

FutureNet MA-E200 Series

ユーザーズガイド

Version 1.1.0



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
FutureNet MA-E200 Series の概要	7
1. FutureNet MA-E210 の概要.....	8
1.1. FUTURENET MA-E210 の特徴	8
1.2. FUTURENET MA-E210 応用例	10
1.3. 外観	11
1.4. RS-232 インターフェース仕様.....	14
1.5. RS-485 インターフェース仕様.....	15
1.6. CONFIG の説明	17
1.7. LED の説明	18
1.7.1. MA-E210 状態表示 LED	18
2. MA-E210 の基本的な操作.....	20
2.1. ソフトウェアの概要	20
2.2. システムへのログイン.....	20
2.3. 起動と停止の方法.....	21
2.3.1. 起動の方法	21
2.3.2. ネットワークで接続する.....	21
2.3.3. コンソールから接続する.....	21
2.3.4. JFFS2 ルート起動	22
2.3.5. NFS ルート起動.....	22
2.3.6. コマンドによる起動方法の変更	22
2.3.7. 停止方法.....	22
2.3.8. 初期化	23
2.4. 基本的な使い方	24
2.4.1. ssh ログイン	24
2.4.2. シェル環境	24
2.4.3. ウェブユーザインターフェース.....	24
2.5. ソフトウェアのバージョンと更新.....	25
2.5.1. ソフトウェアのバージョン	25
2.5.2. ファームウェアのアップデート	26
3. ネットワーク機能の設定.....	30

3.1.	ネットワーク機能の概要	30
3.2.	IP アドレスの設定	30
3.3.	ルーティングの設定	31
3.3.1.	デフォルトゲートウェイ	31
3.3.2.	スタティックルーティング	31
3.3.3.	IP フォワーディング	31
3.4.	IP マスカレードの設定	32
3.5.	パケットフィルタの設定	32
4.	インターフェースの仕様	33
4.1.	インターフェース概要	33
4.2.	シリアルインターフェース(DTE)	33
4.3.	USB HOST	34
4.3.1.	USB serial converter	34
4.3.2.	USB Flash Memory(USB Mass Storage Class 対応)	34
4.4.	LED	35
4.5.	INIT ボタン	35
4.6.	RELEASE ボタン	36
5.	NFS ルートを利用したセルフ開発環境	37
5.1.	NFS ルートの概要	37
5.2.	NFS ルートの設定	37
5.3.	アプリケーションのビルド方法	39
6.	VMware を利用したクロス開発環境	40
6.1.	VMWARE 環境の概要	40
6.2.	クロス開発環境の開発フロー	40
6.3.	クロス開発環境の構成	41
6.4.	VMWARE 仮想マシンイメージのインストール	42
6.5.	NFS ルートファイルシステムの展開	43
6.6.	VMWARE 仮想マシンイメージの LINUX 設定	43
6.7.	カーネルイメージのビルド	46
6.8.	ルートファイルシステムイメージのビルド	47
6.9.	ファームウェアイメージのビルド	50
7.	SD ブートシステムの構築	53
7.1.	SD ブートシステムの概要	53
7.2.	SD カードの初期化	53
7.3.	ルートファイルシステムの構築	56
7.4.	SD カードからの起動	57

8. 仕様一覧	58
8.1. MA-E210 の仕様	58
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	60
9. MA-E210/AD-72 について	61
9.1. AD モデルの概要	61
9.2. 外観	62
9.3. AIN インターフェースの使い方	65
9.3.1. 仕様	65
9.3.2. AIN コネクタとピン配置	65
9.3.3. ハードウェアの接続方法	67
9.3.4. アプリケーションの作成方法	67
9.4. DIO インターフェースの使い方	71
9.4.1. 仕様	71
9.4.2. DIO コネクタとピン配置	72
9.4.3. ハードウェアの接続方法	75
9.4.4. アプリケーションの作成方法	75
9.5. FAIL 接点出力インターフェースの使い方	85
9.5.1. 仕様	85
9.5.2. FAIL 接点出力コネクタとピン配置	85
9.5.3. ハードウェアの接続方法	86
9.5.4. アプリケーションの作成方法	86
9.6. MA-E210/AD-72 の仕様	88
10. MA-E250/F について	90
10.1. FOMA モデルの概要	90
10.2. 外観	91
10.3. 接続と設置	94
10.3.1. FOMA データ通信サービスとアンテナの準備	94
10.3.2. 装置への接続	94
10.3.3. LED 表示の見方	95
10.4. MA-E250/F の仕様	97

FutureNet MA-E200 Series の概要

FutureNet MA-E200 Series は米 Freescale 社の産業・民生品市場向け CPU「**i.MX353**」(ARM11 アーキテクチャ)を搭載した Linux プラットフォーム製品です。**FutureNet MA-E200 Series** は豊富なインターフェースを備え、強力な耐環境性能、低消費電力を活かして、遠隔監視システムの拠点側データ収集装置兼通信装置として、また通信機能付き認証用端末、デジタルサイネージ端末などとして幅広く利用できます。

このシリーズの最初の製品である **FutureNet MA-E210** は、装置との接続用インターフェースとしてイーサネットポート、RS-232、USB 2.0 を備えます。LAN 対応の計測装置やセンサ、設備の監視用シリアルポート等と接続できます。**FutureNet MA-E210/AD-72** ではさらにアナログ入力、デジタル入出力のインターフェースを搭載します。これにより直接接続できるセンサや制御機器の種類が大幅に拡大します。

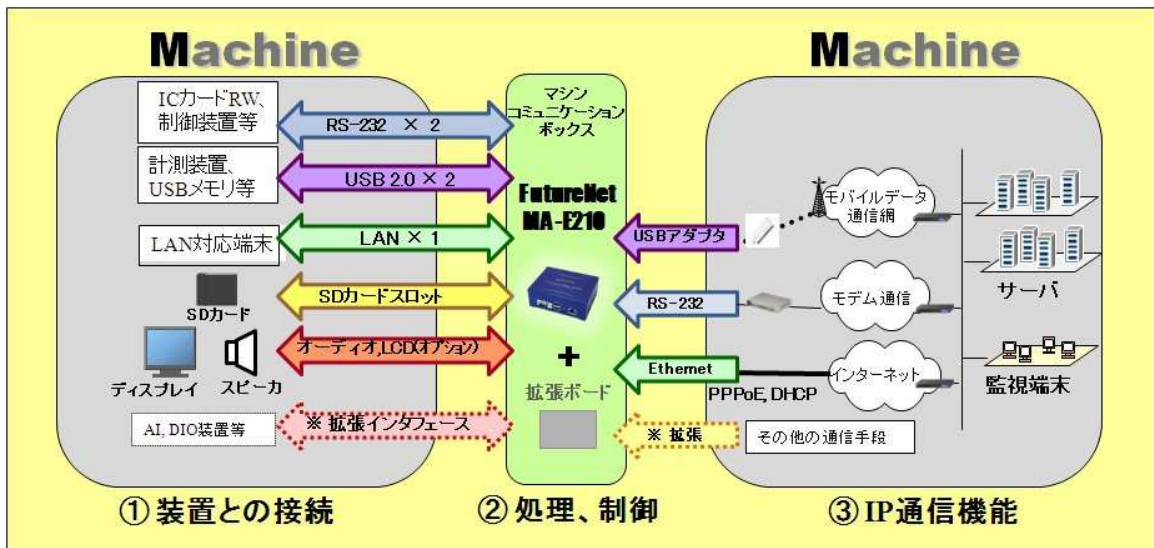
また、通信機能とその設定画面を標準で搭載しており、LAN もしくは USB ポートに接続したモバイルデータ通信端末経由でインターネットや閉域網サービスに接続できます。さらに **FutureNet MA-E250/F** は **FOMA** ユビキタスモジュールを内蔵し、厳しい環境下でも安定したモバイル通信機能を実現できます。

FutureNet MA-E200 Series の OS には Linux カーネル 2.6 を採用しています。そのため、オープンソースで提供されている各種のサーバアプリケーションが利用可能です。Linux アプリケーションの移植や追加、削除、ファームウェアの作成等は標準添付の開発環境 (SDK) を使っておこなえます。

1. FutureNet MA-E210 の概要

1.1. FutureNet MA-E210 の特徴

MA-E210 は、Linux アプリケーションを利用して様々な処理や制御に対応できる超小型 Linux ボックスです。機器どうしを接続する際の通信装置として、またデータの一時保管、変換などの処理装置として利用できます。イーサネットポート、RS-232、USB 2.0、Audio、SD Card の各種インターフェースを標準で実装し、同時に文庫本よりも小さいコンパクト・サイズを実現しました。



以下に **MA-E210** の特長をまとめます。

高性能

MA-E210 は、Freescale Semiconductor i.MX353(動作周波数 532MHz)高性能プロセッサを採用し、このクラスのマイクロサーバーとして最高の性能を発揮します。

多様なインターフェース

MA-E210 は、2つのシリアルポートを備えます。シリアルポートは標準では RS-232 ですが、実装オプションにより RS-485 あるいは内部に IC カードリーダーライター等を接続するための 10ピンコネクタに変更できます。

USB 2.0 ポートには、USB メモリや USB ハードディスク、USB シリアル変換器、データ通信端末などのデバイスを接続できます。USB ファイルシステム(USB メモリ)からの起動も可能です。USB 2.0 のポートは外部接続用に 2 ポート、組み込み接続用の内部コネクタを 2 ポート備えます。

SD カードスロットは、SD メモリカードに対応しています。プログラムやデータの保存、SD メモリカード上に構成したファイルシステムからの起動が可能です。

その他、端末として利用する場面ではスピーカ接続用のライン出力端子やマイク入力を利用できます。また、実装オプションで VGA の液晶ディスプレイモジュールを接続できます。

強力なネットワーク機能

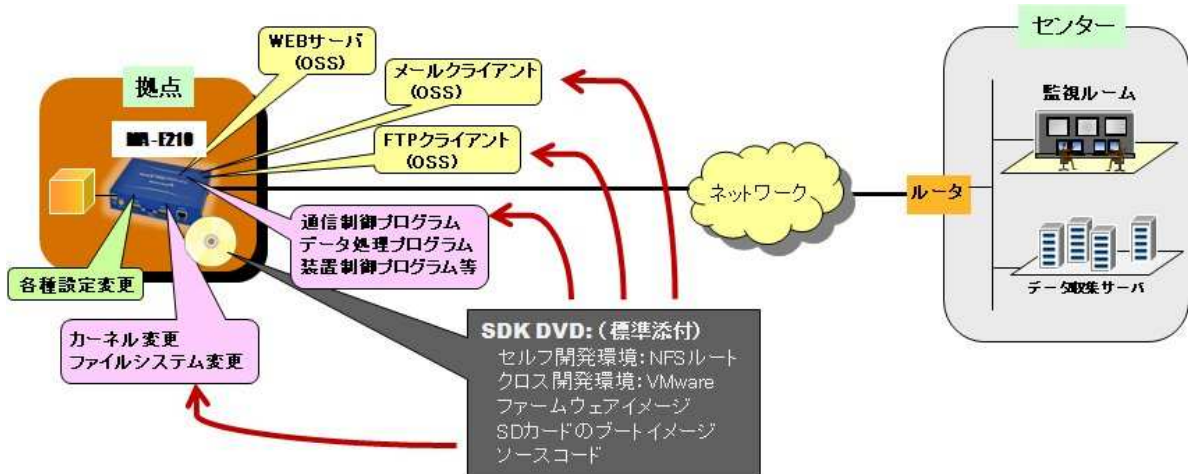
MA-E210 は、PPP や PPPoE 接続機能、経路制御、NAT/NPAT、パケットフィルタ、SSH による暗号化など Linux の優れたネットワーク機能を利用できます。通信手段としてはイーサネットによる有線のネットワーク接続、USB による 3G のモバイルデータ通信、シリアルポートを利用したモデム接続が可能です。また、イーサネットとモバイルデータ通信を組み合わせるとリモートルータとして利用することも可能です。サーバアプリケーションと組み合わせることで、特定用途向けの専用サーバ兼リモートルータを短期間で実現できます。

低消費電力、高信頼性、耐環境性

MA-E210 は、省電力 CPU や電源回路の最適化により、約 3W 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 -20°C ~ 60°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

MA-E210 の OS には、Linux 2.6 を採用しており、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、この製品でセルフコンパイルをおこなうのに必要なソフトウェアをネットワーク上の他の Linux パソコンから提供するための NFS ルート開発環境と、Linux のクロス開発環境を作るのに必要なソフトウェアをインストールした状態の 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、**FutureNet MA-E200 Series** で動作する Linux アプリケーションの開発や、独自の起動用 USB/SD メモリカード(ファイルシステム)の作成、独自のファームウェアの作成が可能です(*)。



* 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。

多様な実装オプションの選択が可能

MA-E210 では、様々な実装オプションをご用意しております。OEM や組込用途等において量産時には、必要のない I/F を部品ごと削除することにより製品単価を抑えることもできます。

カスタマイズサポート

センチュリー・システムズ(株)では FutureNet MA-E210 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、インター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。

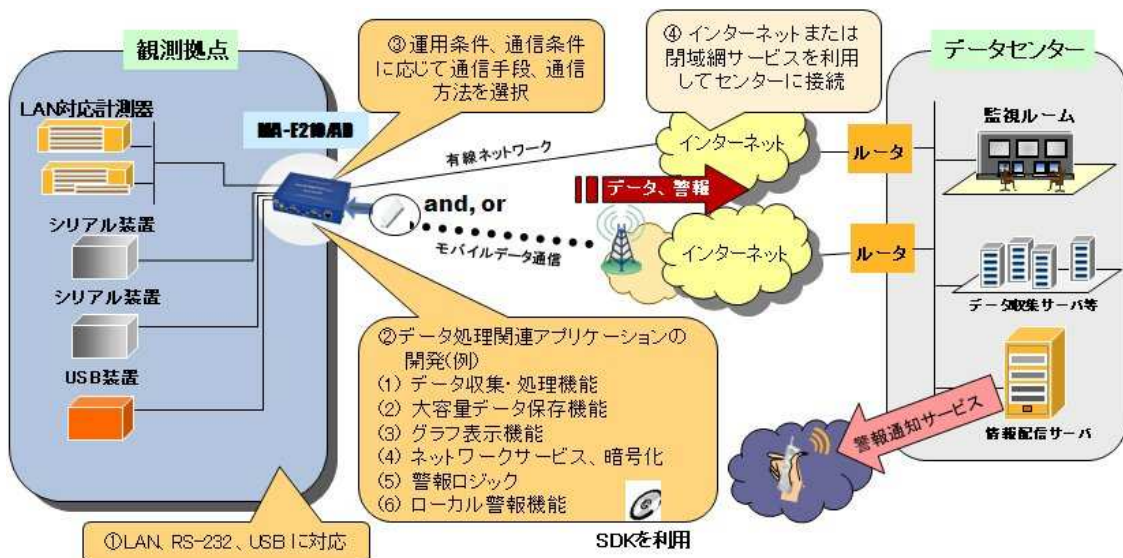
1.2. FutureNet MA-E210 応用例

MA-E210 をベースにソフトウェアを追加することにより次のような応用が可能です。

遠隔監視サーバ

MA-E210 を使って IP ネットワーク経由で遠隔にある装置を監視し、制御することができます。

MA-E210 では、単にデータを送受信するだけでなく、データの変換、暗号化などの処理が可能です。さらに SD メモリカードや USB 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機能の範囲を調整できます。

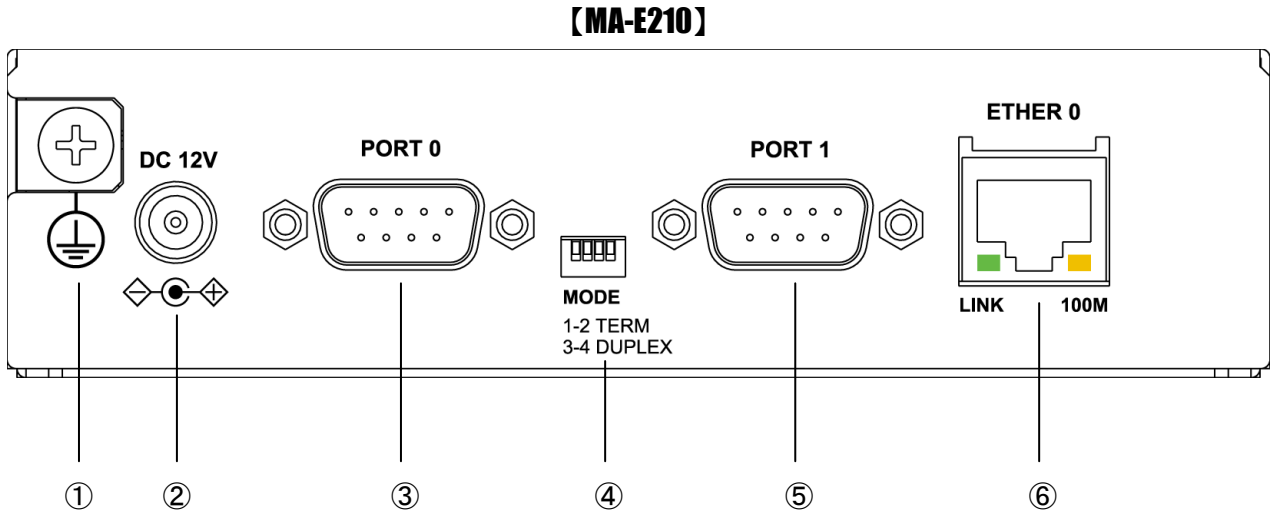


【FutureNet MA-E210 による遠隔監視システム】

1.3. 外観

MA-E210 本体各部の名称は以下のとおりです。

<背面図>



① 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。

② DC 12V 電源コネクタ

製品付属の AC アダプタを接続します。

③ PORT0 ポート

DTE 対応の RS232 ポートが使用可能です。(実装オプションで RS-485 が使用可能です。) 後述するディップ・スイッチの設定により、Linux コンソール用として使用することも可能です。

④ MODE

"PORT0"が RS-232 設定のときは、必ず SW-1~4 をすべて OFF(上側)で使用してください。

"PORT0"が RS-485 設定のときについては、「1.5. RS-485 インターフェース仕様」をご確認ください。



⑤ PORT1 ポート

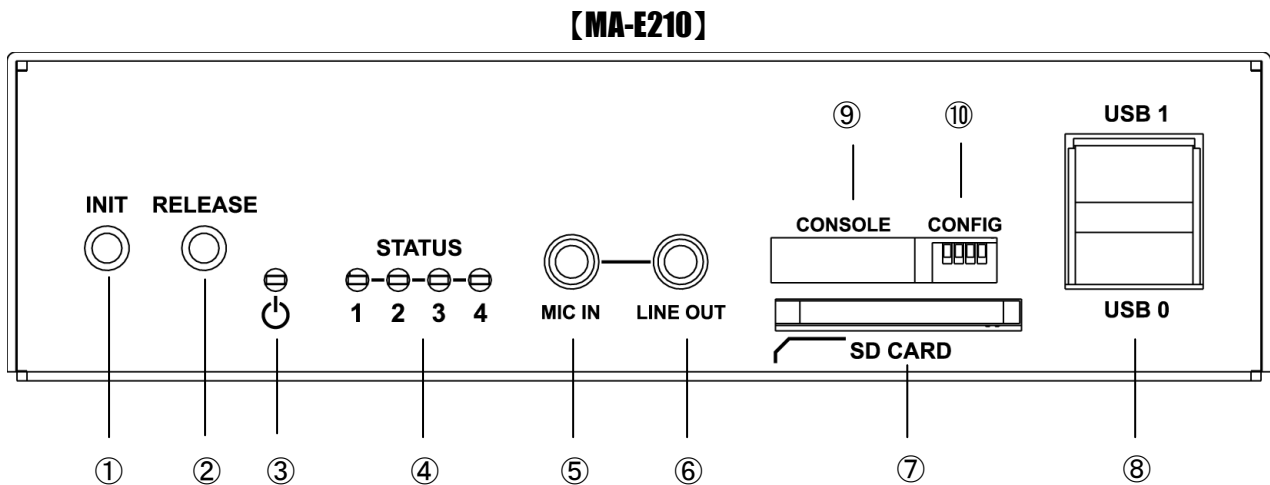
DTE 対応の RS232 ポートが使用可能です。

⑥ Ethernet ポート

10BASE-T/100BASE-TX 対応で、1 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。

<正面図>



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 RELEASE スイッチ

汎用のスイッチとして利用することができます。

③ POWER LED

MA-E210 の電源状態を表示します。

④ STATUS LED

MA-E210 の動作状態等を表示します。設定方法は 1.7 節をご参照ください。

⑤ MIC IN

音声入力を使用可能です。

⑥ LINE OUT

ステレオ音声出力を使用可能です。

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 USB0/USB1 ポート

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

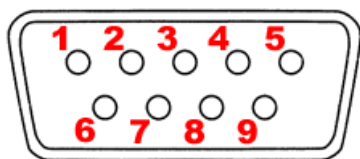
⑨ CONSOLE

FutureNet コンソール・アダプタ(オプション)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。

⑩ 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 1.6 節をご参照下さい。

1.4. RS-232 インターフェース仕様

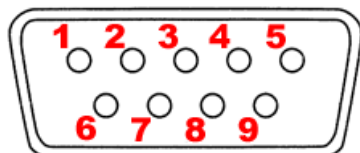


MA-E210にはDTE対応のRS-232ポートが2ポート装備されています。最大ボーレートは、230.4kbpsです。コネクタのピン・アサインは下記の通りです。

番号	信号名	タイプ	説明
1	DCD	I	受信キャリア検出
2	RXD	I	受信データ
3	TXD	O	送信データ
4	DTR	O	データ端末レディ
5	GND	P	信号グラウンド
6	DSR	I	データセットレディ
7	RTS	O	送信要求
8	CTS	I	送信可能
9	RI	I	被呼表示

※ タイプの”I”は入力, ”O”は出力, ”P”は電源を表しています。

1.5. RS-485 インターフェース仕様



MA-E210 は実装オプションで、RS-485 ポートが 1 ポート(PORT0 コネクタ)使用可能です。コネクタのピン・アサイン及びディップ・スイッチは下記の通りです。

(1) Half duplexモード

番号	信号名	タイプ	説明
1	GND	P	信号グラウンド
2	—	—	未使用
3	—	—	未使用
4	Data+	I/O	送受信 Data +
5	Data-	I/O	送受信 Data -
6	—	—	未使用
7	—	—	未使用
8	—	—	未使用
9	—	—	未使用

※ タイプの”I/O”は入出力, ”P”は電源を表しています。

”PORT0”右側に配置されている”MODE”ディップ・スイッチでターミネーション設定とDUPLEXモードを設定します。

”MODE”ディップ・スイッチ			
1	2	3	4
ON/OFF	—	ON	ON

※”MODE”ディップ・スイッチの説明

”MODE”ディップ・スイッチ1 : ターミネーション(120Ω) ON/OFF

”MODE”ディップ・スイッチ2 : 未使用

”MODE”ディップ・スイッチ3 : ON (Half duplex)

”MODE”ディップ・スイッチ4 : ON (Half duplex)

(2) Full duplexモード

番号	信号名	タイプ	説明
1	GND	P	信号グラウンド
2	—	—	未使用
3	—	—	未使用
4	RX+	I	受信 Data +
5	RX-	I	受信 Data -
6	—	—	未使用
7	—	—	未使用
8	TX+	O	送信 Data +
9	TX-	O	送信 Data -

※ タイプの”I”は入力, ”O”は出力, ”P”は電源を表しています。

”PORT0”右側に配置されている”MODE”ディップ・スイッチでターミネーション設定と DUPLEX モードを設定します。

”MODE”ディップ・スイッチ			
1	2	3	4
ON/OFF	ON/OFF	OFF	OFF

※”MODE”ディップ・スイッチの説明

”MODE”ディップ・スイッチ1 : RX 側ターミネーション(120Ω) ON/OFF

”MODE”ディップ・スイッチ2 : TX 側ターミネーション(120Ω) ON/OFF

”MODE”ディップ・スイッチ3 : OFF (Full duplex)

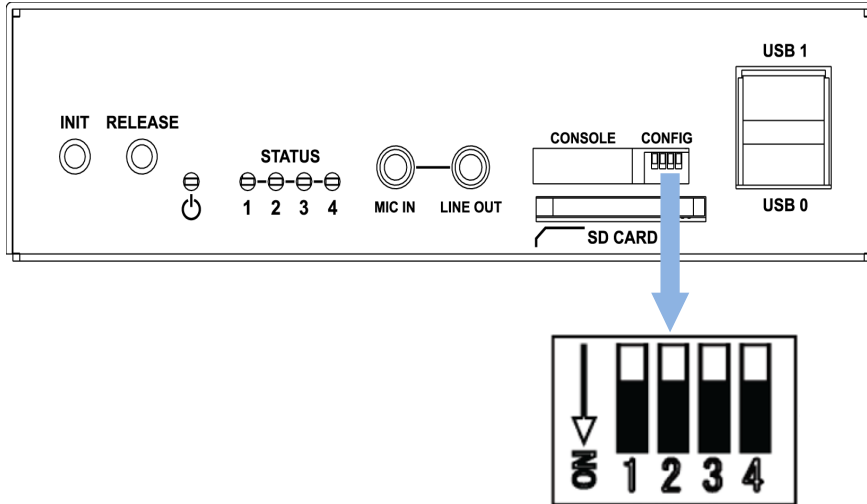
”MODE”ディップ・スイッチ4 : OFF (Full duplex)

1.6. CONFIG の説明

本装置の CONFIG を切り替えることにより、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。
スイッチを下方向にすると ON, 上方向にすると OFF になります。

またスイッチは向かって左から順に 1,2,3,4 の番号が割り当てられています。

<正面図>



CONFIG 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				動作モード
1	2	3	4	
ON	ON	ON	D. C.	TFTP download (ファームウェア更新)
ON	OFF	OFF	D. C.	Linux boot (JFFS2) 'root=/dev/mtdblock4' (デフォルト)
OFF	ON	OFF	D. C.	Linux boot (bootp & NFS) 'root=/dev/nfs ip=dhcp'
ON	ON	OFF	D. C.	Linux boot (USB, sda1) 'root=/dev/sda1'
OFF	OFF	ON	D. C.	Linux boot (SDCard, mmcblk0p1) 'root=/dev/mmcblk0p1'
OFF	OFF	OFF	D. C.	u-boot 'bootargs' 利用

(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。

ディップ・スイッチ 4 により、u-boot, Linux のコンソールを切り替えることができます。

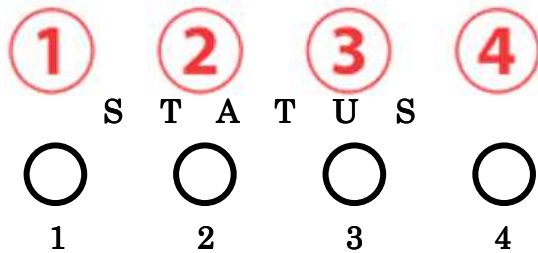
OFF にした場合デバッグ用コンソールポート、ON にした場合は PORT0 が使用されます。

1.7. LED の説明

1.7.1. MA-E210 状態表示 LED

本装置では、**MA-E210** の状態を表示する STATUS LED 1 2 3 4 の 4 つの LED があります。各 LED は 2 色 (red, 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 LED クラスドライバで実装しておりますので、設定により任意の用途での利用が可能です。ここでは説明のため、各 LED を次のように表現します。



1.7.1.1. 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。

電源投入後、Linux が正常に起動後、下記状態で LED の点滅・点灯処理が行われます。

③ red : SDCard アクセス LED, green : CF アクセス LED

④ green : 常に点灯, red : Heartbeat



1.7.1.2. 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。

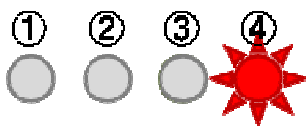
機器の完全停止時

④ green : 点灯



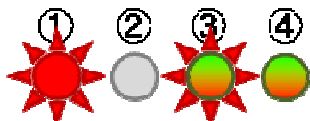
Kernel パニック時

④ red : 点滅 (200msec 毎)



WebUI からファームウェア・アップグレード実行時

- ① red : 点滅(200msec 毎), ③ red/green 交互点灯(500msec 毎),
- ④ green : 点灯, red : Heartbeat = kernel 更新中



- ①③ red/green 交互点灯(500msec 毎)
- ④ green : 点灯, red : Heartbeat = rootfs アップデート時



Init ボタン押しつつ起動した時

- ④ green : 点灯, red : Heartbeat
- ① red/green 交互点灯(200msec 毎) = 起動時



更に5秒間 Init ボタン押し続けた時

- ④ green : 点灯, red : Heartbeat
- ① red 点滅(200msec 点灯/500msec 消灯) = 設定データ初期化



稼働中に Init ボタンを2秒押し続け shutdown を実行した時

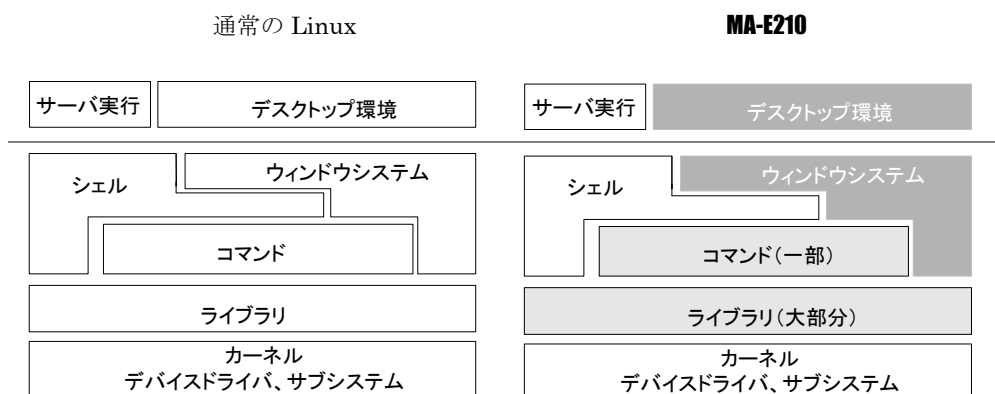
- ④ green : 点灯, red : Heartbeat
- ③ red/green 交互点灯(500msec 毎)



2. MA-E210 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 2.11(NPTL)をサポートしています。Kernel は 2.6.28 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。

Ethernet (eth0) (10/100Base-TX)	192.168.253.253
---------------------------------	-----------------

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、FutureNet コンソール・アダプタ(オプション)を使用することでデバッグ用コンソールポートからログインすることができます。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 ETHER0(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 にオプション品である「FutureNet 開発用コンソール・アダプタ」を接続させる、もしくはデバッグ・スイッチ設定により RS-232 ポート PORT0 をコンソールポートに設定することで、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 FutureNet 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-E210** の NFS 用のファイルシステムを展開することで、**MA-E210** 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

MA-E210 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

```
MAE2xx # shutdown -h now
```

STATUS LED 4 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、STATUS LED 4 のみ点灯したら、電源を切ることができます。

具体的な操作は次の通りです。

(0) 通常稼働時

④ green: 常に点灯, red : Heartbeat



(1) システム・シャットダウン操作

MA-E210 の稼働時に Init ボタンを2秒以上押下すると、シャットダウンが実行され電源を落とせる状態に移行します。(shutdown -h now)

その場合の LED 表示は次の通りです。

④ green : 点灯, red : Heartbeat

③ red/green 交互点灯(500msec 毎)



(2) 電源オフ可能状態

この状態で **MA-E210** の電源をオフにする事が可能です。

④ green : 点灯



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。

設定データ初期化操作

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 LED は下記の様に表示されます。

④ green : 点灯, red : Heartbeat

① red/green 交互点灯(200msec 毎) = 起動時



この状態から更に 5 秒 Init ボタン押下を保持すると、LED 表示は次のように変わります。

④ green : 点灯, red : Heartbeat

① red 点滅(200msec 点灯/500msec 消灯) = 設定データ初期化



この状態で設定データのクリアを受け付けたことを示します。/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-E200 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

MA-E210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:

```
user1@MAE2xx ~ $ cat /proc/version
Linux version 2.6.28.10-Magnolia2 (root@vmware) (gcc version 4.3.4 (Gentoo 4.3.4 p1.1, pie-10.1.5) ) #1 PREEMPT
Mon May 10 16:04:52 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:

```
user1@MAE2xx ~ $ cat /etc/version
MA-E210 firmware version 1.0.0 (Mon May 10 16:00:55 JST 2010)
```

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-E210** 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。

デフォルトではディップ・スイッチは以下のようになっています。



本装置のディップ・スイッチを以下のように変更します。



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 mae2xx {
      hardware ethernet 00:80:6d:XX:XX:XX;      # MA-E210 Mac address
      fixed-address 192.168.252.252;           # MA-E210 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

MA-E210 のディップ・スイッチを以下のように変更します。



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-E200 Series 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
```

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:

```
MAE2xx # 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)

```
MAE2xx # 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

```
MAE2xx # 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。

```
MAE2xx # 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

MAE-E210 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)

```
MAE2xx # iptables -t nat -A POSTROUTING -o ppp0 -s 192.168.253.0/24 -j MASQUERADE
```

- ・設定内容の表示

```
MAE2xx # iptables -t nat -L
```

- ・IP マスカレードを解除する

```
MAE2xx # 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

```
MAE2xx # iptables -P FORWARD DROP
```

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

```
MAE2xx # 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

```
MAE2xx # iptables -L
```

- ・INPUT チェインの設定ルールのみ全て削除する

```
MAE2xx # 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

MA-E210 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。

- DTE:D-sub 9 pin RS-232 × 2 ポート
- 10BASE-T/100BASE-TX × 1 ポート
- USB 2.0 × 2 ポート
- ステレオ・ライン出力 × 1 ポート、マイク入力 × 1 ポート
- SD カードスロット × 1

4.2. シリアルインターフェース (DTE)

DTE ポートのコネクタは D-sub 9 pin オスコネクタ仕様になっています。Linux 上からはシリアルデバイスとしてインターフェースが用意され、以下のように対応しています。

コネクタ	デバイス名
PORT0	/dev/ttymx0
PORT1	/dev/ttymx1

最大ボーレートは 230.4kbps に対応しています。

尚、**MA-E210** は標準でインストールされている RS-232 端末エミュレータ `microcom` で動作確認ができます。

注意事項

アプリケーションでシリアルインターフェースを使用する場合は、`/etc/inittab` のシリアル設定を変更してください。

・外部ターミナルを接続してログインするために使用する場合(デフォルト)

```
# SERIAL CONSOLES
```

```
mxc0:12345:respawn:/sbin/agetty 115200 ttymx0 vt100
```

```
mxc1:12345:respawn:/sbin/agetty 115200 ttymx1 vt100
```

・アプリケーションや PPP のモデム接続でシリアルインターフェースを使用する場合

```
# SERIAL CONSOLES
```

```
#mxc0:12345:respawn:/sbin/agetty 115200 ttymx0 vt100
```

```
#mxc1:12345:respawn:/sbin/agetty 115200 ttymx1 vt100
```

4.3. USB Host

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

4.3.1. USB serial converter

MA-E210 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。

動作確認がされているデバイスは以下のとおりです。

Ratoc	REX-USB60
秋月電子	M-00720
IO DATA	USB-RSAQ2
ELECOM	UC-SGT

※ 完全な動作保証ではありません

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

MA-E210 では、SCSI のストレージ (/dev/sd x) 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

```
MAE2xx ~ # cat /proc/partitions (パーティションの確認)
```

```
major minor #blocks name
```

```
7      0      16244 loop0
31     0       256 mtblock0
31     1       128 mtblock1
31     2       128 mtblock2
31     3      3584 mtblock3
```

```
31      4      28672 mtdblock4
8       0      509695 sda
8       1      509690 sda1
```

```
MAE2xx ~ # mount /dev/sda1 /mnt/usb/ (USB デバイスのマウント)
```

(2) アンマウント

```
MAE2xx ~ # 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
```

```
MAE2xx ~ # umount /dev/usb (USB デバイスのアンマウント)
```

4.4. 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。

使用方法は、LED クラスドライバに準拠していますので、SDK のカーネルソース内のドキュメント /Documentation/leds-class.txt を参照してください。

4.5. 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/psw_init」としてアクセス可能です。

Init ボタンの ON/OFF の状態は、以下の値により表されます。

値	状態
0	OFF
1	ON

通常 pshd_initsw デーモンが状態を監視しており、押下状態を検知した場合

”/sbin/shutdown.sh”スクリプトを呼び出すことにより、本機器を停止します。

(/sbin/shutdown.sh)

```
#!/bin/bash
function ledctrl {
    local leddir='/sys/class/leds/led_'$1
    local mode=$2
    if [ "${mode}" == "on" ]; then
        echo 'default-on' > ${leddir}/trigger
        echo 1 > ${leddir}/brightness
    elif [ "${mode}" == "off" ]; then
        echo 'default-on' > ${leddir}/trigger
    fi
}
```

```

        echo 0 > ${leddir}/brightness
    elif [ "${mode}" == "blink" ]; then
        local ontime=$3
        local offtime=$4
        echo 'timer' > ${leddir}/trigger
        echo $ontime > ${leddir}/delay_on
        echo $offtime > ${leddir}/delay_off
    fi
}
ledctrl g2 blink 500 500
sleep 0.5
ledctrl r2 blink 500 500
/sbin/shutdown -h now

```

4.6. RELEASE ボタン

本装置では、ユーザアプリケーションが RELEASE ボタンを自由に使用することができます。

例えば、ポーリングで状態を監視するプログラムを作成する等により使用することができます。

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

RELEASE ボタンの ON/OFF の状態は、以下の値により表されます。

値	状態
0	OFF
1	ON

```

#!/bin/sh

RELEASE=`cat /proc/driver/psw_eject`
if [ $RELEASE == '0' ]; then
    echo 1 > /sys/class/leds/led_g0/brightness
    echo 0 > /sys/class/leds/led_r0/brightness
else
    echo 1 > /sys/class/leds/led_r0/brightness
    echo 0 > /sys/class/leds/led_g0/brightness
fi

```

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-E200 Series 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。

MA-E210 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。
MA-E210 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。
- **MA-E210** 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 mae2xx_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 MAE2xx_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 MAE2xx_rootfs
# mv MAE2xx_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

MA-E210 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/dhcpd.conf ファイルで行います。

例: /etc/dhcpd.conf

```
# MA-E210
subnet 192.168.252.0 netmask 255.255.255.0 {
  group {
    host mae210 {
      hardware ethernet 00:80:6d:xx:xx:xx;      # MA-E2xx Mac address
      fixed-address 192.168.252.252;             # MA-E2xx IPaddress
      next-server 192.168.252.1;                 # Server IP
      option host-name "mae2xx";                 # 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 mae2xx
- ・ 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-E210 の起動

MA-E210 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

MA-E210 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、ディップ・スイッチを以下のように設定します。



u-boot による変更

MA-E210 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-E210** 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-E200 Series 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。

MA-E210 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。

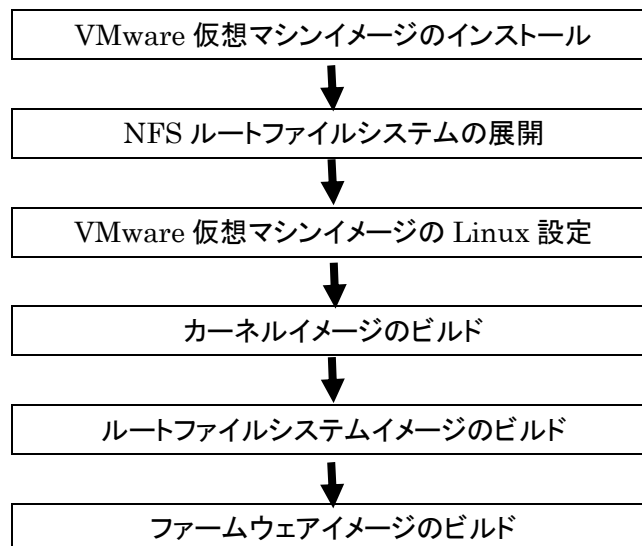
VMware 仮想マシンイメージを利用する場合は、別途 Windows が動作する PC へ VMwarePlayer をインストールし、VMware 仮想マシンイメージを設定する必要があります。

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。



6.4. 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 mae2xx_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 MAE2xx_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 MAE2xx_rootfs
drwxr-xr-x  2 root root        4096 Feb 13  2009 rootfs
# mv MAE2xx_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

MA-E210 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-E2xx
host mae2xx {
    hardware ethernet 00:80:6d:xx:xx:xx;      # MA-E210 の MAC アドレス
    fixed-address 192.168.252.252;           # MA-E210 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 "mae2xx.img";                  # ファームウェアファイル名
    option host-name "mae2xx";              # 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 "mae2xx.img"
- ・ クライアント名は "mae2xx"
- ・ 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 pjxvf linux-2.6.28.x_mae2xx.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、config_xxx_default 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-2.6.28.x_MAE2xx
# make mrproper
# cp config_xxx_default .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-2.6.28.x_MAE2xx/arch/arm/boot/

例:

```
# make uImage
# ls /var/export/rootfs/usr/src/linux-2.6.28.x_MAE2xx/arch/arm/boot
Image Makefile bootp compressed install.sh uImage 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-E2xx**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することが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 mae210_mtdroot_xxxx.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 mae210_mtdroot_xxxx.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-2.6.28.x_MAE2xx
# 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 -ar 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/arm-linux-gnueabi/4.3.4
cp -a ../usr_backup/lib/gcc/arm-linux-gnueabi/4.3.4/libgcc_s.so* usr/lib/gcc/arm-linux-gnueabi/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「mae2xx.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 mae2xx.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-2.6.28.x_MAE2xx
IMAGE_FILE=ulmage
IMAGE_KERNEL=${LINUXROOT}/arch/arm/boot/${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=mae2xx.img

echo "cd embedded_glibc and make rootfs"
rm $ROOTFS_FILE
mkfs.jffs2 -e 0x20000 --pad=0x1c00000 -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-E200 Series 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。

MA-E210 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、**MA-E210** 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

MA-E210 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。

MA-E210 の SD カードのデバイス名は /dev/mmcblk0 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/mmcblk0p1 となります。

例:

```
MAE2xx ~ # ls /dev/mmcblk*  
/dev/mmcblk0    /dev/mmcblk0p1
```

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:

```
MAE2xx ~ # fdisk /dev/mmcblk0
```

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

```
Disk /dev/mmcblk0: 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/mmcblk0p1		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/mmcblk0p1		1	976	1967489+	6	FAT16

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

```
Selected partition 1
```

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

```
Disk /dev/mmcblk0: 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/mmcblk0: 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/mmcblk0p1		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。

MA-E210 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:

```
MAE2xx ~ # mkfs. ext3 /dev/mmcblk0p1
```

ステップ2 SD カード用ルートファイルシステムの展開

MA-E210 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。

MA-E210 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:

```
MAE2xx ~# mount -t ext3 /dev/mmcblk0p1 /mnt/sd
MAE2xx ~# ls /mnt/sd
lost+found2

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

/MA810_rootfs に展開されたファイルを "/" へ移動します。

例:

```
MAE2xx sd # ls
MA810_rootfs lost+found

MAE2xx sd # mv MAE2xx_rootfs/* ./
MAE2xx sd # rmdir MAE2xx_rootfs

MAE2xx 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

MA-E210 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。

□ ディップ・スイッチの変更

MA-E210 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。SD カードで起動する場合、ディップ・スイッチを以下のように設定します。



8. 仕様一覧

8.1. MA-E210 の仕様

製品名		FutureNet MA-E210
CPU		Freescale™ i.MX353®プロセッサ 532MHz (ARM11 コア)
Flash ROM		32Mbyte
RAM		128Mbyte
インターフェース	イーサネット	10BASE-T/100BASE-TX × 1 ポート RJ-45 コネクタ、Auto Negotiation、Full/Half Duplex、Auto MDI/MDIX
	WAN	USB データ通信アダプタ
	シリアル	RS-232 (DTE) D-SUB9ピン オスコネクタ、最大 230.4kbps × 2 ※ 実装オプション: RS-485、基板上10ピンコネクタへの変更可
	USB	USB 2.0 ホスト × 2 ポート (TYPE-A コネクタ)、 ※ 内部インターフェース × 2 ポートと共用
	アナログ入力	なし
	デジタル入出力	なし
グラフィクス (実装オプション)	信号形式、コネクタ形状	LCD 40ピン内部コネクタ ※ 基板上 (予定)
	フレームサイズ	640 × 480
SD カードスロット		SDHC対応 SDカード 1スロット ※ SDメモ리카ードに対応
コンソールポート		Linux コンソール用シリアル (TTL レベル) 6ピンコネクタ ※ アダプタ別売
オーディオ		ステレオ・ライン出力、マイク入力 ※ φ 3.5 ジャック
LED 表示		システム: Power × 1, Status × 4, Ethernet: Link/Active × 1, Speed × 1
基本ソフトウェア	OS	Linux (Kernel 2.6)
	実行時ライブラリ	glibc 2.11(NPTL)
	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、シリアル/Ethernet 変換
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
サイズ・重量	外観寸法(突起物を除く)	139.2mm(W) x 101.5mm(D) x 36.2mm(H)
	本体重量	約 500g
	AC アダプタ	120g
環境	使用電源、電源形状	DC +12V 1A (標準) AC アダプタ
	消費電力 (USB 接続なし)	約 3W

	本体動作温度条件	-20°C~60°C
	AC アダプタ動作温度条件	-10°C~40°C
	動作湿度条件	20%~85%(結露なきこと)
	保存温度	-20°C~60°C、25~90%(結露なきこと)
	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 5 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-E210 製品サポートページ

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

9. MA-E210/AD-72 について

9.1. AD モデルの概要

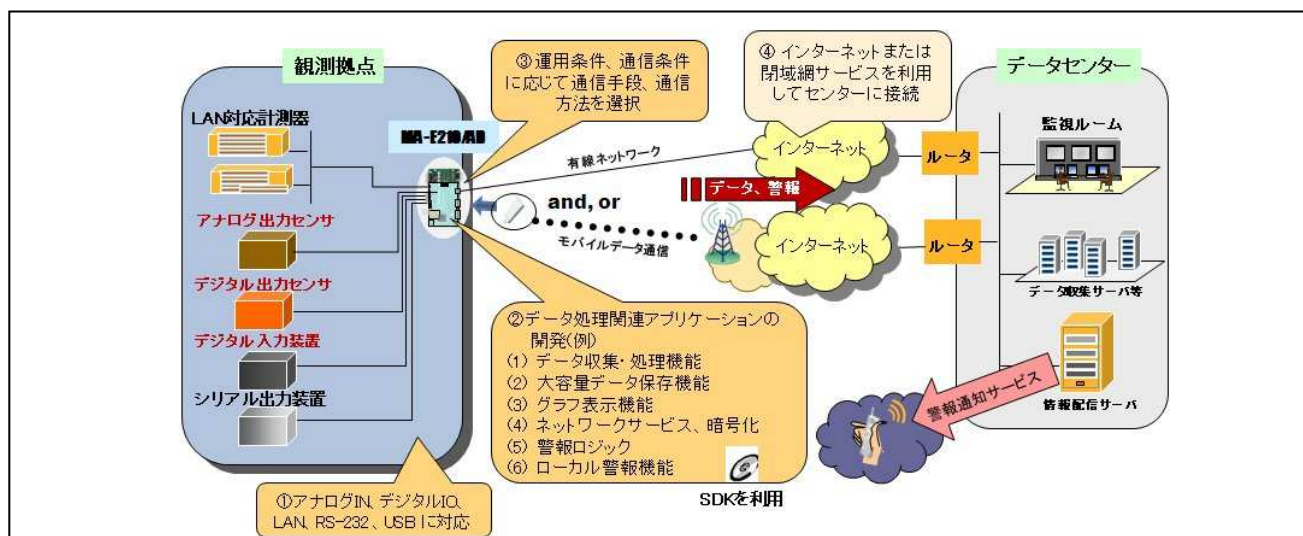
FutureNet **MA-E210/AD-72** は、**MA-E210** にアナログ入力 8 ポート、デジタル入力 32 ポート、デジタル出力 32 ポートの拡張インターフェースボードを搭載したモデルです。

アナログ入力には、温度センサや風力計などのアナログ出力を持つ装置を接続できます。ソフトウェアは Linux で広く使われている hwmon、lm_sensors に対応しており、移植性の高いアプリケーションの開発が可能です。lm_sensors で取得したデータは RRDTOOL などを利用して Linux の標準的な方法でグラフ化などの処理が簡単におこなえます。

デジタル入力は、雨量計などのデジタルパルス出力を持つ装置や ON/OFF の状態出力を持つ装置の接続に利用できます。デジタル出力は警報灯の制御等に利用できます。デジタル入出力は/proc ファイルシステムのインターフェースに対応しています。



【図：FutureNet MA-E210/AD-72 の外観】



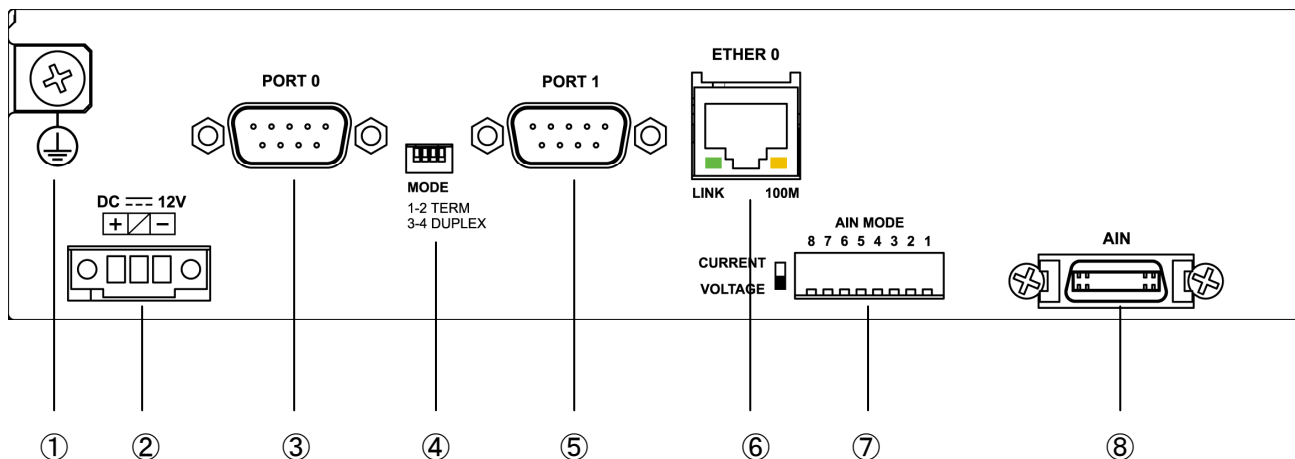
【図：FutureNet MA-E210/AD-72 によるセンサ監視システムの例】 ※ 各種センサを直接収容可能

9.2. 外観

MA-E210/AD-72 本体各部の名称は以下のとおりです。

<背面図>

【MA-E210/AD-72】



① 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。

② DC 12V ねじ止め式 3P コネクタ

DC12V±10%の DC 電源を接続します。(コネクタは Phoenix contact 社製 : MC1.5/3-GF-3.5THT)

③ PORT0 ポート

DTE 対応の RS232 ポートが使用可能です。(実装オプションで RS-485 が使用可能です。) 後述するディップ・スイッチの設定により、Linux コンソール用として使用することも可能です。

④ MODE

"PORT0"が RS-232 設定のときは、必ず SW-1~4 をすべて OFF(上側)で使用してください。

"PORT0"が RS-485 設定のときについては、「1.5. RS-485 インターフェース仕様」をご確認ください。



⑤ PORT1 ポート

DTE 対応の RS232 ポートが使用可能です。

⑥ Ethernet ポート

10BASE-T/100BASE-TX 対応で、1 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。

⑦ AIN MODE

アナログ差動入力 of 電圧／電流モードの切り替えを行います。

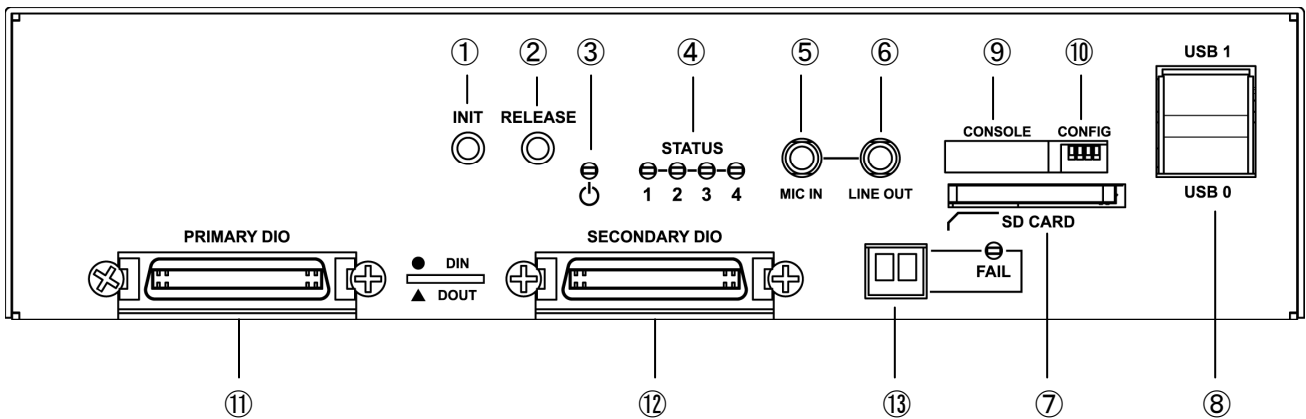
アナログ入力を電流モードで使用する場合はスイッチを CURRENT ポジションに設定します。電圧モードで使用する場合は VOLTAGE ポジションに設定します。

⑧ AIN

アナログ差動入力 that 8 チャンネル使用可能です。

<正面図>

【MA-E210/AD-72】



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 RELEASE スイッチ

汎用のスイッチとして利用することができます。

③ **POWER LED**

MA-E210/AD-72 の電源状態を表示します。

④ **STATUS LED**

MA-E210/AD-72 の動作状態等を表示します。設定方法は 1.7 節をご参照ください。

⑤ **MIC IN**

音声入力を使用可能です。

⑥ **LINE OUT**

ステレオ音声出力を使用可能です。

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おりません。

⑧ **USB0/USB1 ポート**

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

⑨ **CONSOLE**

FutureNet コンソール・アダプタ(オプション)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。

⑩ **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 1.6 節をご参照下さい。

⑪ **PRIMARY DIO**

DI 接点入力が 16 チャンネル(CH0~CH15)、DO 接点出力が 16 チャンネル(CH0~CH15)使用できます。

⑫ **SECONDARY DIO**

DI 接点入力が 16 チャンネル(CH16~CH31)、DO 接点出力が 16 チャンネル(CH16~CH31)使用できます。

⑬ **FAIL**

フェイル接点出力が 1 チャンネル使用できます。

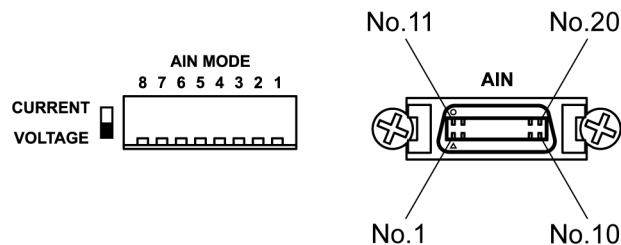
9.3. AIN インターフェースの使い方

9.3.1. 仕様

AI アナログ入力	マルチプレクサ式逐次変換型(SAR)
ポート数	差動入力 8CH
絶対最大入力	DC±15V
入力レンジ	電圧モード 0V~+5V, 0V~+10V, -5V~+5V, -10V~+10V (ソフトウェアによる切り替え) 電流モード 0mA~20mA (差動入力対応・スイッチ切り替え)
許容コモンモード電圧	DC±5V 以下(0V~5V レンジ) DC±1V 以下(-10V~10V レンジ)
入力インピーダンス	電圧入力モード 2MΩ 以上(差動入力時) 電流入力時 250Ω
分解能	12bit
変換精度	±0.8%(フルスケール)
変換時間	100 μ sec
絶縁方式	デジタルアイソレータ絶縁
絶縁耐圧	DC500V 1分間, 外部端子~内部回路間
コネクタ	ハーフピッチ 20ピン コネクタ 3M 社製 MDR タイプ : 10220-52A2PL

9.3.2. AIN コネクタとピン配置

<AIN コネクタピン配置図>



(1) AIN コネクタピン配置

PIN No.	名称	機能	PIN No.	名称	機能
1	AIN COM	アナログコモン*	11	AIN COM	アナログコモン*
2	AIN0 (+)	アナログ差動入力	12	AIN4 (+)	アナログ差動入力
3	AIN0 (-)	CH0 (hwmon0/in0)	13	AIN4 (-)	CH4 (hwmon1/in0)
4	AIN1 (+)	アナログ差動入力	14	AIN5 (+)	アナログ差動入力
5	AIN1 (-)	CH1 (hwmon0/in1)	15	AIN5 (-)	CH5 (hwmon1/in1)
6	AIN2 (+)	アナログ差動入力	16	AIN6 (+)	アナログ差動入力
7	AIN2 (-)	CH2 (hwmon0/in2)	17	AIN6 (-)	CH6 (hwmon1/in2)
8	AIN3 (+)	アナログ差動入力	18	AIN7 (+)	アナログ差動入力
9	AIN3 (-)	CH3 (hwmon0/in3)	19	AIN7 (-)	CH7 (hwmon1/in3)
10	NC	未使用	20	NC	未使用

*Note: Pin No.1 と No.11 のアナログコモンは内部で接続されています。

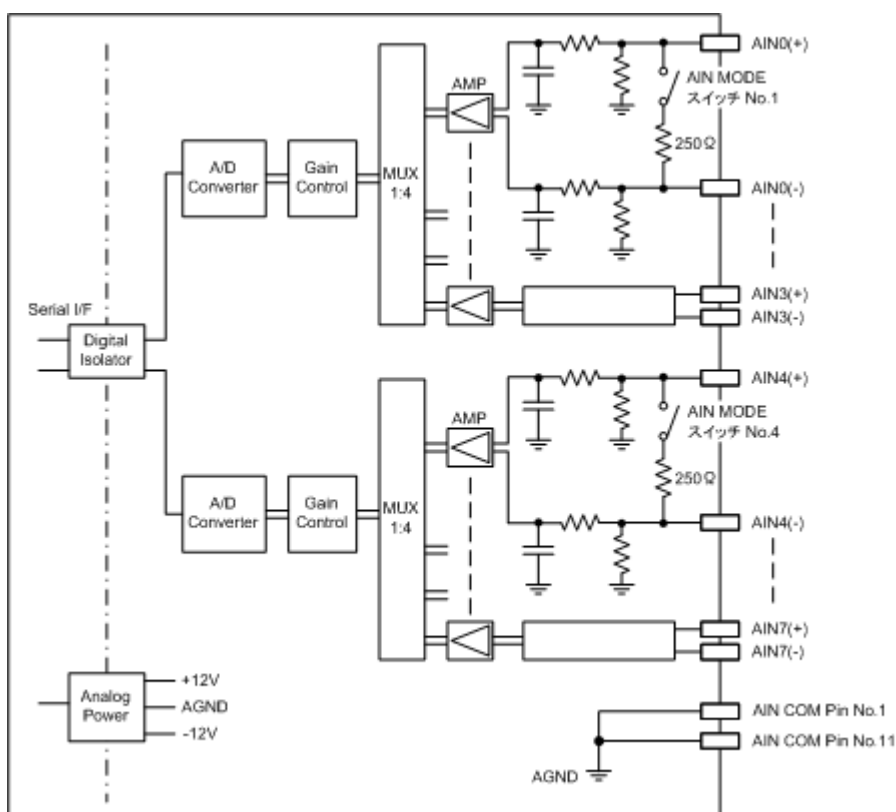
(2) AIN MODE スイッチ設定

アナログ差動入力の電圧／電流モードの切り替えを行います。アナログ入力を電流モードで使用する場合、AIN MODE スイッチを CURRENT ポジションに切り替えます。電流モードに切り替えると AINn(+) ~ AINn(-)間に 250Ω の抵抗が接続されます。電流モードを使用する場合は、アナログ入力レンジを 0~5V に設定してください。

AIN MODE スイッチ設定

SW No.	AIN チャンネル	スイッチポジション	
		VOLTAGE	CURRENT
1	CH0	電圧モード	電流モード
2	CH1		
3	CH2		
4	CH3		
5	CH4		
6	CH5		
7	CH6		
8	CH7		

9.3.3. ハードウェアの接続方法



9.3.4. アプリケーションの作成方法

アナログ入力(以後 AIN とします)インターフェースのプログラミング方法について記載します。

AIN のアクセスは、Linux の"/sys"ディレクトリ内のファイルアクセスにより行います。

AIN への設定はファイルへの書き込みにより、AIN からのデータ取得はファイルの読み込みにより行います。

下表の名称「AIN0～AIN7」は、前述の「AIN コネクタとピン配置」の「(1)AIN コネクタピン配置」一覧表に準じます。この一覧表の機能欄には、各チャンネルに割り当てるディレクトリとファイル名との対応関係が記載されています。

(1) AIN チャンネルに割り当てられたディレクトリとファイルの対応関係

・AIN0～AIN3

AIN0～AIN3 のファイルは、「/sys/class/hwmon/hwmon0/device/」ディレクトリに割り当てられている。

名称	ファイル
AIN0	in0_input, in0_input_raw, in0_max, in0_min, in0_rang
AIN1	in1_input, in1_input_raw, in1_max, in1_min, in1_rang
AIN2	in2_input, in2_input_raw, in2_max, in2_min, in2_rang
AIN3	in3_input, in3_input_raw, in3_max, in3_min, in3_rang

・AIN4～AIN7

AIN4～AIN7 のファイルは、「/sys/class/hwmon/hwmon1/device/」ディレクトリに割り当てられている。

名称	ファイル
AIN4	in0_input, in0_input_raw, in0_max, in0_min, in0_rang
AIN5	in1_input, in1_input_raw, in1_max, in1_min, in1_rang
AIN6	in2_input, in2_input_raw, in2_max, in2_min, in2_rang
AIN7	in3_input, in3_input_raw, in3_max, in3_min, in3_rang

・内蔵温度センサ

内蔵温度センサのファイルは、「/sys/class/hwmon/hwmon2/device/」ディレクトリに割り当てられている。

名称	ファイル
内蔵温度センサ	temp1_input, temp1_max, temp1_min, temp1_max_alarm, temp1_min_alarm, temp1_crit, temp1_crit_alarm

[例] AIN0～AIN3 のファイル 「/sys/class/hwmon/hwmon0/device/」ディレクトリ

```
MAE2xx din # cd /sys/class/hwmon/hwmon0/device/
MAE2xx device # ls
driver          in0_range      in2_input      in3_input_raw  power
hwmon           in1_input      in2_input_raw  in3_max        ranges
in0_input       in1_input_raw  in2_max        in3_min        subsystem
in0_input_raw  in1_max        in2_min        in3_range      uevent
in0_max         in1_min        in2_range      modalias
in0_min         in1_range      in3_input      name
```

(2)使用方法

・AIN

下表の in*_xxxxx の*には 0 から 3 の数字が対応します。

ファイル	意味
in*_input	<p>入力電圧 (mV 表示, Kernel で計算)</p> <p>[例] 乾電池を接続</p> <pre># cat in0_input 1604</pre>
in*_input_raw	<p>入力(AD コンバータ出力, 換算なし)</p> <p>10 進数, 16 進数 で返します。</p> <p>AD コンバータ出力値を生で取得したい場合に利用します。</p> <p>[例] 乾電池を接続</p> <pre># cat in0_input_raw 10496 0x2900</pre>
in*_max	<p>入力電圧 最大値 (mV 表示)</p> <p>[例]</p> <pre># cat in0_max 5000</pre>
in*_min	<p>入力電圧 最小値 (mV 表示)</p> <p>[例]</p> <pre># cat in0_min -5000</pre>
in*_rang	<p>AD コンバータ 入力電圧レンジ指定</p> <p>設定する値と入力電圧レンジの関係は下記の通りです。</p> <p>0: -5V to +5V</p> <p>1: -10V to +10V</p> <p>2: 0V to +5V</p> <p>3: 0V to +10V</p> <p>[例] 現在のレンジ取得</p> <pre># cat in0_range [UNI:0 GAIN:0] -5V to +5V</pre> <p>[例] レンジを変更</p> <pre># echo 2 > in0_range</pre> <div style="border: 1px dashed black; padding: 5px; margin-top: 10px;"> <p>※ ranges というファイルで確認できます。</p> <pre># cat ranges 0: [UNI:0 GAIN:0] -5V to +5V 1: [UNI:0 GAIN:1] -10V to +10V 2: [UNI:1 GAIN:0] 0V to +5V 3: [UNI:1 GAIN:1] 0V to +10V</pre> </div>

・内蔵温度センサ用

ファイル	意味
templ_input	温度表示 (mV 表示) [例] # cat templ_input 38500
templ_max	温度上限アラーム設定値 (mV 表示) 64.0°C (default) [例] # cat templ_max 64000
templ_min	温度下限アラーム設定値 (mV 表示) 10°C (default) [例] # cat templ_min 10000
templ_max_alarm	温度上限アラーム状態表示 [例] # cat templ_max_alarm 0
templ_min_alarm	低温下限アラーム状態表示 [例] # cat templ_min_alarm 0
templ_crit	高温クリティカルアラーム設定値 (mV 表示) 80.0°C (default) [例] # cat templ_crit 80000
templ_crit_alarm	高温クリティカルアラーム状態表示 [例] # cat templ_crit_alarm 0

9.4. DIO インターフェースの使い方

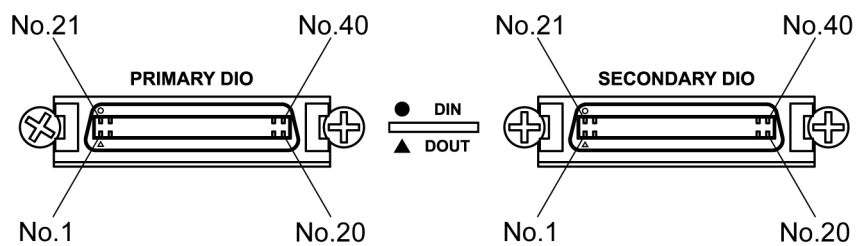
9.4.1. 仕様

DI 接点入力	
ポート数	Primary DI 16CH (CH0~CH15) Secondary DI 16CH (CH16~CH31)
入力信号	DC電圧
コモン	4CH/コモン (マイナスコモン・プラスコモン共用)
入力電圧	DC 24V±10%
入力閾値	ON : DC 20V 以上 OFF : DC 5V 以下
入力電流	約 2mA
入力インピーダンス	約 12kΩ
パルス応答時間	25 μs (Typical)
入力フィルタ	Through/50 μs/1ms/20ms (コモン毎に設定)
ソフト割り込み	チャンネル毎に設定 (立ち上がり・立ち下がリエッジ選択可)
絶縁方式	フォトカプラ絶縁
絶縁耐圧	DC500V 1分間, 外部端子~内部回路間
パルスカウント機能	16CH (Primary DI CH0~CH7, Secondary DI CH16~CH23)
パルスカウンタ	16bit ハードウェア アップカウンタ(32bit ソフトウェア アップカウンタ) コンペアマッチング割り込み/オーバフローフラグ
パルス周波数	20kHz 以下
DO 接点出力	
ポート数	Primary DO 16CH (CH0~CH15) Secondary DO 16CH (CH16~CH31)
出力信号	MOS FET 出力(シンク)
コモン	8CH/コモン
負荷電圧	DC 24V±10%
負荷電流	100mA/CH (ただしコモン毎最大 500mA まで)
応答時間	50 μs (Typical)
ON 電圧	DC 0.5V 以下
OFF 時漏洩電流	0.1mA 以下
保護機能	アクティブクランプ 過電流保護
フェイル時動作	保持/プリセット出力 (ソフトウェアの実装による)
絶縁方式	フォトカプラ絶縁

	絶縁耐圧	DC500V 1分間, 外部端子～内部回路間
	外部供給電源	DC 24V±10%, 20mA/コモン
DI/DO コネクタ		
	Primary channel	ハーフピッチ 40ピン コネクタ (Primary DI CH0～CH15, Primary DO CH0～CH15) 3M 社製 MDR タイプ : 10240-52A2PL
	Secondary channel	ハーフピッチ 40ピン コネクタ (Secondary DI CH16～CH31, Secondary DO CH16～CH31) 3M 社製 MDR タイプ : 10240-52A2PL

9.4.2. DIO コネクタとピン配置

<DIO コネクタピン配置図>



(1)PRIMARY DIO コネクタピン配置

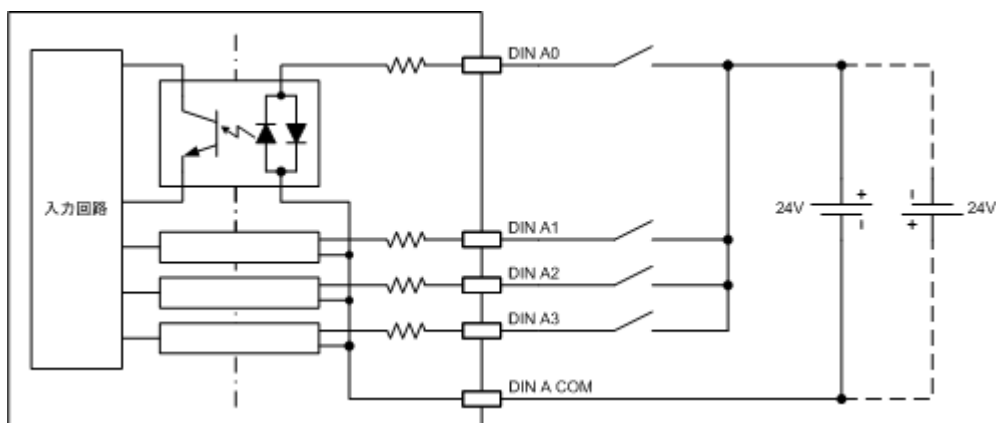
△列 No.	接点出力 グループ	接点出力 ピン名称	機能	○列 No.	接点入力 グループ	接点入力 ピン名称	機能
1	A	DOUT A +24V	+24V 電源 A	21	A	DIN A COM	コモン A
2		DOUT A0	dout/val00	22		DIN A0	din/port00
3		DOUT A1	dout/val01	23		DIN A1	din/port01
4		DOUT A2	dout/val02	24		DIN A2	din/port02
5		DOUT A3	dout/val03	25		DIN A3	din/port03
6		DOUT A COM	コモン A	26	B	DIN B COM	コモン B
7		DOUT A4	dout/val04	27		DIN B0	din/port04
8		DOUT A5	dout/val05	28		DIN B1	din/port05
9		DOUT A6	dout/val06	29		DIN B2	din/port06
10		DOUT A7	dout/val07	30		DIN B3	din/port07
11	B	DOUT B +24V	+24V 電源 B	31	C	DIN C COM	コモン C
12		DOUT B0	dout/val08	32		DIN C0	din/port08
13		DOUT B1	dout/val09	33		DIN C1	din/port09
14		DOUT B2	dout/val10	34		DIN C2	din/port10
15		DOUT B3	dout/val11	35		DIN C3	din/port11
16		DOUT B COM	コモン B	36	D	DIN D COM	コモン D
17		DOUT B4	dout/val12	37		DIN D0	din/port12
18		DOUT B5	dout/val13	38		DIN D1	din/port13
19		DOUT B6	dout/val14	39		DIN D2	din/port14
20		DOUT B7	dout/val15	40		DIN D3	din/port15

(2)SECONDARY DIO コネクタピン配置

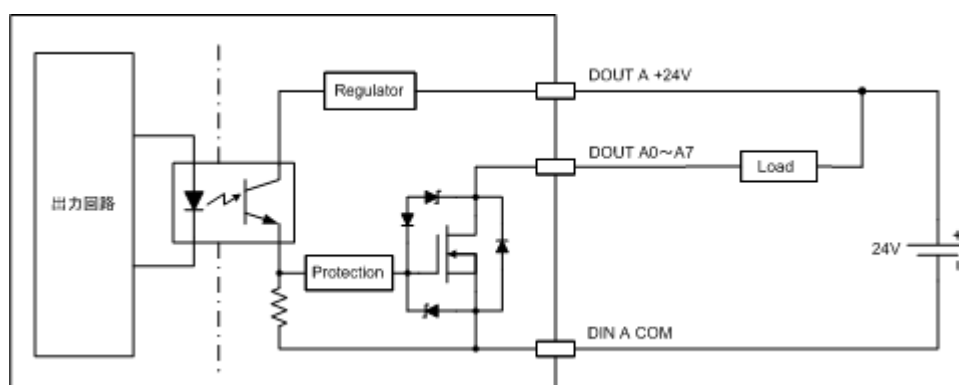
△列 No.	接点出力 グループ	接点出力 ピン名称	機能	○列 No.	接点入力 グループ	接点入力 ピン名称	機能
1	C	DOUT C +24V	+24V 電源 C	21	E	DIN E COM	コモン A
2		DOUT C0	dout/val16	22		DIN E0	din/port16
3		DOUT C1	dout/val17	23		DIN E1	din/port17
4		DOUT C2	dout/val18	24		DIN E2	din/port18
5		DOUT C3	dout/val19	25		DIN E3	din/port19
6		DOUT C COM	コモン C	26	F	DIN F COM	コモン B
7		DOUT C4	dout/val20	27		DIN F0	din/port20
8		DOUT C5	dout/val21	28		DIN F1	din/port21
9		DOUT C6	dout/val22	29		DIN F2	din/port22
10		DOUT C7	dout/val23	30		DIN F3	din/port23
11	D	DOUT D +24V	+24V 電源 D	31	G	DIN G COM	コモン C
12		DOUT D0	dout/val24	32		DIN G0	din/port24
13		DOUT D1	dout/val25	33		DIN G1	din/port25
14		DOUT D2	dout/val26	34		DIN G2	din/port26
15		DOUT D3	dout/val27	35		DIN G3	din/port27
16		DOUT D COM	コモン D	36	H	DIN H COM	コモン D
17		DOUT D4	dout/val28	37		DIN H0	din/port28
18		DOUT D5	dout/val29	38		DIN H1	din/port29
19		DOUT D6	dout/val30	39		DIN H2	din/port30
20		DOUT D7	dout/val31	40		DIN H3	din/port31

9.4.3. ハードウェアの接続方法

(1)DIN



(2)DOUT



9.4.4. アプリケーションの作成方法

デジタル入力/出力(以後 DIN/DOUT とします)インターフェースのプログラミング方法について記載します。

DIN/DOUT のアクセスは、Linux の"/proc"ディレクトリ内のファイルアクセスにより行います。

DIN/DOUT への設定はファイルへの書き込みにより、DIN/DOUT からのデータ取得はファイルの読み込みにより行います。

下記の名称「DOUT A0~DOUT D7、DIN A0~DIN H3」は、前述の「DIO コネクタとピン配置」の「(1)PRIMARY DIO コネクタピン配置」、「(2)SECONDARY DIO コネクタピン配置」一覧表に準じます。この一覧表の機能欄には、各チャンネルに割り当てるディレクトリとファイル名との対応関係が記載されています。

9.4.4.1. DIN

(1)DIN チャンネルに割り当てられたディレクトリとファイルの対応関係

DIN チャンネルは 8 つのグループに分割されています。各グループ単位に後述する"filter"値を設定することができます。

最初に DIN チャンネルとディレクトリの対応関係を示します。

・DINA

接点入力 グループ	接点入力 ピン名称	ディレクトリ(またはファイル)
A		filter (ファイル)
	DINA0	/proc/driver/din/group0/port00 (または/proc/driver/din/port00)
	DINA1	/proc/driver/din/group0/port01 (または/proc/driver/din/port01)
	DINA2	/proc/driver/din/group0/port02 (または/proc/driver/din/port02)
	DINA3	/proc/driver/din/group0/port03 (または/proc/driver/din/port03)

・DINB

接点入力 グループ	接点入力 ピン名称	ディレクトリ(またはファイル)
B		filter (ファイル)
	DINB0	/proc/driver/din/group1/port04 (または/proc/driver/din/port04)
	DINB1	/proc/driver/din/group1/port05 (または/proc/driver/din/port05)
	DINB2	/proc/driver/din/group1/port06 (または/proc/driver/din/port06)
	DINB3	/proc/driver/din/group1/port07 (または/proc/driver/din/port07)

・DINC

接点入力 グループ	接点入力 ピン名称	ディレクトリ(またはファイル)
C		filter (ファイル)
	DINC0	/proc/driver/din/group2/port08 (または/proc/driver/din/port08)
	DINC1	/proc/driver/din/group2/port09 (または/proc/driver/din/port09)
	DINC2	/proc/driver/din/group2/port10 (または/proc/driver/din/port10)
	DINC3	/proc/driver/din/group2/port11 (または/proc/driver/din/port11)

・DIND

接点入力グループ	接点入力ピン名称	ディレクトリ(またはファイル)
D		filter (ファイル)
	DIND0	/proc/driver/din/group3/port12 (または/proc/driver/din/port12)
	DIND1	/proc/driver/din/group3/port13 (または/proc/driver/din/port13)
	DIND2	/proc/driver/din/group3/port14 (または/proc/driver/din/port14)
	DIND3	/proc/driver/din/group3/port15 (または/proc/driver/din/port15)

・DINE

接点入力グループ	接点入力ピン名称	ディレクトリ(またはファイル)
E		filter (ファイル)
	DINE0	/proc/driver/din/group4/port16 (または/proc/driver/din/port16)
	DINE1	/proc/driver/din/group4/port17 (または/proc/driver/din/port17)
	DINE2	/proc/driver/din/group4/port18 (または/proc/driver/din/port18)
	DINE3	/proc/driver/din/group4/port19 (または/proc/driver/din/port19)

・DINF

接点入力グループ	接点入力ピン名称	ディレクトリ(またはファイル)
F		filter (ファイル)
	DINF0	/proc/driver/din/group5/port20 (または/proc/driver/din/port20)
	DINF1	/proc/driver/din/group5/port21 (または/proc/driver/din/port21)
	DINF2	/proc/driver/din/group5/port22 (または/proc/driver/din/port22)
	DINF3	/proc/driver/din/group5/port23 (または/proc/driver/din/port23)

・DING

接点入力グループ	接点入力ピン名称	ディレクトリ(またはファイル)
G		filter (ファイル)
	DING0	/proc/driver/din/group6/port24 (または/proc/driver/din/port24)
	DING1	/proc/driver/din/group6/port25 (または/proc/driver/din/port25)
	DING2	/proc/driver/din/group6/port26 (または/proc/driver/din/port26)
	DING3	/proc/driver/din/group6/port27 (または/proc/driver/din/port27)

・DINH

接点入力 グループ	接点入力 ピン名称	ディレクトリ(またはファイル)
H		filter (ファイル)
	DINF0	/proc/driver/din/group7/port28 (または/proc/driver/din/port28)
	DINF1	/proc/driver/din/group7/port29 (または/proc/driver/din/port29)
	DINF2	/proc/driver/din/group7/port30 (または/proc/driver/din/port30)
	DINF3	/proc/driver/din/group7/port31 (または/proc/driver/din/port31)

[例] DIN のディレクトリ 「/proc/driver/din/」ディレクトリ

```

MAE2xx group0 # cd /proc/driver/din/group0
AE2xx group0 # ls -la
-rw-r--r--  1 root  root           0 Jun  4 15:36 filter
dr-xr-xr-x  2 root  root           0 Jun  4 15:36 port00
dr-xr-xr-x  2 root  root           0 Jun  4 15:36 port01
dr-xr-xr-x  2 root  root           0 Jun  4 15:36 port02
dr-xr-xr-x  2 root  root           0 Jun  4 15:36 port03

MAE2xx group0 # cd /proc/driver/din
MAE2xx din # ls
all  group4 port01 port06 port11 port16 port21 port26 port31
group0 group5 port02 port07 port12 port17 port22 port27
group1 group6 port03 port08 port13 port18 port23 port28
group2 group7 port04 port09 port14 port19 port24 port29
group3 port00 port05 port10 port15 port20 port25 port30

```

次に、DIN チャンネルとファイルの対応関係を示します。

各 port##(##は数字)ディレクトリには、下記の例に示すファイルが存在します。これらのファイルへの読み書きでプログラミングを行います。

[例] DIN のディレクトリ 「/proc/driver/din/group0/port00」ディレクトリ

```
MAE2xx din # cd group0/port00
MAE2xx port00 # ls -la
-rw-r--r--  1 root  root           0 Jun  4 15:01 counter_ctrl
-rw-r--r--  1 root  root           0 Jun  4 15:01 counter_val
-r-----   1 root  root           0 Jun  4 15:01 counter_val_diff
-rw-rw-rw-  1 root  root           0 Jun  4 15:01 event
-rw-r--r--  1 root  root           0 Jun  4 15:01 hwcounter_ctrl
-rw-r--r--  1 root  root           0 Jun  4 15:01 hwcounter_val
-rw-r--r--  1 root  root           0 Jun  4 15:01 polarity
-r--r--r--  1 root  root           0 Jun  4 15:01 val
```

(2)使用方法

・filter

ファイル	意味
filter	(R)入力フィルタ設定値の読み出し (W)入力フィルタ設定値の設定 入力回路のチャタリング除去用フィルタの時定数を設定する。 “through”(フィルタ無し), “50us”, “1ms”, “20ms” が選択可能。 [例] 入力フィルタ設定値の読み出し <pre>group0 # cat filter through</pre> [例] 入力フィルタ設定値の設定 <pre>group0 # echo 50us > filter group0 # cat filter 50us</pre>

・DOUT

ファイル	意味
counter_ctrl	<p>(R)ソフトウェアカウンタコントロール読出 (W)ソフトウェアカウンタコントロール設定 'on' or '1': enable 'off' or '0': disable [例] ソフトウェアカウンタコントロール読出 port00 # cat counter_ctrl off [例] ソフトウェアカウンタコントロール設定 (on) port00 # echo 1 > counter_ctrl port00 # cat counter_ctrl on [例] ソフトウェアカウンタコントロール設定 (off) port00 # echo 0 > counter_ctrl port00 # cat counter_ctrl off</p>
counter_val	<p>(R)ソフトウェアカウンタ値読出 (W)ソフトウェアカウンタ値クリア (任意の値) [例] ソフトウェアカウンタ値読出 port00 # cat counter_val 123 0x0000007b [例] ソフトウェアカウンタ値クリア port00 # echo 100 > counter_val port00 # cat counter_val 0 0x00000000</p>

ファイル	意味
event	<p>(R)入力データがマッチするまで待ち (W)指定時間 ([ms])待ち [例] タイムアウトの場合 port00 # echo 5000 > event -su: echo: write error: Connection timed out port00 # echo \$? 1 [例] 時間内にイベントがあった場合 <code> port00 # echo 5000 > event port00 # echo \$? 0</p>
hwcounter_ctrl	"counter_ctrl"の HW サポート版 ただし"group0"と"group1"のみサポート
hwcounter_val	"counter_val"の HW サポート版 ただし"group0"と"group1"のみサポート
polarity	<p>(R)割込極性読出 (W)割込極性設定 'rising' or '0': rising edge 'falling' or '1': falling edge [例] 割込極性読出 port00 # cat polarity rising [例] 割込極性設定 port00 # echo falling > polarity port00 # cat polarity falling</p>
val	<p>(R)入力状態読出 [例] port00 # cat val 0</p>

ファイル	意味
hwcounter_ctrl	"counter_ctrl"の HW サポート版 ただし"group0"と"group1"のみサポート
hwcounter_val	"counter_val"の HW サポート版 ただし"group0"と"group1"のみサポート

9.4.4.2. DOUT

(1)DOUT チャンネルに割り当てられたディレクトリとファイルの対応関係
DOUT チャンネルは 4 つのグループに分割されています。

接点出力 グループ	接点出力 ピン名称	ファイル
A	DOUTA0	/proc/driver/dout/val00
	DOUTA1	/proc/driver/dout/val01
	DOUTA2	/proc/driver/dout/val02
	DOUTA3	/proc/driver/dout/val03
	DOUTA4	/proc/driver/dout/val04
	DOUTA5	/proc/driver/dout/val05
	DOUTA6	/proc/driver/dout/val06
	DOUTA7	/proc/driver/dout/val07

接点出力 グループ	接点出力 ピン名称	ファイル
B	DOUTB0	/proc/driver/dout/val08
	DOUTB1	/proc/driver/dout/val09
	DOUTB2	/proc/driver/dout/val10
	DOUTB3	/proc/driver/dout/val11
	DOUTB4	/proc/driver/dout/val12
	DOUTB5	/proc/driver/dout/val13
	DOUTB6	/proc/driver/dout/val14
	DOUTB7	/proc/driver/dout/val15

接点出力 グループ	接点出力 ピン名称	ファイル
C	DOUTC0	/proc/driver/dout/val16
	DOUTC1	/proc/driver/dout/val17
	DOUTC2	/proc/driver/dout/val18
	DOUTC3	/proc/driver/dout/val19
	DOUTC4	/proc/driver/dout/val20
	DOUTC5	/proc/driver/dout/val21
	DOUTC6	/proc/driver/dout/val22
	DOUTC7	/proc/driver/dout/val23

接点出力 グループ	接点出力 ピン名称	ファイル
D	DOUDD0	/proc/driver/dout/val24
	DOUDD1	/proc/driver/dout/val25
	DOUDD2	/proc/driver/dout/val26
	DOUDD3	/proc/driver/dout/val27
	DOUDD4	/proc/driver/dout/val28
	DOUDD5	/proc/driver/dout/val29
	DOUDD6	/proc/driver/dout/val30
	DOUDD7	/proc/driver/dout/val31

[例] DOUT のディレクトリ 「/proc/driver/dout/」ディレクトリ

MAE2xx # cd /proc/driver/dout										
MAE2xx dout # ls										
val00	val03	val06	val09	val12	val15	val18	val21	val24	val27	val30
val01	val04	val07	val10	val13	val16	val19	val22	val25	val28	val31
val02	val05	val08	val11	val14	val17	val20	val23	val26	val29	

(2)使用方法

・DOUT

下表の val##の##には 0 から 31 の数字が対応します。

ファイル	意味
val##	(R)出力データの読み出し (W)出力データの書き込み 'on' or '1' 'off' or '0' [例] 出力データの読み出し dout # cat val00 0 off [例] 出力データの書き込み On dout # echo on > val00 dout # cat val00 1 on [例] 出力データの書き込み Off dout # echo 0 > val00 dout # cat val00 0 off

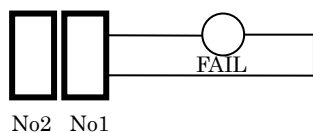
9.5. FAIL 接点出力インターフェースの使い方

9.5.1. 仕様

フェイル接点出力	
ポート数	1CH
出力信号	フォトモスリレー出力
負荷電圧	DC 24V±10%
負荷電流	100mA/CH
ON 電圧	DC 0.5V 以下
OFF 時漏洩電流	0.1mA 以下
開閉条件	オープン:システム起動時/異常発生時/ソフト設定 クローズ:正常動作時 (ソフトウェアの実装による)
保護機能	過電流保護
絶縁方式	フォトカプラ絶縁
絶縁耐圧	DC500V 1分間, 外部端子~内部回路間
フェイル状態表示	2色 LED(赤:オープン時点灯、緑:クローズ時点灯)
コネクタ	ロック付き 2P コネクタ Phoenix contact 社製 : MC1.5/2-G-3.5THT

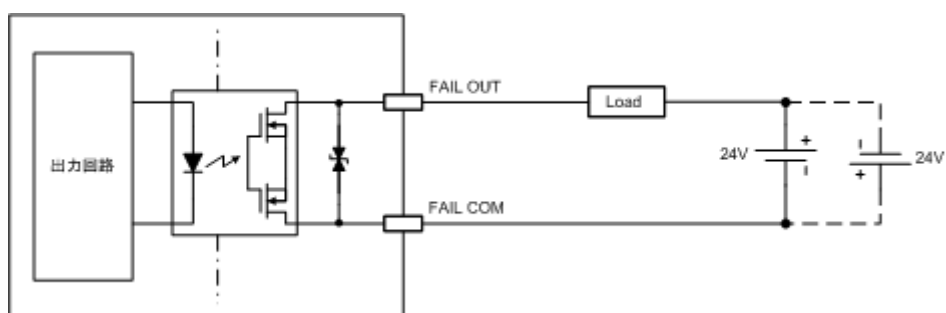
9.5.2. FAIL 接点出力コネクタとピン配置

<FAIL 接点出力コネクタピン配置図>



No.	ピン名称	機能
1	FAIL OUT	dout/fail
2	FAIL COM	

9.5.3. ハードウェアの接続方法



9.5.4. アプリケーションの作成方法

(1) FAIL 接点出力に割り当てられたディレクトリとファイルの対応関係

FAIL 接点出力のファイルは、「/proc/driver/dout/」ディレクトリに割り当てられている。

名称	ファイル
fail	fail

[例] FAIL のディレクトリ 「/proc/driver/dout/」ディレクトリ

```
MAE2xx # cd /proc/driver/dout
MAE2xx dout # ls fail
fail
```

(2)使用方法

・fail

ファイル	意味
fail	<p>(R)出力データの読み出し (W)出力データの書き込み 'on' or '1' 'off' or '0' [例] 出力データの読み出し dout # cat fail 0 off [例] 出力データの書き込み On dout # echo on > fail dout # cat fail 1 on [例] 出力データの書き込み Off dout # echo 0 > fail dout # cat fail 0 off</p>

9.6. MA-E210/AD-72 の仕様

製品名		FutureNet MA-E210/AD-72
CPU		Freescale™ i.MX353®プロセッサ 532MHz (ARM11 コア)
Flash ROM		32Mbyte
RAM		128Mbyte
インターフェース	イーサネット	10BASE-T/100BASE-TX × 1 ポート RJ-45 コネクタ、Auto Negotiation、Full/Half Duplex、Auto MDI/MDIX
	WAN	USB データ通信アダプタ
	シリアル	RS-232 (DTE) D-SUB9ピン オスコネクタ、最大 230.4kbps × 2 ※ 実装オプション: RS-485、基板上10ピンコネクタへの変更可
	USB	USB 2.0 ホスト × 2 ポート (TYPE-A コネクタ)、 ※ 内部インターフェース × 2 ポートと共用
	アナログ入力	アナログ入力 × 8 点 ※ 入力システム間絶縁
	デジタル入出力	接点入力 × 32 点 接点出力 × 32 点 ※ フォトカプラ絶縁
グラフィクス (実装オプション)	信号形式、コネクタ形状	LCD 40ピン内部コネクタ ※ 基板上 (予定)
	フレームサイズ	640 × 480
SD カードスロット		SDHC対応 SDカード 1スロット ※ SDメモ리카ードに対応
コンソールポート		Linux コンソール用シリアル (TTL レベル) 6ピンコネクタ ※ アダプタ別売
オーディオ		ステレオ・ライン出力、マイク入力 ※ φ 3.5 ジャック
LED 表示		システム: Power × 1, Status × 4, Ethernet: Link/Active × 1, Speed × 1
基本ソフトウェア	OS	Linux (Kernel 2.6)
	実行時ライブラリ	glibc 2.11(NPTL)
	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、シリアル/Ethernet 変換、データロガー
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
サイズ・重量	外観寸法(突起物を除く)	199.2mm(W) x 101.5mm(D) x 47.8mm(H)
	本体重量	約 850g
環境	使用電源、電源形状	DC+12V/+24V/+48V 入力、3P ねじ止め式コネクタ
	消費電力 (USB 接続なし)	約 6W
	本体動作温度条件	-20°C ~ 60°C
	AC アダプタ動作温度条件	—

	動作湿度条件	20%~85%(結露なきこと)
	保存温度	-20°C~60°C、25~90%(結露なきこと)
	添付品	DVD(取扱説明書、開発キット含む)、保証書
	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 5 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。

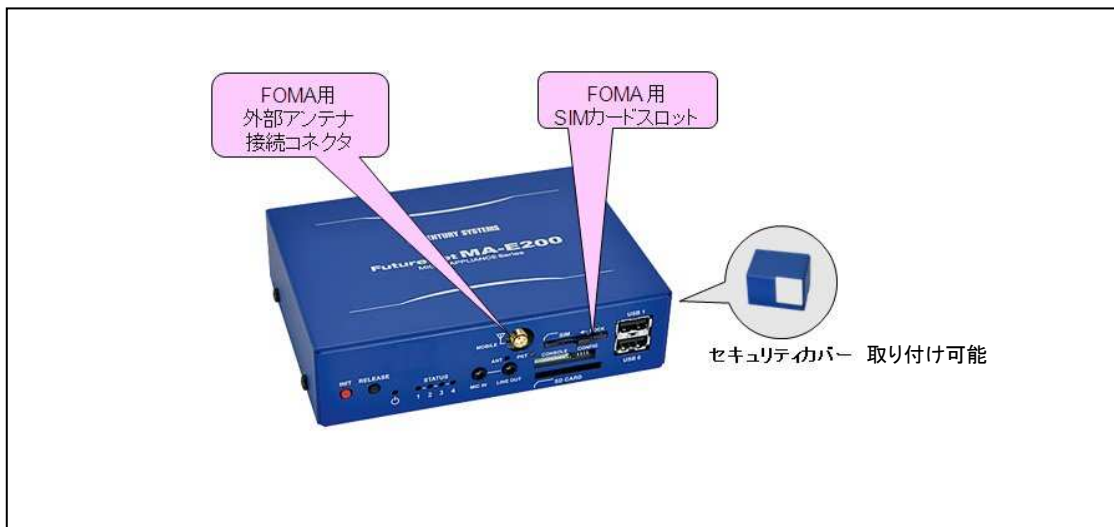
10. MA-E250/F について

10.1. FOMA モデルの概要

FutureNet MA-E250/F は、**MA-E210** に FOMA ユビキタスモジュールを搭載したモデルです。

モジュールタイプは、外部アンテナを接続できるため、別装置の中に組み込んでも安定した通信が可能です。また、USB タイプと比べて対応温度範囲が広いので、屋外設置など温度条件が厳しい環境にも対応できます。

閉域網接続サービス(ビジネス mopera アクセスプレミアム)にも対応しており、センターからの着信にも対応できます。また、Linux の機能を利用して通信の頻度やタイミングを柔軟に設定することが可能です。

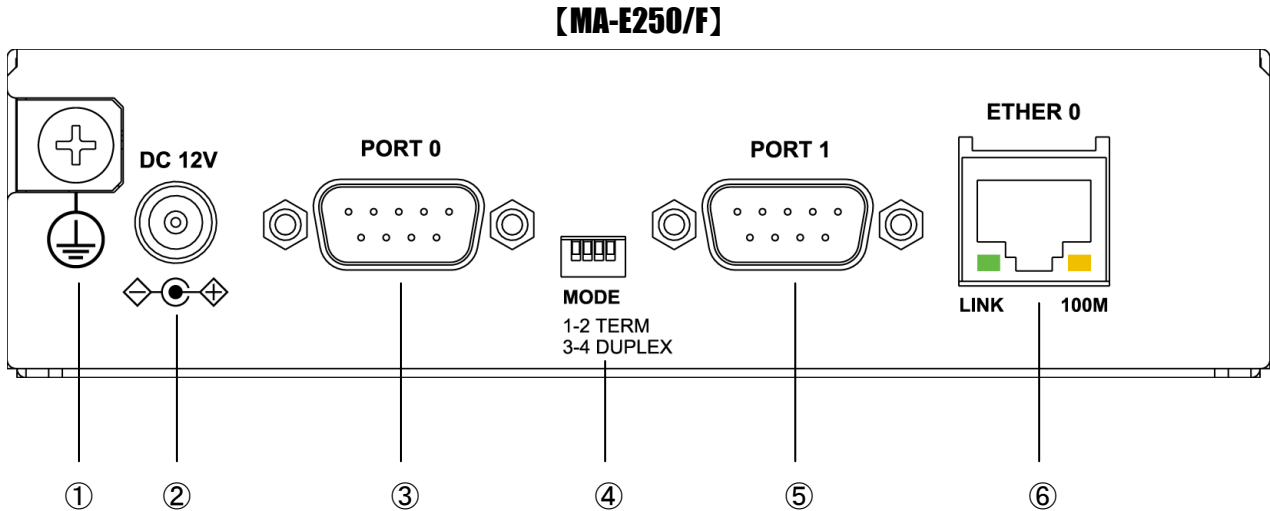


【図: FutureNet MA-E250/F の外観】 ※ 赤の部分が MA-E210 との違い

10.2. 外観

MA-E250/F 本体各部の名称は以下のとおりです。

<背面図>



① 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。

② DC 12V 電源コネクタ

製品付属の AC アダプタを接続します。

③ PORT0 ポート

DTE 対応の RS232 ポートが使用可能です。(実装オプションで RS-485 が使用可能です。) 後述するディップ・スイッチの設定により、Linux コンソール用として使用することも可能です。

④ MODE

"PORT0"が RS-232 設定のときは、必ず SW-1~4 をすべて OFF(上側)で使用してください。

"PORT0"が RS-485 設定のときについては、「1.5. RS-485 インターフェース仕様」をご確認ください。



⑤ PORT1 ポート

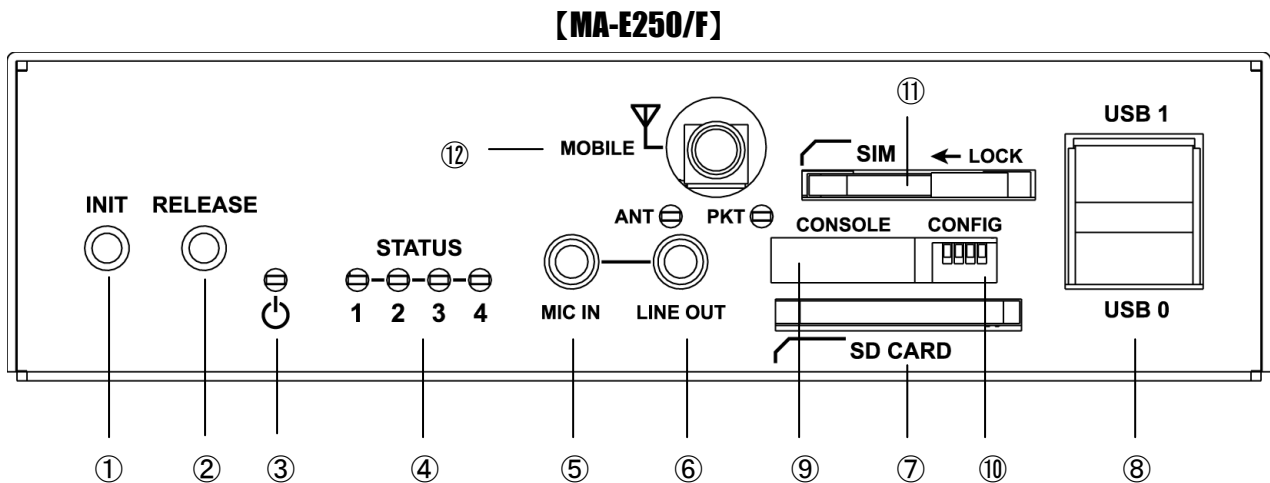
DTE 対応の RS232 ポートが使用可能です。

⑥ Ethernet ポート

10BASE-T/100BASE-TX 対応で、1 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。

<正面図>



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 RELEASE スイッチ

汎用のスイッチとして利用することができます。

③ POWER LED

MA-E250/F の電源状態を表示します。

④ STATUS LED

MA-E250/F の動作状態等を表示します。設定方法は 1.7 節をご参照ください。

⑤ MIC IN

音声入力を使用可能です。

⑥ LINE OUT

ステレオ音声出力を使用可能です。

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 USB0/USB1 ポート

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

⑨ CONSOLE

FutureNet コンソール・アダプタ(オプション)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。

⑩ 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 1.6 節をご参照下さい。

⑪ SIM カードソケット

FOMA カードを取り付けます。

⑫ MOBILE FOMA 用アンテナ

FOMA 用アンテナを取り付けます。

10.3. 接続と設置

10.3.1. FOMA データ通信サービスとアンテナの準備

FutureNet MA-E250/FをFOMA ネットワークへ接続するためには、「FOMA データ通信サービス」をご契約頂き、MA-E250/F へ FOMA カードを取り付けていただく必要があります。

また、MA-E250/F へ取り付けるアンテナは、本装置が適合する外付けアンテナ(別売)をご使用ください。

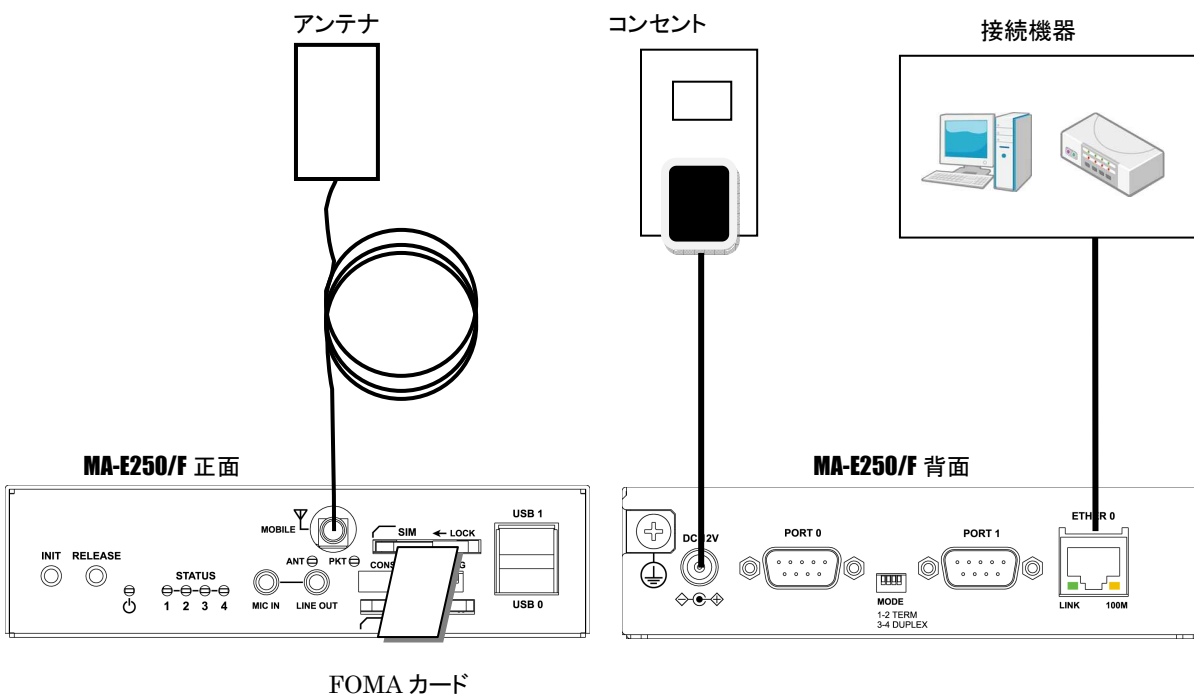
MA-E250/F 単体での接続はできませんことをご了承ください。

外付けアンテナのご購入、および「FOMA データ通信サービス」のご契約については、センチュリー・システムズ株式会社、もしくは弊社代理店様にお問い合わせください。

※FOMA カードとは、FOMA 用の SIM カードを言います。FOMA カードは、電話番号などのお客様情報が入った IC カードで、MA-E250/F のような FOMA サービス端末に挿入して使用します。FOMA カードは FOMA データ通信サービス契約後に NTT ドコモ、もしくはその代理店から発行されます。

10.3.2. 装置への接続

FutureNet MA-E250/F へ取り付け方法について説明します。



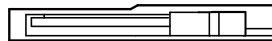
(1)作業前の準備

MA-E250/F、および **MA-E250/F** に接続するすべての接続装置の電源が OFF の状態で作業を行ってください。

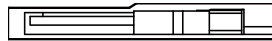
(2)FOMA カードの取り付け

MA-E250/F 正面にある"SIM"(以後 SIM カードソケットと呼ぶ)へ FOMA カードを挿入してください。SIM カードソケットにはスライドカバーが付いています。FOMA カード挿入後、スライドカバーを閉じて SIM カードソケットのロックを行ってください。

SIM カードアンロック状態



SIM カードロック状態



(3)アンテナの取り付け

MA-E250/F 正面にある"MOBILE"へ FOMA 用アンテナを接続してください。

FOMA 用アンテナの設置については、FOMA アンテナに添付の取扱説明書を参照して作業を行ってください。

(4)接続装置の取り付け

MA-E250/F 背面にある"ETHER0"、"PORT0"、および"PORT1"へ PC、デバイス等の接続装置をケーブルで接続してください。

(5)AC アダプタの取り付け

MA-E250/F 背面にある"DC 12V"へ AC アダプタを挿入してください。

全ての接続が完了しましたら、**MA-E250/F** と各接続機器の電源を投入してください。

10.3.3. LED 表示の見方

FutureNet MA-E250/F の FOMA に関する LED 表示の見方について説明します。

(1)"ANT" LED

アンテナレベル表示	LED 点灯
圏外	消灯
圏内(アンテナ 1 本)	赤点灯
圏内(アンテナ 2 本)	緑点灯
圏内(アンテナ 3 本)	緑点灯

(2)"PKT" LED

パケット圏内表示	LED 点灯
圏外	赤点灯
圏内	緑点灯

10.4. MA-E250/F の仕様

製品名		FutureNet MA-E250/F
CPU		Freescale™ i.MX353®プロセッサ 532MHz (ARM11 コア)
Flash ROM		32Mbyte
RAM		128Mbyte
インターフェース	イーサネット	10BASE-T/100BASE-TX × 1 ポート RJ-45 コネクタ、Auto Negotiation、Full/Half Duplex、Auto MDI/MDIX
	WAN	FOMA ユビキタス モジュール
	シリアル	RS-232 (DTE) D-SUB9ピン オスコネクタ、最大 230.4kbps × 2 ※ 実装オプション: RS-485、基板上10ピンコネクタへの変更可
	USB	USB 2.0 ホスト × 2 ポート (TYPE-A コネクタ)、 ※ 内部インターフェース × 2 ポートと共用
	アナログ入力	なし
	デジタル入出力	なし
グラフィクス (実装オプション)	信号形式、コネクタ形状	LCD 40ピン内部コネクタ ※ 基板上 (予定)
	フレームサイズ	640 × 480
SD カードスロット		SDHC対応 SDカード 1スロット ※ SDメモ리카ードに対応
コンソールポート		Linux コンソール用シリアル (TTL レベル) 6ピンコネクタ ※ アダプタ別売
オーディオ		ステレオ・ライン出力、マイク入力 ※ φ 3.5 ジャック
LED 表示		システム: Power × 1, Status × 4, Ethernet: Link/Active × 1, Speed × 1
基本ソフトウェア	OS	Linux (Kernel 2.6)
	実行時ライブラリ	glibc 2.11(NPTL)
	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、シリアル/Ethernet 変換
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
サイズ・重量	外観寸法(突起物を除く)	139.2mm(W) x 101.5mm(D) x 36.2mm(H)
	本体重量	約 550g
	AC アダプタ	120g
環境	使用電源、電源形状	DC +12V 1A(標準) AC アダプタ ※ 組込向けに 3ピンコネクタへの変更可
	消費電力 (USB 接続なし)	約 5W
	本体動作温度条件	-20°C~50°C
	AC アダプタ動作温度条件	-10°C~40°C

	動作湿度条件	20%~85%(結露なきこと)
	保存温度	-20°C~60°C、25~90%(結露なきこと)
	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 5 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。

FutureNet MA-E200 Series ユーザーズガイド Ver 1.1.0

2010年6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.
