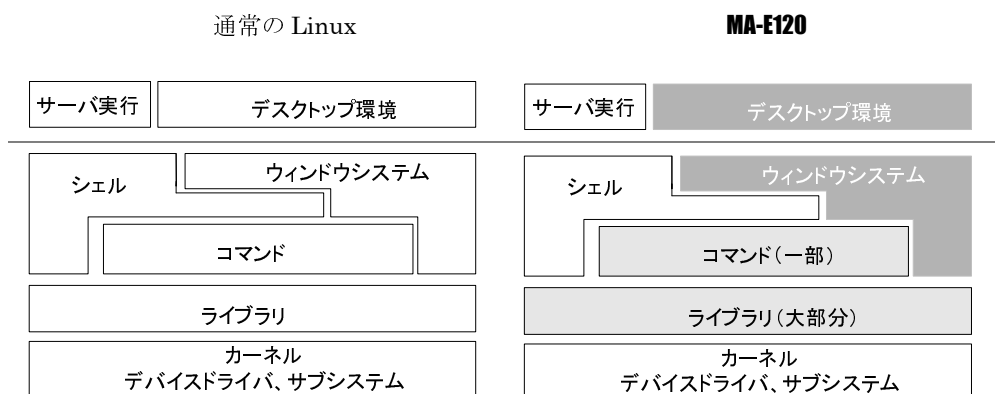


shutdown 完了後、0、1 が消灯します。DC コネクタを抜くことができます。

2. MA-E120 の基本的な操作

2.1. ソフトウェアの概要

通常の Linux ディストリビューションがコマンドからウィンドウシステム、ネットワーク機能などを含むシステムのオペレーション環境全体を提供するのに対し、本装置のソフトウェアは、ネットワークデバイスとして、あるいはサーバ機能を提供するベースとして最低限必要なプログラムだけを収めた Linux のサブセットとなっています。



本装置には、ウィンドウシステムのためのソフトウェアは含まれません。シェルとしては bash が利用できます。メインボードのみの構成の場合、コマンドはシステムの起動に必要なものや基本的なファイル操作のためのコマンドだけを提供します。エディタは nano が利用できます。

ライブラリについてはウィンドウシステム関連を除くほとんどのライブラリを提供しています。glibc は 2.10(NPTL)をサポートしています。Kernel は 2.6.33 を使用しています。

2.2. システムへのログイン

システムへのログインはイーサネットポートを経由して行います。

本装置にはデフォルトで次のような IP アドレスが設定されています。

Ether0 (10/100Base-TX)	192.168.253.253
Ether1 (10/100Base-TX)	192.168.254.254

なおログインには以下の 1 つのアカウントを使用します。

ログイン: アカウント・パスワード共に user1
su パスワード: root

イーサネットポートからシステムへログインする場合、マシンの IP アドレスを本装置に合わせて、本装置のデフォルトの IP アドレスに ssh で接続して下さい。

尚、コンソール・アダプタを使用することで CONSOLE からログインする事ができます。詳しくは 2.3.3 節をご参照ください。

2.3. 起動と停止の方法

2.3.1. 起動の方法

本装置には電源スイッチがありません。DCコネクタの抜き差しで電源の ON/OFFを行います。電源を入れれば自動的に Linux が起動します。

各種機能の設定はネットワーク、またはコンソール経由で本体にログインした後で行います。本体にログインする場合は次の手順に従って下さい。

2.3.2. ネットワークで接続する

ステップ 1 ツイストペア・ケーブルの接続

本装置の ETHER(10/100Base-TX)と LAN を接続します。

ステップ 2 電源オン

付属する AC アダプタを 100V 電源に接続し、続いて本体に AC アダプタを接続して下さい。

ステップ 3 ログイン

ネットワーク上のマシンから ssh コマンドを使用してネットワーク経由でログインします。マシンの IP アドレスのネットワーク部を本装置に合わせて(192.168.253.1 など)、本装置のデフォルトの IP アドレス (Ether0=192.168.253.253)に ssh で接続してください。工場出荷時はユーザ名、パスワード共に user1 でログインできます。

2.3.3. コンソールから接続する

本装置の CONSOLE にコンソール・アダプタを接続させることで、コンソールからログインできます。

ステップ 1 RS-232 ケーブルの接続

RS-232 のクロスケーブルで、本装置に接続させたコンソール・アダプタとターミナルエミュレータソフトを動かすマシンの RS-232 ポートを接続します。

ステップ 2 通信条件の設定

ログインするマシン側ではターミナルエミュレータを起動し、RS-232 の通信条件を以下に合わせて下さい。

データ長	8bit
パリティ	なし
ボーレート	115200bps
ストップビット	1
フロー制御	None

ステップ 3 電源オン

付属する AC アダプタを 100V 電源に接続し、続いて本体に AC アダプタを接続して下さい。

ステップ 4 ログイン

接続が正しければ起動した段階で login: のプロンプトが表示されます。デフォルトではユーザ名、パスワード共に root でログインできます。

2.3.4. JFFS2 ルート起動

FLASH メモリから Linux が起動できる状態になっている場合は電源投入後、自動的に Linux が起動します。

2.3.5. NFS ルート起動

Linux がインストールされている PC に **MA-E120** の NFS 用のファイルシステムを展開することで、本装置の DHCP 上で root-path を渡して NFS ルートで起動することができます。

NFS ルート設定については、「5 NFS ルートを利用」に示します。

2.3.6. コマンドによる起動方法の変更

本装置では、u-boot のコマンドラインで起動方法を変更することができます。起動変更時のコマンドの使用方法については、u-boot の各種ドキュメントを参考にして下さい。

2.3.7. 停止方法

DC コネクタを抜くと電源が切れますが、その前に必ずシャットダウンの手続きをおこなって下さい。シャットダウンの方法には以下の方法があります。

shutdown コマンドを実行する

```
MA-E120 # shutdown -h now
```

POWER のみ点灯したら、電源を切ることができます。

Init ボタンを押す

Init ボタンには2つの役割があり、停止時または設定データの初期化を実行する際に使用します。(初期化に関しては 2.3.8 節を参照願います。)

本装置では通常稼働中に Init ボタンを押すことにより shutdown することができます。

Init ボタンを押した後、POWER LED のみ点灯したら、電源を切ることができます。

2.3.8. 初期化

本装置では Init ボタンと電源投入の組み合わせで設定データの初期化を行うことができます。

電源を投入するまえに Init ボタンを押下したまま、電源を入れしばらくすると設定データのクリアを受け付けます。/etc 下の内容(WebUI の設定データ、パスワード等)は製品出荷時の状態に戻りますが、/opt 等の jffs2 ファイルシステムの状態は保持されています。完全に初期化を行いたい場合は、ファームウェアの更新を行うことで初期化可能です。

2.4. 基本的な使い方

2.4.1. ssh ログイン

ネットワークに接続し ssh でログインして下さい。前述にもある通り、工場出荷時はユーザ名、パスワード共に user1 でログインできます。ログインしたら安全のためすぐにパスワードを変更して下さい。

2.4.2. シェル環境

シェルは IEEE POSIX Shell and Tools 仕様に準拠した Bash です。

/bin/sh は/bin/bash へのシンボリックリンクです。

コマンドは主に以下のディレクトリに格納されています。

```
/bin
/sbin
/usr/bin
/usr/sbin
/usr/local/sbin
/usr/local/bin
```

2.4.3. ウェブユーザインターフェース

概要

本装置の各種システム設定を、ウェブユーザインターフェースから行うことができます。設定できる項目は次のとおりです。

- PPP 発信・着信設定
- 日付・時刻・NTP の設定
- ルータの設定
 - ・ ネットワークインターフェース(Ethernet)の設定
 - ・ GRE トンネル
 - ・ 静的ルーティング
 - ・ ゲートウェイの設定
 - ・ DNS 設定
 - ・ DHCP サーバ設定

- ・ IP フィルタリング
- ・ NATP、ファイアウォール

- 死活監視
- 運用管理設定
 - ・ 設定ファイル入出力
 - ・ ファームウェア更新
 - ・ パスワードの変更

ウェブユーザインターフェースは、FLASH メモリ(JFFS2 ルートファイルシステム)にインストールされています。ウェブユーザインターフェースの操作仕様は、「FutureNet MA-E120 Series ウェブユーザインターフェース操作マニュアル」に示します。

2.5. ソフトウェアのバージョンと更新

2.5.1. ソフトウェアのバージョン

本装置では、基板上のフラッシュ ROM に格納されている Linux kernel、ファームウェアバージョンを以下の方法で確認することができます。

Linux kernel のバージョン

ファイル /proc/version に記述されています。以下のコマンドで確認できます。

例:

```
MA-E120 ~ # cat /proc/version
Linux version 2.6.33.7 (root@vmware) (gcc version 4.3.4 (Gentoo 4.3.4 p1.1, pie-10.1.5) ) #5 Fri Oct 1 13:16:36
JST 2010
```

ファームウェアのバージョン

ファイル /etc/version に記述されています。以下のコマンドで確認できます。

例:

```
MA-E120 ~ # cat /etc/version
MA-E120 firmware version 1.0.0 build 2 (Tue, 05 Oct 2010 14:58:14 +0900)
```

2.5.2. ファームウェアのアップデート

FLASH メモリに格納されているファームウェアのアップデート方法には以下の 3 通りの方法があります。

- Windows95/98/NT/2000/XP/Vista の PC で付属のソフトを使用する方法

- Linux で bootp,tftp サーバを使用する方法
- ウェブユーザインターフェースを使用する方法

2.5.2.1. Windows95/98/NT/2000/XP/Vista の PC で付属のソフトを使用する方法

ファームウェアのアップデートには、dhcp サーバと tftp サーバソフトが必要になります。

- **MA-E120** ファームウェアイメージファイル

dhcp サーバと tftp サーバソフトに関しましては、以下のソフトで動作の確認がとれています。WEB サイト等からダウンロードして使用して下さい。

<http://tftpd32.jounin.net/>

ファームウェアのアップデート手順は次の通りです。

本装置の CONFIG スイッチ 1、2 を両方 ON に設定します。

ダウンロード用 Windows 機上で dhcp サーバおよび tftp サーバソフトを起動します。操作の詳細は、ソフトウェアに添付の説明書を参照してください。

dhcp サーバおよび tftp サーバソフトの設定が終了したら、電源ケーブルを本装置に接続して電源を入れます。自動的にファームウェアのアップデートが始まります。LED が全て消灯したらアップデート完了です。電源ケーブルを抜いて下さい。

ダウンロード完了後、先程変更したディップ・スイッチを元に戻し(1、2 を両方 OFF)電源を ON にします。

2.5.2.2. Linux で dhcp サーバ,tftp サーバを使用する方法

dhcp サーバ,tftp サーバがインストールされた Linux マシンを利用するファームウェア更新について示します。これ以降は dhcp サーバ,tftp サーバがインストールされていることが前提となります。

ファームウェアをアップデートする前に

dhcp サーバ,tftp サーバを設定する前に Linux PC 上にファームウェアイメージファイルを用意します。ここでは、/tftpboot に mae210_firm-vXX.img というファイルネームで保存します。

DHCP サーバの設定

設定ファイル/etc/dhcpd.conf を編集します。

例:

```
subnet 192.168.252.0 netmask 255.255.255.0{
    group {
        host mae120 {
            hardware ethernet 00:80:6d:XX:XX:XX;      # MA-E120 Mac address
```

```

fixed-address 192.168.252.252;      # MA-E120 IPAddress
next-server 192.168.252.1;        # Server IP
filename "mae210_firm-vXX.img";    # firmware file name
option host-name "mae2xx";        # 割り当てるホスト名
}
}
}

```

- ・ 接続しているネットワークは 192.168.252.0/24
- ・ サブネットマスクは 255.255.255.0
- ・ クライアントの MAC アドレスは、00:80:6d:XX:XX:XX
- ・ クライアントの IP アドレスは 192.168.252.252
- ・ DHCP サーバの IP アドレスは 192.168.252.1
- ・ ファームウェアの名前は mae210_firm-vXX.img
- ・ クライアント名は mae2xx

設定終了後は DHCP サーバを起動します。

例:

```
# etc/init.d/dhcpd start
```

inetd からの起動

tftpd サーバはコマンドラインからも起動できますが、ここでは inetd から起動するように設定します。

/etc/inetd.conf ファイルを編集する前に、/etc/services を確認します。以下の行がコメントアウト(#)されている場合は、行頭の「#」を削除して下さい。

例:

```
tftp      69/tcp      #TFTP  server
tftp      69/udp
```

/etc/inetd.conf ファイルで tftp,bootps の行がコメントアウト(#)されている場合、行頭の#を削除して下さい。

例:

```
tftp dgram udp wait nobody /usr/sbin/tcpd in.tftpd /tftpboot
```

変更した設定を反映させます。

例:

```
# /etc/init.d/inetd reload
```

ファームウェアのアップデート

本装置の CONFIG スイッチ 1、2 を両方 ON に設定します。

電源ケーブルを本装置に接続して電源を入れます。

自動的にファームウェアのアップデートが始まります。LED が全て消灯したらアップデート完了です。電源ケーブルを抜いて下さい。

ダウンロード完了後、先程変更したディップ・スイッチを元に戻し、電源を ON にします。

2.5.2.3. ウェブユーザインターフェースを使用する方法

ウェブユーザインターフェースを使用して、ファームウェアを更新できます。この場合、外部に TFTP サーバ等を立てたり、ディップ・スイッチの変更の必要はありません。操作方法は、「FutureNet MA-E120 ウェブユーザインターフェース操作マニュアル」を参照してください。

3. ネットワーク機能の設定

3.1. ネットワーク機能の概要

ネットワーク設定は、ウェブユーザインターフェースから行うことができます。ウェブユーザインターフェースは、ネットワーク設定情報を独自のシステム設定ファイルに保存し、Gentoo Linux 形式のネットワーク設定 (/etc/conf.d 下の設定ファイル)にテンプレートを使用し書き出します。

エディタにより GentooLinux 形式のネットワーク設定を行う場合、設定した内容はウェブユーザインターフェースには反映されませんのでご注意ください。

以下、GentooLinux 形式の設定ファイルによるネットワーク設定について記述します。

設定項目	設定方法	関連する主なファイル
	コマンド	
ネットワークデバイスの設定	ifconfig	/etc/conf.d/net
IP アドレスの設定	ifconfig	/etc/conf.d/net
ネットマスクの設定	ifconfig	/etc/conf.d/net
静的ルーティングの設定	route	/etc/conf.d/net
デフォルトゲートウェイの IP	route	/etc/conf.d/net
パケットフィルタ	iptables	/etc/shorewall/*
IP フォワーディング	iptables	/etc/shorewall/*
IP マスカレード	iptables	/etc/shorewall/masq

詳細については各コマンドやファイルの man ページなどを参照して下さい。

なお、本装置には man ページを付属していませんので、インターネット上のドキュメントを参照して下さい。

3.2. IP アドレスの設定

本装置にはデフォルトで次のような IP アドレスが設定されています。

```
Ether0 : 192.168.253.253
```

```
Ether0 : 192.168.254.254
```

このアドレスはそのまま使うこともできますし、ネットワーク環境に合わせて変更することもできます。IP アドレスを変更する場合は ifconfig コマンドで行えます。

設定例:

```
MA-E120 # ifconfig eth0 192.168.0.1 netmask 255.255.255.0
```

次回以降も有効にしたい場合は、/etc/conf.d/net ファイルに設定します。

/etc/conf.d/net ファイルの設定例:

```
modules_eth0="iproute2"  
config_eth0="192.168.0.1/24"
```

3.3. ルーティングの設定

3.3.1. デフォルトゲートウェイ

デフォルトゲートウェイは、route コマンドを使って設定できます。route コマンドは次のように使用します。
(192.168.253.254 がルータの場合)

```
MA-E120 # route add default gw 192.168.253.254
```

route コマンドを使って設定した内容は、再起動後は反映されませんのでご注意ください。再起動後も有効にしたい場合は、/etc/conf.d/net ファイルに設定します。

/etc/conf.d/net ファイルの設定例:

```
routes_eth0="default via 192.168.253.254"
```

3.3.2. スタティックルーティング

スタティックルーティングは route コマンドで設定できます。/etc/conf.d/net ファイルに記述することで起動時に自動的に設定されます。

192.168.200.0 のネットワークへの静的経路を追加する例

```
MA-E120 # route add -net 192.168.200.0/24 gw 192.168.1.10
```

このコマンドは 192.168.200.0 のネットワークに対するルータとして 192.168.1.10 のゲートウェイを指定します。

/etc/conf.d/net ファイルの設定例:

```
routes_eth0="192.168.200.0/24 via 192.168.1.10"
```

3.3.3. IP フォワーディング

工場出荷時はネットワークインターフェース間の IP パケットのフォワーディングが有効になっています。IP パケットのフォワーディングを無効にするには、次のコマンドを使用します。

```
MA-E120 # echo 0 > /proc/sys/net/ipv4/ip_forward
```

再起動後も IP フォワーディングを無効にしたい場合は、/etc/sysctl.conf の下記エントリを編集してください。0 はフォワーディング無効、1 は有効です。

```
net.ipv4.ip_forward = 0
```

3.4. IP マスカレードの設定

本装置では iptables コマンドを使って IP マスカレードやパケットフィルタの設定を行うことができます。

- ・ppp0 から出力されるパケットをマスカレードする
(送信元 IP が 192.168.253.0/24 であるパケットをマスカレードする)

```
MA-E120 # iptables -t nat -A POSTROUTING -o ppp0 -s 192.168.253.0/24 -j MASQUERADE
```

- ・設定内容の表示

```
MA-E120 # iptables -t nat -L
```

- ・IP マスカレードを解除する

```
MA-E120 # iptables -t nat -D POSTROUTING -o ppp0 -s 192.168.253.0/24 -j MASQUERADE
```

3.5. パケットフィルタの設定

パケットフィルタは iptables コマンドを使用することで設定できます。

主な使用例を示します。

- ・FORWARD チェインの基本ポリシーを拒否

```
MA-E120 # iptables -P FORWARD DROP
```

- ・192.168.253.253 への ICMP でのアクセスを拒否する

```
MA-E120 # iptables -A INPUT -d 192.168.253.253 -p icmp -j DROP
```

- ・設定されているフィルタルールを表示する

```
MA-E120 # iptables -L
```

- ・INPUT チェインの設定ルールのみ全て削除する

```
MA-E120 # iptables -F INPUT
```

4. インターフェースの仕様

4.1. インターフェース概要

本装置では、以下の I/O および通信インターフェースを持ちます。

- 10BASE-T/100BASE-TX × 2 ポート
- USB 2.0 × 2 ポート
- ExpressCard スロット × 1
- SD カードスロット × 1
- CF カードスロット × 1 (筐体内蔵)

4.2. USB Host

USB Host は USB2.0 対応の USB0、USB1 の 2 ポートが使用可能です。

4.2.1. USB serial converter

MA-E120 では、PL2303 チップを使用した USB シリアル変換器に対応するドライバ「pl2303」がカーネルに組み込まれています。

4.2.2. USB Flash Memory(USB Mass Storage Class 対応)

MA-E120 では、SCSI のストレージ (/dev/sdx) として認識し、使用可能になります。

動作確認がされている USB フラッシュメモリは以下のとおりです。

CFD 販売	CUFD-H4G
A-DATA	Nobility Series PD7 200x (Turbo Speed) Flash Drive 4GB
	Classic Series PD9 Flash Drive 2GB
	Reader Series microSD Trio(USB+SD interface)

※ 完全な動作保証ではありません

ディップ・スイッチの設定により、USB デバイスをルートとして起動することも可能です。

本装置では、USB フラッシュメモリは自動マウントされませんので、下記のコマンド例をご参考に mount および umount を実行して下さい。また、USB フラッシュメモリからのブート時は途中で取り外さないで下さい。

(1) マウント

```
MA-E120 ~ # cat /proc/partitions (パーティションの確認)
```

```
major minor #blocks name
```

```
7 0 16244 loop0
```

```

31      0      256 mtdblock0
31      1      128 mtdblock1
31      2      128 mtdblock2
31      3     3584 mtdblock3
31      4     28672 mtdblock4
 8      0     509695 sda
 8      1     509690 sda1

```

```
MA-E120 ~ # mount /dev/sda1 /mnt/usb/ (USB デバイスのマウント)
```

(2) アンマウント

```
MA-E120 ~ # cat /proc/mounts | egrep sd[a-z] (マウント済みデバイスの確認)
```

```
/dev/sda1 /mnt/usb vfat rw, fmask=0022, dmask=0022, codepage=cp932, iocharset=euc-jp 0 0
```

```
MA-E120 ~ # umount /dev/usb (USB デバイスのアンマウント)
```

4.3. LED

本装置では、ユーザアプリケーションが LED を変化させることができます。

4.3.1. /dev/led インターフェース

/dev/led インタフェースを使用し STATUS LED の制御が可能です。echo コマンドで /dev/led へ出力することで LED の操作ができます。

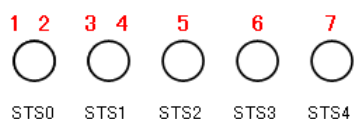
以下の例では STS0 を緑点灯させます。

```
MA-E120 ~ # echo 1 1 > /dev/led
```

echo (LED 番号) (制御番号) > /dev/led

LED 番号

- 1 : STS0 緑
- 2 : STS0 赤
- 3 : STS1 緑
- 4 : STS1 赤
- 5 : STS2 緑
- 6 : STS3 緑
- 7 : STS4 緑



制御番号

0 : 消灯

1 : 点灯

2 : 点滅

4.4. INIT ボタン

本装置では、Init ボタンを押すことによりシステムをシャットダウンすることができます。

Init ボタンの状態は、/proc ファイルシステム上のファイル「/proc/driver/ initsw」としてアクセス可能です。

Init ボタンの状態が on/off により表されます。

5. NFS ルートを利用したセルフ開発環境

5.1. NFS ルートの概要

MA-E120 は、SDK として NFS ルートファイルシステムを提供しています。

本装置では、NFS ルートにより、このルートファイルシステムを利用することができます。NFS ルートを利用する場合は、別途 Linux が動作する PC に NFS ルートファイルシステムを構築する必要があります。

NFS ルートファイルシステムを利用することで、主に次のことが出来るようになります。

- セルフ開発環境によるユーザランドアプリケーションを開発することができます。
本装置のすべてのユーザランドアプリケーションの開発は、NFS ルートファイルシステムを利用して開発します。
本ルートファイルシステムは、Gentoo Linux を利用しています。binutils, gcc 等のセルフ開発環境として必要なパッケージ類がインストール済みです。
- 本装置の外部ストレージとして使用することができます。
FLASH メモリや SD カードに収まらないアプリケーションや、多量のデータを扱うことができます。
- SD カード用のルートファイルシステムとして使用することができます。

【ファームウェアイメージを開発する場合】

FLASH メモリへ格納するカーネルとルートファイルシステムを含むファームウェアイメージの開発環境は、後述する「VMware を利用したクロス開発環境」を利用します。

ユーザランドアプリケーションを開発し、それをファームウェアイメージへ組み込む場合は、VMware 仮想マシン上の Linux に NFS ルートファイルシステムを構築すると開発効率が向上します。

5.2. NFS ルートの設定

NFS ルートを使用するときは、Linux が動作する PC に、NFS サーバの設定、DHCP サーバの設定、および NFS ルートファイルシステムを用意する必要があります。

ステップ1 NFS ルートファイルシステムの展開

NFS ルートファイルシステムを Linux が動作する PC に展開しておきます。

展開場所は /var/export とします。

例:

```
# cd /var/export
# tar pjxvf mae120_rootfs_2010XXXX.tar.gz2
```

ステップ2 NFS ルートファイルシステムの移動

SDK の NFS ルートファイルシステムを展開して作成された MAE120_rootfs ディレクトリ名を rootfs というディレクトリ名へ変更しておきます。

以後、NFS ルートファイルシステムの NFS ルートの配置場所を /var/export/rootfs とします。

例:

```
# ls -la
drwxr-sr-x 20 root  250      4096 May 11 14:37 MAE120_rootfs
# mv MAE120_rootfs/ rootfs/
# ls -la
drwxr-sr-x 20 root  250      4096 May 11 14:37 rootfs
```

ステップ3 NFS サーバの設定

ステップ1を行った Linux が動作する PC で、NFS サーバの設定を行います。NFS の設定は/etc/exports ファイルで行います。ここでは/var/export/rootfs/ を NFS ルートファイルシステムのトップパスとします。

例: /etc/exports

```
/var/export/rootfs/ 192.168.252.*(rw,no_root_squash,insecure,syncs)
```

NFS サーバを起動させます。

例:

```
# /etc/init.d/nfs start
```

ステップ4 DHCP サーバの設定

動時に IP アドレスをリースし、root-path パラメータを渡す設定を行います。DHCP サーバの設定は /etc/dhcpd.conf ファイルで行います。

例: /etc/dhcpd.conf

```
# MA-E120
subnet 192.168.252.0 netmask 255.255.255.0 {
  group {
    hostMA-E120 {
      hardware ethernet 00:80:6d:xx:xx:xx;      # MA-E120 Mac address
      fixed-address 192.168.252.252;           # MA-E120 IPaddress
      next-server 192.168.252.1;               # Server IP
      option host-name "mae120";               # 割り当てるホスト名
    }
  }
}
```



```
option root-path "/var/export/rootfs"; # NFS
}
}
}
```

- ・ 接続しているネットワークは 192.168.252.0/24
- ・ サブネットマスクは 255.255.255.0
- ・ クライアントの MAC アドレスは、00:80:6d:xx:xx:xx
- ・ クライアントの IP アドレスは 192.168.252.252
- ・ 自ホストの IP アドレスは 192.168.252.1
- ・ クライアント名は mae120
- ・ NFS ルートファイルシステムのマウンティングポイントは"/var/export/rootfs/"

設定終了後は DHCP サーバを起動させます

例:

```
# /etc/init.d/dhcpd start
```

ステップ5 MA-E120 の起動

本装置を NFS ルートで起動します。NFS ルートで起動するには以下の方法があります。

ディップ・スイッチの変更

本装置ではディップ・スイッチで起動モードを変更することができます。NFS ルートで起動する場合、CONFIG ディップ・スイッチを 1:OFF、2:ON に設定します。

u-boot による変更

本装置では u-boot でも起動モードを変更することができます。

設定の方法については、u-boot のドキュメントを参照して下さい。

5.3. アプリケーションのビルド方法

通常の Linux 上でユーザランドのアプリケーションを開発する場合と同様に、**MA-E120** の NFS ルートファイルシステム上でアプリケーションのビルド、実行等を行う事ができます。

6. VMware を利用したクロス開発環境

6.1. VMware 環境の概要

MA-E120 は、SDK としてクロス開発環境をインストールした Linux (Gentoo Linux) の VMware 仮想マシンイメージ、カーネルソース、およびビルドキットを提供しています。

本装置では、クロス開発環境と、その環境上でカーネルソースとビルドキットを利用することができます。

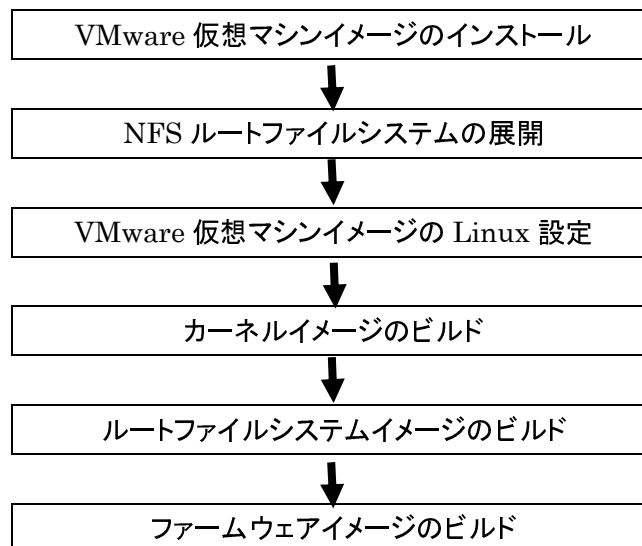
VMware 仮想マシンイメージを利用する場合は、別途 Windows が動作する PC へ VMwarePlayer をインストールし、VMware 仮想マシンイメージを設定する必要があります。

VMware を利用したクロス開発環境を利用することで、主に次のことができるようになります。

- FLASH メモリに格納するカーネルイメージをビルドすることができます。
- FLASH メモリに格納するルートファイルシステムのイメージをビルドすることができます。
セルフ開発環境でビルドしたユーザランドアプリケーションを、このルートファイルシステムに組み込むことが出来るようになります。
- カーネルとルートファイルシステムを含むファームウェアイメージをビルドすることができます。

6.2. クロス開発環境の開発フロー

VMware を利用したクロス開発環境の開発フローの全体像を次に示します。



6.3. クロス開発環境の構成

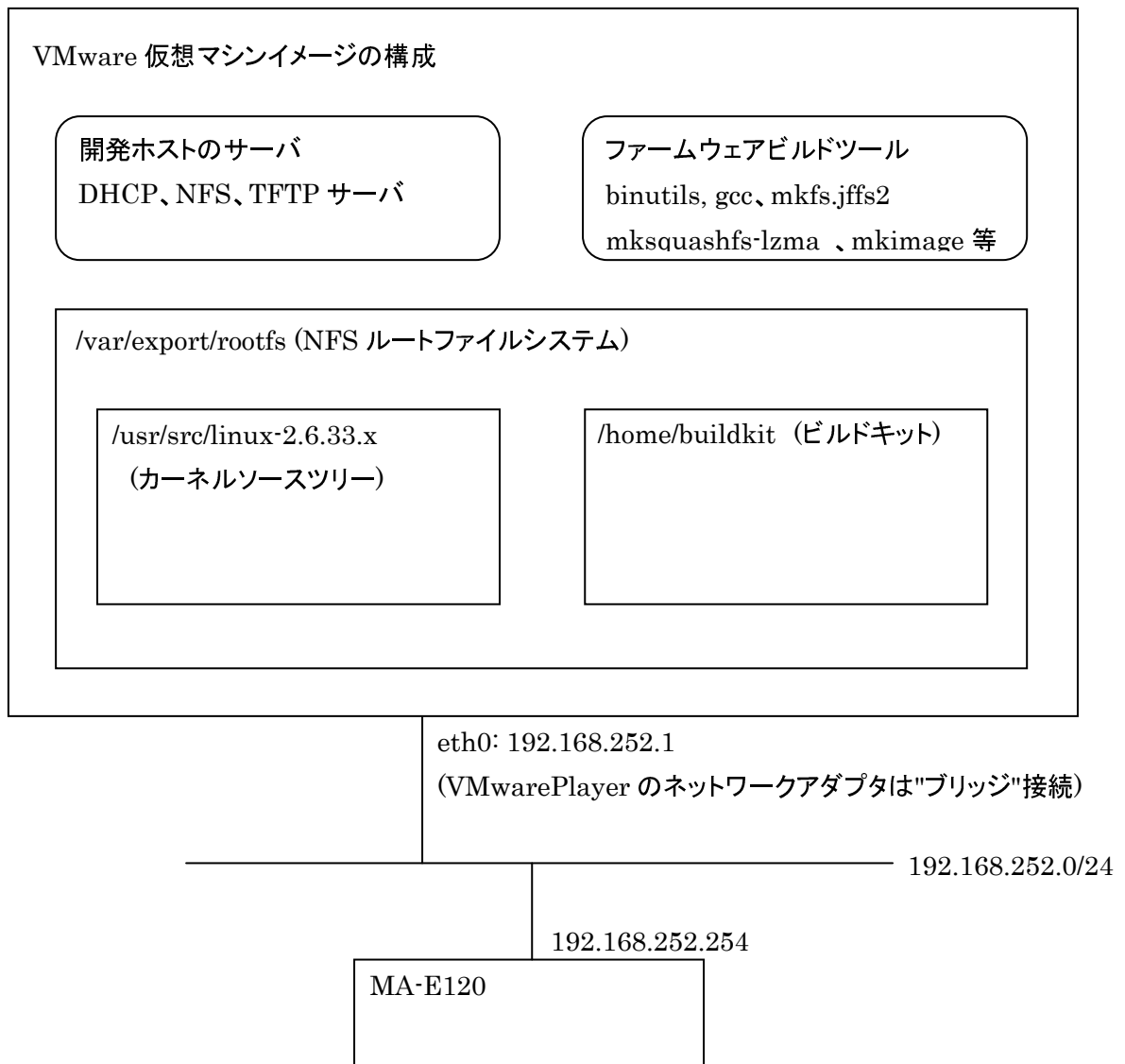
VMware を利用したクロス開発環境のシステムおよびディレクトリ構成を次に示します。

これ以降の手順では、このディレクトリ構成に従って説明をします。

- ・ `/var/export/rootfs` NFS ルートファイルシステム
- ・ `/var/export/rootfs/usr/src/ linux-2.6.33.x` カーネルソースツリー
- ・ `/var/export/rootfs/home/buildkit` ビルドキット

「注意事項」

このディレクトリ構成と IP アドレス以外でクロス開発を行う場合は、適宜説明を読み替えてクロス開発を行ってください。また、ビルドキットに収録されているスクリプトは、編集してから使用してください。



6.4. VMware 仮想マシンイメージのインストール

事前に、PC へ VMwarePlayer をインストールしてください。添付の VMware 仮想マシンイメージは、下記の VMware player で動作確認をしています。

・VMware player 3.0.0 build-203739、 3.0.1 build-227600

ステップ1 仮想マシンイメージの展開

SDK の仮想マシンイメージ「masdk2010XXXX.zip」を PC 上で展開してください。

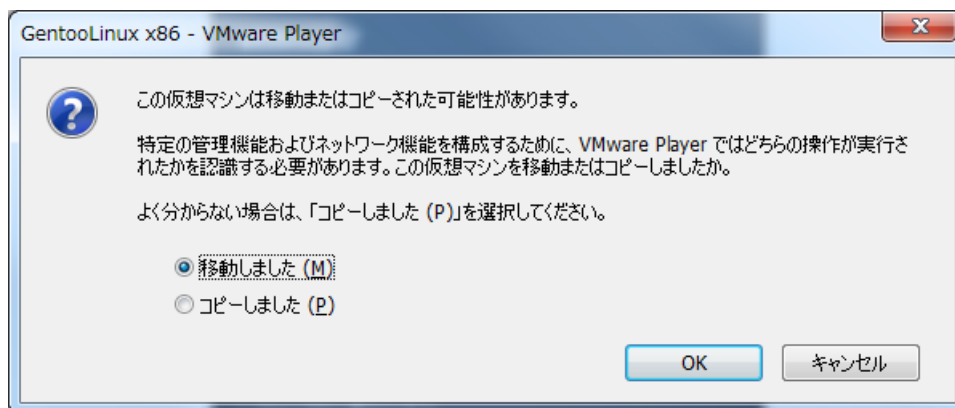
例えば、VMwarePlayer が仮想マシンイメージを管理する「C:\Virtual Machines」フォルダへ「masdk2010XXXX.zip」をコピーし、そこでこのファイルを展開します。

VMwarePlayer の「仮想マシンを開く」コマンドを使用して、展開した仮想マシンイメージを開きます。

ステップ2 仮想マシンイメージの実行

VMwarePlayer の「仮想マシンの再生」コマンドを使用して、VMwarePlayer に登録した仮想マシンを再生してください。

「この仮想マシンは移動またはコピーされた可能性があります。」とダイアログが表示されたら、「移動しました」を選択してください。



・ログイン

root ユーザー(パスワード=root)でログインできます。

・IP アドレス他

eth0 に 192.168.252.1/24 が設定されています。

sshd が自動起動します。

6.5. NFS ルートファイルシステムの展開

ステップ1 NFS ルートファイルシステムの展開

VMware 仮想マシンとして実行した Linux に、SDK の NFS ルートファイルシステムを展開しておきます。展開場所は /var/export とします。

例:

```
# cd /var/export
# tar pjxvf mae120_rootfs_2010XXXX.tar.bz2
```

ステップ2 NFS ルートファイルシステムの移動

SDK の NFS ルートファイルシステムを展開して作成された MAE120_rootfs ディレクトリ配下のすべてのファイルを、rootfs ディレクトリへ移動しておきます。

以後、NFS ルートファイルシステムの NFS ルートの配置場所を /var/export/rootfs とします。

例:

```
# ls -la
drwxr-sr-x 20 root portage      4096 May 11 14:37 MAE120_rootfs
drwxr-xr-x  2 root root        4096 Feb 13  2009 rootfs
# mv MAE120_rootfs/* rootfs/
```

6.6. VMware 仮想マシンイメージの Linux 設定

ステップ1 DHCP サーバの設定

本装置起動時に IP アドレスをリースし、root-path パラメータを渡す設定を行います。DHCP サーバの設定は /etc/dhcp/dhcpd.conf ファイルで行います。

例: /etc/dhcp/dhcpd.conf

```
# MA-E120
host mae120 {
    hardware ethernet 00:80:6d:xx:xx:xx;      # MA-E120 の MAC アドレス
    fixed-address 192.168.252.252;           # MA-E120 に割り当てる IPアドレス
    next-server 192.168.252.1;              # TFTP サーバの IPアドレス
    filename "mae120.img";                  # ファームウェアファイル名
    option host-name "mae120";              # 割り当てるホスト名
    option root-path "/var/export/rootfs";   # NFS ルートの配置場所
#    option routers 192.168.252.1;          # デフォルトルート
    option subnet-mask 255.255.255.0;      # サブネットマスク
    option broadcast-address 192.168.252.255; # ブロードキャストアドレス
#    option domain-name-servers 192.168.252.2; # DNS サーバ IP アドレス
}
```

- ・ 接続しているネットワークは 192.168.252.0/24
- ・ サブネットマスクは 255.255.255.0
- ・ クライアントの MAC アドレスは、00:80:6d:xx:xx:xx
- ・ クライアントの IP アドレスは 192.168.252.252
- ・ TFTP サーバの IP アドレスは 192.168.252.1
- ・ ファームウェアのファイル名は "mae120.img"
- ・ クライアント名は "mae120"
- ・ NFS ルートファイルシステムの NFS ルートの配置場所は "/var/export/rootfs"

設定終了後は DHCP サーバを起動させます

例:

```
# /etc/init.d/dhcpd start
```

ステップ2 NFS サーバの設定

NFS サーバの設定を行います。NFS の設定は/etc/exports ファイルで行います。ここでは/var/export/rootfs/ を NFS ルートファイルシステムの NFS ルートの配置場所とします。

例: /etc/exports

```
/var/export/rootfs 192.168.252.*(rw,no_root_squash,insecure, sync,no_subtree_check)
```

NFS サーバを起動させます。

例:

```
# /etc/init.d/nfs start
```

ステップ3 TFTP サーバの設定

仮想マシンイメージの Linux には TFTP サーバが起動されています。TFTP ルートは/tftpboot です。ここにファームウェアを格納してください。

ファイル名を dhcpd.conf の filename オプションで指定します。ファイル名を変更する場合は、dhcpd.conf を編集し、DHCP サーバおよび、xinetd を再起動してください。

DHCP サーバおよび、xinetd を再起動させます。

例:

```
# /etc/init.d/dhcpd restart
# /etc/init.d/xinetd restart
```

TFTP サーバの設定を変更する場合

TFTP サーバの設定は、/etc/xinetd.d/tftp ファイルで行います。

例: /etc/xinetd.d/tftp

```
service tftp
{
    protocol = udp
    port = 69
    socket_type = dgram
    wait = yes
    user = root
    server = /usr/sbin/in.tftpd
    server_args = /tftpboot
    only_from = 192.168.252.0/24
    disable = no
}
```

6.7. カーネルイメージのビルド

ステップ1 カーネルソースの展開

VMware 仮想マシンとして実行した Linux に、SDK のカーネルソースを展開しておきます。
展開場所は /var/export/rootfs/usr/src とします。

例:

```
# cd /var/export/rootfs/usr/src
# tar pxvf linux-2.6.33.x_mae120.tar.bz2
```

ステップ2 カーネルの config 編集

Linux カーネルツリー中の、dot.config.mae120 がカーネルコンフィグです。これを.config にコピーして、"make oldconfig"、または必要に応じて"make menuconfig"で編集してください。

例:

```
# cd linux-2.6.33.x_mae120
# make mrproper
# cp dot.config.mae120 .config
# make oldconfig      (または# make menuconfig)
```

ステップ3 カーネルイメージのビルド

カーネルをビルドします。カーネルイメージ uImage は次のパスに生成されます。
/var/export/rootfs/usr/src/linux-2.6.33.x_MAE2xx/arch/powerpc/boot/

例:

```
# make ulmage
# ls /var/export/rootfs/usr/src/linux-2.6.33.x_mae120/arch/powerpc/boot
Image Makefile bootp compressed install.sh ulmage zImage
```


ステップ4 カーネルモジュールのビルドとインストール

カーネルモジュールをビルドします。

ビルド後、カーネルモジュールを NFS ルートファイルシステムへインストールします。これにより、**MA-E120** の新しいカーネルで、本 NFS ルートファイルシステムを利用できるようになります。

なお、カーネルモジュールは、後述する手順でビルドキットのルートファイルシステムへもインストールします。

例:

```
# make modules
# make modules_install INSTALL_MOD_PATH=/var/export/rootfs/
```

6.8. ルートファイルシステムイメージのビルド

ステップ1 ビルドキットの展開

VMware 仮想マシンとして実行した Linux に、SDK のビルドキットを展開しておきます。

展開場所は /var/export/rootfs/home/buildkit とします。

例:

```
# cd /var/export/rootfs/home
# mkdir buildkit
# cd buildkit
# tar pjxvf mae120_mtdroot_vxxx.tar.bz2
```

例:

```
# ls -la
-rwxr-xr-- 1 root root 1086 Apr 12 14:33 build.sh
drwxr-xr-x 19 root root 4096 Apr 14 10:18 embedded_glibc
-rw-r--r-- 1 root root 34910694 May 10 12:05 mae120_mtdroot_vxxx.tar.bz2
-rwxr-xr-- 1 root root 757 Apr 12 14:33 rootfs.sh
```

·embedded_glibc	FLASH メモリ用ルートファイルシステムの先頭ディレクトリ
·rootfs.sh	ルートファイルシステムのイメージ作成スクリプト
·build.sh	ファームウェアのイメージ作成スクリプト

ステップ2 カーネルモジュールのインストール

カーネルモジュールをビルドキットのルートファイルシステムへインストールします。

例:

```
# cd /var/export/rootfs/usr/src/linux-2.6.33.x_mae120
# make modules_install INSTALL_MOD_PATH=/var/export/rootfs/home/buildkit/embedded_glibc/
```

ステップ3 アプリケーションのインストール

ユーザが開発したアプリケーションをビルドキットのルートファイルシステムツリー上へインストールします。

例:

```
# cd /var/export/rootfs/home/buildkit/embedded_glibc
# tar pjxvf user_application.tar.bz2
```

ステップ4 ルートファイルシステムイメージの作成

ビルドキットのルートファイルシステムイメージを作成します。

rootfs.sh スクリプトを実行してください。実行後は /var/export/rootfs/home/buildkit/embedded_glibc はルートファイルシステムイメージへ変換されます。

rootfs.sh スクリプトでは、最初にビルドキットのルートファイルシステムをバックアップします。このバックアップは、ルートファイルシステムイメージを再度作成しなおす場合に利用します。

例:

```
# cd /var/export/rootfs/home/buildkit
# ./rootfs.sh
```

rootfs.sh スクリプト:

```
#!/bin/bash

echo "backup embedded_glibc"
cp -r embedded_glibc embedded_glibc_backup

echo "cd embedded_glibc"
cd embedded_glibc
ROOTFS_DIR=$PWD

echo "cd usr/lib/python2.6"
cd usr/lib/python2.6

find . -name "*.py[co]" | xargs rm
find . -name "test" | xargs rm -rf
find . -name "tests" | xargs rm -rf
rm -rf idlelib
python compileall.py .
find . -name "*.py" | xargs rm

cd ../../../../
if [ ${ROOTFS_DIR} != $PWD ]; then
    exit 1
fi
rm -rf var/db/pkg
rm -rf var/cache/edb
mv var .var.rom
mkdir var

mksquashfs_lzma usr opt/system/usr.sqfs

mkdir -p ../usr_backup
rm -rf ../usr_backup/*
mv usr/* ../usr_backup
mkdir -p usr/lib/gcc/powerpc-unknown-linux-gnu/4.3.4
cp -a ../usr_backup/lib/gcc/powerpc-unknown-linux-gnu/4.3.4/libgcc_s.so* usr/lib/gcc/
powerpc-unknown-linux-gnu/4.3.4/
```

ステップ5 ルートファイルシステムを繰り返し作成する場合

(1)初期状態から作成する場合

本章の「**ステップ1 ビルドキットの展開**」からファイルシステムイメージの作成を行ってください。

(2)ユーザが開発したアプリケーションを更新する場合

バックアップしたルートファイルシステムを使用してアプリケーションの更新を行います。

アプリケーションを更新後 rootfs.sh スクリプト実行し、ファイルシステムイメージの作成を行ってください。

例:

```
# cd /var/export/rootfs/home/buildkit
# rm -r embedded_glibc
# mv embedded_glibc_backup embedded_glibc
# ユーザアプリケーションの更新
# ./rootfs.sh
```

6.9. ファームウェアイメージのビルド

ステップ1 ファームウェアイメージの作成

ファームウェアイメージを作成します。

スクリプト build.sh を実行してください。実行後は、ファームウェアイメージ「mae120.img」がカレントディレクトリに作成されます。

※build.sh

ディレクトリ構成が本ドキュメントと異なる場合は、お客様の環境に合わせて内容を変更してください。

例:

```
# cd /var/export/rootfs/home/buildkit
# ./build.sh

# ls -la
-rwxr-xr-- 1 root root 1086 Apr 12 14:33 build.sh
drwxr-xr-x 19 root root 4096 Apr 14 10:18 embedded_glibc
drwxr-xr-x 18 root root 4096 Jun 2 2009 embedded_glibc_backup
-rw-r--r-- 1 root root 33030144 Apr 14 10:19 mae120.img
-rw-r--r-- 1 root root 537616 Apr 14 10:19 mtd3.kernel.pad
-rw-r--r-- 1 root root 29360128 Apr 14 10:19 root.jffs2
-rwxr-xr-- 1 root root 757 Apr 12 14:33 rootfs.sh
drwxr-sr-x 7 root root 4096 Apr 14 10:19 usr_backup
```

build.sh スクリプト:

```
#!/bin/sh
#set -x

LINUXROOT=/var/export/rootfs/usr/src/linux-2.6.33.x_mae120
IMAGE_FILE=ulmage
IMAGE_KERNEL=${LINUXROOT}/arch/powerpc/boot/${IMAGE_FILE}
IMAGE_SIZE=3670016
IMAGE_DIR=$PWD
IMAGE_PAD=${IMAGE_DIR}/mtd3.kernel.pad
ROOTFS_DIR=${IMAGE_DIR}/embedded_glibc
ROOTFS_FILE=root.jffs2
FIRM_FILE=mae120.img

echo "Make rootfs"
rm $ROOTFS_FILE
mkfs.jffs2 -b -e 0x20000 --pad=0x1be0000 -r $ROOTFS_DIR -o $ROOTFS_FILE
if [ $? -ne 0 ]; then
    echo "Make $ROOTFS_FILE error"
    exit -1
fi

echo "Make firmware"
KERNEL_SIZE=`stat -c %s $IMAGE_KERNEL`
if [ $KERNEL_SIZE -gt $IMAGE_SIZE ]; then
    echo "WARNING: kernel size is more than $IMAGE_WARNING($KERNEL_SIZE)"
    exit 1
fi

rm $IMAGE_PAD
KERNEL_PAD_SIZE=$(( $IMAGE_SIZE - $KERNEL_SIZE ))
dd if=/dev/zero of=$IMAGE_PAD bs=1 count=$KERNEL_PAD_SIZE > /dev/null 2>&1
if [ $? -ne 0 ]; then
    echo "Make $IMAGE_PAD error"
    exit -1
fi
```

```
rm $FIRM_FILE
cat $ROOTFS_FILE $IMAGE_KERNEL $IMAGE_PAD > $FIRM_FILE
if [ $? -ne 0 ]; then
    echo "Make $FIRM_WARE error"
    exit -1
fi

# Build succeeds!
echo "Make firmware succeeds!"
```

ステップ2 ファームウェアイメージの更新

「2.5.2. ファームウェアのアップデート」に従ってファームウェアを更新することができます。

7. SD ブートシステムの構築

7.1. SD ブートシステムの概要

MA-E120 は、SDKとしてSDカード用のルートファイルシステムを提供しています。

本装置では、SDカードを初期化しルートファイルシステムを構築する必要があります。

なお、本手順では、本装置を使用してSDカードの初期化からルートファイルシステムを構築までを行います。

SDカード用ルートファイルシステムを利用することで、主に次のことが出来るようになります。

- ユーザ独自のルートファイルシステムとして使用することができます。
FLASHメモリに収まらないアプリケーションや、多量のデータを扱うことができます。

【注意事項】

SDカード製品には、書き換え回数の上限という寿命があります。SDカード製品の寿命は、メーカー、製品種類（仕様）ごとにそれぞれ異なっています。

SDブートシステムの構築にあたり、お客様でSDカード製品の寿命を把握したうえでご使用ください。

7.2. SDカードの初期化

ステップ1 SDカードのデバイスの確認

本装置のSD CARD スロットへSDカードを挿入します。挿入後、lsコマンドでSDカードのデバイス名を確認します。

本装置のSDカードのデバイス名は /dev/sdX となります。パーティションが作成されている場合は、デバイス名は/dev/sdX1 となります。

例:

```
MA-E120 ~ # ls /dev/sdX*  
/dev/sdX    /dev/sdX1
```

ステップ2 パーティションの確認

SDカードにパーティションが作成されている場合は、いったんすべてを削除します。その後、SDブートシステム用のパーティションを1つ作成します。

なお、SDブートシステムは、Linuxファイルシステムを使用します。その他のファイルシステムは使用できません。

パーティションの確認の確認を行います。確認にはfdiskでCommand pを使用します。

例:

```
MA-E120 ~ # fdisk /dev/sdX
```

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

```
Disk /dev/sdX: 2014 MB, 2014838784 bytes
64 heads, 63 sectors/track, 976 cylinders
Units = cylinders of 4032 * 512 = 2064384 bytes
Disk identifier: 0x00000000
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdX1		1	976	1967489+	6	FAT16

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

ステップ3 パーティションの削除

パーティションの削除を行います。削除には fdisk で Command d を使用します。

例:

Device	Boot	Start	End	Blocks	Id	System
/dev/sdX1		1	976	1967489+	6	FAT16

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

```
Selected partition 1
```

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

```
Disk /dev/sdX: 2014 MB, 2014838784 bytes
64 heads, 63 sectors/track, 976 cylinders
Units = cylinders of 4032 * 512 = 2064384 bytes
Disk identifier: 0x00000000
```

Device	Boot	Start	End	Blocks	Id	System
--------	------	-------	-----	--------	----	--------

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

ステップ4 パーティションの作成

パーティションの作成を行います。作成には fdisk で Command n を使用します。

例:

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-976, default 1): 1
Last cylinder, +cylinders or +size[K,M,G] (1-976, default 976):
Using default value 976

Command (m for help): p

Disk /dev/sdX: 2014 MB, 2014838784 bytes
64 heads, 63 sectors/track, 976 cylinders
Units = cylinders of 4032 * 512 = 2064384 bytes
Disk identifier: 0x00000000

   Device Boot      Start         End      Blocks   Id  System
/dev/sdX1          1           976     1966064    83  Linux

Command (m for help):
```

ステップ4 パーティションの書き込み

パーティションの書き込みを行います。書き込みには fdisk で Command w を使用します。

例:

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

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

7.3. ルートファイルシステムの構築

ステップ1 SDカードのフォーマット

SDカードのフォーマットを行います。

本装置にSDカードを挿入し、その後にフォーマットコマンドを使用します。

なお、SDブートシステムは、LinuxファイルシステムのEXT2かEXT3を使用します。お客様のご利用環境にあったファイルシステムを選択してください。ここではEXT3を使用します。

例:

```
MA-E120 ~ # mkfs.ext3 /dev/sdX1
```

ステップ2 SDカード用ルートファイルシステムの展開

本装置の/tmpへSDカード用ルートファイルシステムのアーカイブを準備しておきます。

本装置に、SDカードをマウントし、その後にSDカードへSDカード用ルートファイルシステムを展開します。

例:

```
MA-E120 ~ # mount -t ext3 /dev/sdX1 /mnt/sd
MA-E120 ~ # ls /mnt/sd
lost+found2

MA-E120 ~ # cd /mnt/sd
MAE2xx sd # tar pjxvf /tmp/xxxxxx_2010XXXX.tar.bz2
```

/MAE120_rootfs に展開されたファイルを "/" へ移動します。

例:

```
MAE120 sd # ls
MA120_rootfs lost+found

MAE120 sd # mv MAE120_rootfs/* ./
MAE120 sd # rmdir MAE120_rootfs

MAE120 sd # ls
bin dev home lost+found opt rc sbin sys usr
boot etc lib mnt proc root service tmp var
```

7.4. SD カードからの起動

本装置を SD カードのルートファイルシステムで起動します。SD カードで起動するには以下の方法があります。

□ デイップ・スイッチの変更

本装置は CONFIG デイップ・スイッチで起動モードを変更することができます。SD カードで起動する場合、デイップ・スイッチを 1:OFF、2:ON に設定し INIT スイッチを押しながら電源を投入して下さい。

8. 仕様一覧

8.1. MA-E120 の仕様

製品名		FutureNet MA-E120
CPU		Freescale™ MPC8313E®プロセッサ 333MHz (PowerQUICC II Pro)
Flash ROM		32Mbyte
RAM		128Mbyte
LAN/WAN	Gigabit Ethernet インタフェース	Gigabit Ethernet × 2 ポート 1000BASE-T/100BASE-TX/10BASE-T(RJ-45)
	適応回線・適応接続サービス	NGN、Ethernet-WAN、ADSL/SDSL、FTTH、CATV、モバイルデータ通信 固定 IP/PPPoE/DHCP ブロードバンドサービス
	動作モードの固定設定	○
モバイルデータ通信端末接続インタフェース		USB 型 × 2 ポート(USB メモリ共用) ExpressCard 型 × 1 スロット ※同時に利用できるのは1つのみ
外部メモリ用インタフェース		SD カードスロット × 1 CF カードスロット × 1 (筐体内蔵、SD カードと排他利用を推奨) USB2.0 × 2 ポート(モバイルデータ通信共用) ※USB メモリ: Memory Media USB-128(別売)
コンソールポート		※ 設定用RS-232ポート<RJ-45コネクタ> ※ 変換アダプタ、ケーブル付属
LED 表示		システム: Power × 1, Status × 5, Ethernet: Link/Active × 2, Speed × 2
基本ソフトウェア	OS	Linux Kernel 2.6
	実行時ライブラリ	glibc 2.10.1-r1
	起動方法	FlashROM boot、NFS Root(dhcp)、SD boot、USB boot
	PPP 接続	○
	ネットワーク機能	デフォルトルーティング、スタティックルーティング、 iptables によるパケットフィルタ IP マスカレード
サンプルアプリケーション		LAN 機器の死活監視/メール通知
運用管理	設定手段	WEB 設定画面、Linux ログイン(シェル)、SSH
	ファームウェア更新	○ ※ WEB 設定画面、tftp
	構成定義情報	WEB 設定の設定内容ダウンロード、アップロード
	ログ監視	Syslog(metalog)による監視
	その他	DHCP サーバ、時刻設定、NTP クライアント/サーバ
認定/準拠	VCCI	Class A 準拠
	RoHS 指令	○
サイズ・重量	外観寸法(突起物を除く)	174mm(W) x 111mm(D) x 33mm(H)
	本体重量	約 240g
環境	使用電源、電源形状	DC 5V AC アダプタ(入力 AC 100V±10% 50Hz/60Hz、平行 2 ピン)
	消費電力 (USB 接続なし)	最大約 5.5W
	本体動作温度条件	0°C~50°C
	AC アダプタ動作温度条件	-10°C~40°C
	動作湿度条件	10%~85%(結露なきこと)
	保存環境	温度: -20°C~60°C、10~85%(結露なきこと)

	冷却方式	自然空冷
添付品		DVD(取扱説明書、開発キット含む)、保証書、 AC アダプタ
開発キット(開発環境)		gcc 4.3.4、binutils 2.20.1

以上の内容は 2010 年 10 月時点のものです。改良のため予告なく内容・仕様を変更することがあります。

8.2. オープン・ソース・ソフトウェアのライセンスについて

本製品は、各種オープン・ソース・ソフトウェアを使用しており、各々のライセンス条件に従ってソース・コードの入手、改変、再配布の権利があることをお知らせします。

オープン・ソースとしての性格上著作権による保証はなされていませんが、本製品については保証書記載の条件により弊社による保証がなされています。

ライセンスについては弊社 Web 製品サポートページをご覧ください。

MA-E120 製品サポートページ

<http://www.centurysys.co.jp/support/>

FutureNet MA-E120 Series ユーザーズガイド Ver 1.0.1

2010年10月版

発行 センチュリー・システムズ株式会社

Copyright© 2010 Century Systems Co., Ltd. All rights reserved.
