

AP4010 (ARM926EJ-S) CPU BOARD

EMP-ARM9

ソフトウェアマニュアル

TOPPERS 版

1 版

ダイジェスト版



eclipse

ALPHA PROJECT co.,LTD

<http://www.apnet.co.jp>

目次

1. はじめに	1
1.1 TOPPERS について	1
1.2 TOPPERS/JSP カーネルについて	1
1.3 TINET について	1
1.4 GNU と FSF について	1
1.5 TOPPERS ライセンスについて	1
1.6 RedBoot について	2
1.7 GCC について	2
1.8 GPL と LGPL について	2
1.9 保証とサポート	2
2. EMP-ARM9 のシステム	3
2.1 RedBoot	3
2.2 TOPPERS/JSP	4
2.3 TINET	9
2.4 Eclipse 概要	11
3. クロス開発環境	12
3.1 クロス開発環境の概要	12
3.2 動作環境	13
3.3 インストール CD の構成	14
3.4 インストール手順	15
3.5 インストールディレクトリの構成	21
4. EMP-ARM9 の起動	23
4.1 動作環境	23
4.2 ターミナルと Ethernet の接続	24
4.3 ターミナルの設定	25
4.4 起動と動作確認	27
5. 開発手順	29
5.1 Eclipse の起動	29
5.2 新規プロジェクト作成	30
5.3 ビルド方法	37
5.4 プログラムの転送と動作確認	39

6. サンプルアプリケーション 40

- 6.1 サンプルアプリケーションの概要 40
- 6.2 プログラムの配置イメージ 42

7. ハードウェア依存設定 43

- 7.1 レジスタ設定 43
- 7.2 IRQ 番号 45

- 製品サポートのご案内 45
- エンジニアリングサービスのご案内 46
- 改定履歴 47

2. EMP-ARM9 のシステム

EMP-ARM9 のシステムを大別すると、RedBoot とユーザプログラム (TOPPERS カーネル+アプリケーション) の 2 つから構成されます。RedBoot は、RedHat 社で開発されユーザプログラムのダウンロードや起動するための Bootloader として機能します。カーネルは、TOPPERS/JSP カーネルを採用しています。また、TCP/IP プロトコルスタックとしては ITRON TCP/IP API 仕様に準拠した TINET を採用しています。

2.1 RedBoot

RedBoot の機能には、ユーザプログラムのダウンロードを行なう機能、ユーザプログラムを起動、実行を行なう機能があります。ユーザプログラムのダウンロードは、Ethernet 又はシリアル経由で外部 SDRAM へ書き込むことが出来ます。ユーザプログラムの起動は、プログラムの開始アドレスにジャンプして動作を開始します。RedBoot 自身は、出荷時に書き込み済みですが、付属 CD にも添付されておりますので JTAG ICE 等を利用して書き込むことも可能です。

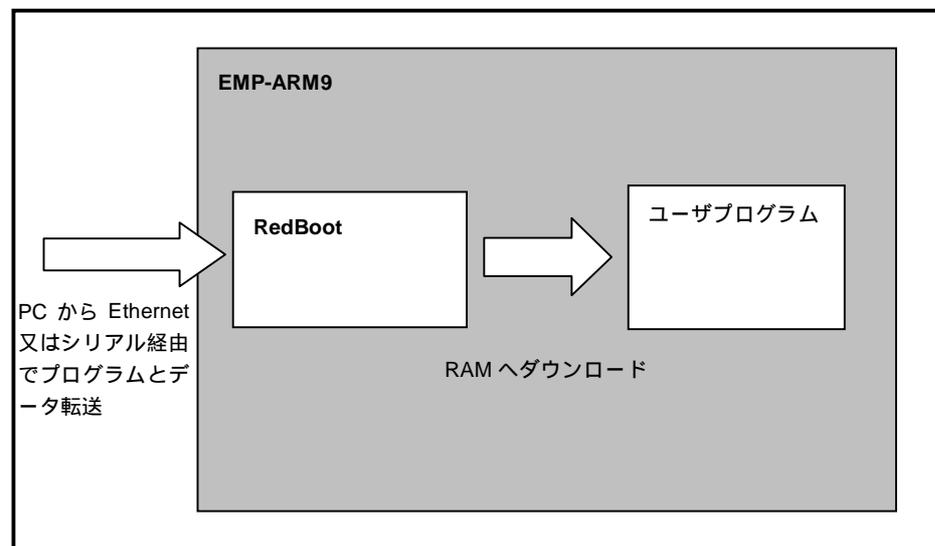


Fig 2.1-1 RedBoot の動作イメージ (ユーザプログラムダウンロード)

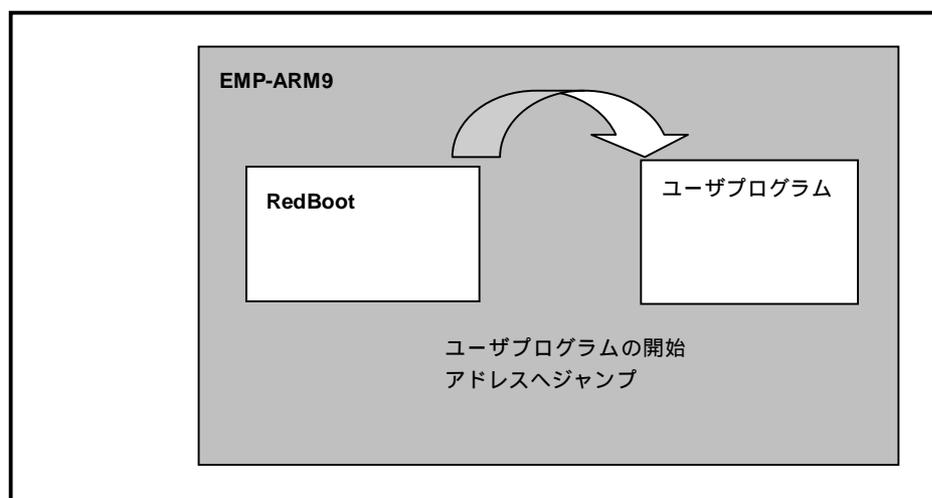


Fig 2.1-2 RedBoot の動作イメージ (ユーザプログラム起動モード)

2.2 TOPPERS/JSP

TOPPERS/JSP は、リアルタイム性を必要とするアプリケーションに最適なスケジューリング方式を採用している RTOS(リアルタイムオペレーティングシステム)の1つです。

RTOS のリアルタイムという言葉の意味は、応答速度が速いというわけではなく、システムに課せられた時間的制約を満足できることを表します。

RTOS では、リアルタイム性の実現も含めた主な機能として、以下の機能があります。

- ・マルチタスク機構
- ・タスク間同期、通信機構
- ・割り込み管理
- ・例外管理
- ・時間管理
- ・メモリ管理

2.2.1 マルチタスク機能

複数のタスクを同時に実行させる環境を提供するもので、CPU の仮想化 / 多重化を行う機能です。

タスクとは、マルチプログラミング環境における独立した並行実行単位のことを言います。

複数あるタスクに以下の規則 (スケジューリングポリシー) にしたがって CPU の使用权を与え、マルチタスク環境を実現しているとともにリアルタイム性も確保しています。

- ・その時点でもっとも高い優先度のタスクを実行する。
- ・その時点でもっとも高い優先度のタスクの処理は中断されてはいけない。
- ・同一優先度のタスクの場合は、先着順とする。
- ・優先度の高いタスクが実行可能になった場合、それよりも低い優先度のタスクが処理の途中でも、タスクを切り替えて高い優先度のタスクを実行する。

上記のような、優先度の低いタスクを中断させて高い優先度のタスクを動作させるスケジューリングポリシーをプリエンティブな優先度ベースのスケジューリングといい、多くのリアルタイム OS がサポートするスケジューリングポリシーです。

タスクの動作状態として、 μ ITRON では以下の5つを定義しています。また、状態遷移図を Fig 2.2-1 に示します。

1) 実行状態

CPU が割り当てられタスクを実行している状態。この状態のタスクを特にカレントタスクと呼びます。

2) 実行可能状態

タスクを実行する条件は整っているが、他の優先度が高いタスク、もしくは同一優先度で先に実行可能状態となったタスクが実行状態であるため、実行できない状態。

3) 広義の待ち状態

実行できる条件が整わないため実行できない状態。広義の待ち状態には、以下の3つの状態に分類されます。

・待ち状態

何らかの条件が整うまでタスクみずから実行を中断した状態。

・強制待ち状態

他のタスクによって強制的に実行を中断させられた状態。

・二重待ち状態

待ち状態と強制待ち状態が重なった状態。

4) 休止状態

タスクが未だ起動されていないか、終了した状態。

5) 未登録状態

タスクが未生成か、削除された後のシステムに存在しない仮想的な状態。

2.3 TINET

TINET は苫小牧高等専門学校情報工学科で研究・開発されたオープンソースの組み込み用 TCP/IP プロトコルスタックです。組み込みシステムの使用が考慮され、厳しいメモリ容量でも動作するように設計されています。アプリケーションプログラムとのインターフェース API は ITRON TCP/IP API が採用されています。

Fig 2.3-1 に示すように、TINET は、TOPPERS カーネルに組み込まれるのではなく、TOPPERS カーネルとアプリケーションの間で位置するミドルウェアです。

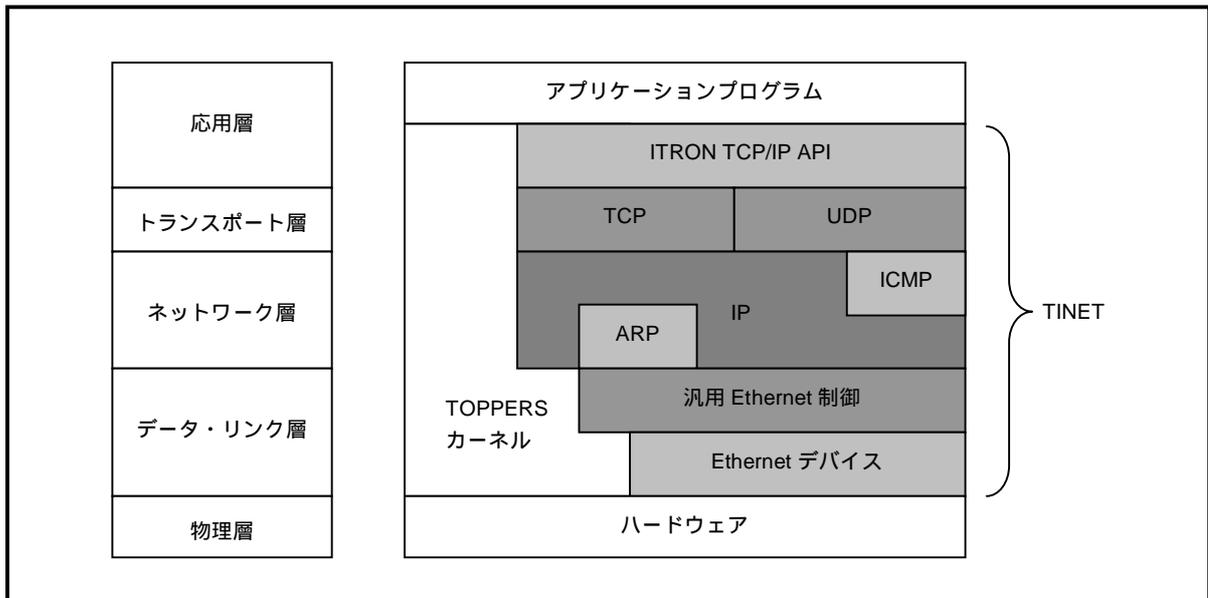


Fig 2.3-1 TINET の構成

TINET では、固定長メモリプールから割り当てるネットワークバッファを用いて、プロトコルスタック内における各層間のデータの受け渡しを行います。

送受信ともに割り当て前にデータサイズを確定し、プロトコルスタック内では、ネットワークバッファの動的メモリ操作は行いません。

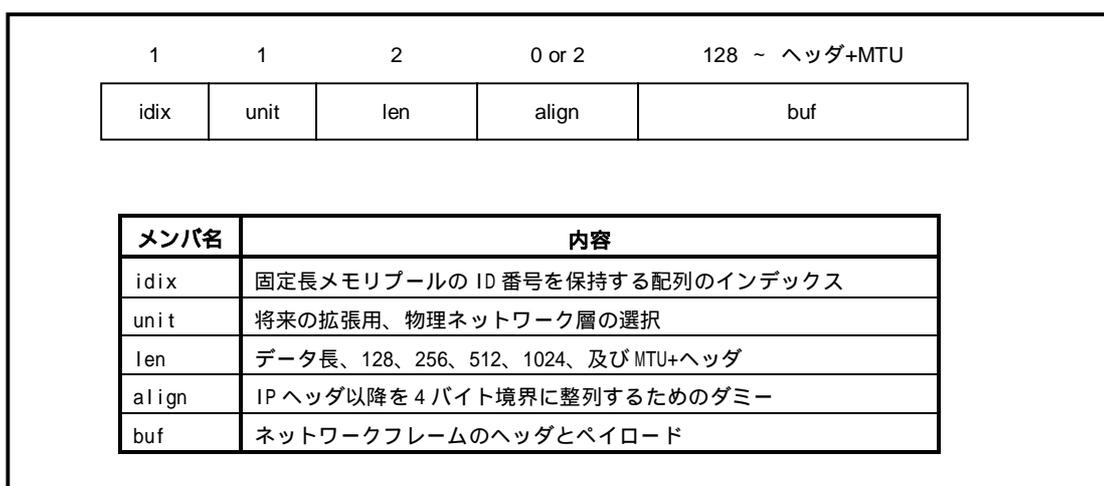


Fig 2.3-2 ネットワークバッファの構造

2.4 Eclipse 概要

Eclipse は IBM が開発したフリーの統合開発環境で、Java や C/C++、PHP、XML などのプラグインを使用することによって様々な言語の開発環境として使用することができます。Eclipse 自体は Java で書かれているため、Eclipse を動作させるには Java 仮想マシン (JVM) をはじめとする Java ランタイム環境 (JRE) が必要となります。

TOPPERS の開発では、C 言語を使用するため、C/C++ のプラグインである C/C++ Development Toolkit (CDT) を利用します。

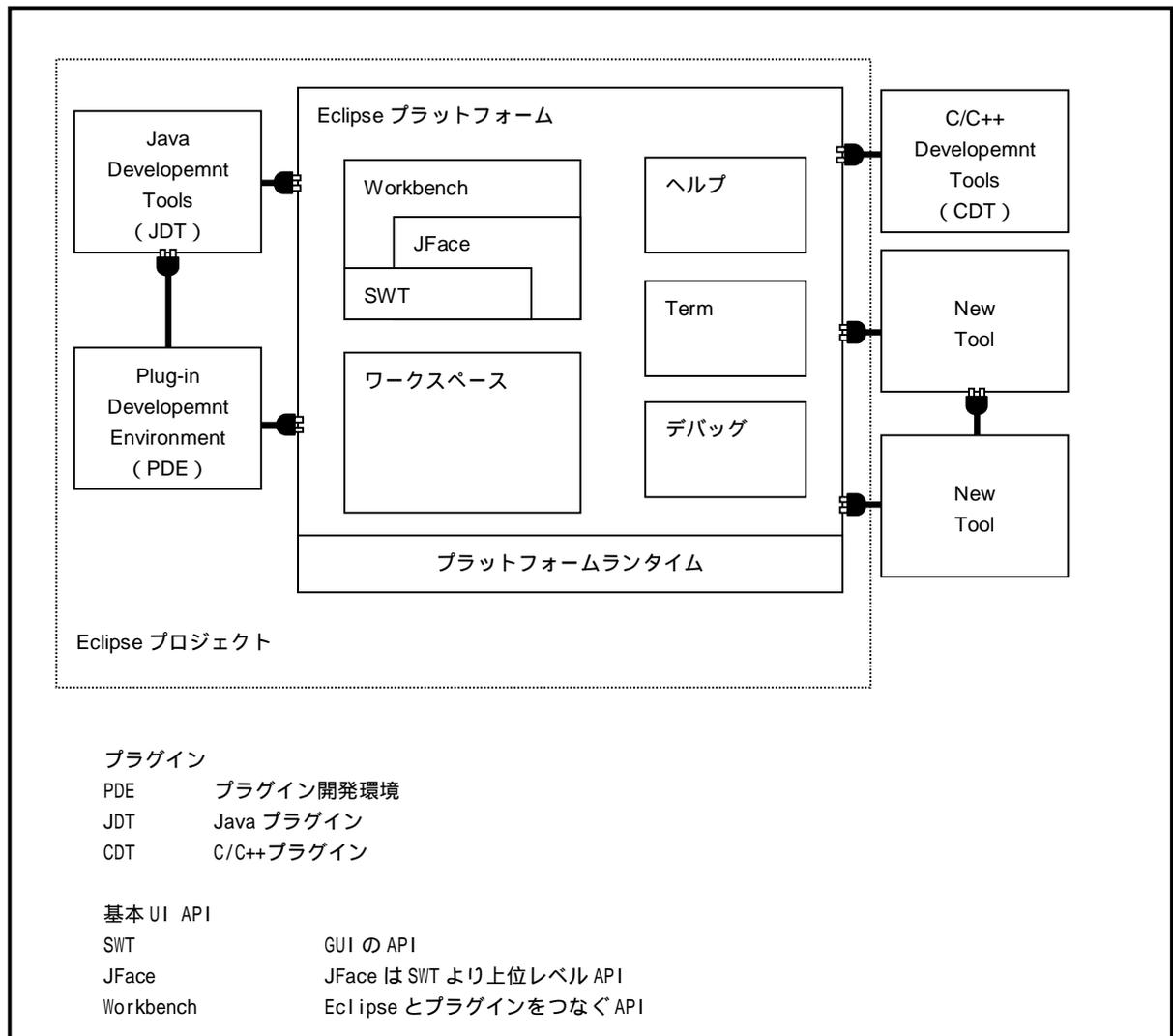


Fig 2.4-1 Eclipse 概要

3. クロス開発環境

3.1 クロス開発環境の概要

EMP-ARM9 上で動作する TOPPERS プログラムを作成するには Windows 上で動作する PC/AT 互換機上にクロス開発環境を構築する必要があります。本システムでは添付 CD のインストーラにより全ての開発環境が一括して構築されます。

一般的に、クロス開発環境を構築するにはホスト PC (Windows) 上にターゲット用の下記のパッケージをインストールします。

- ・ GNU binary utilities (アセンブラ、リンカ等)
- ・ GNU Compiler Collection (クロスコンパイラ・プリプロセッサ、標準ライブラリ等)
- ・ newlib (C 標準ライブラリ等)

これらパッケージによりターゲット用の実行ファイルを作成することができます。実行ファイルはホスト PC からターゲットシステムにダウンロードし動作を確認します。

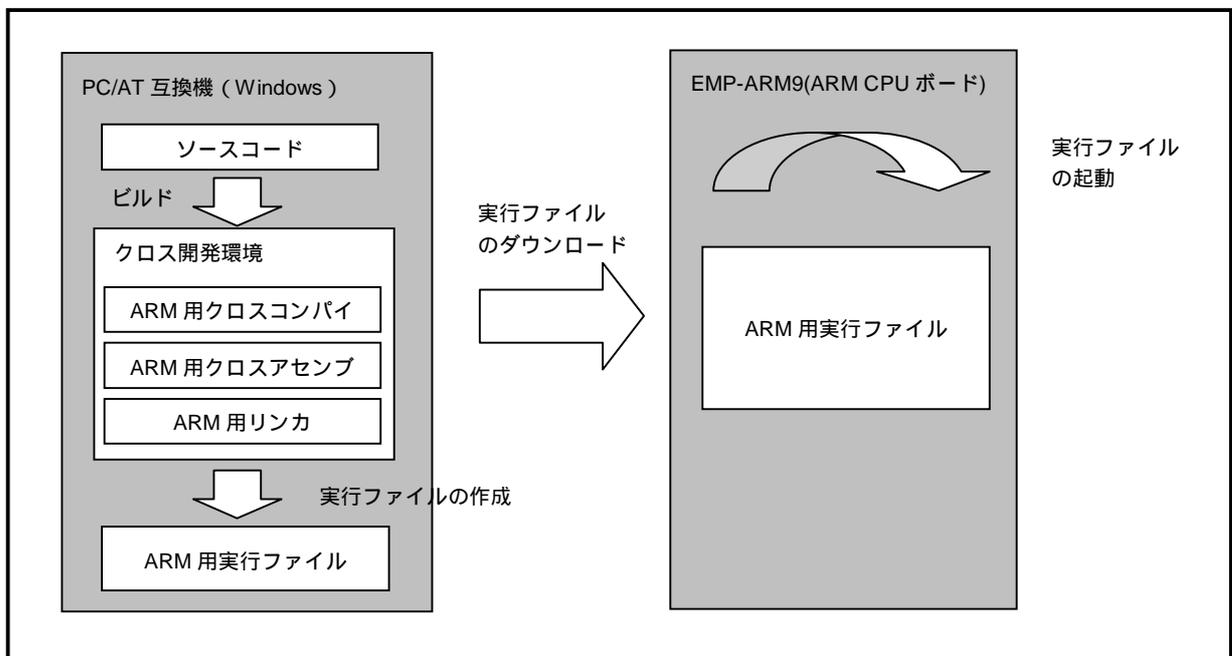


Fig 3.1-1 クロス開発環境

3.2 動作環境

クロス開発環境に必要な環境を次表に示します。

使用機器等	必要環境
パーソナルコンピュータ	PC/AT 互換機
OS	Windows2000 / XP
空き容量	600MB 以上
メモリ	128MByte 以上を推奨
ソフトウェア	ターミナルソフト
CD ドライブ	CD-R 読み込み可能なドライブ
その他	シリアルポート 1CH LAN ポート 1CH

Table 3.2-1 クロス開発の動作環境

3.3 インストール CD の構成

EMP-ARM9 のクロス開発には、TOPPERS カーネルソース、クロスコンパイラ等が必要です。

これらは、弊社ホームページ及び関連リンクからダウンロードするか、本 CD-ROM から入手することができます。

以下では、本 CD-ROM を対象に説明します。

(TOPPERS 開発環境は CD-ROM の¥Toppers ディレクトリ以下に収録されています)

インストール CD	
¥Toppers ディレクトリ	
-- ¥installer	
-- setup.exe	・開発環境インストールプログラム
-- ¥license	・ライセンスファイル
-- ¥module	
-- sample_x_x.srec	・サンプルプログラムの S レコードファイル
-- ¥source	
-- sample.zip	・サンプルプログラムのソースファイル
-- toppers.zip	・TOPPERS/JSP カーネル、TINET のソースファイル
-- ¥specifications	・µITRON4.0 仕様書, TOPPERS/JSP マニュアル, 内蔵デバイス API マニュアル
-- ¥manual	・マニュアル
-- index.html	・ブラウザ表示用 index
-- ¥index_images	・ブラウザ表示用 イメージファイル
`-- EMP-ARM9_update.txt	・EMP-ARM9 関連更新履歴

Table 3.3-1 CD-ROM 収録内容

4. EMP-ARM9 の起動

本章では、EMP-ARM9 に搭載されている TOPPERS/JSP の起動から簡単な動作確認までを説明します。

4.1 動作環境

TOPPERS/JSP の起動を確認するためには、EMP-ARM9 と EMP-ARM9 COM ボード以外に以下の環境が必要です。

ホスト PC

EMP-ARM9 用の TOPPERS/JSP では PC をコンソール端末として使用しますので、TOPPERS/JSP の起動を確認するためには、シリアルポートが使用可能な PC が必要となります。PC では、ハイパーターミナル等のターミナルソフトウェアを動作させます。

電源

EMP-ARM9 本体に必要な電源は DC5V ± 5% です。単体で動作させる場合には 1A 程度の電源を用意してください。

LAN

EMP-ARM9 をネットワークに接続する場合は、LAN ケーブルを接続してください。直接ホスト PC と接続する際はクロスケーブル、ハブを介してネットワークに接続する際はストレートケーブルをご使用ください。

LAN ケーブルは、10/100BASE-TX 対応 (UTP カテゴリ 5) ケーブルをご利用ください。

使用機器等	環 境
CPU ボード	EMP-ARM9, EMP-ARM9 COM
HOST PC	PC/AT 互換機
OS	Windows2000 / XP
メモリ	使用 OS による
ソフトウェア	ターミナルソフト
ドライブ	CD-R 読み込み可能なドライブ
USB ポート	1 ポート
LAN ポート	10Base-T 1 ポート
シリアルポート	1 ポート
RS232C ケーブル	クロスケーブルを使用
シリアル変換コネクタ	EMP-ARM9 COM 付属品
ミニ USB ケーブル	サンプルプログラムの動作確認に必要
LAN ケーブル	ホスト PC と接続時はクロスケーブルを使用 ハブと接続時はストレートケーブルを使用
電源	DC5V ± 5% 1A 以上

Table 4.1-1 EMP-ARM9 の推奨動作環境

4.2 ターミナルと Ethernet の接続

ホスト PC と EMP-ARM9 を下図のように接続してください。

LAN ネットワークに直接接続する場合は、ネットワーク管理者と相談し、設定に注意して接続してください。

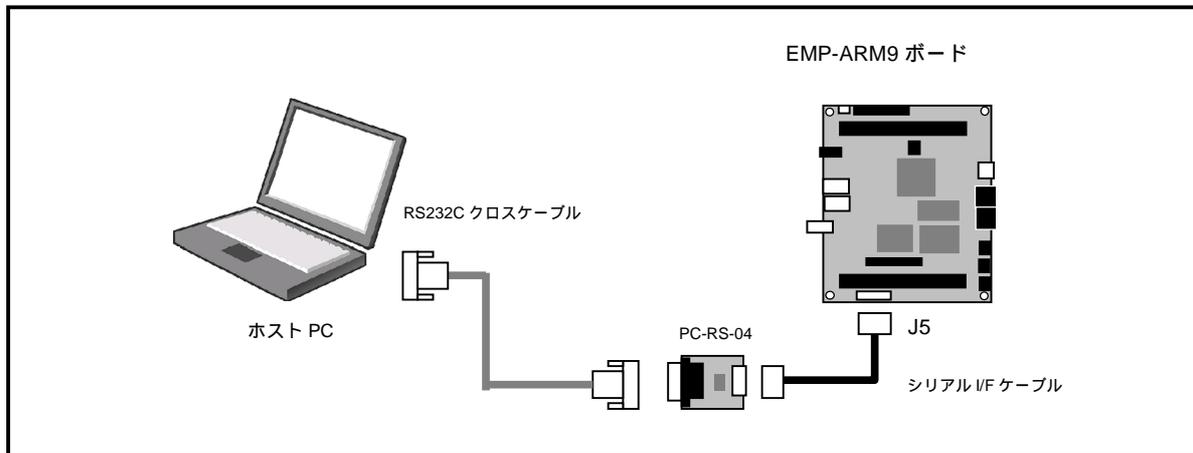


Fig 4.2-1 EMP-ARM9 接続例

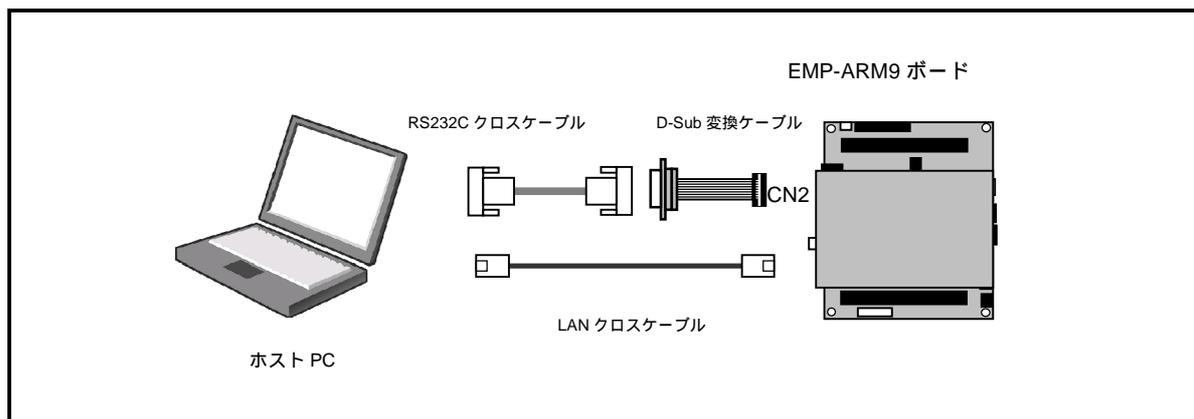


Fig 4.2-2 EMP-ARM9COM 基盤付 EMP-ARM9 接続例 (PC に接続する場合)

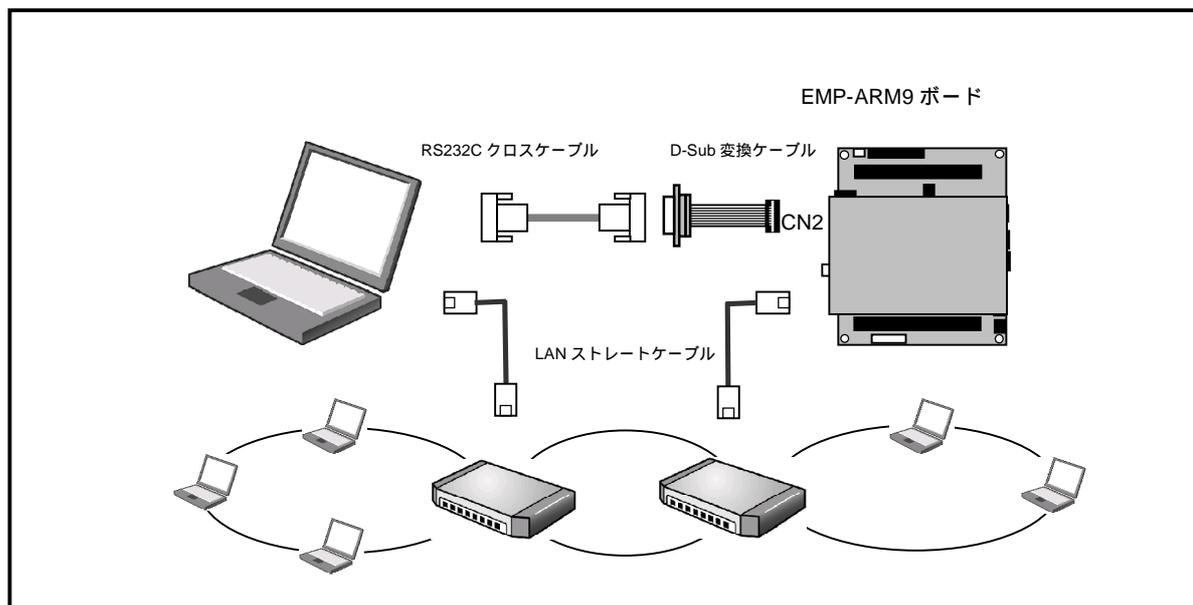


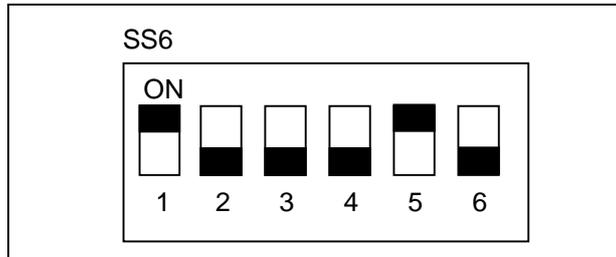
Fig 4.2-3 EMP-ARM9COM 基盤付 EMP-ARM9 接続例 (HUB に接続する場合)

4.4 起動と動作確認

EMP-ARM9 は出荷時の設定状態にしてください。出荷時状態にするには RedBoot のマニュアルを参照してください。

「4.2 ターミナルと Ethernet の接続」にしたがって、ホスト PC と EMP-ARM9 ボードを接続します。
PC のハイパーターミナルを起動します。

EMP-ARM9 ボードのディップスイッチを以下の設定にします。



EMP-ARM9 ボードの電源を投入し、電源 SW (SW5) を押します。

EMP-ARM9 ボードが正常に起動した場合には、RedBoot のログが出力されコマンド入力待ち状態になります。

ログ出力画面

```
+DP83902A - eeprom ESA: 00:0c:7b:22:01:09
Ethernet eth0: MAC address 00:0c:7b:22:01:09
IP: 192.168.5.181/255.255.240.0, Gateway: 192.168.1.253
Default server: 0.0.0.0

RedBoot(tm) bootstrap and debug environment [ROM]
Non-certified release, version UNKNOWN - built 10:25:30, Apr 19 2006

Platform: EMP-ARM9 system (ARM926E-JS<AP4010>) Version 1.0
Copyright (C) 2000, 2001, 2002, 2003, 2004 Red Hat, Inc.

RAM: 0x00000000-0x02000000, [0x000142a8-0x01fed000] available
FLASH: 0x10000000 - 0x10400000, 64 blocks of 0x00010000 bytes each.
RedBoot>
```

サンプルプログラムのダウンロード

添付 CD-ROM の¥Toppers¥module ディレクトリより、sample_jsp.srec を XMODEM にてダウンロードし実行します。
下記手順に従い実行して下さい。

```

RedBoot> load -t -m ymodem
CCCCCCCC Entry point: 0x2010003c, address range: 0x00100000-0x00106cdf
xyzModem - CRC mode, 654(SOH)/0(STX)/0(CAN) packets, 8 retries
RedBoot> act 0x2010003c
physical start address 0x2010003c
JSP Kernel Release 1.4 (patchlevel = 2) for EMP-ARM9+COM (Apr 24 2006, 19:35:49)
Copyright (C) 2000-2003 by Embedded and Real-Time Systems Laboratory
                        Toyohashi Univ. of Technology, JAPAN
System logging task is started on port 1.
Sample program starts (exinf = 0).
task1 is running (001). |
task1 is running (002). |
task1 is running (003). |
task1 is running (004). |
task1 is running (005). |

```

(A) “load -t -m ymodem ” と入力しリターンキーを押します。

次にハイパーターミナルのメニューより「転送」「ファイルの送信」を選択し、プロトコルを YMODEM、ファイル名を参照ボタンにより “sample_jsp.srec” を選択し送信を実行します。

(B) 転送が終了し、プロンプトが表示されたら “act [スタートアドレス] ” と入力し、リターンキーを押します。

[スタートアドレス] は YMODEM 転送後に表示された ,Entry point の値を入力します。

(C) コンソール上に TOPPERS/JSP の開始ログが表示され、サンプルプログラムの動作状態がコンソールに出力されます。サンプルプログラムは TOPPERS/JSP の基本的な動作が確認できます。キーボードよりコマンドを入力することにより、タスクの動作を制御できます。操作の詳細については sample1.c のコメントを参照して下さい。

5. 開発手順

以下に統合開発環境 Eclipse を使用した、TPPERS/JSP のサンプルアプリケーションの開発手順を記載致します。

TPPERS/JSP カーネルの仕様並びに CPU 内蔵デバイスの API 関数等については付属 CD の¥Toppers¥specifications フォルダ内の資料を参照して下さい。また、µiTRON 全般に関わる詳細は、別途書籍等を参照いただきますようお願い致します。

5.1 Eclipse の起動

Windows のスタートメニューから、[プログラム] - [AlphaProject] - [EMP-ARM9] - [Eclipse] を選択します。
(インストール時にグループをデフォルトの¥AlphaProject を指定した場合です。)

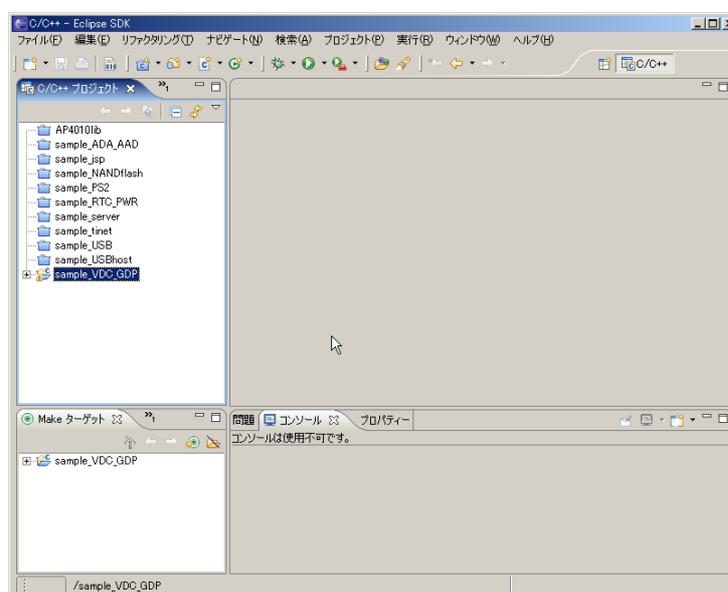
起動ロゴの表示後に使用するワークスペースの選択ダイアログが表示されます。

ここでは、C:¥AlphaProject¥EMP-ARM9¥Eclipse¥workspace を指定して[OK]ボタンを押します。

(インストール時にインストール先をデフォルトの C:¥AlphaProject に指定した場合です。)



初回起動時には、以下のような状態で Eclipse が起動します。



プロジェクト説明

インストール直後では、13 のプロジェクトがあります。

- AP4010lib : AP4010 内蔵デバイスライブラリです。*1
- Sample_xxx : サンプルプログラムを作成することができるプロジェクトです。
AP4010 の内蔵デバイス制御サンプルが 11 種類あります。
- Base : このプロジェクトは、新しいプロジェクトを作成する時に利用します。

*1 コンパイラバージョン等の変更があった場合、再ビルドします。

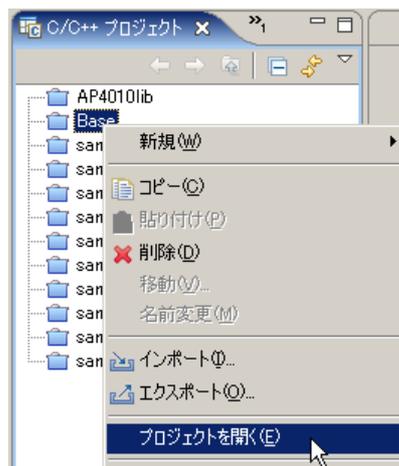
5.2 新規プロジェクト作成

Base プロジェクトを利用して EMP-ARM9 ボードの LED1 が 1 秒おきに点滅するアプリケーションを作成します。

Base プロジェクトから新規プロジェクト作成

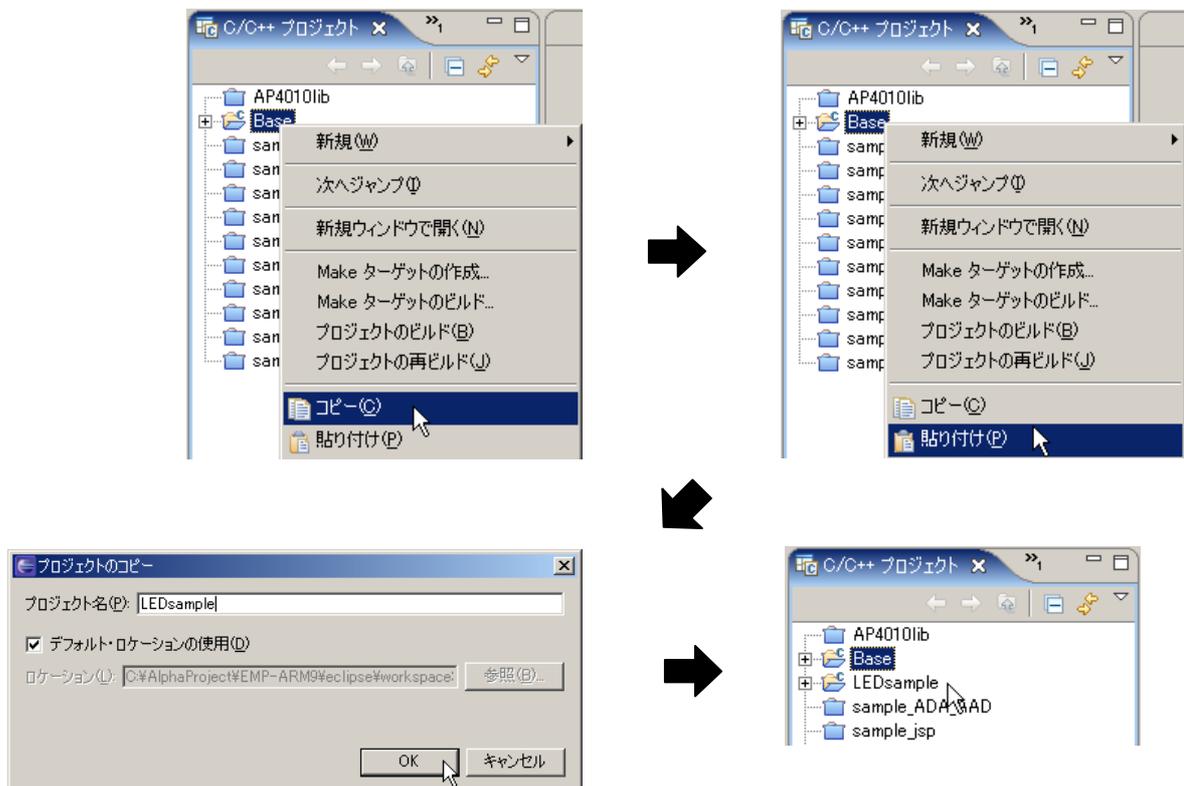
Base プロジェクトを開きます。

ウィンドウの左側にあるツリーウィンドウの Base の文字列部分を左クリックして選択し、右クリックで表示されるメニューから「プロジェクトを開く(E)」を選択します。



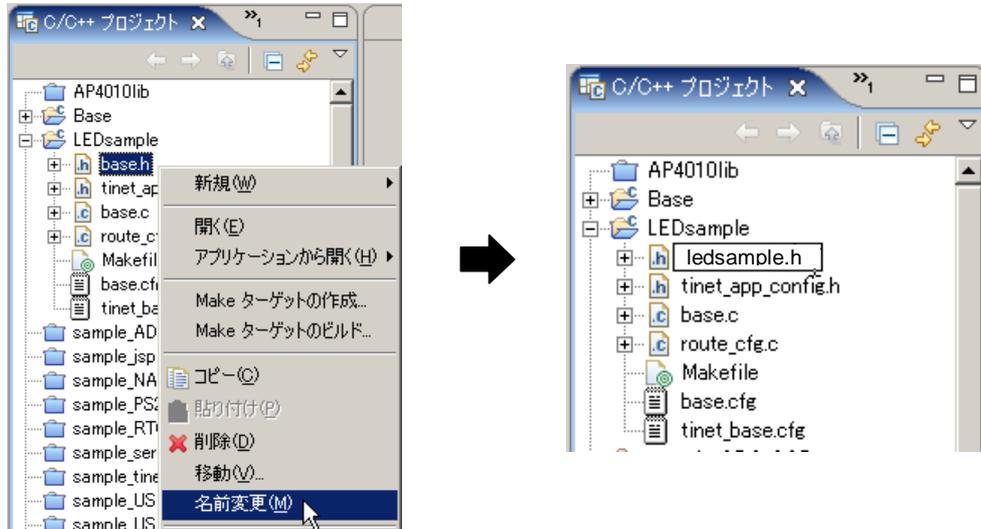
Base プロジェクトをコピーします。

手順 と同様な作業を行い、メニューから「コピー(C)」を選択します。その後、もう一度メニューを表示し「貼り付け(P)」を選択します。「プロジェクトのコピー」というダイアログが表示されて、コピー先のプロジェクト名の入力を求められますので、LEDsSample と入力して[OK]ボタンを押します。正常に作業が終了した場合、ツリーウィンドウに LEDsSample が追加されます。



ファイル名を変更します。

まず、変更するファイル名を左クリックして選択し、右クリックで表示されるメニューから「名前変更(M)」を選択します。その後、選択した名前に枠が表示されますので、新しいファイル名を入力しEnterキーで決定します。



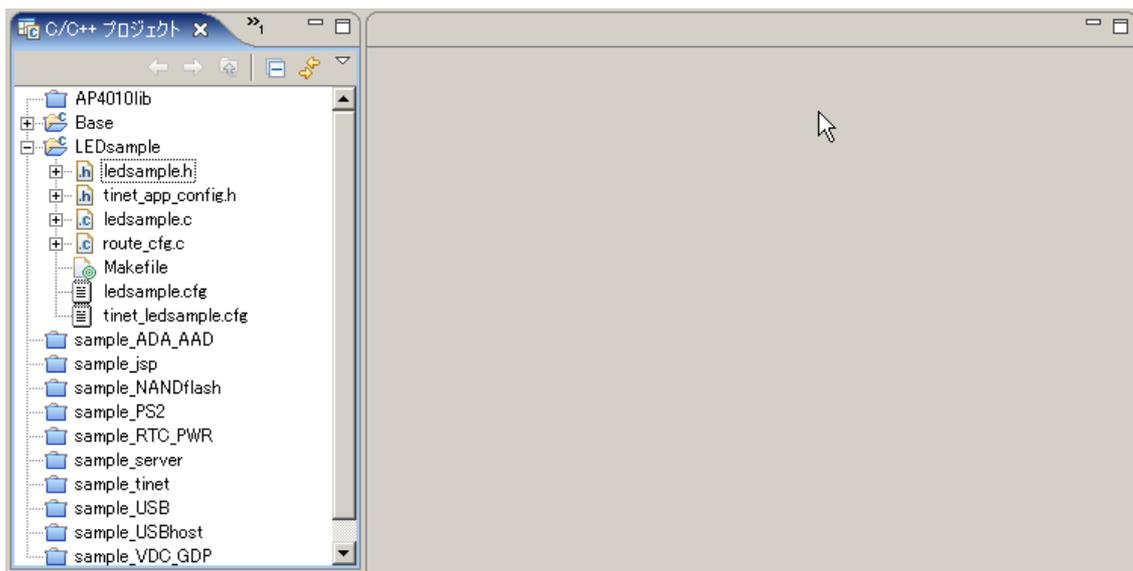
同様の作業を続けて行います。ファイル名は以下の名前に変更します。

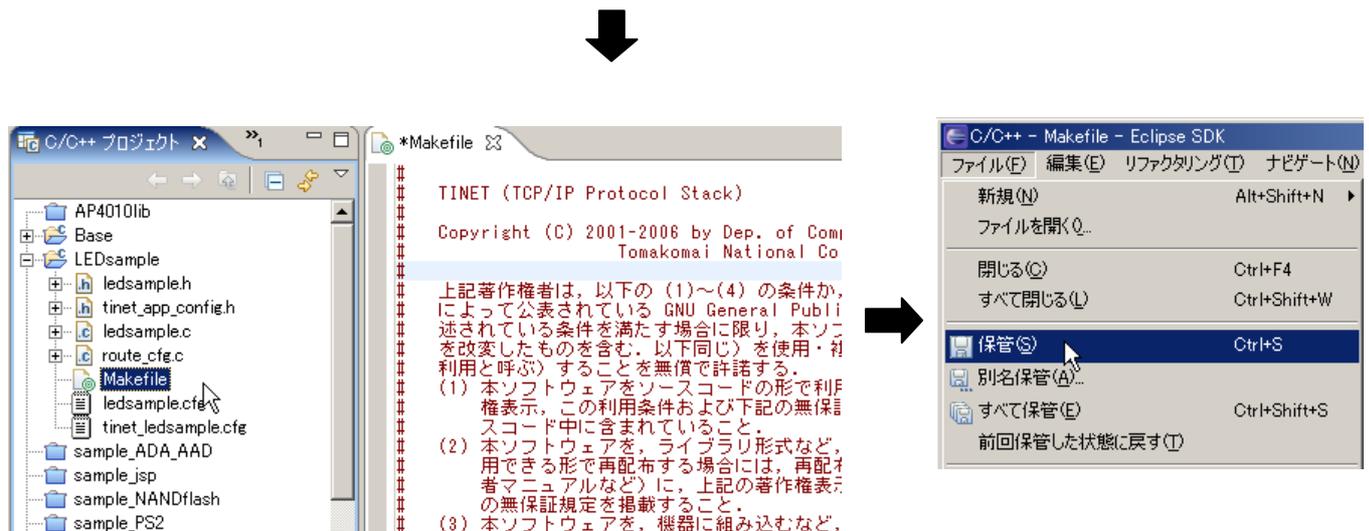
base.c	ledsample.c
base.cfg	ledsample.cfg
base.h	ledsample.h
tinet_base.cfg	tinet_ledsample.cfg

ファイル内容を変更します。

ファイル名を変更したので、ファイル内の影響する記述部分を修正します。

修正するファイル名をダブルクリックをすると、右側にファイル内容が表示されますので、必要な箇所を修正して、「ファイル(F)」メニューから「保管(S)」を選択して、保存します。

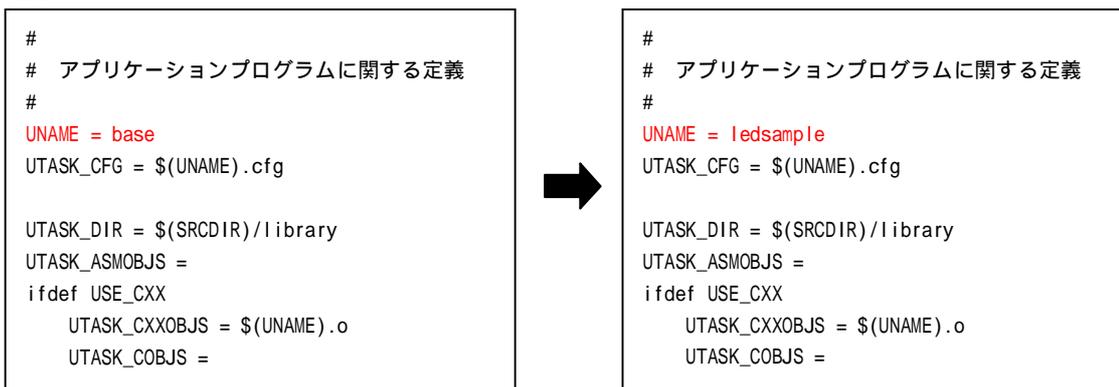




以下に修正を加える箇所をすべて記述します。

左側が修正前の内容で、右側が修正後の内容です。赤字で記述されている部分が該当箇所です。

Makefile



5.3 ビルド方法

新規に作成した LEDsample プロジェクトを、Eclipse によってビルドする方法を説明します。

プロジェクトを開きます。

LEDsample の左側に表示されているアイコンが閉じているアイコンの場合には、プロジェクトを開く必要があります。

LEDsample プロジェクトを左クリックで選択し、右クリックで表示されるメニューから、「プロジェクトを開く(E)」を選択します。

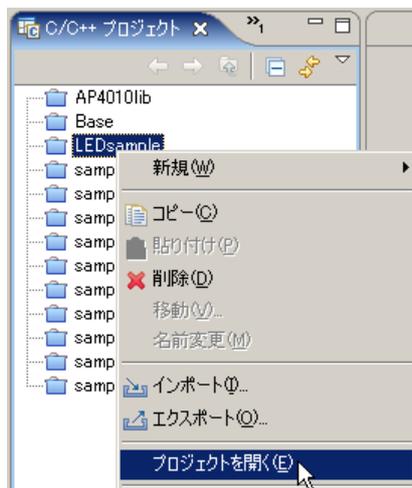


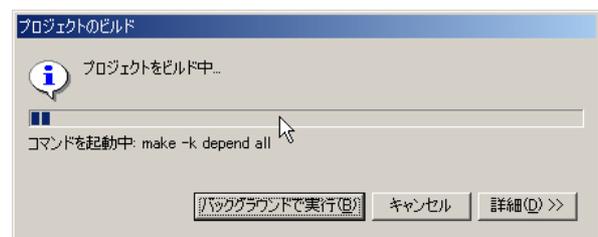
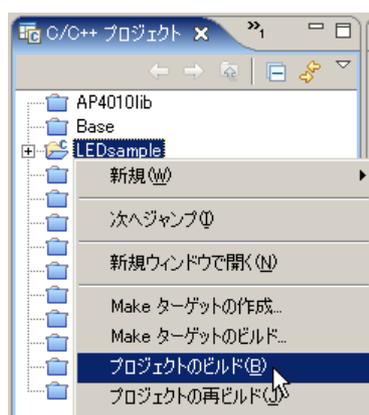
Fig 5.3-1 閉じてるアイコン



Fig 5.3-2 開いてるアイコン

LEDsample プロジェクトをビルドします。

LEDsample プロジェクトを左クリックで選択し、右クリックで表示されるメニューから「プロジェクトのビルド(B)」を選択します。



備考

・プロジェクトのビルドにより依存ファイルの作成をします。もしビルドに失敗した場合は、エラーのあったファイルを修正し「プロジェクト」メニューより「クリーン」を選択し実行して下さい。その後、再度ビルドを実行して下さい。

・初期状態や中間ファイルを削除した状態でビルドを行った場合、正常にビルドできないことがあります。その時には、もう一度「プロジェクトの再ビルド(B)」を行うと正常にビルドできます。

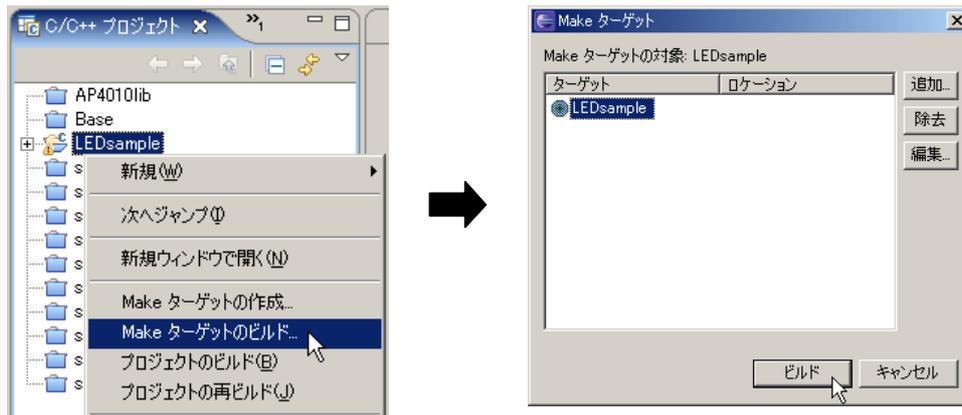
Make ターゲットのビルド。

LEDsample プロジェクトを左クリックで選択し、右クリックで表示されるメニューから「Make ターゲットのビルド...」を選択します。

「Make ターゲット」ダイアログが表示され、ターゲット「LEDsample」を選択し[ビルド]ボタンをクリックします。

正常にビルド(リンク)が終了すれば、以下のファイルが C:\AlphaProject\EMP-ARM9\workspace\LEDsample フォルダに作成されます。

```
ledsample.exe      : ELF フォーマットのデバッグ情報付きバイナリファイル
ledsample.srec    : Sレコードファイル
```



コンソールウィンドウに下記のようなメッセージが表示されたら、ビルドは正常に終了しています。

```

問題  コンソール  プロパティ
C-Build [LEDsample]
mv tmpfile2 ledsample.chk
arm-elf-gcc -mcpu=arm9tdmi -mlittle-endian -g -O2 -mno-apcs-float
-mstructure-size-boundary=8 -DEMP_ARM9 -DSUPPORT_INET4 -DSUPPORT_ETHER
-DSUPPORT_TCP -DTCP_CFG_LIBRARY -DUDP_CFG_LIBRARY -I. -I../jsp/include
-I../jsp/config/armv4/emp_arm9 -I../jsp/config/armv4 -I../jsp/tinet/netdev/if_ed
-I../jsp/tinet -nostdlib -mcpu=arm9tdmi -N -mlittle-endian
-Wl,-section-start,.vector=0x20100000 -Wl,-Ttext,0x2010003c
-Wl,-Tdata,0x21000000 -T ../jsp/config/armv4/emp_arm9/emp_arm9.ld -o
ledsample.exe ¥
start.o ledsample.o net_buf.o net_count.o net_subr.o net_timer.o
in_subr.o tinet_cfg.o route_cfg.o ip_input.o ip_output.o ip_icmp.o
tinnet_sys_config.o if.o ethernet.o if_ether.o if_ed.o tcp_input.o tcp_output.o
tcp_timer.o tcp_subr.o tcp_subr_cs.o tcp_subr_ncs.o timer.o serial.o logtask.o
log_output.o vasyslog.o t_perror.o strerror.o kernel_cfg.o ../AP4010lib/apiflib.a
libtinnet.a -lc libkernel.a -lgcc
arm-elf-nm ledsample.exe > ledsample.syms
arm-elf-objcopy -O srec -S ledsample.exe ledsample.srec
../jsp/cfg/chk -m ledsample.syms,ledsample.srec ¥
-obj -cs ledsample.chk -cpu armv4 -system emp_arm9

```

もし、ビルドに失敗した場合、問題の発生した部分を修正し「ファイル」メニューより「保管」を実行し、再度「Make ターゲットのビルド」を行なって下さい。

6. サンプルアプリケーション

Eclipse を使用した統合開発環境には、TPPERS/JSP ならびに TINET、アプローズテクノロジーズ社提供の AP4010 内蔵デバイスの API 関数を使用したサンプルアプリケーションが含まれています。以降ではサンプルアプリケーションの概要を説明致します。

EMP-ARM9 COM 基板が搭載された状態で動作することを前提としています。(COM 基板無しの場合は 6.2 章を参照)

尚、サンプルアプリケーションはその動作を保证するものではありません。

6.1 サンプルアプリケーションの概要

サンプルプログラムを動作させるには、EMP-ARM9 ボードと PC をシリアルで接続する必要があります。

PC 側のターミナルソフトは、ハイパーターミナルや TeraTerm 等をご使用下さい。(詳細は第 4 章を参照して下さい)

Eclipse よりサンプルプログラムの「プロジェクトのビルド」実行後、「Make ターゲットのビルド」を実行して下さい。

サンプルプログラムをビルド後に生成されるオブジェクトファイル名は jsp.exe(ELF 形式), jsp.srec(S フォーマット)です。

接続条件：115200bps,8bit,パリティなし,STOP 1bit,フロー制御無し

接続コネクタ：COM 基板無し = J5 (レベル変換必用：弊社 PC-RS-04 が接続可能です)

COM 基板有り = COM 基板上の CN2 (レベル変換不要：添付変換コネクタを使用)

No	プロジェクト名	機能概要
1	sample_ADA_AAD	内蔵 A/D、D/A コンバータのサンプルプログラムです。 CN4 AUDIO 出力 R/L に、それぞれ 440Hz のサイン波と MIC 入力(CN3)信号を出力します。 コンソールより割込み処理方式を選択して下さい。
2	sample_jsp	TPPERS/JSP のサンプルプログラムです。 ターミナルより、task の動作を制御可能です。Task の制御方法についてはソースファイルを参照して下さい。
3	sample_NANDflash	NAND フラッシュメモリのテストプログラムです。ターミナルより下記制御が可能です。 ページリード(r)、ページライト(w)、ブロック消去(e)、ブランクチェック(b) Bad Block チェック(c)、ID リード(i)、全ブロック消去(a) 本 NAND フラッシュメモリは、BadBlock が含まれます。事前に BadBlock をチェックしアプリケーション側で対処する必要があります。 BadBlock に対しては消去動作を行わないで下さい。
4	sample_PS2	PS2 マウスとキーボードのサンプルプログラムです。 CN2 にマウスを接続後、CN1 にキーボードを接続して下さい。デバイスからのデータをターミナルに出力します。 すべてのデバイスをチェックしておりませんので、修正が必要かもしれません。
5	sample_RTC_PWR	リアルタイムクロックと電源制御のサンプルです。 基板上の JSW3 は XSHT 側に設定して下さい。 ターミナルより時刻の設定が可能です。設定後は一秒ごとにターミナルに時刻が表示されます。 電源 SW(SW5)を押すとメッセージ表示 1 秒後に電源が切れます。
6	sample_server	拡張 COM 基板が必要です。TINET 付属のサンプルプログラムです。 WWW、TELNET 等が動作します。詳細は TINET のマニュアルを参照して下さい。 デフォルト IP アドレス 192.168.5.181,サブネットマスク 255.255.240.0 に設定されています。 tinet_appconfig.h に定義されています。
7	sample_tinet	拡張 COM 基板が必要です。TINET 付属のサンプルプログラムです。 エコーバックサーバが動作します。詳細は TINET のマニュアルを参照して下さい。 デフォルト IP アドレス 192.168.5.181,サブネットマスク 255.255.240.0 に設定されています。 tinet_appconfig.h に定義されています。

8	sample_usb	<p>USB バルク転送のサンプルプログラムです。</p> <p>本サンプルプログラムは汎用 USB ドライバを使用しています。ドライバ作者の意図を十分に理解した上でご使用下さい。</p> <ol style="list-style-type: none"> 1 . http://www.otto.to/~kasiwano/ より汎用 USB ドライバ(uusbdrv10.zip)をダウンロードしてください 2 . アーカイブファイル内の uusbdrv.dll と uusbdrv.sys を sample_USB\UUSB にコピーします 3 . サンプルプログラムを起動し、EMP-ARM9 とホスト PC を USB ケーブルで接続します 4 . ドライバが要求されたら UUSB ホルダ内の uusbdrv.inf ファイルにてドライバを登録します 5 . UUSB ホルダ内の p.bat を実行することにより EMP-ARM ボード内のメモリを表示します。
9	sample_USBhost	<p>拡張 COM 基板が必要です。HUDI デバイスのサンプルプログラムです。</p> <p>USB コネクタにマウス、キーボードを接続するとターミナルにデバイスからのデータを出力します。すべてのデバイスをチェックしておりませんので、修正が必要かもしれません。</p>
10	sample_VDC_GDP	<p>VDC と GDP のサンプルプログラムです。</p> <p>LCD または CN5 に TV モニタ等を接続すると、カラーバー表示がされます。SW1 を押すとマントヒビが表示されます。BitBlt とアルファブレンディングにより描画されています。</p>
11	Sample_LTC	<p>タッチパネルのサンプルプログラムです。</p> <p>キャリブレーション実行後、入力座標を LCD 上に表示します。</p>

注) サンプルプログラムは全て最適化オプションが無効の状態で作成されます。必要に応じて MakeFile の最適化オプションを修正して下さい。

著作権について

- ・本文書の著作権は（株）アルファプロジェクトが保有します。
- ・本文書の内容を無断で転載することは一切禁止します。
- ・本文書の内容は、将来予告なしに変更されることがあります。
- ・本文書の内容については、万全を期して作成いたしました。万が一不審な点、誤りなどお気づきの点がありましたら弊社までご連絡下さい。
- ・本文書の内容に基づき、アプリケーションを運用した結果、万が一損害が発生しても、弊社では一切責任を負いませんのでご了承下さい。

商標について

- ・Windows®の正式名称は Microsoft®Windows®Operating System です。
Microsoft、Windows、Windows NT は、米国 Microsoft Corporation.の米国およびその他の国における商標または登録商標です。
Windows®XP、Windows®2000 Professional、Windows®Millennium Edition、Windows®98 は、米国 Microsoft Corporation.の商品名称です。
本文書では下記のように省略して記載している場合がございます。ご了承ください。
Windows®XP は Windows XP もしくは WinXP
Windows®2000 Professional は Windows 2000 もしくは Win2000
Windows®Millennium Edition は Windows Me もしくは WinMe
Windows®98 は Windows 98 もしくは Win98
- ・その他の会社名、製品名は、各社の登録商標または商標です。



株式会社アルファプロジェクト
〒431-3114
静岡県浜松市東区積志町 834
<http://www.apnet.co.jp>
E-MAIL : sales@apnet.co.jp
