SUZAKU-S
ソフトウェアマニュアル

SZ010-U00
SZ030-U00
SZ130-U00

Version 1.5.7-d308169
2009/08/03

株式会社アットマークテクノ [http://www.atmark-techno.com]

SUZAKU 公式サイト [http://suzaku.atmark-techno.com]

SUZAKU-S
ソフトウェアマニュアル

株式会社アットマークテクノ

060-0035 札幌市中央区北 5 条東 2 丁目 AFT ビル 6F
TEL 011-207-6550 FAX 011-207-6570

製作著作 © 2008 Atmark Techno, Inc

Version 1.5.7-d308169
2009/08/03

目次

1. はじめに 7
1.1. マニュアルについて 7
1.2. フォントについて 7
1.3. コマンド入力例の表記について 7
1.4. 謝辞 8
1.5. ソフトウェアに関する注意事項 8
2. 作業の前に 9
2.1. 準備するもの 9
2.2. 接続方法 10
2.3. ジャンパピンの設定について 11
3. 使用方法 12
3.1. 起動の前に 12
3.2. 起動モード 12
3.2.1. オートブートモード 12
3.2.2. ブートローダモード 12
3.3. 起動 13
3.4. ディレクトリ構成 15
3.5. 終了 15
3.6. ネットワーク設定 15
3.6.1. ネットワーク設定の確認 15
3.7. telnet ログイン 16
3.8. ファイル転送 16
3.9. Web サーバ 16
4. ブートローダ 18
4.1. ブートローダの種類 18
4.1.1. bboot 18
4.1.2. hermit 18
4.2. ブートシーケンス 19
4.2.1. 第 1 ステージ (bboot) 19
4.2.2. 第 2 ステージ (hermit) 19
4.2.3. 第 3 ステージ (カーネル) 19
4.2.4. 第 4 ステージ (ユーザーランド) 20
5. フラッシュメモリ 21
5.1. リージョンについて 21
5.2. メモリマップ 21
5.3. フラッシュメモリを書き換える (ダウンロード) 22
5.3.1. hermit による方法 24
5.3.2. netflash による方法 27
5.3.3. モトローラ S レコード形式による方法 27
5.3.4. flatfsd による方法 28
6. 開発環境の準備 30
6.1. Linux 環境の構築 30
6.2. クロス開発環境パッケージのインストール 30
6.2.1. クロス開発環境パッケージ 30
6.2.2. クロスデバッガパッケージ 31
6.2.3. uClinux-dist のビルドに必要なパッケージ 31
6.2.4. Linux 開発環境の依存関係について 32
7. uClinux-dist でイメージを作成 33
7.1. ソースコードアーカイブの展開 33
7.2. 設定 33

7.3. ビルド 35
8. Flat Binary Format 36
8.1. Flat Binary Format の特徴 36
8.2. 実行ファイルの圧縮 36
8.2.1. コンパイル済バイナリファイルを圧縮 37
8.3. スタックサイズの指定 38
8.3.1. コンパイル済バイナリファイルスタックサイズを変更 38
8.3.2. コンパイル時にスタックサイズを指定 39
8.4. ZFLAT 対応カーネルを作る 39
A. Appendix 41
A.1. シリアルコンソールソフト(minicom)のインストール 41
A.2. ダウンローダ(hermit)のインストール 41
A.3. 固定 IP アドレスで使用する場合 42

図目次

2.1. SUZAKU 接続例 10
2.2. ジャンパの位置 11
3.1. bboot 起動画面 12
3.2. bboot メニュー画面 13
3.3. hermit 起動画面 13
3.4. 起動ログ 13
3.5. ifconfig によるネットワーク設定の確認 15
3.6. ifconfig で IP アドレスを設定 16
3.7. Web サーバ 17
5.1. hermit コマンド入力例 25
5.2. Download 画面 25
5.3. Download 進捗ダイアログ 26
5.4. Download 終了画面 26
5.5. netflash 実行例 27
5.6. netflash ヘルプコマンド 27
5.7. モトローラ S レコード形式ダウンロード開始画面 28
5.8. モトローラ S レコード形式ダウンロード終了画面 28
5.9. flatfsd 28
6.1. クロス開発用パッケージのインストール 31
6.2. 複数パッケージのインストール 31
6.3. 開発用パッケージのインストール 31
6.4. 環境変数 PATH の設定例 31
7.1. dist アーカイブの展開 33
7.2. dist のコンフィギュレーション 33
7.3. Vendor の選択 34
7.4. Product の選択 34
7.5. Library の選択 34
7.6. デフォルト設定の選択 34
7.7. Customize および Update の選択 35
7.8. ビルド 35
8.1. 通常の Flat Binay Format 37
8.2. 圧縮された Flat Binary Format 37
8.3. スタックサイズの変更 38
8.4. FLTFLAGS によるスタックサイズの指定 39
8.5. ZFLAT support を Enable する 39
A.1. minicom のインストール 41
A.2. hermit のインストール 42
A.3. dhcpcd-new を外す 42
A.4. ifconfig ファイルの編集 42
A.5. dhcpcd-new の追加 43

表目次

1.1. 使用しているフォント 7
1.2. 表示プロンプトと実行環境の関係 7
2.1. ジャンパの設定と起動時の動作 11
3.1. シリアル通信設定 12
3.2. ログイン時のユーザ名とパスワード 15
3.3. ディレクトリ構成の一覧 15
3.4. telnet ログイン時のユーザ名とパスワード 16
3.5. ftp のユーザ名とパスワード 16
4.1. ブートローダ覧 18
4.2. frob コマンド 19
5.1. フラッシュメモリマップ (SZ130-U00) 21
5.2. フラッシュメモリマップ (SZ030-U00) 22
5.3. フラッシュメモリマップ (SZ010-U00) 22
5.4. リージョンと書き換え方法の対応 23
5.5. 各リージョン用のイメージバイナリファイル名 23
5.6. 各リージョン用 Flash デバイス名 27
6.1. クロス開発環境パッケージ一覧 30
6.2. uClinux-dist のビルドに必要なパッケージ一覧 32
6.3. SUZAKU-S 開発環境の組合せ 32

1.はじめに

1.1. マニュアルについて

このマニュアルは、SUZAKU を使用する上で必要な情報のうち、以下の点について記載されています。

- 基本的な使い方
- ブートローダ
- フラッシュメモリの書き換え方法
- 開発環境の構築
- カーネルとユーザーランドのビルド

アプリケーションやデバイスドライバ開発については、「uClinux-dist Developpers Guide」および「SUZAKU スターターキットガイド (Linux 開発編) 」を参照してください。

SUZAKU の機能を最大限に引き出すために、ご活用いただければ幸いです。

1.2. フォントについて

このマニュアルでは以下のようにフォントを使っています。

表 1.1. 使用しているフォント

フォント例	説明
本文中のフォント	本文
`[PC ~]$ ls`	プロンプトとユーザ入力文字列 ソースファイルのコード、ファイル名、ディレクトリ名など

1.3. コマンド入力例の表記について

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

表 1.2. 表示プロンプトと実行環境の関係

プロンプト	コマンドの実行環境
`[PC /]#`	作業用 PC 上の特権ユーザで実行
`[PC /]$`	作業用 PC 上の一般ユーザで実行
`[SUZAKU /]#`	SUZAKU 上の特権ユーザで実行
`[SUZAKU /]$`	SUZAKU 上の一般ユーザで実行

1.4. 謝辞

SUZKAU で使用しているソフトウェアは Free Software / Open Source Software で構成されています。Free Software / Open Source Software は世界中の多くの開発者の成果によって成り立っています。この場を借りて感謝の意を示します。

1.5. ソフトウェアに関する注意事項

本製品に含まれるソフトウェア(付属のドキュメント等も含みます)は、現状のまま(AS IS)提供されるものであり、特定の目的に適合することや、その信頼性、正確性を保証するものではありません。また、本製品の使用による結果についてもなんら保証するものではありません。

2.作業の前に

2.1. 準備するもの

SUZAKU を使用する前に、以下のものを準備してください。

作業用 PC	Linux もしくは Windows が動作し、1 ポート以上のシリアルポートを持つ PC です。
シリアルクロスケーブル	D-Sub9 ピン(メス－メス)のクロス接続用ケーブルです。
D-Sub 9 ピン-10 ピン変換ケーブル	D-Sub9 ピンと SUZAKU のピンヘッダ(10 ピン)を接続するケーブルです。
付属 CD-ROM(以降、付属 CD)	SUZAKU に関する各種マニュアルやソースコードが収納されています。
シリアル通信ソフト	minicom や Tera Term などのシリアル通信ソフトです。
DC3.3V 電源	DC3.3V 出力の電源です。

2.2. 接続方法

下図を参照して、シリアルクロスケーブル、電源、そして LAN ケーブルを SUZAKU に接続してください。D-Sub9 ピン-10 ピン変換ケーブルを SUZAKU に接続する際には、お互いの三角マークを合わせるようにして接続してください。

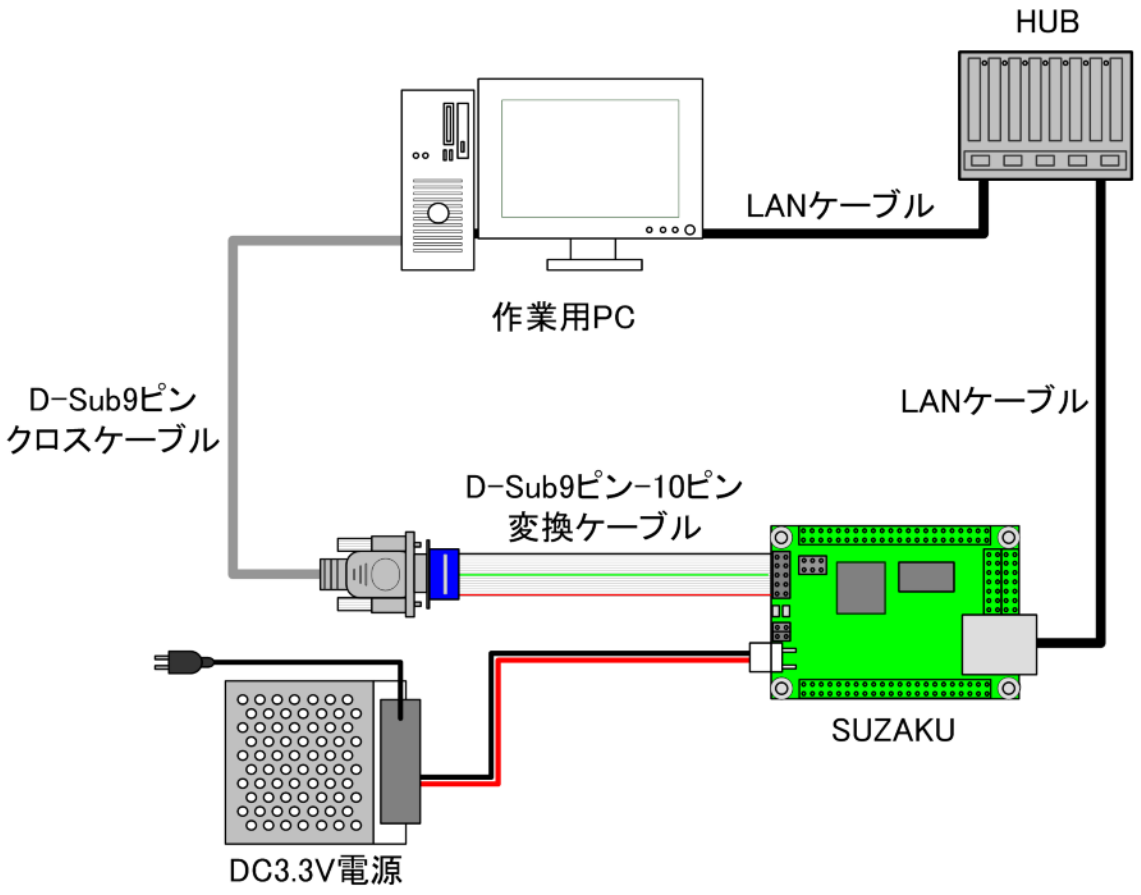

図 2.1. SUZAKU 接続例

2.3. ジャンパピンの設定について

SUZAKU ではジャンパピン(JP)の設定を変えることで、起動時の動作を変更することができます。以下の表に設定と動作の関連を記載します。起動モードについては、「3.2. 起動モード」を参照してください。

表 2.1. ジャンパの設定と起動時の動作

JP1	JP2	起動時の動作	起動モード
オープン	オープン	Linux カーネルを起動	オートブートモード
ショート	オープン	第 1 ステージ・ブートローダを起動	ブートローダモード
—	ショート	FPGA コンフィギュレーション [1]	—

[1] 詳しくは、「SUZAKU ハードウェアマニュアル」を参照してください。

ジャンパピンのオープン、ショートとは

・オープン： ジャンパピンにジャンパソケットを挿さない状態
・ショート： ジャンパピンにジャンパソケットを挿した状態

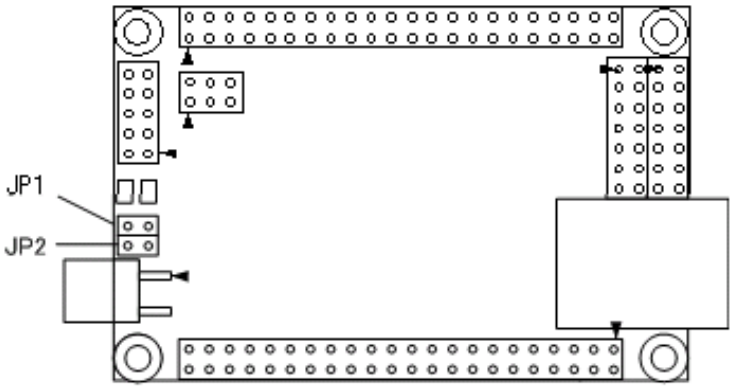

図 2.2. ジャンパの位置

3.使用方法

この章では SUZAKU の基本的な使用方法について説明します。

3.1. 起動の前に

SUZAKU のシリアルポート 1(CON1)と作業用 PC をシリアルケーブルで接続し、シリアル通信ソフトを起動します。次のように通信設定を行ってください。

表 3.1. シリアル通信設定

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

3.2. 起動モード

SUZAKU の起動モードには、オートブートモード、ブートローダモード、モトローラ S レコード形式ダウンロードモードの 3 つがあります。各モードの切り換えは、ジャンパピン 1 (JP1) で行います。ジャンパピンについては、「2.3. ジャンパピンの設定について」を参照してください。

3.2.1. オートブートモード

オートブートモードでは、自動的に Linux が起動されます。このモードで起動するには、ジャンパピン (JP1, JP2)をオープンにして、SUZAKU に電源を入れます。

3.2.2. ブートローダモード

ブートローダモードは、ブートローダを起動する場合に使用します。出荷時の SUZAKU のフラッシュメモリには hermit と呼ばれる高機能なブートローダ兼ダウンローダが入っています。hermit を起動すると、ユーザからのコマンド入力を受け付けるプロンプトが表示されます。コマンドには、メモリダンプ機能や、Linux イメージをフラッシュメモリへ書き込む機能などがあります。

ブートローダモードに移行するには、ジャンパピン JP1 をショート、JP2 をオープンにして電源を入れます。シリアル通信ソフトに、第 1 ステージ・ブートローダの bboot の起動ログと bboot メニューへの移行する指示が表示されます。

```
BBoot v2.x (microblaze)
Press ‘z’ or ‘Z’ for BBoot Menu.
```

図 3.1. bboot 起動画面

ここで、z キーを押し、bboot メニュー画面へ移動します。

メッセージ"Press `z' or `Z' for BBoot Menu"が表示されてから z キーを押した場合、キーの受付に間に合わないことがあります。その際は、z キーを押しながら電源を入れるようにしてください。

下図のような bboot メニュー画面がシリアルコンソールに表示されます。

```
Please choose one of the following and hit enter.
a: activate second stage bootloader (default)
s: download a s-record file
```

図 3.2. bboot メニュー画面

ここで、a キーまたは Enter キーを押してください。SUZAKU のデフォルトブートローダである hermit が起動し、プロンプトが表示されます。

```
Hermit-At v1.1.12(suzaku/microblaze) compiled at 21:41:20, Oct 18 2007
hermit>
```

図 3.3. hermit 起動画面

hermit を使ってフラッシュメモリにデータを書き込む場合には一度 SUZAKU をこのモードで起動しておかなければなりません。フラッシュメモリの詳しい書き換え方法については「5.3.1. hermit による方法」を参照してください。

なお、bboot メニュー画面へ移らなかった場合は、自動的に hermit 起動画面に移行します。

3.3. 起動

オートブートモードで電源を入れると、Linux が起動します。正常に起動した場合、シリアル通信ソフトに次のようなログが出力されます (SUZAKU-S SZ130-U00, ディストリビューション: uClinux-dist-20051110-suzaku4) 。

```
Linux version 2.4.32-uc0 (build@sarge) (gcc version 3.4.1 ( Xilinx EDK 8.1 Build
EDK_I.17 090206 ))
#1 Thu Oct 10 19:42:17 JST 2006
On node 0 totalpages: 8192
zone(0): 8192 pages.
zone(1): 0 pages.
zone(2): 0 pages.
CPU: MICROBLAZE
Kernel command line:
Console: xmbserial on UARTLite
Calibrating delay loop... 25.39 BogoMIPS
Memory: 32MB = 32MB total
```

```
Memory: 29524KB available (986K code, 1896K data, 44K init)
Dentry cache hash table entries: 4096 (order: 3, 32768 bytes)
Inode cache hash table entries: 2048 (order: 2, 16384 bytes)
Mount cache hash table entries: 512 (order: 0, 4096 bytes)
Buffer cache hash table entries: 1024 (order: 0, 4096 bytes)
Page-cache hash table entries: 8192 (order: 3, 32768 bytes)
POSIX conformance testing by UNIFIX
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Initializing RT netlink socket
Microblaze UARTlite serial driver version 1.00
ttyS0 at 0xffff2000 (irq = 1) is a Microblaze UARTlite
Starting kswapd
xgpio #0 at 0xFFFFA000 mapped to 0xFFFFA000
Xilinx GPIO registered
RAMDISK driver initialized: 16 RAM disks of 4096K size 1024 blocksize
eth0: LAN9115 (rev 1150001) at ffe00000 IRQ 2
Suzaku MTD mappings:
 Flash 0x800000 at 0xff000000
flash: Found an alies 0x800000 for the chip at 0x0, ST M25P64 device detect.
Creating 7 MTD partitions on "flash":
0x00000000-0x00800000 : "Flash/All"
0x00000000-0x00100000 : "Flash/FPGA"
0x00100000-0x00120000 : "Flash/Bootloader"
0x007f0000-0x00800000 : "Flash/Config"
0x00120000-0x007f0000 : "Flash/Image"
0x00120000-0x00420000 : "Flash/Kernel"
0x00420000-0x007f0000 : "Flash/User"
FLASH partition type: spi
uclinux[mtd]: RAM probe address=0x8012ba38 size=0x1a6000
uclinux[mtd]: root filesystem index=7
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 2048 bind 4096)
VFS: Mounted root (romfs filesystem) readonly.
Freeing init memory: 44K
Mounting proc:
Mounting var:
Populating /var:
Running local start scripts.
Mounting /etc/config:
Populating /etc/config:
flatfsd: Created 4 configuration files (149 bytes)
Setting hostname:
Setting up interface lo:
Starting DHCP client:
Starting inetd:
Starting thttpd:

SUZAKU-S.SZ130 login:
```

図 3.4. 起動ログ

ログインユーザは root で、パスワードの初期設定は root です。

表 3.2. ログイン時のユーザ名とパスワード

ユーザ名	パスワード	権限
root	root	特権ユーザ

3.4. ディレクトリ構成

ディレクトリ構成は次のようになっています。

表 3.3. ディレクトリ構成の一覧

ディレクトリ名	説　明
/bin	アプリケーション用
/dev	デバイスノード用
/etc	システム設定用
/etc/config	flatfsd 向け設定用
/lib	共有ライブラリ用
/mnt	マウントポイント用
/proc	プロセス情報用
/sbin	システム管理コマンド用
/usr	ユーザ共有情報用
/home	ユーザホームディレクトリ
/home/httpd	WEB サーバホームディレクトリ用
/tmp	テンポラリ保存用
/var	変更データ用

3.5. 終了

電源を切断し、SUZAKU を終了します。

3.6. ネットワーク設定

SUZAKU の工場出荷時には、DHCP を使用して IP アドレスを取得する設定になっています。ネットワークの設定を変更する場合は、イメージを再作成する必要があります。「7. uClinux-dist でイメージを作成」および「uClinux-dist 開発者ガイド」もあわせて参照してください。

3.6.1. ネットワーク設定の確認

ネットワークの設定は ifconfig コマンドで確認することができます。詳しくは、ifconfig コマンドのマニュアルを参照してください。DHCP にて IP アドレスの取得に成功した場合は、以下のような結果が表示されます。

```
[SUZAKU /]# ifconfig
eth0   Link encap:Ethernet HWaddr ##:##:##:##:##:##
       inet addr:192.168.1.## Bcast:192.168.1.255 Mask:255.255.255.0
       UP BROADCAST NOTRAILERS RUNNING MULTICAST MTU:1500 Metric:1
       RX packets:6 errors:0 dropped:0 overruns:0 frame:0
       TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
```

```
lo     Link encap:Local Loopback
       inet addr:127.0.0.1 Mask:255.0.0.0
       UP LOOPBACK RUNNING MTU:16436 Metric:1
       RX packets:0 errors:0 dropped:0 overruns:0 frame:0
       TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:0
```

図 3.5. ifconfig によるネットワーク設定の確認

DHCP サーバがない場合、IP アドレスの取得に失敗します。その際は、以下のように ifconfig コマンドで固定 IP アドレスを設定することができます。設定する前に、DHCP クライアントである dhcpcd プロセスを停止 (kill) する必要があります。設定が完了したら、ifconfig コマンドで正しく設定されているかを確認してください。

```
[SUZAKU /]# ifconfig eth0 192.168.1.##
```

図 3.6. ifconfig で IP アドレスを設定

但し、この方法では SUZAKU の電源を落とした際に、設定がクリアされてしまいます。クリアされないようにする方法は、「A.3. 固定 IP アドレスで使用する場合」を参照してください。

3.7. telnet ログイン

次のユーザ名とパスワードで telnet ログインが可能です。

表 3.4. telnet ログイン時のユーザ名とパスワード

ユーザ名	パスワード
root	root

3.8. ファイル転送

ftp によるファイル転送が可能です。次のユーザ名とパスワードでログインしてください。ホームディレクトリは/(ルート)です。`/var/tmp` ディレクトリに移動することでデータの書き込みが可能になります。

表 3.5. ftp のユーザ名とパスワード

ユーザ名	パスワード
root	root

3.9. Web サーバ

thttpd という小さな HTTP サーバが起動しており、Web ブラウザから SUZAKU をブラウズすることが出来ます。データディレクトリは`/home/httpd` です。URL は http://(SUZAKU の IP アドレス)/になります(例 http://192.168.1.100/)。

Welcome to SUZAKU - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
戻る
検索
お気に入り
アドレス(D)
http://192.168.1.100/
移動
リンク
Welcome to SUZAKU
This is a placeholder page in the SUZAKU, running uClinux release of the thttpd Web server.
Atmark Techno, Inc., Sapporo, Japan April 18, 2004
インターネット

図 3.7. Web サーバ

4.ブートローダ

この章では、SUZAKU のブートローダとブートシーケンスについて説明します。

4.1. ブートローダの種類

SUZAKU で使われているブートローダは、以下のとおりです。

表 4.1. ブートローダ覧

ブートローダ名	説明
bboot	第 1 ステージ・ブートローダ
hermit	第 2 ステージ・ブートローダ

4.1.1. bboot

bboot は、SUZAKU に電源を投入した際に、最初に起動するプログラムです。そのため、第 1 ステージ・ブートローダと呼ばれます。bboot は、FPGA の Block RAM と呼ばれる内部メモリに配置されています。

bboot の主な役割は、以下の通りです。

- 第 2 ステージ・ブートローダの起動
- 第 2 ステージ・ブートローダの書き換え(復旧)

第 2 ステージ・ブートローダの書き換え機能は、フラッシュメモリ上の第 2 ステージ・ブートローダが何らかの理由で動作しなくなった場合に、復旧(フラッシュメモリに第 2 ステージ・ブートローダを再書き込み)する際に使用します。復旧の方法については、「5.3.3. モトローラ S レコード形式による方法」を参照してください。

4.1.2. hermit

hermit は、Mike Touloumtzis 氏がメインテナンスを行っている高機能ブートローダ兼ダウンローダです。これにアットマークテクノがオリジナルのカスタマイズ、アットマークテクノ製品の対応を行い派生させたものを hermit-at として、SUZAKU では利用しています。なお、このマニュアルでは特に断りがない限り、hermit-at を hermit として表記しています。

hermit は、bboot の Block RAM とは違い、フラッシュメモリに配置されています。フラッシュメモリについては、「5. フラッシュメモリ」を参照してください。 hermit の一番の役割はカーネルをブートすることですが、その他にも多数の機能が存在します。以下にそれらの機能をいくつか紹介します。いずれも、ブートローダモードで起動し、hermit プロンプト(「図 3.3. hermit 起動画面」)からコマンドを入力し実行します。

boot　Linux をブートします。

frob　指定したアドレスへの読み書きが可能です。

表 4.2. frob コマンド

ブートローダ名	説明
peek <addr>	指定されたアドレスから 32bit のデータを読み出します。
peek8 <addr>	指定されたアドレスから 8bit のデータを読み出します。
peek16 <addr>	指定されたアドレスから 16bit のデータを読み出します。
poke <addr> <value>	指定されたアドレスに 32bit のデータを書き込みます。
poke8 <addr> <value>	指定されたアドレスに 8bit のデータを書き込みます。
poke16 <addr> <value>	指定されたアドレスに 16bit のデータを書き込みます。
quit	frob コマンドを終了します。
help	利用できる frob コマンド一覧を表示します。

help　hermit で利用できるコマンド一覧を表示します。

なお、第 1 ステージ・ブートローダである bboot から起動されるため、第 2 ステージ・ブートローダと呼ばれます。

4.2. ブートシーケンス

4.2.1. 第 1 ステージ (bboot)

リセット時にプログラム・カウンタが初期化されます。SUZAKU-S では 0x0 番地に FPGA の Block RAM がマップされており、bboot という第 1 ステージ・ブートローダが格納されています。

bboot は SUZAKU の JP1 の設定により、第 2 ステージ・ブートローダを起動するか、bboot 起動画面を表示します。JP1 をショートした場合、bboot 起動画面 (「図 3.1. bboot 起動画面」) が表示され、bboot メニュー画面 (「図 3.2. bboot メニュー画面」) へ移動することができます。bboot メニュー画面では、モトローラ S レコード形式のダウンロードなど、bboot で利用可能な機能が一覧表示され、機能を選択し実行することができます。

4.2.2. 第 2 ステージ (hermit)

hermit が JP1 を調べ、オートブートモードで起動するのかブートローダモードで起動するのかを判断します。オートブートモードの場合は、カーネルイメージをフラッシュメモリから SDRAM (オフセット 0x0 番地) にコピーしてカーネルに制御を渡します。

hermit をブートローダモードで起動した場合 (bboot メニュー画面から選択可能) 、hermit は独自のバイナリ転送方式で Linux のイメージをフラッシュメモリに書き込みます。

4.2.3. 第 3 ステージ (カーネル)

カーネルはブートローダから制御を渡された後、システムの初期化を行います。システムの初期化の多くは一般的な Linux と同じです。スケジューリングに必要なタイマの初期化や割り込みベクタの初期化、メモリ・マネージメント・サブシステムによる有効な RAM 領域の初期化などが行われます。そして、カーネルは最後にルート・ファイル・システムをマウントして、ユーザーランドに制御を渡します。

uClinux 特有の機能として、カーネルイメージの最後尾にルート・ファイル・システムを結合することができます。SUZAKU では、ROM の使用量を抑えるために、ルート・ファイル・システムはカーネルの BSS セクションを上書きするように結合しています。カーネルは初期化時にルート・ファイル・システムのイメージを発見すると、ルート・ファイル・システムをシフトし、BSS セクションを 0 で初期化します。

4.2.4. 第 4 ステージ (ユーザーランド)

カーネルはデフォルトで最初に/sbin/init を実行します。/sbin/init は、/etc/inittab に従ってシリアルコンソールからのログイン用に getty を起動したり、システムが機能するための設定やデーモンの起動を行います。使用している/sbin/init の実装は、デスクトップやサーバ用のディストリビューションとは異なりますが、シェル・スクリプトを順次実行するという基本的な動作は同じです。

5.フラッシュメモリ

SUZAKU ボードには、記憶装置としてフラッシュメモリが搭載されています。このフラッシュメモリには、ブートローダを始め Linux カーネルやユーザーランドのデータが格納されています。この章では、フラッシュメモリをどのように利用しているかと、フラッシュメモリの書き換え方法について説明します。

5.1. リージョンについて

SUZAKU では、フラッシュメモリをリージョンと呼ばれる複数の領域に分割して利用しています。以下にその 4 つのリージョンと用途を説明します。

fpga リージョン　FPGA のコンフィギュレーションデータを格納する領域です。

bootloader リージョン　ブートローダと呼ばれる、電源投入後、最初に実行されるソフトウェアのイメージを格納する領域です。ブートローダは、OS を起動する機能や、シリアル経由でフラッシュメモリを書き換える機能などを持ちます。ここでのブートローダは、第 2 ステージ・ブートローダの hermit を指します。

image リージョン　Linux のカーネルおよびユーザーランドのイメージを格納する領域です。この領域に格納されたカーネルはブートローダによって起動されます。

config リージョン　ユーザーアプリケーションなど各種設定ファイルなどの保存に利用する領域です。passwd ファイルもここに保存されています。

5.2. メモリマップ

SUZAKU ボードの種類によって、搭載しているフラッシュメモリの容量が異なります。そのため、各リージョンに割り当てられているアドレスやサイズも、製品によって異なります。ここでは各ボードのフラッシュメモリのリージョンとその領域を示します。

表 5.1. フラッシュメモリマップ (SZ130-U00)

アドレス	リージョン名	サイズ
0x00000000 | 0x000FFFFF	fpga	1MB
0x00100000 | 0x0011FFFF	bootloader	128KB
0x00120000 | 0x007EFFFF	image	約 6.81MB
0x007F0000 | 0x007FFFFF	config	64KB

表 5.2. フラッシュメモリマップ (SZ030-U00)

アドレス	リージョン名	サイズ
0x00000000 | 0x0000FFFF	free1	64KB
0x00010000 | 0x0007FFFF	free2	448KB
0x00080000 | 0x000FFFFF	fpga	512KB
0x00100000 | 0x0011FFFF	bootloader	128KB
0x00120000 | 0x007EFFFF	image	約 6.81MB
0x007F0000 | 0x007FFFFF	config	64KB

表 5.3. フラッシュメモリマップ (SZ010-U00)

アドレス	リージョン名	サイズ
0x00000000 | 0x0007FFFF	fpga	512KB
0x00080000 | 0x0009FFFF	bootloader	128KB
0x000A0000 | 0x003EFFFF	image	約 3.31MB
0x003F0000 | 0x003FFFFF	config	64KB

5.3. フラッシュメモリを書き換える (ダウンロード)

新しく開発したデバイスドライバやユーザーアプリケーションを SUZAKU に反映させるには、新しいイメージでフラッシュメモリを書き換える必要があります。また、何らかの原因で SUZAKU が正常に動作しなくなった場合も同様に、正常なイメージに書き直さなければなりません。この節では、フラッシュメモリの書き換え方法について説明します。なお、フラッシュメモリを書き換える作業は、作業用 PC から SUZAKU にデータを転送することからダウンロードとも言います。

フラッシュメモリを書き換える方法は、4 種類存在します。以下に各方法の特徴と書き換え可能なリージョンの関係を示します。

表 5.4. リージョンと書き換え方法の対応

リージョン名	書き換え方法			
	hermit	netflash	s-record	flatfsd
fpga	○	○	—	—
bootloader	○	○	○	—
image	○	○	—	—
config	—	○	—	○

hermit による方法　ダウンローダ hermit を使って、シリアル経由でフラッシュメモリを書き換えます。書き換えるイメージは、バイナリ形式のファイルを用います。

netflash による方法　Linux アプリケーションの netflash により、ネットワーク経由でフラッシュメモリを書き換えます。ネットワークを利用するため、シリアル経由の hermit に比べ高速です。しかし、不具合などにより Linux が動作しない状態では利用できません。書き換えるイメージは、hermit による方法と同じく、バイナリ形式のファイルを用います。

モトローラ S レコード形式による方法　bboot (ファーストブートローダ) に実装されているフラッシュメモリ書き換え機能です。書き換えるイメージは、モトローラ S レコード形式の ASCII ファイルです。hermit が異常状態に陥ったときにリカバリする手段となります。

flatfsd による方法　flatfsd と呼ばれる Linux アプリケーションで、SUZAKU のファイルシステム上のデータをフラッシュメモリに書き込んだり、フラッシュメモリからファイルシステム上に読み込んだりすることができます。

付属 CD の `suzaku/image` ディレクトリには、各リージョン向けのイメージバイナリファイルが格納されています。リカバリ用にお使いください。

表 5.5. 各リージョン用のイメージバイナリファイル名

リージョン	ファイル名	備考
fpga	fpga-sz###.bin	###には型番が入ります
bootloader	loader-suzaku-microblaze-v#.#.#.bin	使用する CPU によって、ファイルが異なります #.#.#にはバージョン番号は入ります
image	image-sz###.bin	###には型番が入ります

以降の説明では、ファイル名を簡略化するために、それぞれのリージョン用のファイルを `fpga.bin`、`loader-suzaku.bin`、`image.bin` とします。適時読み替えてください。

何らかの原因により書き換えイメージの転送に失敗した場合、SUZAKU が正常に起動しなくなる場合があります。書き換えの最中には次の点に注意してください。

- SUZAKU の電源を切らない。
- SUZAKU と作業用 PC を接続しているシリアルケーブルを外さない。

fpga リージョンに不適切なデータを書き込んだ場合、SUZAKU の異常動作により SUZAKU 及び周辺機器が発熱、劣化、破損する可能性があります。その際は、FPGA コンフィギュレーションデータの再プログラミングが必要になります。詳しくは「SUZAKU ハードウェアマニュアル」を参照してください。

bootloader リージョンに誤ったイメージを書き込んでしまった場合、オンボードフラッシュメモリからの起動ができなくなります。この場合は「5.3.3. モトローラ S レコード形式による方法」を参照してブートローダを復旧してください。

5.3.1. hermit による方法

hermit を使用したフラッシュメモリの書き換えについて説明します。hermit がインストールされていない場合は、「A.2. ダウンローダ(hermit)のインストール」を参照し、作業用 PC に hermit をインストールしてください。

まず、ブートローダモードで起動します。作業用 PC と SUZAKU のシリアルポート 1 (CON1) をシリアルケーブルで接続し、SUZAKU のジャンパピンを次のように設定します。

- JP1：ショート
- JP2：オープン

電源を投入後、bboot メニュー画面からブートローダモードに移動します。詳しくは、「3.2.2. ブートローダモード」を参照してください。

minicom などのシリアル通信ソフトがシリアルポートを使用している状態では、hermit がシリアルポートを使用できないためダウンロードに失敗します。必ずシリアル通信ソフトを終了 (シリアルポートを使用できる状態) してから hermit を起動してください。

ダウンロードの方法を説明します。以降の手順は、作業用 PC の OS によって異なります。

1. Linux の場合

   Linux が動作する作業用 PC でターミナルを起動し、イメージファイルとリージョンを指定して hermit コマンドを入力します。

   下の図では fpga のイメージ (`fpga.bin`) と hermit のイメージ (`loader-suzaku.bin`) 、uClinux のイメージ (`image.bin`) をダウンロードする例を示しています。-i オプションでファイル名を、-r オプションでリージョン名を指定します。

```
[PC ~]$ hermit download -i fpga.bin -r fpga --force-locked
[PC ~]$ hermit download -i loader-suzaku.bin -r bootloader --force-locked
[PC ~]$ hermit download -i image.bin -r image
```

図 5.1. hermit コマンド入力例

作業用 PC で使用するシリアルポートが ttyS0 以外の場合、オプション`--port "ポート名"`を追加してください。

FPGA および bootloader リージョンを書き換える際は、オプション--force-locked を追加する必要があります。指定しなかった場合、警告が表示され書き込みは実行されません。

書き換え終了後、JP1、JP2 をオープンに設定して SUZAKU を再起動すると、書き込んだイメージで起動されます。

2. Windows の場合

「A.2. ダウンローダ(hermit)のインストール」にてファイルを展開したフォルダにある、「Hermit-At WIN32 (hermit.exe) 」を起動します。

「Download」ボタンをクリックすると Download 画面が表示されます。

"Serial Port" には、SUZAKU と接続しているシリアルポートを設定してください。

"Image" には、書き込むイメージファイルを指定します。ファイルダイアログによる指定も可能です。

"Region" には、書き込むリージョンまたは、アドレスを指定します。

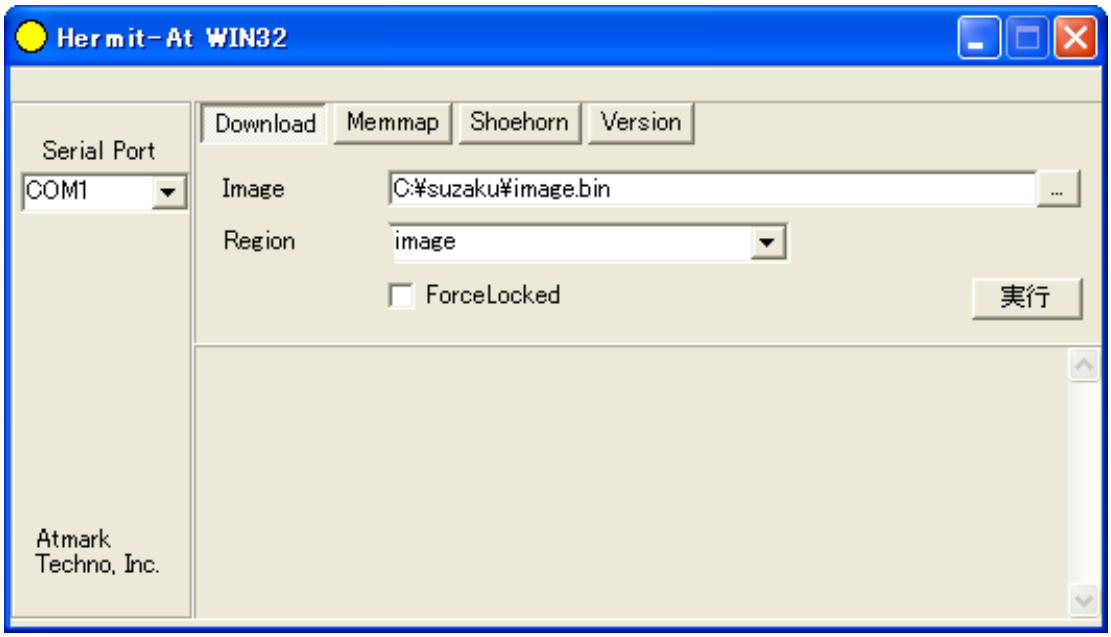

図 5.2. Download 画面

「実行」ボタンをクリックすると、フラッシュメモリの Download が開始されます。Download 中は、進捗状況が下図のように表示されます。ダイアログは、Download が終了すると自動的にクローズし、「図 5.4. Download 終了画面」のような Download 終了画面が表示されます。

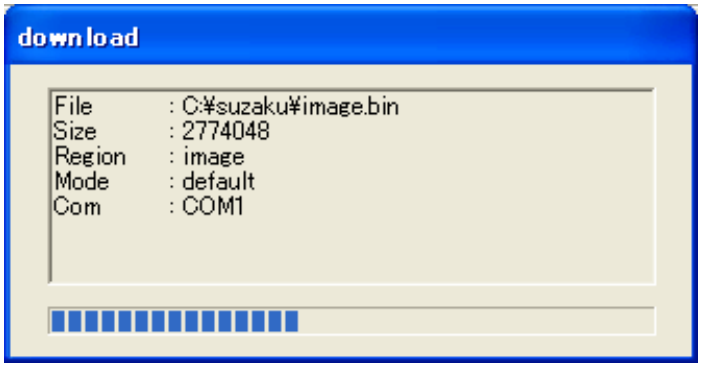

図 5.3. Download 進捗ダイアログ

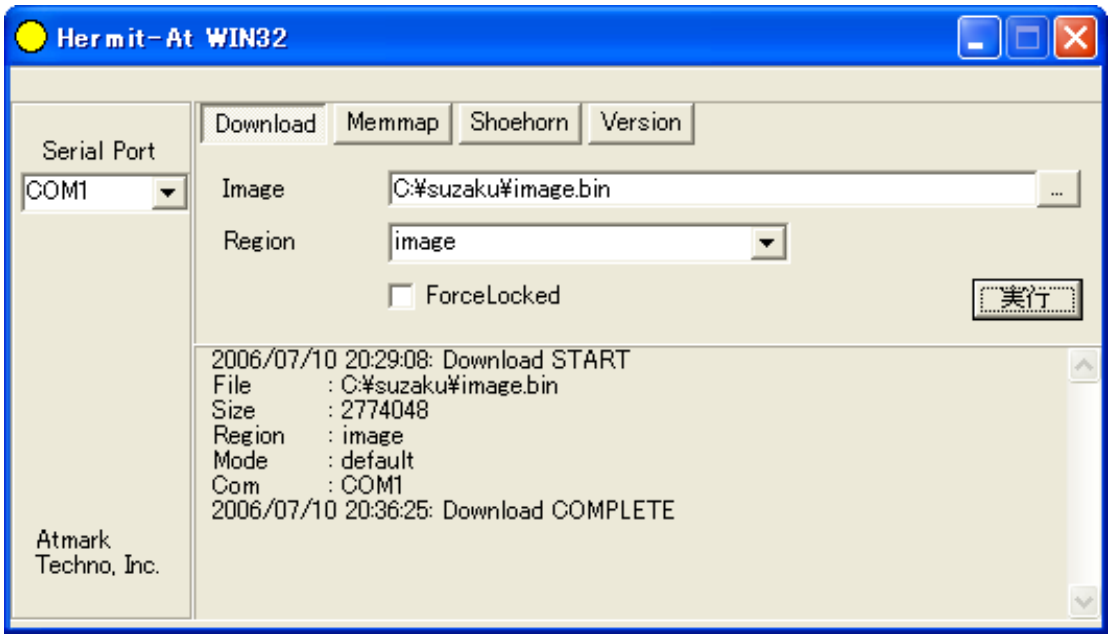

図 5.4. Download 終了画面

FPGA およびブートローダ領域を書き換える際は、「ForceLocked」をチェックする必要があります。これを選択しない場合、警告が表示されブートローダ領域への書き込みは実行されません。

書き換え終了後、JP1、JP2 をオープンに設定して SUZAKU を再起動すると、新たに書き込んだイメージで起動します。

5.3.2. netflash による方法

フラッシュメモリの内容を書き換える方法として、ユーザーアプリケーションの netflash を使用することも可能です。netflash を使用してフラッシュメモリを書き換える方法を説明します。

netflash は、HTTP や FTP サーバからファイルを取得し、フラッシュメモリへ書き込みます。予め、HTTP や FTP サーバにイメージファイルを置いておく必要があります。

SUZAKU 上でのイメージ (image.bin) を変更するコマンド例です。

```
[SUZAKU /]# netflash http://local.server.name/suzaku/image.bin
```

図 5.5. netflash 実行例

-r オプションで Flash デバイス名を指定します (デフォルトは`/dev/flash/image` です) 。Flash デバイス名の指定は下記表を参照してください。

表 5.6. 各リージョン用 Flash デバイス名

リージョン	Flash デバイス名
fpga	/dev/flash/fpga
bootloader	/dev/flash/boot
image	/dev/flash/image
config	/dev/flash/config

netflash のヘルプは以下のコマンドで参照することができます。

```
[SUZAKU /]# netflash -h
```

図 5.6. netflash ヘルプコマンド

netflash については、「uClinux-dist 開発者ガイド」にも説明があります。あわせて参照してください。

5.3.3. モトローラ S レコード形式による方法

モトローラ S レコード形式でのダウンロードは、bboot の機能として実装されており、hermit や Linux が起動しなくても使用することが可能です。ここでは、minicom を使ってモトローラ S レコード形式の hermit イメージを SUZAKU にダウンロードする方法を説明します。

モトローラ S レコード形式の hermit イメージは、付属 CD の `suzaku/bootloader/s-record` ディレクトリに以下のファイル名で格納されています。以降、SZ010-U00 の場合を例に説明します。他の製品をお使いの場合は適宜読み替えてください。

- loader-suzaku-microblaze-v#.#.#-4M.srec (SZ010-U00)
- loader-suzaku-microblaze-v#.#.#-8M.srec (SZ030-U00, SZ130-U00)

フラッシュメモリのサイズによって、hermit が書き込まれるアドレスが異なります。モトローラ S レコード形式ではファイル内に書き込みアドレスが書かれているために、同じ SUZAKU-S でもフラッシュメモリのサイズによってファイルを分けてあります。

まず、シリアルクロスケーブルで SUZAKU と作業用 PC を接続し、作業用 PC 上で minicom を起動します。次に SUZAKU の JP1 をショートして SUZAKU の電源を入れ、bboot メニュー画面に移行します。モトローラ S レコード形式のダウンロードを実行するには、bboot メニュー画面で s キーを入力します。以下のようなメッセージが表示されます。

```
Start sending S-Record!
```

図 5.7. モトローラ S レコード形式ダウンロード開始画面

このメッセージの後、SUZAKU はモトローラ S レコード形式のファイルを待ちます。minicom のメニュー画面から「Send files」を選択します。アップロードサブメニューの中から ASCII を選択し、ファイル選択画面に移動します。ファイル選択画面で bootloader-suzaku-microblaze-v#.#.#-4M.srec を選択して転送を開始します。転送中は minicom の中に小さな画面が表示されます。転送が終了すると

```
Erasing SPI ...
Programming SPI ...
Original checksum: 00XXXXXX
 Program checksum: 00XXXXXX
done.
Reboot.
```

図 5.8. モトローラ S レコード形式ダウンロード終了画面

と表示され、SUZAKU が再起動されます。

Original checksum は転送前のファイルのチェックサム、Program checksum は転送後のフラッシュメモリのチェックサムとなっています。もしこの二つのチェックサムの値が一致しない場合はファイルの転送に失敗しています。再度ファイルを転送しなおしてください[1]。

5.3.4. flatfsd による方法

ユーザ設定を保持するファイル (以降、設定ファイル) などをフラッシュメモリの config リージョンに保存する方法を紹介します。

フラッシュメモリ内に設定ファイルを保存するには、flatfsd プロセスに SIGUSR1 シグナルを送信します(「図 5.9. flatfsd 」参照) 。

```
[SUZAKU /]# killall -USR1 flatfsd
```

図 5.9. flatfsd

[1] チェックサムは BBoot v2.6 以降の機能となっております。

SIGUSR1 シグナルを受信した flatfsd プロセスは、`/etc/config` ディレクトリの内容を`/dev/flash/config` に書き込みます。つまり、設定ファイルは`/etc/config` に用意する必要があります。

fltfsd に関する説明は、「uClinux-dist 開発者ガイド」にもあります。詳しくは、そちらを参照してください。

6.開発環境の準備

この章では、SUZAKU の Linux ソフトウェアを開発するために、作業用 PC にクロス開発環境を構築する方法について説明します。

6.1. Linux 環境の構築

SUZAKU の Linux ソフトウェアを開発するためには、作業用 PC に Linux 環境を用意する必要があります。既に作業用 PC で Linux が稼動している場合は、次節に進んでください。

ここでは、Linux 環境をお持ちでない方のために、Windows 環境上に仮想的な Linux 環境を構築する方法として、「VMware」を紹介します。VMware を使用する方のために、開発に必要なソフトウェアがインストールされた状態の OS イメージ「ATDE (Atmark Techno Development Environment)」を提供しています。初めて開発される方や、すぐに開発に着手したい場合には、こちらをお勧めします。ATDE の使用に関する詳細は、「ATDE Install Guide」を参照してください。

6.2. クロス開発環境パッケージのインストール

付属 CD の `suzaku/cross-dev` ディレクトリにクロス開発環境パッケージが用意されているので、これらを全て作業用 PC にインストールします。インストールは必ず root 権限で行ってください。以下のパッケージが用意されています。

6.2.1. クロス開発環境パッケージ

クロス開発環境パッケージは付属 CD の `suzaku/cross_dev/microblaze` ディレクトリにあります。パッケージファイルは deb (Debian 系ディストリビューション向け)、tgz(インストーラ非使用)が用意されています。お使いの OS にあわせて、いずれか 1 つを選択してご利用ください。

表 6.1. クロス開発環境パッケージ一覧

パッケージ名	バージョン	説明
atde-essential-microblaze	7	Atmark Techno Development Environment for MicroBlaze
binutils-microblaze	2.10.1-mb-1	The GNU Binary utilities
cpp-microblaze	3.4.1-mb-1	The GNU C preprocessor
gcc-microblaze	3.4.1-mb-1	The GNU C compiler
elf2flt-microblaze	20070228-1	Elf2flt with PIC, ZFLAT and full reloc support.

クロス開発用パッケージのインストール例を「図 6.1. クロス開発用パッケージのインストール」に示します。

```
▼deb パッケージを使用する場合
[PC ~]# dpkg -i binutils-microblaze_2.10.1-mb-1_i386.deb

▼tgz を使用する場合
[PC ~]# tar zxf binutils-microblaze-2.10.1-mb-1.tgz -C /
```

図 6.1. クロス開発用パッケージのインストール

インストール時に依存関係でエラーになる場合は、以下のように複数のパッケージを同時に指定してください。ワイルドカードによる指定も可能です。

```
[PC ~]# dpkg -i xxx.deb yyy.deb zzz.deb
[PC ~]# dpkg -i *.deb
```

図 6.2. 複数パッケージのインストール

6.2.2. クロスデバッガパッケージ

クロスデバッガパッケージは、gzip で圧縮された tar アーカイブ形式になっています。作業用 PC の/usr/local/microblaze-gdb/ に展開してください。

```
[PC ~]$ su -
[PC ~]# mkdir -p /usr/local/microblaze-gdb/
[PC ~]# cd /usr/local/microblaze-gdb/
[PC microblaze-gdb]# tar zxvf microblaze-gdb-20060213.tar.gz
[PC microblaze-gdb]# ls
bin include info lib share
[PC microblaze-gdb]# exit
[PC ~]$
```

図 6.3. 開発用パッケージのインストール

次に、パッケージの実行ファイルが入っているディレクトリを環境変数 PATH に追加します。シェルによって設定方法が異なりますので、詳しくはお使いのシェルのマニュアルを参照してください。

ここでは、bash の設定例を示します。.bashrc ファイルに記述しておくと、次回、ログイン時にも有効になります。

```
[PC ~]$ export PATH=$PATH:/usr/local/microblaze-gdb/bin
[PC ~]$
```

図 6.4. 環境変数 PATH の設定例

6.2.3. uClinux-dist のビルドに必要なパッケージ

uClinux-dist をビルドするためには、作業用 PC に「表 6.2. uClinux-dist のビルドに必要なパッケージ一覧」に記されているパッケージがインストールされている必要があります。作業用 PC の環境に合わせて適切にインストールしてください。

表 6.2. uClinux-dist のビルドに必要なパッケージ一覧

パッケージ名	バージョン	説明
file	4.12-1 以降	Determines file type using "magic" numbers
genromfs	0.5.1-3 以降	This is the mkfs equivalent for romfs filesystem
libncurses5-dev	5.4-4 以降	Developer's libraries and docs for ncurses
perl	5.8.4-8 以降	Larry Wall's Practical Extraction and Report Language
sed	4.1.2-8 以降	The GNU sed stream editor
zlib1g-dev	1.2.2-4 以降	compression library - development

6.2.4. Linux 開発環境の依存関係について

SUZAKU にて Linux (uClinux)の開発を行うには、クロス開発パッケージと呼ばれる開発ツールと SUZAKU 用 Linux ディストリビューション (uClinux-dist-YYYYMMDD-suzakuX)が必要です。これらは、プロセッサを含め依存関係が存在します。下表に、利用可能な推奨する組合せを示します。

表 6.3. SUZAKU-S 開発環境の組合せ

ISE / EDK	クロスコンパイラ	ブートローダ	Linux ディストリビューション
10.1 (MicroBlaze 4.00)	gcc-3.4.1	Hermit-At v1.1.18	uClinux-dist-20051110- suzaku13

7.uClinux-dist でイメージを作成

この章では、uClinux-dist を使用して、Linux カーネルおよびユーザーランドのイメージを作成する方法を説明します。uClinux-dist に関する詳しい使用方法は、「uClinux-dist 開発者ガイド」を参照してください。

uClinux-dist を使用した開発作業では、基本ライブラリ・アプリケーションやシステム設定ファイルの作成・配置を行います。各ファイルは uClinux-dist ディレクトリ配下で作成・配置作業を行いますが、作業ミスにより誤って作業用 PC 自体の OS を破壊しないために、すべての作業は root ユーザではなく一般ユーザで行ってください。

7.1. ソースコードアーカイブの展開

付属 CD の suzaku/dist ディレクトリに uClinux-dist-YYYYMMDD-suzakuX.tar.gz というファイル名のソースコードアーカイブがあります。このファイルを任意のディレクトリに展開します。ここでは、ユーザのホームディレクトリ (~) に展開することとします。

```
[PC ~]$ tar zxf uClinux-dist-YYYYMMDD-suzakuX.tar.gz
```

図 7.1. dist アーカイブの展開

7.2. 設定

ターゲットボード用に uClinux-dist をコンフィギュレーションします。以下の例のようにコマンドを入力し、コンフィギュレーションを開始します。

```
[PC ~/uClinux-dist-YYYYMMDD-suzakuX]$ make config
```

図 7.2. dist のコンフィギュレーション

続いて、使用するボードのベンダー名を聞かれます。"AtmarkTechno"と入力してください。

```
*
* Vendor/Product Selection
*
*
* Select the Vendor you wish to target
*
Vendor (3com, ADI, Akizuki, Apple, Arcturus, Arnewsh, AtmarkTechno, Atmel,
Avnet, Cirrus, Cogent, Conexant, Cwlinux, CyberGuard, Cytek, EMAC, EPSD,
Exys, Feith, Future, GDB, Hitachi, Imt, Insight, Intel, KendinMicrel,
LEOX, Mecel, Midas, Motorola, NEC, NetSilicon, Netburner, Nintendo,
OPENcores, OpenGear, Philips, Promise, SNEHA, SSV, SWARM, Samsung,
SecureEdge, Signal, SnapGear, Soekris, Sony, StrawberryLinux, TI, TeleIP,
Triscend, Triscend, Via, Weiss, Xilinx, senTec) [SnapGear] (NEW)
AtmarkTechno
```

図 7.3. Vendor の選択

次に製品名を聞かれます。お使いの SUZAKU ボードを入力してください。Product は、製品名と型番を連結した文字列で表現されています (uClinux-dist-20051110-suzaku6 以降) 。

```
*
* Select the Product you wish to target
*
AtmarkTechno Products (SUZAKU-S.SZ010, SUZAKU-S.SZ030, SUZAKU-S.SZ130,
SUZAKU-S.SZ130-SIL, SUZAKU-UQ-XIP, SUZAKU-V.SZ310, SUZAKU-V.SZ310-SIL)
[SUZAKU-S.SZ010] (NEW) SUZAKU-S.SZ130-SIL
```

図 7.4. Product の選択

使用する C ライブラリを指定します。SUZAKU では、"uClibc"を選択します。

```
*
* Kernel/Library/Defaults Selection
*
*
* Kernel is linux-2.4.x
*
Libc Version (None, glibc, uC-libc, uClibc) [uClibc] (NEW) uClibc
```

図 7.5. Library の選択

デフォルトの設定にするかどうか質問されます。"y"(Yes)を選択してください。

```
Default all settings (lose changes) (CONFIG_DEFAULTS_OVERRIDE) [N/y/?]
(NEW) y
```

図 7.6. デフォルト設定の選択

最後の３つの質問は全て"n"(No)と答えてください。

```
Customize Kernel Settings (CONFIG_DEFAULTS_KERNEL) [N/y/?] n
Customize Vendor/User Settings (CONFIG_DEFAULTS_VENDOR) [N/y/?] n
Update Default Vendor Settings (CONFIG_DEFAULTS_VENDOR_UPDATE) [N/y/?] n
```

図 7.7. Customize および Update の選択

質問事項が終わるとビルドシステムの設定を行います。すべての設定が終わるとプロンプトに戻ります。

7.3. ビルド

ビルドします。以下のコマンドを入力してください。

```
[PC ~/uClinux-dist-YYYYMMDD-suzakuX]$ make dep all
```

図 7.8. ビルド

選択した製品によっては、make の途中で一時停止し、未設定項目の問合せが表示される場合があります。通常はデフォルト設定のままで構いませんので、そのような場合はそのままリターンキーを入力して進めてください。

ビルドが終了すると、uClinux-dist-YYYYMMDD-suzakuX/images ディレクトリに、カーネルとユーザーランドを合わせたイメージファイル (image.bin) が作成されます。作成したイメージファイルを SUZAKU に書き込む方法は「5.3. フラッシュメモリを書き換える (ダウンロード) 」を参照してください。

dep ターゲットは依存関係の解決を行います。2.4 系までの Linux カーネルのビルドシステムでは、make の前に依存関係の解決を行わなければなりません。

8.Flat Binary Format

SUZAKU に MMU を持たないソフトコアプロセッサ Microblaze を搭載したときには、MMU が必要な一般的な Linux は動作しません。このため出荷状態の SUZAKU-S では uClinux を利用しています。

uClinux が対象としている組み込み機器では実行ファイルのサイズは大きな問題です。一般の Linux が採用している ELF は柔軟性に富んだフォーマットですが、サイズが大きくなってしまいます。そこで uClinux では昔ながらの a.out フォーマットに似た新たなバイナリーフォーマットを採用しています。この章では、その一つである Flat Binary Format について説明します。

はじめに Flat Binary Format の特徴を説明した後、実行ファイルの圧縮方法とスタックサイズの変更方法を説明します。

8.1. Flat Binary Format の特徴

Flat Binary Format には次のような特徴があります。

シンプル	シンプルな設計はバイナリファイルの実行速度と大きさに貢献しています。もちろん ELF に比べ柔軟性は劣りますが、組み込み機器にとっては必要なトレードオフと言えます。
圧縮可能	Flat Binary Format は圧縮可能なフォーマットです。圧縮には２種類あり、ファイル全体の圧縮とデータ領域のみの圧縮が可能です。実行ファイルはロードされる時に伸長されるため、起動速度は非圧縮の実行ファイルに比べ低速です。起動後は非圧縮の実行ファイルとの差はなく、常駐プロセスのように起動停止を繰り返さないプログラムには適しています。
スタックサイズフィールド	再コンパイルせずに変更可能なスタックサイズのフィールドを持っています。MMU を持たない CPU では動的にスタック領域を拡張することが難しいため、固定サイズのスタック領域を持ちます。このフィールドは flthdr と呼ばれるツールで変更することが可能です。また、コンパイル時にサイズを指定することも可能です。
XIP 対応	Flat Binary Format は XIP にも対応しています。XIP とは eXecute In Place (その場実行) の略で、一般的には RAM に実行バイナリをコピーすることなく、保存されている ROM 上で実行する機能のことです。

8.2. 実行ファイルの圧縮

例として「uCliux-dist 開発者ガイド」で作成した Hello World プログラムを圧縮します。後半ではコンパイル時に圧縮を指定する方法を説明します。

デフォルトのカーネルでは、圧縮された Flat Binary Format (ZFLAT) のファイルを実行できません。カーネルを ZFLAT 対応にするには、「8.4. ZFLAT 対応カーネルを作る」を参照してください。

8.2.1. コンパイル済バイナリファイルを圧縮

flthdr を使ってコンパイル済のバイナリファイルを圧縮します。flthdr は Flat Binary Format のファイルを編集・表示するプログラムです。Microblaze ツールチェーンには mb-flthdr という名前で存在していますので、以降 mb-flthdr として説明します。

通常のコンパイルで生成された実行ファイルでは以下のようになります。

```
[PC ~/hello]$ make
[PC ~/hello]$ mb-flthdr hello                                    --- ❶
hello
   Magic:       bFLT
   Rev:         4
   Build Date:  Fri Jun 30 18:33:00 2006
   Entry:       0x50
   Data Start:  0x4f40
   Data End:    0x5cf0
   BSS End:     0x7d10
   Stack Size:  0x1000
   Reloc Start: 0x5cf0
   Reloc Count: 0x51
   Flags:       0x1 ( Load-to-Ram )
[PC ~/hello]$
```

❶ mb-flthdr コマンドで実行ファイルのヘッダ部を表示させる

図 8.1. 通常の Flat Binay Format

次に mb-flthdr で圧縮します。

```
[PC ~/hello]$ mb-flthdr -z hello                                 --- ❶
zflat hello --> hello
[PC ~/hello]$ mb-flthdr hello                                    --- ❷
   Magic:       bFLT
   Rev:         4
   Build Date:  Fri Jun 30 18:35:00 2006
   Entry:       0x50
   Data Start:  0x4f40
   Data End:    0x5cf0
   BSS End:     0x7d10
   Stack Size:  0x1000
   Reloc Start: 0x5cf0
   Reloc Count: 0x51
   Flags:       0x5 ( Load-to-Ram Gzip-Compressed )             --- ❸
[PC ~/hello]$
```

❶ mb-flthdr に圧縮オプション-z を渡す

❷ mb-flthdr コマンドで実行ファイルのヘッダ部を表示させる

❸ Gzip-Compressed フラグが確認できる

図 8.2. 圧縮された Flat Binary Format

8.3. スタックサイズの指定

スタックサイズを指定する方法を 2 種類紹介します。

8.3.1. コンパイル済バイナリファイルスタックサイズを変更

mb-flthdr の-s オプションでスタックサイズを指定します。アーキテクチャによりますが、スタックサイズのデフォルト値は 4096 (0x1000) が多いです。

```
[PC ~/hello]$ mb-flthdr -s 8192 hello                    --- ❶
Zflat hello --> hello
[PC ~/hello]$ mb-flthdr hello                            --- ❷
hello
   Magic:       bFLT
   Rev:         4
   Build Date:  Fri Jun 30 18:34:00 2006
   Entry:       0x50
   Data Start:  0x4f40
   Data End:    0x5cf0
   BSS End:     0x7d10
   Stack Size:  0x2000                                   --- ❸
   Reloc Start: 0x5cf0
   Reloc Count: 0x51
   Flags:       0x1 ( Load-to-Ram )
[PC ~/hello]$
```

❶ mb-flthdr にスタックサイズ変更オプション-s とスタックサイズ (10 進数)を渡す

❷ mb-flthdr コマンドで生成された実行ファイルのヘッダ部を表示させる

❸ 8192byte に変更されているのが確認できる

図 8.3. スタックサイズの変更

8.3.2. コンパイル時にスタックサイズを指定

コンパイル時にスタックサイズを指定するには、FLTFLAGS 環境変数を使います。以下に Hello World での例を示します。

```
[PC ~/hello]$ make FLTFLAGS='-s 8192'                                  --- ❶
[PC ~/hello]$ mb-flthdr hello                                          --- ❷
hello
   Magic:       bFLT
   Rev:         4
   Build Date:  Fri Jun 30 18:34:00 2006
   Entry:       0x50
   Data Start:  0x4f40
   Data End:    0x5cf0
   BSS End:     0x7d10
   Stack Size:  0x2000                                                 --- ❸
   Reloc Start: 0x5cf0
   Reloc Count: 0x51
   Flags: 0x1 ( Load-to-Ram )
[PC ~/hello]$
```

❶ FLTFLAGS 環境変数に'-s 8192'をセットして make を実行する

❷ mb-flthdr コマンドで生成された実行ファイルのヘッダ部を表示させる

❸ 8192byte に変更されたスタックが確認できる

図 8.4. FLTFLAGS によるスタックサイズの指定

8.4. ZFLAT 対応カーネルを作る

ZFLAT とは圧縮された Flat Binary Format の実行ファイルです。実際に ZFLAT に対応したカーネルを作ってみます。

uClinux-dist のディレクトリで、make menuconfig を実行し、「Kernel/Library/Defaults Selection」メニューから「Customize Kernel Settings」を選択します。メインメニューから Exit を選択し、設定を保存し終了します。「Linux Kernel Configuration」のメインメニューの中から、「General setup」に移動します。「Kernel support for flat binaries」の下にある「Enable ZFLAT support」を選択します。メインメニューから Exit を選択し、設定を保存し終了します。

```
[PC ~/uClinux-dist]$ make menuconfig
Main Menu
  Kernel/Library/Defaults Selection --->
    [*] Customize Kernel Settings

Main Menu
  General setup --->
    <*> Kernel support for flat binaries
    [*] Enable ZFLAT support                                  ← チェックを追加
```

図 8.5. ZFLAT support を Enable する

make clean dep all を実行し、image.bin を生成します。生成されたイメージのカーネルでは ZFLAT ファイルを実行することができます。

付録 A. Appendix

A.1. シリアルコンソールソフト(minicom)のインストール

作業用 PC に「シリアルコンソールソフト(minicom)」をインストールします。

付属 CD よりパッケージファイルを用意し、インストールします。必ず root 権限で行ってください。パッケージファイルは deb(Debian 系ディストリビューション向け)、rpm(Red Hat 系ディストリビューション向け)が用意されています。お使いの OS にあわせて、いずれか 1 つを選択してご利用ください。

▼deb パッケージを使用する場合
```
[PC ~]# dpkg -i minicom_2.1-4.woody.1_i386.deb
```

▼rpm パッケージを使用する場合
```
[PC ~]# rpm -i minicom_2.1-1-rh7.3.i386.rpm
```

図 A.1. minicom のインストール

minicom に-s オプションを付けて起動します。-s を指定することで、設定画面に移行します。シリアルポートの通信設定を行ってください。

minicom の初期設定では、起動時にモデムの初期化を行うようになっていることが多いようです。設定で初期化用の AT コマンドを外すか、minicom に-o オプションを付けて起動することでモデム初期化コマンドを省略することができます。また、使用するシリアルポートの読み込みと書き込み権限が無い場合、minicom の起動に失敗します。使用するシリアルポートの権限を確認してください。詳しくは minicom のマニュアルまたはご使用の OS のマニュアルをご覧ください。

A.2. ダウンローダ(hermit)のインストール

作業用 PC に、ダウンローダである hermit をインストールします。ダウンローダは SUZAKU のフラッシュメモリの書き換えに使用します。

1. Linux の場合

   付属 CD よりパッケージファイルを用意し、インストールします。必ず root 権限で行ってください。パッケージファイルは deb(Debian 系ディストリビューション向け)、rpm(Red Hat 系ディストリビューション向け)が用意されています。お使いの OS にあわせて、いずれかを選択してください。また、tar.gz(インストーラ非使用、要コンパイル)が用意されています。

```
▼deb パッケージを使用する場合
[PC ~]# dpkg -i hermit-at_#.#.#_i386.deb

▼rpm パッケージを使用する場合
[PC ~]# rpm -i hermit-at-#.#.#-#.i386.rpm

▼tar.gz を使用する場合
[PC ~]# tar zxf hermit-at-#.#.#-source.tar.gz -C /
```

図 A.2. hermit のインストール

2. Windows の場合

付属 CD より「Hermit-At WIN32(hermit-at-win-YYYYMMDD.zip)」を任意のフォルダに展開します。

A.3. 固定 IP アドレスで使用する場合

固定 IP アドレスで起動するイメージの作成方法を説明します。まず、DHCP クライアントを起動させないようにするために、コンフィギュレーションにて dhcpcd-new (2.0/2.4)を外します。

```
[PC ~/uClinux-dist]$ make menuconfig
Main Menu
  Kernel/Library/Defaults Selection --->
    [*] Customize Vendor/User Settings

Main Menu
  Network Applications --->
    [ ] dhcpcd-new (2.0/2.4)                                  ← チェックを外す
```

図 A.3. dhcpcd-new を外す

次に、固定 IP アドレスを設定します。設定は、uClinux-dist を展開したディレクトリの下の vendors/AtmarkTechno/SUZAKU-#.SZ###/etc/rc/ifconfig ファイルを編集します。#の部分はお使いの SUZAKU ボードに合わせて選択してください。ifconfig ファイルの上部に記述されている IP_ADDRESS に適用する IP アドレスを記述します。

```
[PC ~/uClinux-dsit]$ vi vendors/AtmarkTechno/SUZAKU-#.SZ###/etc/rc/ifconfig
#!/bin/sh

IP_ADDRESS=192.168.10.54                                      ← IP アドレスを記述

PATH=/bin:/sbin:/usr/bin:/usr/sbin

echo “Setting up interface lo: “
ifconfig lo up 127.0.0.1

echo “Setting up interface eth0: “
ifconfig eth0 $IP_ADDRESS
```

図 A.4. ifconfig ファイルの編集

最後にイメージファイルを再作成し、フラッシュメモリを書き換えてください。フラッシュメモリの書き換え方については、「5.3. フラッシュメモリを書き換える (ダウンロード) 」を参照してください。

なお、固定 IP アドレスに変更したあと DHCP を使用する設定に戻したい場合には、以下のように DHCP クライアントデーモン (dhcpcd-new) を組み込みます。追加した後は、イメージファイルを再作成し、フラッシュメモリを書き換えてください。

```
[PC ~/uClinux-dist]$ make menuconfig
Main Menu
  Kernel/Library/Defaults Selection --->
    [*] Customize Vendor/User Settings

Main Menu
  Network Applications --->
    [*] dhcpcd-new (2.0/2.4)                                  ← チェックを追加
```

図 A.5. dhcpcd-new の追加

改訂履歴

バージョン	年月日	改訂内容
1.0.0	2004/04/29	・初版発行
1.0.1	2004/06/04	・Minicom のインストールを apt から dpkg に変更 ・ツールチェーン名の誤字を修正 ・OTC の Makefile を汎用化
1.0.2	2004/12/15	・会社住所変更
1.1.0	2005/01/31	・8MB Flash の記述を追加
1.2.0	2005/03/01	・SUZAKU-V 用の記述を追加
1.3.0	2006/07/14	・全体構成の見直し ・ネットワーク関連情報 (設定、telnet、ftp) の追加 ・ダウンローダ (Hermit) の Windows 版の記述を追加 ・SZ130-U00 Flash メモリマップの記述を追加 ・Windows 上に開発環境を構築する方法 (coLinux) を追加
1.3.1	2006/08/11	・Flash メモリの書き換え方法に fpga リージョンの記述を追加
1.3.2	2006/09/15	・uClinux-dist-20051110-suzaku3 用に書き換え ・誤記訂正
1.3.3	2006/10/20	・「1.6. 保証に関する注意事項」を追加 ・uClinux-dist-20051110-suzaku4 用に書き換え ・telnet ログインの説明修正
1.4.0	2007/01/19	・「5. フラッシュメモリの書き換え方法」と「8. フラッシュメモリについて」を統合し、「5. フラッシュメモリ」を追加 ・「4.2.3. モトローラ S レコード形式ダウンロードモード」を削除 ・coLinux のインストール方法を「coLinux Guide」に移動 ・章構成の見直し ・表示デザイン変更 ・誤記、誤植修正
1.4.1	2007/04/20	・「6.2.4. Linux 開発環境の依存関係について」を追加
1.5.0	2007/10/19	・coLinux から VMware に移行 ・「6.1. Linux 環境の構築」を ATDE 向けの記述に書き換え ・「6.2. クロス開発環境パッケージのインストール」を deb/tgz パッケージ使用するように書き換え ・SUZAKU-V 用の記述を削除
1.5.1	2008/02/15	・「4.2.2. 第 2 ステージ (hermit) 」に kernel の先頭アドレス追記
1.5.2	2008/03/14	・「6.2. クロス開発環境パッケージのインストール」に atde-essential-microblaze を追加
1.5.3	2008/04/22	・「5.3.3. モトローラ S レコード形式による方法」にファイルのチェックサム機能について追記
1.5.4	2008/09/26	・タイトルを英語表記からカタカナ表記に ・「2.2. 接続方法」にシリアルケーブル接続時の注意について追記 ・「表 6.3. SUZAKU-S 開発環境の組合せ」を最新情報に更新
1.5.5	2009/03/19	・参照先を記述する際の表記を統一 ・表紙に対応型番を記載 ・表記ゆれを修正
1.5.6	2009/07/17	・本文のレイアウト統一 ・文字化け修正

1.5.7	2009/07/29	・製品保証に関する記載を http://www.atmark-techno.com/support/warranty-policy に移動(2009/08/03 適用)

SUZAKU-S ソフトウェアマニュアル
Version 1.5.7-d308169
2009/08/03

株式会社アットマークテクノ
060-0035 札幌市中央区北 5 条東 2 丁目 AFT ビル 6F TEL 011-207-6550 FAX 011-207-6570