

# Multi-Media ARM9

## Linux-2.6.29版 Eclipse + GCC + open-JTAG

## マニュアル

株式会社日昇テクノロジー

http://www.csun.co.jp

info@csun.co.jp

2010/3/17



USB で ARM7 と接続

## copyright@2010-2011



第一章 MINI2440 ボードの概要	7
1.1 仕様	7
1.2 使えるデバイス例	10
1.3 付属アプリケーション例	12
第二章 インターフェースの説明	12
2.1 電源	12
2.2 ユーザーボタン	13
2.3 シリアルポート	13
2.4 液晶 LCD インターフェース	14
2.5 JTAG	14
2.6 GPIO	15
2.7 CMOS CAMERA	16
2.8 システムバス	17
2.9 リセット	
2.10 AD	
2.11 PWM ブザー	19
第三章 初体験(GUI)	20
3.1 タッチパネルの校正	20
3.2 日本語の設定	20
<b>3.3 MP3</b> の再生	22
3.4 ビデオの再生	23
3.5 ピクチャのビューと編集	24
3.6 SD と USB メモリの自動認識	25
3.7 ターミナル	26
3.8 ネットワークの設定	27
3.9 ping	27
3.10 LED テスト	
3.11 EEPROM テスト	29
3.12 PWM ブザー	
3.13 音声のレコーダー	31
3.14 USB カメラ	32
3.15 CMOS イメージセンサー	32
3.16 AD テスト	33
3.17 ボタン	33
3.18 手書き	34



3.19 Watch dog	35
3.20 回転	36
3.21 スタートアップ	37
3.22 USB GPS	
第四章 初体験(コンソール)	
4.1 パソコン側のハイパーターミナルの設定	
<b>4.2 MP3</b> の再生	41
4.3 USB メモリと外付けハードデスク	41
4.4 SD/MMC カード	42
4.5 シリアルポートでファイルを ARM9 にダウンロード	43
4.6 シリアルポートで ARM9 のファイルを PC に保存	45
4.7 LED 制御	46
<b>4.8</b> ボタンのテスト	47
4.9 シリアルポートのテスト	48
4.10 ブザー(PWM)のテスト	49
4.11 LCD のバックライト	49
4.12 I2C-EEPROM	50
4.13 AD テスト	50
4.14 CMOS イメージセンサー	51
4.15 ネットワーク機能	51
4.15.1 ウェブサーバー	51
4.15.2 Telnet と Ftp 機能	52
4.15.3 DNS と gateway の設定	52
4.15.4 MAC アドレスの設定	52
4.15.5 ネットワーク・ファイルシステム(NFS)のマウント	52
4.16. RTC の設定	52
<b>4.17 液晶(LCD)</b> 画面を取ります	53
4.18 USB 無線 LAN	53
第五章 Linux のクロス開発環境	55
5.1 クロス開発環境を構築	55
5.2 NFS サーバーを構築	56
5.3 NFS はルートファイルシステムとして起動	57
第六章 Linux 環境のアプリケーションを開発	57
6.1 Hello, World!	57
6.2 Hello,World をコンパイル	58
6.3 Hello,World を ARM9 ボードで実行	58



6.4 ほかのサンプル	
6.5 Qt/Embedded GUI プログラムを作る	59
第七章 Linux カーネルを再構築	60
7.1 カーネルのソースコードを解凍	60
7.2 Linux を再コンパイル	61
7.3 ドライバの場所	64
7.4 Linux カーネルのコンフィグ	66
7.4.1 LCD 液晶とバックライト	67
7.4.2 タッチパネル	70
7.4.3 USB マウスとキーボード	71
7.4.4 USB メモリ	72
7.4.5 汎用 USB カメラ	74
7.4.6 CMOS イメージセンサー(OV9650)	76
7.4.7 イーサネット	
7.4.8 USB 無線 LAN	
7.4.9 オーディオ	
7.4.10 SD/MMC メモリカード	
7.4.11 Watchdog	
7.4.12 LED	
7.4.13 ボタン	
7.4.14 PWM ブザー	
7.4.15 AD	
7.4.16 シリアルポート	90
7.4.17 リアルタイマーRTC	
7.4.18 I2C - EEPROM	
7.4.19 yaff2s ファイルシステム	
7.4.20 EXT2/VFAT/ NFS ファイルシステム	
7.4.21 USB-RS232 シリアルポート	
7.4.22 ARM7TDMI/LPC2148 との通信	
7.5 Linux 起動ロゴを作る	
7.6 yaffs ルートファイルシステムのイメージを生成	
7.7 Linux ドライバの開発入門	
7.7.1 簡単なドライバのソースコード	
7.7.2 コンフィグファイルを編集します	
7.7.3 Makefile を編集	
7.7.4 ドライバをコンパイルします	



7.7.5 ARM9 ボードでドライバをインストールします	
第八章 生成されたファイルを書き込む	
8.1 NOR Flash から起動	
8.2 USB ドライバのインストール	
8.3 NAND Flash のパーティション	
8.4 ブートロードの書き込み	
8.5 Linux のカーネルの書き込み	
8.6 ルート・ファイルシステムの書き込み	
8.7 NAND Flash のバックアップ	
8.8 NAND Flash のリストア	
8.9 メモリで Linux カーネルを直接に実行	
第九章 NOR Flash のブートロードを更新	
9.1 簡易 JTAG で書き込み	
9.1.1 H-JTAG をダウンロードとインストールします	
9.1.2 NOR Flash を書き込む	
9.2 Open-JTAG で書き込み	
9.2.1 ドライバをインストールする	140
9.2.2 書き込み	143
第十章 Web カメラストリーミング配信	145
10.1 MJPG-streamer のダウンロードとコンパイル	145
10.2 MJPG-streamer を mini/micro2440 ボードにインストール	145
10.3 Web ブラウザで Web カメラを見ましょう	146
第十一章 Eclipse + GCC + Open-JTAG	147
11.1 GCC ツールチェーン	147
11.2 Integrated Development Environment(Eclipse)	149
11.3 プロジェクトを作る	153
11.4 Eclipse プラグイン(Zylin Embedded CDT)インストール	155
11.5 ビルドの設定	
11.6 ビルド	
11.7 GDB の設定	165
11.8 OpenOCD の設定	171
11.9 デバッグ	175
11.10 デバッグ終了	



- ※ 使用されたソースコードは<u>http://www.csun.co.jp/</u>からダ ウンロードできます。
- ※ この文書の情報は、事前の通知なく変更されることがあり ます。
- ※(株)日昇テクノロジーの書面による許可のない複製は、い かなる形態においても厳重に禁じられています。



## 第一章 MINI2440 ボードの概要

#### 1.1 仕様



※ 液晶は 3.3V/5V 二種類があります。MINI2440 の電圧選択ジャンパ ー(J2)は必ず正しく設定されなければなりません。

CPU プロセッサー

 ARM920Tコアを採用したサムソン(SAMSUNG)社のS3C2440A、周波数400MHz、 最高周波数533MHz。



メモリ

- 64MB SDRAM, 32 ビット幅データ・バス, SDRAM の最高周波数 100MHz
- 64MB NAND Flash メモリ
- 2MB NOR Flash メモリ

#### 液晶(LCD)

- 4線式抵抗膜方式のタッチパネルのインターフェース
- 標準の LCD I/F を持って、3.5"から 12.1"までの各種液晶パネル(黒白、STN、TFT、 最高分解能 1024\*768)に対応します。

インターフェース

- 10M/100MBase-T Ethernet RJ45(DM9000) x 1
- RS232 シリアルポートx3
- USB1.1 ホストx1
- USB1.1 スレーブ x 1
- MMC/SDメモリカードのソケットx1
- ステレオ・オーディオの出力x1
- マイクの入力x1
- 10 ピンの JTAG(2mm DIP ピッチ)
- ユーザーLED x 4
- ユーザーボタンx6
- PWM 制御の圧電ブザーx1
- 可変抵抗、A/D のテストの為に x1
- I2C バスの AT24C08、I2C バスのテストの為に x 1
- 20 ピン CMOS カメラのインターフェース(2mm DIP ピッチ)
- RTC のバッテリーバックアプ
- 34 ピン GPIO(2mm DIP ピッチ)
- 44 ピンのシステムバス(2mm DIP ピッチ)

#### 搭載した OS

- Linux2.6.29 + Qtopia2.2.0
- WindowsCE.NET 5.0
- uCOSII



#### 外形寸法

• 100×100(mm) 突起物は除く

#### 供給電源

 5VDC 電源、プラグ 1.3mmφ、極性はセンタープラス ⊖ ● ● です。 電源ス イッチと電源指示 LED 付き

スイッチ S2 はボードの動作モデルを選択します。一つは Nor Flash から起動です。もう一つは Nand Flash から起動です。この二つの起動モデルでシステムのアドレスが異なります。



デフォルトの設定は Nand Flash から Linux を起動します。



## 1.2 使えるデバイス例



USB カメラ (SPACXX 又は UVC に対応)



USB 無線 LAN 装置



USB マウスとキーボード



外付けハードディスク





SD/MMCメモリ



USB HUB



USB シリアルポート





7インチ液 晶



1024X768 VGA ※ 付属のドライバ以外は、使えない可能性があります。



## 1.3 付属アプリケーション例



## 第二章 インターフェースの説明

### 2.1 電源





## 2.2 ユーザーボタン



	K1	K2	K3	K3	K4	K5
割り込み	EINT8	EINT11	EINT13	EINT14	EINT15	EINT19
GPIO	GPG0	GPG3	GPG5	GPG6	GPG7	GPG11
他の機能	なし	nSS1	SPIMISO1	SPIMOSI1	SPICLK1	TCLK1
CON12	CON12.1	CON12.2	CON12.3	CON12.4	CON12.5	CON12.6

※ CON12.7 は 3.3V 電源、CON12.8 は GND です。

## 2.3 シリアルポート





## 2.4 液晶 LCD インターフェース



LCD インターフェースは最大 RGB(888)の液晶をサポートします。37,38,39,40 ピンは四線 抵抗式のタッチパネルの入力です。J2 は液晶給電の選択、5V 又は 3.3V の液晶に対応しま す。

#### 2.5 JTAG





## 2.6 GPIO

CON4				12	
VDD5V	,	VDD33V			
GND					
AIN0 AIN2					
EINTO					
EINT2		0 11 12 EINT3	_		
EINT4 EINT6		C 13 14 C EINTS		·	
EINT9		0 15 16 EINT11		8	
EINT13	3	0 19 20 D EINT14	[		
EINT1	8	0 21 22 D EINT19		10 mm	
SPIMIS	50	O 23 24 D SPIMOS	1		
SPICL	ĸ	C 27 28 C 135_SPI			
GPBO	-	0 29 30 0 12030A GPB1			
CLKOU	TO	CLKOUT	r1		
		HEADER 17X2		10	
CON4	名前	説明	CON4	名前	説明
1	VDD5V	5V 電源	2	VDD33V	3.3V 出力
3	GND	GND	4	nRESET	リセット出力
5	ANI0	AD ch0	6	AIN1	AD ch1
7	ANI2	AD ch2	8	AIN3	AD ch3
9	EINT0	EINT0/GPF0	10	EINT1	EINT1/GPF1
11	EINT2	EINT2/GPF2	12	EINT3	EINT3/GPF3
13	EINT4	EINT4/GPF4	14	EINT5	EINT5/GPF5
15	EINT6	EINT6/GPF6	16	EINT8	EINT8/GPG0
17	EINT9	EINT9/GPG1	18	EINT11	EINT11/GPG3/nSS1
19	EINT13	EINT13/GPG5/SPIMISO1	20	EINT14	EINT14/GPG6/SPIMOSI1
21	EINT15	EINT15/GPG7/SPICLK1	22	EINT17	EINT17/GPG9/nRST1
23	EINT18	EINT18/GPG10/nCTS1	24	EINT19	EINT19/GPG11
25	SPIMISO	SPIMISO /GPE11	26	SPIMOSI	SPIMOSI /EINT14/GPG6
27	SPICLK	SPICLK /GPE13	28	nSS_SPI	nSS_SPI /EINT10/GPG2
29	I2CSCL	I2CSCL/GPE14	30	I2CSDA	I2CSDA/GPE15
31	GPB0	TOUT0/ GPB0	32	GPB1	TOUT1/ GPB1
33	CLKOUT0	CLKOUT0/GPH9	34	CLKOUT1	CLKOUT1/GPH10



#### 2.7 CMOS CAMERA



CAMERA	名前	他の機能	CAMERA	名前	他の機能
1	I2CSDA	GPE15	2	I2CSCL	GPE14
3	EINT20	GPG12	4	CAMRST	GPJ12
5	CAMCLK	GPJ11	6	CAM_HREF	GPJ10
7	CAM_VSYNC	GPJ9	8	CAM_PCLK	GPJ8
9	CAMDATA7	GPJ7	10	CAMDATA6	GPJ6
11	CAMDATA5	GPJ5	12	CAMDATA4	GPJ4
13	CAMDATA3	GPJ3	14	CAMDATA2	GPJ2
15	CAMDATA1	GPJ1	16	CAMDATA0	GPJ0
17	VDD33V	3.3V 電源	18	VDD_CAM	VDD_CAM
19	VDD18V	1.8V 電源	20	GND	GND



## 2.8 システムバス

	CONS	1	1 2
VDD5V	1 2	6 GND	1. 22 1
EINT17	3 4	EINT18	244
EINT3	5 6	EINT9	
nGCS1	7 0	nGCS2	5
nGCS3	, o 0 10	nGC85	
LnOE	11 12	LuWE	
nWAIT	12 14	nRESET	
nXDACK0	15 14	nXDREQ0	1
LADDRO	17 10	LADDR1	
LADDR2	10 20	LADDR3	
LADDR4	21 22	LADDR5	
LADDR6	22 24	LADDR24	
LDATAO	25 26	LDATA1	
LDATA2	27 20	LDATA3	
LDATA4	20 20	LDATA5	: ::::::::::::::::::::::::::::::::::::
LDATA6	29 30	LDATA7	88
LDATAS	22 24	LDATA9	
LDATA10	25 26	LDATA11	
LDATA12	27 20	LDATA13	
LDATA14	20 40	LDATA15	
	- 5 <del>9</del> - 40	γ	
	HEADER 20	X2	

CON5	名前	説明	CON5	名前	説明
1	VDD5V	5V 電源	2	GND	GND
3	EINT17	割り込み17	4	EINT18	割り込み18
5	EINT3	割り込み3	6	EINT9	割り込み9
7	nGCS1	0x08000000	8	nGCS2	0x10000000
9	nGCS3	0x18000000	10	nGCS5	0x28000000
11	LnOE		12	LnWE	
13	nWAIT		14	nRESET	
15	nXDACK0		16	nXDREQ0	
17	LADDR0		18	LADDR1	
19	LADDR2		20	LADDR3	
21	LADDR4		22	LADDR5	
23	LADDR6		24	LADDR24	
25	LDATA0		26	DATA1	
27	LDATA2		28	DATA3	
29	LDATA4		30	DATA5	
31	LDATA6		32	DATA7	
33	LDATA8		34	DATA9	



35	LDATA10	36	DATA11	
37	LDATA12	38	DATA13	
39	LDATA14	40	DATA115	

## 2.9 リセット



## 2.10 AD





2.11 PWM ブザー





## 第三章 初体験(GUI)

#### 3.1 タッチパネルの校正



GUI システムを再インストールした後、自動的にタッチパネルの校正画面が出てきます。 ペンで"十"字の中心をタッチします。四角と中心、すべて五つの"十"字が順番に出て きます。タッチパネルがずれた場合は、マウスで設定タブのリカリブレートを選択して、 校正画面も出てきます。



## **3.2 日本語の設定**



"友善之臂"タブの"语言设置"をクリックして、"Japanese"を選択し、「OK」ボタンを 押します。



「Yes」ボタンを押すと、日本語の画面が出てきますが、フォントの原因なので、ある文字 が表示できません。"ロ定"タブの"外ロ"をクリックして、



「Font」タブで「Unifont」を選択して、右上の「OK」ボタンを押します。綺麗な日本語の画面が出てきます。



#### 3.3 MP3 の再生





「アプリケーション」タブの「ミュージック」を選 択し、自動的に「ドキュメント」の MP3 ファイル をさがします。探した MP3 をチェックして、再生 ボタンを押します。



## 3.4 ビデオの再生





「アプリケーション」タブの「ビデオ」を選択し、 自動的に「ドキュメント」のビデオファイルを探し ます。探したビデオをチェックして、再生ボタンを 押します。



## 3.5 ピクチャのビューと編集







「アプリケーション」タブの「ピクチャ」を選択し、 自動的に「ドキュメント」のピクチャを探します。 探したビデオをクリックして、ビューと編集できま す。



## 3.6 SD と USB メモリの自動認識



SD カードめるいは USB メモリを ARM9 ホードに 挿入すると、システムは自動的に SD/USB メモリ を認識して、アイコンが出てきます。「アプリケー ション」タブの「システム情報」をクリックして、 「Storage」タブを選択すると、SD/USB メモリの 情報を表示します。





## 3.7 ターミナル



「アプリケーション」タブの「ターミナル」をクリックします。パソコンがなくでも、タ ーミナルでコマンドを入力できます。

			Terminal 🛛	
			Font Options	
			l° ⇒ ⇒ 9 8 4 10	CASE
	HanRun HR911103A 08/16		R moncontest -d /dem-ttusiCM)	Cattoring St
	1229		-	
	- <u>ģ</u>		* [	
1			HI 1 2 3 4 5 6 7 8 9 0 +	
			Caps a s d f g h j k i z ' Ret	
			Shift z x c v b n m , + / Shift   Ctrl Alt Alt Ctrl Alt Ctrl Alt Ctrl Alt Ctrl Ctrl Alt Ctrl	
÷			- M 🔍 🛱 🕊 🚺 13:34	
		0	FriendlyARM	· * . *

ターミナルでほかのシステム(ARM7TDMI/LPC2148)を通信する様子。



## 3.8 ネットワークの設定



「FriendlyARM」タブの「ネットワーク設定」を選択して、ネットワークを設定きます。

## 3.9 ping



「FriendlyARM」タブの「ネットワーク設定」を選択して、pingをします。



## 3.10 LED テスト

E 🐼 Fr	iendlyARM		LED Testing
A/Dコン バート	Auto Start Setting	CMOS Camera ≡	LED Control
File Browser	I2C-EEPR OM	CDデスト	LED 2 All Off
LEDS	<mark>)</mark> Penpad	) Ping	Stop led-player Close
рым 7° 4₽ авс∕∕ ▲	Carial Ref		Designed by FriendlyARM All rights reserved.   www.arm9.net

LED Testing		×
LED Control		
LED 0		
LED 1	All On	
LED 2	All Off	
LED 3		
Start led-player	Close	
Designed by FriendlyARM All rights reserved.   www.arm9.net		
<mark></mark>	L 🗠 🖻 🖉	03:35

「FriendlyARM」タブの「LEDs」を選択して、 LED の点灯制御ができます。起動した後、「Stop led-player」ボタンを押します。



## 3.11 EEPROM テスト





## 3.12 PWM ブザー





## 3.13 音声のレコーダー





## 3.14 USB カメラ



3.15 CMOS イメージセンサー





## 3.16 AD テスト



ARM9 ボードの可変抵抗を回ると、ADの数値が変化します。

## 3.17 ボタン





## 3.18 手書き



🋂 abc 🖉 🔺 🤳 💦 📲 📲 🚺 07:49



#### 3.19 Watch dog





### 3.20 回転





ほしい方向を選択して、「OK」ボタンを押します。 システムの再起動が必要かもしれません。




# 3.21 スタートアップ





起動の時、アプリケーションを自動的に実行させ ます。Windows のスタートアップにみたい機能で す。

例は起動の時、時計を自動的に実行させます。



# 3.22 USB GPS



秋月電子が販売している USB GPS





# 第四章 初体験(コンソール)

• DB9 メス-メス型のストレートケーブルで mini2240 とパソコンを繋ぐ。



- クロス LAN ケーブルで mini2240 とパソコンを繋ぐ。
- mini2240 のオーディオ出力とスピーカーを繋ぐ。
- 5V電源、極性はセンタープラス ⊖- €- ⊕ です。

# 4.1 パソコン側のハイパーターミナルの設定

パソコンのメニュー : スタート → すべてのプログラム → アクセサリ → 通信 → ハイ パーターミナルを選ぶと、次の画面が出てきます。

接続の設定	? 🗙
動しい接続	
名前を入力し、アイコンを選んでください: 名前( <u>N</u> ): 	
بر د ۲۹ کرد ۲۹ 🚳 🏹 🍣 🏂	
< OK +++>	

このハイパーターミナルの名前を入力して、"OK"ボタンを押すと。



接続の設定	? 🛛
🦓 н8	
電話番号の情報を	入力してください。
国/地域番号( <u>C</u> ):	日本 (81)
市外局番(E):	03
電話番号(P):	
接続方法( <u>N</u> ):	COM1 💌
	OK キャンセル

使用したいシリアルポートを選んでください。

:0M3のプロパティ	?
ポートの設定	
ビット/秒(图):	115200
データ ビット( <u>D</u> ):	8
パリティ( <u>P</u> ):	なし
ストップ ビット( <u>S</u> ):	1
フロー制御(E):	73U
	既定値(z戻す(R)
	K 適用( <u>A</u> )

シリアル通信速度を 115200bps に設定します。フロー制御はなしです。 "OK"ボタンを押すと、設定が完了します。



### 4.2 MP3 の再生

コマンド	madplay
ソースコード	madplay.tgz
コンパイル	Arm-linux-gcc-4.3.2 with EABI

#### # madplay your.mp3

このコマンドは your.mp3 というファイルを再生します。自分で your.mp3 ファイルを用意 してください。「Ctrl+c」で停止させます。

# 4.3 USB メモリと外付けハードデスク

USBメモリを USB ホスト又は USB ハブに挿入すると



自動的にこのような情報が出てきます。USB メモリのデバイス名は/dev/udisk です。シス テムは自動的に/udisk にマウントします。

#### ※ FAT32/VFAT だけの USB メモリが認識できます。

/udisk に移動して、USB メモリのファイルをリストします。

# cd /udisk

# ls



日昇テク	ファ	ジーな	6月	能に	する

🗞 arm9 ー ハイパーターミナル	
ファイル(E) 編集(E) 表示(V) 通信(C) 転送(T) ヘルブ(H)	
[root@FriendlyARM /]# cd /udisk [root@FriendlyARM /udisk]# ls ffsample.zip test.txt [root@FriendlyARM /udisk]#	
接続 0.0022 ANSIW 115200 8-N-1 SCROLL CAPS NUM キャ エコーを印	

# 4.4 SD/MMC カード

🗞 arm9 - ハイパーターミナル 📃 🗆 🔀
ファイル(E) 編集(E) 表示(V) 通信(C) 転送(T) ヘルプ(H)
Iroot@Friendlu0PM /iW s3c2440-sdi s3c2440-sdi: running at 0kHz (requested: 0kHz) s3c2440-sdi s3c2440-sdi: running at 198kHz (requested: 197kHz). s3c2440-sdi s3c2440-sdi: running at 16875kHz (requested: 25000kHz). s3c2440-sdi s3c2440-sdi: running at 16875kHz (requested: 25000kHz). mmc0: new 39 card at address 3f7b mmcblk0: ml FAT: utf8 is not a recommended IO charset for FAT filesystems, filesystem will b e case sensitive!
接続 0:00:44 ANSIW 115200 8-N-1 SCRULL CAPS NUM キャ エコーを印



SD/MMC カードを ARM9 に挿入すると、この情報が出てきます。システムは自動的に /sdcard フォルダを生成します。

🗞 arm9 - ハイパーターミナル 🔳 🗖 🔀
ファイル(E) 編集(E) 表示(V) 通信(C) 転送(E) ヘルブ(H)
<pre>[root@FriendlyARM /]# s3c2440-sdi s3c2440-sdi: running at 0kHz (requested: 0kHz) s3c2440-sdi s3c2440-sdi: running at 198kHz (requested: 197kHz). s3c2440-sdi s3c2440-sdi: running at 16875kHz (requested: 25000kHz). s3c2440-sdi s3c2440-sdi: running at 16875kHz (requested: 25000kHz). s3c2440-sdi s3c2440-sdi: running at 16875kHz (requested: 25000kHz). mmc0: new SD card at address 3f7b mmcblk0: m1 FAT: utf8 is not a recommended IO charset for FAT filesystems, filesystem will b e case sensitive! Troot@FriendlyARM /]# 1s sdcard ffsample.zip test.txt Froot@FriendlyARM /]#</pre>
接続 0.05:06 ANSIW 115200 8-N-1 SCROLL CAPS NUM キャ エコーを印

/sdcard フォルダのファイルをリストします。

### # ls /sdcard

# 4.5 シリアルポートでファイルを ARM9 にダウンロード







■ ファイルの送信	? 🗙
フォルダ: C:¥Documents and Settings¥zqing ファイル名(F):	
C¥Documents and Settings¥zqing¥PRODUCTS.GIF	照(图)
プロトコル( <u>P</u> ):	
Zmodem クラッシュ回復機能付き)	*
送信(S) 閉じる(C) キ	ャンセル

送信したいファイルを選んで、「送信」ボタンを押します。

Zmodem クラッシュ回復機能付き)ファイル送信 – arm9						
送信中:	C:¥Documents and Settings¥zging¥PRODUCTS.GIF					
最新のイベント	送信中	ファイル: 1/1				
状態:	送信中	再試行: 0				
ファイル		35K / 299K				
経過時間:	00:00:03 残り: 00:00:22	スループット: 11946 cps				
		キャンセル cps/bps(©)				

[root@FriendlyARM /]# md5sum PRODUCTS.GIF
ae9cf1d3da214a6c985d9de9ece071b9 PRODUCTS.GIF
[root@FriendlyARM /]# _

			COROLL I	0.100	a station of the	Color Color	And and
接続 0:03:58 AN	NSIW 1	15200 8-N-1	SURULL	CAPS	NUM	77	T

転送が正しいかどうか、md5sum コマンドで検証します。



# 4.6 シリアルポートで ARM9 のファイルを PC に保存

🧠 arm9 ー ハイパーターミナル			
ファイル(E) 編集(E) 表示(V) 通信(C)	転送(1) ヘルプ(円)		
口戸 🖉 🖀 🗈 🎢 😭 🌘	ファイルの送信( <u>S</u> )…		
	ファイルの受信(B)		
s3c2410-rtc s3c2410-r	テキストファイルの送信(① )M		
183) FAT: utf8 is not a re	キャプチャして印刷( <u>P</u> ) COmmended IO Char'S		
■ ファイルの受信	? 🛛		
受信するファイルの保存先生			
C:¥Documents and Settings¥zging	参照( <u>B</u> )		
使用するプロトコル(ロ):			
Zmodem クラッシュ回復機能付き)	~		
受信®	閉じる(2) キャンセル		

保存先を設定して、「閉じる」ボタンを押します。

#### # sz PRODUCTS.GIF

コマンドで転送が開始します。

FAT: uť	Zmodem のう	ッシュ回復機能付き) ファイル受信 – arm9
le case [02/Apr	受信中:	PRODUCTS.GIF
[02/Apr	保存先:	C*Documents and Settings¥zging¥PRODUCTS.C ファイル: 1/1
Try to	最新のイベント	受信中 再試行:
Done	状態:	受信中
Please [root@F	ファイル	194K / 299K
- [root PRODUCT	経過時間:	00:00:17 残り: 00:00:09 スループット: 11223 cps
bin dev		キャンセル スキップ(S) cps/bps(C)
etc [root@Fi - <u>0</u> 0000	riendlyAR 0000000	M /]# sz PRODUCTS.GIF



# 4.7 LED 制御

コマンド	led-player leds
ソースコード	led-player.c leds.c
パッケージ	examples.tgz
コンパイル	Arm-linux-gcc-4.3.2 with EABI
デバイス名	/dev/leds
ドライバ	Linux-2.6.29/drivers/char/mini2440_leds.c

### 1) LED サーバ

システム起動の時、自動的に LED サーバ(**led-player**)を起動させます(**/etc/rc.d/init.d/leds**)。 LED を点滅させています。**led-player** を実行した後、/tmp/led-control というパイプを生成しま す。

#### #echo 0 0.2 > /tmp/led-control

LED が 0.2 秒周期で流れます。

### #echo 1 0.2 >/tmp/led-control

LED が 0.2 秒周期で累計します。

### #/etc/rc.d/init.d/leds stop

LED を停止します。

### #/etc/rc.d/init.d/leds start

LED が点滅をスタートします。

### 2) 単独 LED 制御

### #/etc/rc.d/init.d/leds stop

LED サーバをストップさせます。

#### # led

Usage: leds led\_no 0|1 led\_no は LED 番号(0,1,2,3)です。

### #led 2 1

LED2 を点灯させます。



# 4.8 ボタンのテスト

コマンド	buttons
ソースコード	buttons_test.c
パッケージ	examples.tgz
コンパイル	Arm-linux-gcc-4.3.2 with EABI
デバイス名	/dev/buttons
ドライバ	Linux-2.6.29/drivers/char/mini2440_buttons.c
★ ttySD - 超级终端 文件で) 編輯で) 查看 (火) 呼叫で) 传送で) → ○ ○ ○ ○ ○ □ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	帮助 ① his console. s decoded (0:03:11.1), +0.0 dB peak amplitude, 6
	L CAPS NUM 捕打印

#### #buttons

このコマンドを入力してください。



# 4.9 シリアルポートのテスト

コマンド	armcomtest
ソースコード	Comtest.c
パッケージ	examples.tgz
コンパイル	Arm-linux-gcc-4.3.2 with EABI
デバイス名	/dev/ttySAC0,1,2 或は
	/dev/ttyUSB0,1,2 或は
	/dev/ttyACM0,1,2

- ※ ARM9 は自分の三つのシリアルポートが/dev/ttySAC0,1,2 です。
- ※ ARM9 は弊社が販売している USB-RS232 変換ケーブルを直接使え ます。デバイス名は/dev/ttyUSB0,1,2 です。
- ※ ARM9 は弊社が販売している ARM7TDMI/LPC2148 を USB で通 信できます。 ARM7TDMI/LPC2148 のデバイス名は /dev/ttyACM0,1,2です。

このコマンドは LPC2148 と通信します。

#### #armcomtest -d /dev/ttyACM0 -o

	Terminal X	
	■ ⇒ ⇒ ↑ ↓ № 1s -	200 F
lanRun 98/1103A 08/16	ne Tio (Prizing)	
	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	
	Shift         z         x         c         v         b         n         n         i         i         i         i         Shift           Ctri         Alt         Alt         Ctri         Alt         Ctri         Alt         Ctri         Alt         Al	
C C	FriendlyARM	* *
State State	Maria Maria Maria Maria Maria	1 22 10 X



# 4.10 ブザー(PWM)のテスト

コマンド	pwm_tset
ソースコード	pwm_tset.c
パッケージ	examples.tgz
コンパイル	Arm-linux-gcc-4.3.2 with EABI
デバイス名	/dev/pwm
ドライバ	Linux-2.6.29/drivers/char/pwm.c

#pwm\_test



"+"と"-"キーは音声の周波数を変更させます。"ESC"キーは音声を停止させます。

# 4.11 LCD のバックライト

#echo 0 > /dev/backlight	;バックライト消灯
#echo 1 > /dev/backlight	;バックライト点灯



## 4.12 I2C-EEPROM

コマンド	i2c
ソースコード	eeprom.c 24cXX.c
パッケージ	examples.tgz
コンパイル	Arm-linux-gcc-4.3.2 with EABI
デバイス名	/dev/i2c/0
ドライバ	Linux-2.6.29/drivers/i2c/busses/i2c-s3c2440.c

### #i2c-w ;データ(0~255)をボードの 24C08 に書き込む

#i2c -r ;ボードの 24C08 からデータを読み出す

## 4.13 AD テスト

コマンド	adc-test
ソースコード	adc-test.c
パッケージ	examples.tgz
コンパイル	Arm-linux-gcc-4.3.2 with EABI
デバイス名	/dev/adc
ドライバ	Linux-2.6.29/drivers/char/mini2440_adc.c

#### #adc-test

ボードの可変抵抗をまわして、AD 数値の変化が見えます。

```
[root@FriendlyARM /]# adc-test
press Ctrl-C to stop
ADC Value: 60
ADC Value: 187
ADC Value: 267
ADC Value: 312
ADC Value: 368
ADC Value: 444
ADC Value: 422
ADC Value: 337
ADC Value: 260
ADC Value: 211
ADC Value: 190
ADC Value: 190
ADC Value: 190
ADC Value: 190
```



## 4.14 CMOS イメージセンサー

コマンド	camtest
ソースコード	camtest.c
パッケージ	examples.tgz
コンパイル	Arm-linux-gcc-4.3.2 with EABI
デバイス名	/dev/camera
ドライバ	Linux-2.6.29/drivers/media/video/s3c2440camif.c

#### #camtest



# 4.15 ネットワーク機能

### 4.15.1 ウェブサーバー

Linux でウェブサーバー(boa)をインストールしました。パソコンのブラウザーで http://192.168.1.230 を入力すると、mini2440 のホームページが見えます。このホームペ ージを通じて、ユーザーLED と USB カメラ(\*)をアクセスできます。

> ※ mjpg-streamer というソフトウェアをインストールすることが必要 です。



#### 4.15.2 Telnet と Ftp 機能

Linux でクライント側とサーバー側の Telnet/Ftp をインストールしました。ご利用してく ださい。 デフォルトの設定: Telnet のユーザーネームは root です、password がありません。 Ftp のユーザーネームは plg です、password も plg です。

### 4.15.3 DNS と gateway の設定

DNSのIPアドレスを/etc/resolv.confファイルに書き込みます。 gatewayの設定:#route add default gw 192.168.1.1

### 4.15.4 MAC アドレスの設定

# ifconfig eth0 down # ifconfig eth0 hw ether 00:11:AA:BB:CC:DD(新 MAC アドレス) # ifconfig eth0 up 新 MAC アドレスが有効のため、これらのコマンドを起動スクリプト/etc/init.d/rcS に書き 込みます。

### 4.15.5 ネットワーク・ファイルシステム(NFS)のマウント

まず、ネットワーク・ファイルシステムのサーバーを構築します。 #mount -t nfs -o nolock 192.168.1.111:/root\_nfs /mnt 192.168.1.111 はネットワーク・ファイルシステムのサーバーの IP アドレスです。 マウント成功すれば、ARM9 は大きなリモート・ハードディスク(/mnt)を直接にアクセス できます。プログラムを開発する時が便利です。 #umount /mnt #リモート・ハードディスクを ARM9 システムから外します。

# 4.16. RTC の設定

(1)#date -s 042916352007 #今の時間を設定します: 2007-04-29 16:34
(2)#hwclock -w #今の時間を S3C2440 の RTC に保存します。
(3)#hwclock -s #起動の時、Linux 時間を S3C2440 の RTC から回復します。
※ hwclock -s コマンドは起動スクリプト(/etc/init.d/rcS)に書き込みま

した。起動の時、自動的に実行します。



# 4.17 液晶(LCD)画面を取ります

#### #snapshot pic.png

液晶(LCD)で表示された画面を pic.png というファイルに保存します。

## 4.18 USB 無線 LAN

Linux-2.6.29にはrt73という USB 無線 LAN のドライバを実装しました。この類型の USB 無線 LAN を ARM9 ボードの USB ホストに挿入すると、次の情報が出てきます。 [root@FriendlyARM /]# usb 1-1: new full speed USB device using s3c2410-ohci and address 2 usb 1-1: New USB device found, idVendor=148f, idProduct=2573 usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=0 usb 1-1: Product: 54M.USB...... usb 1-1: Manufacturer: Ralink usb 1-1: configuration #1 chosen from 1 choice wmaster0 (rt73usb): not using net\_device\_ops yet wlan0 (rt73usb): not using net\_device\_ops yet

[root@FriendlyARM /]# ifconfig eth0 down 有線 LAN をストップ [root@FriendlyARM /]# ifconfig wlan0 up 無線 LAN を起動 rt73usb 1-1:1.0: firmware: requesting rt73.bin 利用可能の無線ネットを探す [root@FriendlyARM /]# iwlist scanning | grep ESSID lo Interface doesn't support scanning. eth0 Interface doesn't support scanning. wmaster0 Interface doesn't support scanning. ESSID:"FRIENDLY-ARM" ESSID:"NETGEAR"

### 無線ネットの ESSID を入力

[root@FriendlyARM /]# iwconfig wlan0 essid "FRIENDLY-ARM"

### 無線ネットのパスワードを入力

[root@FriendlyARM /]# iwconfig wlan0 key s:12345



### 無線ネットのルータに接続

[root@FriendlyARM /]# iwconfig wlan0 ap auto

### IP アドレスの設定

[root@FriendlyARM /]# ifconfig wlan0 192.168.1.120

## 無線ネットのテスト

[root@FriendlyARM /]# ping 192.168.1.1 PING 192.168.1.1 (192.168.1.1): 56 data bytes 64 bytes from 192.168.1.1: seq=0 ttl=64 time=42.804 ms 64 bytes from 192.168.1.1: seq=1 ttl=64 time=5.020 ms 64 bytes from 192.168.1.1: seq=2 ttl=64 time=5.021 ms ^C --- 192.168.1.1 ping statistics ---3 packets transmitted, 3 packets received, 0% packet loss round-trip min/avg/max = 5.020/17.615/42.804 ms [root@FriendlyARM /]#



# 第五章 Linux のクロス開発環境

# 5.1 クロス開発環境を構築

ARM9のクロス開発環境は Fedora9/11 をお勧めします。

Fedora で ARM のクロス開発ツールをインストールします。

\$ su スーパーユーザーに切り替え

#### # tar xvzf arm-linux-gcc-4.3.2.tgz -C /

※ 使いやすいため、「export PATH=\$PATH:/usr/local/arm/3.4.1/bin」 を.bashrc ファイルに入れます。

#### \$ vi ~/.bashrc

			zqing@	localhost:~		-	÷	×
ファイル( <u>F</u> )	編集( <u>E</u> )	表示( <u>V</u> )	端末( <u>T</u> )	ヘルプ( <u>H</u> )				
# .bashrc								$\sim$
# Source gl if [ -f /et . / fi	obal def: c/bashrc etc/bash	initions ]; then rc						
# User spec	ific ali	ases and	function	15				
export PATH	=\$PATH:/	usr/local	/arm/4.	3.2/bin				
~								
~								
~								
~								
~								
~								
~								
~								
~								
~								
~ "~/.bashrc"	10L, 16	BC	<u>л</u>		10,0-1		全て	$\downarrow$



再び Fedora のターミナルを開き、次のコマンド:

#### \$ arm-linux-gcc –v

zqing@localhost:~		+	×
ファイル( <u>F</u> ) 編集( <u>E</u> ) 表示( <u>V</u> ) 端末( <u>T</u> ) ヘルプ( <u>H</u> )			
<pre>[zqing@localhost ~]\$ vi ~/.bashrc [zqing@localhost ~]\$ arm-linux-gcc -v Using built-in specs. Target: arm-none-linux-gnueabi Configured with: /scratch/julian/lite-respin/linux/src/gcc-4.3/configur =i686-pc-linux-gnuhost=i686-pc-linux-gnutarget=arm-none-linux-gnu able-threadsdisable-libmudflapdisable-libsspdisable-libstdcxx- h-gnu-aswith-gnu-ldenable-languages=c,c++enable-sharedenabl =gnuenablecxa_atexitwith-pkgversion='Sourcery G++ Lite 2008q3- h-bugurl=https://support.codesourcery.com/GNUToolchain/disable-nls opt/codesourcerywith-sysroot=/opt/codesourcery/arm-none-linux-gnueab with-build-sysroot=/scratch/julian/lite-respin/linux/install/arm-none-l abi/libcwith-gmp=/scratch/julian/lite-respin/linux/obj/host-libs-200 m-none-linux-gnueabi-i686-pc-linux-gnu/usrwith-mpfr=/scratch/julian/ in/linux/obj/host-libs-2008q3-72-arm-none-linux-gnueabi-i686-pc-linux-g disable-libgompenable-poison-system-directorieswith-build-time-to tch/julian/lite-respin/linux/install/arm-none-linux-gnueabi Thread model: posix gcc version 4.3.2 (Sourcery G++ Lite 2008q3-72) [zqing@localhost ~]\$</pre>	<pre>&gt;  &gt;abi &gt;ch &gt;- sy /2' pre i/lil inux }q3 lite nu/u &gt;ls=, puil(</pre>	buil ( mvel w: fix= bc gnu 72-a -re: sr - /sct n	ld ant still Je ar and still

この画面が出たら、ARM 用のクロス開発環境をインストール成功しました。

### 5.2 NFS サーバーを構築

ネットワーク・ファイルシステム(NFS)を使用すれば、ARM9 は大きなホストのハードディ スクを直接にアクセスできます。プログラムを開発する時が便利です。次は NFS サーバー を構築手順です。

(1) NFS のルートシステムファイルを解凍します。

# tar xvzf root\_qtopia.tgz -C /

(2) /etc/exports ファイルを編集します。

```
「/ root_qtopia *(rw,sync,no_root_squash)」を/etc/exports というファイルに入れま
```

す。



(3) NFS サーバを起動します。# /etc/init.d/nfs start

(4) NFSf ファイルシステムを確認します。
# mount -t nfs localhost:/root\_nfs /mnt/
# ls /mnt

※ firewall の設定によって、外部から NFS ヘアクセスできない可能性があります。

# 5.3 NFS はルートファイルシステムとして起動

ARM9 ボードが起動、又はリセットの時、ハイパーターミナルにスペースキーを押します。 次のコマンドを入力します。

Supervivi>param set linux\_cmd\_line "console=ttySAC0 root=/dev/nfs nfsroot=192.168.1.111:/root\_nfs

ip=192.168.1.70:192.168.1.111:192.168.1.111:255.255.255.0:MINI2440.arm9. net:eth0:off"

param set linux\_cmd\_line は linux 起動のパラメーターです。パラメーターの意味は: nfsroot は NFS サーバーの IP アドレス。

"ip="の後ろ:

第一項(192.168.1.70)は ARM9 ボードの IP;

第二項(192.168.1.111)はホストの IP;

第三項(192.168.1.111)はゲットウェイの IP;

第四項(255.255.255.0)はネットマスク;

第五項はホストのドメイン(自由的に入力でも大丈夫です)

eth0はLAN デバイスの名前。

Boot コマンドを入力すると Supervivi>boot

ARM9 ボードは NFS からブートします。

# 第六章 Linux 環境のアプリケーションを開発

## 6.1 Hello, World!

Hello, World のソースコードは examples.tgz にあります。 #include <stdio.h>



int main(void) {

printf("hello, FriendlyARM!\fu");

}

# 6.2 Hello,World をコンパイル

#cd examples/hello #arm-linux-gcc –o hello main.c 又は #make 実行できるhelloを生成します。

# 6.3 Hello,World を ARM9 ボードで実行

生成された実行コードhelloをARM9ボードに入れて、ARM9のコンソールで実行します。 #./hello hello,FriendlyARM!

ARM9ボードに入れるのは幾つの方法があります。USB・SDメモリ、シリアルポート、FTPなど。一番便利な方法はNFSです。ARM9ボードは直接にホスト側の実行ファイルを実行できます。

# 6.4 ほかのサンプル

examples.tgzに	幾つのサンプルがあります。
adc-test	ADCテスト
buttons	ボタンテスト
<b>c++</b>	C++サンプル
camtest	CMOSイメージセンサーテスト
comtest	シリアルポートテスト
hello	
i2c	i2c EEPROMテスト
led-player	pipeサンプル
leds	LED点灯
math	数学処理
pthread	スレッド



pwmPWMテストudptalkUDP通信サンプル

# 6.5 Qt/Embedded GUI プログラムを作る

詳細の QT 組込開発手順は「QT-Embedded\_DEV\_manual.pdf」を参照

# tar zxvf arm-qtopia-2.2.0.tar.gz -C /opt/FriendlyARM/mini2440

Qtopia-2.2.0 のパッケージをディレクトリ/opt/FriendlyARM/mini2440 に展開します。

# cd /opt/FriendlyARM/mini2440/arm-qtopia

#./build-all 長い時間がかかりますので、我慢してください

#./mktarget ターゲットに書き込むファイル target-qtopia-konq.tgz を生成 サンプル hello をコンパイルします

# cd /opt/FriendlyARM/mini2440/arm-qtopia/hello

# ./build

コンパイル完了すれば、ディレクトリ arm-qtopia/qtopia-2.2.0-FriendlyARM/qtopia/bin に実 行ファイル hello を生成させます。実行ファイル hello と arm-qtopia/hello/hello.desktop を ARM9 ボードにダウンロードして、ARM9 の適当なディレクトリに移転します。

#chmod +x hello

### #mv hello /opt/Qtopia/bin

#mv hello.desktop /opt/Qtopia/apps/Applications



再起動すれば、デスクトップで hello のアイコンが見えます。実行させると...



# 第七章 Linux カーネルを再構築

# 7.1 カーネルのソースコードを解凍

### \$ tar xvzf linux-2.6.29-mini2440-20090609.tgz

※ このカーネルはQQ2440v3/MINI2440に両対応する

			zqing@	localho	st:~			-	÷	×
ファイル( <u>F</u> )	編集( <u>E</u> )	表示(⊻)	端末( <u>T</u> )	ヘルプ(	<u>H</u> )					
linux-2.6.29	/Documen	tation/	SM501.txt	t						_
linux-2.6.29	/Documen	tation/	sh/							
linux-2.6.29	/Documen	tation/	sh/clk.t>	(t						
linux-2.6.29	/Documen	tation/	sh/regist	ter-banl	ks.txt					
linux-2.6.29	/Documen	tation/	sh/new-ma	achine.	txt					
linux-2.6.29	/Documen	tation/:	isapnp.t>	(t						
linux-2.6.29	/Documen	tation/1	IPMI.txt							
linux-2.6.29	/Documen	tation/\	wimax/							
linux-2.6.29	/Documen	tation/\	wimax/REA	ADME.wi	nax					
linux-2.6.29	/Documen	tation/	wimax/REA	ADME.124	100m					
linux-2.6.29	/Documen	tation/	kbuild/							- <b>3</b>
linux-2.6.29	/Documen	tation/	kbuild/ko	config-	language	e.txt				
linux-2.6.29	/Documen	tation/	kbuild/ma	akefile	s.txt					
linux-2.6.29	/Documen	tation/	kbuild/mo	dules.	txt					
linux-2.6.29	/Documen	tation/	kbuild/ko	config.	txt					
linux-2.6.29	/Documen	tation/	kbuild/00	)-INDEX						
linux-2.6.29	/Documen	tation/	kbuild/kk	build.t	ct					
linux-2.6.29	/Documen	tation/:	ide/							
linux-2.6.29	/Documen	tation/:	ide/ide-t	tape.tx	t					
Linux-2.6.29	/Documen	tation/:	ide/ide.t	txt .						
Linux-2.6.29	/Documen	tation/:	ide/Chang	geLog.id	de-flopp	y.1996-	2002			-
Linux-2.6.29	/Documen	tation/:	ide/Chang	geLog.id	de-tape.	1995-20	02			::
linux-2.6.29	/Documen	tation/:	1de/Chang	geLog.id	de-cd.19	94-2004				
										$\sim$



# 7.2 Linux を再コンパイル

\$ cd linux-2.6.29

\$ ls

ファイルをリストします

	zqing@localhost:~/lin	ux-2.6.29		-	+ ×
ファイル( <u>F</u> ) 編集	፤( <u>E)</u> 表 <del>示(⊻) 端末(王)</del> ヘルプ( <u>H</u> )	)			
[zqing@localhos [zqing@localhos COPYING CREDITS Documentation Kbuild MAINTAINERS Makefile README REPORTING-BUGS arch [zqing@localhos	t []\$ cd linux-2.6.29 t linux-2.6.29]\$ ls block config_mini2440_a70 config_mini2440_t35 config_mini2440_vga1024x768 config_q2440_a70 config_q2440_n35 config_q2440_vga1024x768 crypto t linux-2.6.29]\$	drivers firmware fs include init ipc kernel lib mm	net samples scripts security sound usr virt		
	•				$\downarrow$

**config\_mini2440\_xxx**はmini/micro2440用コンフィグファイルです。 **config\_qq2440\_xxx**はmini2440用コンフィグファイルです。

a70は7インチ用コンフィグファイルです。 n35は3.5インチ用コンフィグファイルです。 vga1024X768はVGA用コンフィグファイルです。



#### \$ cp config\_mini2440\_n35 .config

あるコンフィグファイルを選択します

#### \$ make menuconfig





何も変更しません、「Exit」を選択します。



この画面が出てきたら、「Yes」を押します。make zImageでコンパイルします。







	root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29	- + X
<u>File Edit</u>	: <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp	
CC	init/version.o	^
LD	init/built-in.o	
LD	.tmp_vmlinux1	
KSYM	.tmp_kallsyms1.S	
AS	.tmp_kallsyms1.o	
LD	.tmp_vmlinux2	
KSYM	.tmp_kallsyms2.S	
AS	.lmp_kallsyms2.o	
LD	Vmlinux	
SYSMAP	System.map	
SYSMAP	.tmp System.map	
OBJCOPY	arch/arm/boot/Image	
Kernet:	arch/arm/boot/image is ready	
AS	arch/arm/boot/compressed/nead.o	
GZIP	arch/arm/boot/compressed/piggy.gz	
AS	arch/arm/boot/compressed/piggy.o	
	arch/arm/boot/compressed/wilipux	
OBICOPY	arch/arm/boot/zimage	
Kernel	arch/arm/boot/zImage is ready	
[root@tom	linux-2.6.291# ls arch/arm/boot/	
bootp co	mpressed Image install.sh Makefile zImage	11
[root@tom	linux-2.6.29]#	

コンパイル完了すると、arch/arm/bootフォルダにLinuxのカーネルzImageを生成さ せます。

# 7.3 ドライバの場所

#### (1)DM9000 10/1000Mイーサネット

Linux-2.6.29/drivers/net/dm9000.c (2)シリアルポート(三つのシリアルポート0,1,2, デバイス名/dev/ttySAC0,1,2) Linux-2.6.29/drivers/serial/s3c2440.c (3)リアルタイマーRTC Linux-2.6.29/drivers/rtc/rtc-s3c.c (4)LED Linux-2.6.29/drivers/char/mini2440\_leds.c (5)ボタン Linux-2.6.29/drivers/char/mini2440\_buttons.c (6)タッチパネル Linux-2.6.29/drivers/input/touchscreen/s3c2410\_ts.c (7)yaffs2ファイルシステム Linux-2.6.29/fs/yaffs2 (8)USBマウス、キーボード Linux-2.6.29/drivers/usb/hid



## (9)SD/MMCメモリカード(最大32GB)

Linux-2.6.29/drivers/mmc

## (10)Nand Flash

Linux-2.6.29/drivers/mtd/nand

# (11)UDA1341オーディオ

Linux-2.6.29/sound/soc/s3c24xx

## (12)LCD液晶

Linux-2.6.29/drivers/video/s3c2410fb.c

## (13)USBメモリ

Linux-2.6.29/drivers/usb/storage

# (14)gspca類USBカメラ

Linux-2.6.29/drivers/media/video/gspca

# (15)I2C-EEPROM

inux-2.6.29/drivers/i2c

(16)バックライト

Linux-2.6.29/drivers/video/mini2440\_backlight.c

# (17)PWMブザー

Linux-2.6.29/drivers/char/mini2440\_pwm.c

# (18)Watchdog

Linux-2.6.29/drivers/watchdog/s3c2410\_wdt.c

## (19)ADC

Linux-2.6.29/drivers/char/mini2440\_ad.c

# (20)CMOSイメージセンサー

Linux-2.6.29/drivers/media/video/s3c2440camif.c

## (21)USB無線LAN

Linux-2.6.29/drivers/net/wireless/rt2x00

## (22)USB-RS232変換ケーブル

Linux-2.6.29/drivers/usb/serial/pl2302.c

# (23)CDC ACM(ARM7TDMI/LPC2148通信)

Linux-2.6.29/drivers/usb/class/



# 7.4 Linux カーネルのコンフィグ

\$ cd linux-2.6.29

\$ cp config\_mini2440\_n35.config あるコンフィグファイルを選択します

**\$** make menuconfig



初のコンフィグ画面です。



### 7.4.1 LCD 液晶とバックライト



初の画面で「Device Drivers」を選択、



「Enter」キーで「Graphics support」に入ります。





「support for frame buffer devices」に入ります。

	root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29 🔤 🕈	x
<u>F</u> ile	<u>E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp	
. CO	onfig - Linux Kernel v2.6.29 Configuration	
	Support for frame buffer devices         Arrow keys navigate the menu. <enter> selects submenus&gt;.         Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes,         <mbodularizes <esc="" features.="" press=""> to exit, <? > for Help,  tor Search. Legend: [*] built-in [ ] excluded <m> module &lt;&gt;         (-)       Imable Tile Blitting Support         **** Frame buffer hardware drivers ***         &lt;&gt; Epson S1D13XXX framebuffer support         <h>CD select (3.5 incn 240X320 NEC LCD)&gt;         []       S3C2410 lcd debug messages         &lt;*&gt;       Backlight support for FriendlyARM board         &lt;&gt;&gt;       Virtual Frame Buffer support (ONLY FOR TESTING!)         &lt;&gt;       E-Ink Metronome/8track controller support         v(+)</h></m></mbodularizes></n></y></enter>	

まず、スペースキーで「Backlight support for FriendlyARM board」に**<\*>**を入れます。「LCD select」に入ります。



	root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29 📃 🕈	×
<u>F</u> ile	<u>E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp	
. CDI	nfig - Linux Kernel v2.6.29 Configuration	Â
	I(D salect	
	Use the arrow keys to navigate this window or press the hotkey of	
	the item you wish to select followed by the <space bar="">. Press</space>	
	(X) 3.5 inch 240X320 NEC LCD	
	() VGA 1024x768	111
	<pre><select> &lt; Help &gt;</select></pre>	

3.5インチのLCDを選択します。コンフィグ完了したら、「Exit」で「Device Drivers」メニューに戻ります。



### 7.4.2 タッチパネル

「Device Drivers」メニューの「Input device support」に入ります。



< Exit > < Help >

「Touchscreens」に入ります。

Keyboards ---> Mice --->

Tablets --->
Touchscreens --->

Joysticks/Gamepads --->

Miscellaneous devices ---> Hardware I/O ports --->

<Select>



root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29	. *	x
<u>File E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp		
.config - Linux Kernel v2.6.29 Configuration		
		-
Arrow keys navigate the menu <pre>Enters selects submenus&gt;</pre>	1	
Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes</n></y>		
<pre><m> modularizes features. Press <esc><esc> to exit, <?> for Help, <!--/pre--></esc></esc></m></pre>	>	
for Search. Legend: [*] built-in [ ] excluded <m> module &lt; &gt;</m>		
⊲*> Samsung S3C2410 touchscreen input driver		
rujitsu seriat touchscreen		
<pre>&lt; &gt; Gunze AHL-51S touchscreen</pre>		13
<pre>&lt; &gt; Elo serial touchscreens</pre>		
MicroTouch serial touchscreens		
Incrorouch serial touchscreens		
< > ICS MicroClock MK712 touchscreen		
L V(+)		
Selects < Exit > < Hetp >		
		¥.

画面のように選択します。「Exit」で「Device Drivers」メニューに戻ります。

## 7.4.3 USB マウスとキーボード

「Device Drivers」メニューの「HID Devices」に入ります。





🔲 root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29 🔤 🍕	×
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp	
.config - Linux Kernel v2.6.29 Configuration	
HID Devices         Arrow keys navigate the menu. <enter> selects submenus ···&gt;.         Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes,         <d><d><d>modularizes features. Press <esc><fsc> to exit, &lt;2&gt; for Help, &gt; for Search. Legend: [*] built-in [] excluded <m> module &lt;&gt;         HID Devices         -*- Generic HID support         [] HID debugging support         [] /dev/hidraw raw HID device support         *** USB Input Devices ***         Image: Press Second (full HID) support         [] PID device support         [] /dev/hiddev raw HID device support         Special HID drivers +&gt;</m></fsc></esc></d></d></d></n></y></enter>	

画面のように選択して、「Exit」で「Device Drivers」メニューに戻ります。

### 7.4.4 USB メモリ

「Device Drivers」メニューの「SCSI device support」に入ります。

		ro	ot@tom:/	opt/Fr	iendly	yARM/mi	ini24	40/li	nux-2.	6.29		1820	4	R
File	<u>E</u> dit	View	Terminal	Ta <u>b</u> s	<u>H</u> elp									
. coi	nfig -	Linux	Kernel v	2.6.29	9 Conf	figuratio	on							-
	Arrow Highl <m> m for S [ [ [ [ [ [</m>	keys ighted odular earch. > Par > Blo ] Mis > ATA SCS > Ser ] Mul *] Net	navigale letters izes feat Legend: ory Techr allel por ck devices /ATAPI/MF I device ial ATA ( tiple dev work devi	the me are ho ures. [*] t nology t supp s M/RLL Suppor prod) rices c ce sup	Devi otkeys Pres ouilt- Devic ort support and P friver oport	<pre>ce Drive <enler> s. Press ss <esc>- in [] ce (MID) &gt; ort&gt; varallel - support &gt;</esc></enler></pre>	ers sele sing <fscs excl supp &gt; AIA</fscs 	ecls <y> &gt; to luded port (exp AID a</y>	submen includ exit, <m> &gt; erimen nd LVM</m>	tal)	->. N> excl or Help e < > drivers	udes,		
			<	elect>	<	< Exit >		< Hel	p >					


		ro	ot@tom:/	opt/Fr	iendl	yARM/m	ini 24	140/li	nux-2.	5.29		1	•	X
<u>F</u> ile	<u>E</u> dit	View	Terminal	Ta <u>b</u> s	<u>H</u> elp									
. CO	nfig -	Linux	Kernel N	2.6.29	9 Cont	figurati	.on							
	Arrow Highl <m> m for S</m>	keys ighted odular earch.	navigate letters izes feat Legend:	the me are ho ures. [*] t	SCSI ( enu. otkeys Pres Duilt	device s <enter> s. Pres ss <esc> -in []</esc></enter>	sele sing <esc> excl</esc>	rt ects <y> &gt; to luded</y>	submenu include exit, < <m> r</m>	us: es, <n: <? > for nodule</n: 	>. > exclu - Help, < >	udes, ,		
	< < (	<pre>&gt; PAI *&gt; SCS &gt; SCS ] leg *** *&gt; SCS</pre>	D Transpo I device I target acy /proc SCSI sup I disk su	ort Cla suppor suppor support t	ass rt / supp type (	port (disk, t	ape,	CD-R	OM) ***	ĸ				-
		> SCS > SCS > SCS (+)	<del>I tape st</del> I OnStrea I CDROM s	m SC-> support	k0 tap t	pe suppo	rt							
			<	elect		< <mark>E</mark> xit >		< Hel	p >					
*														2

画面のように選択して、「Exit」で「Device Drivers」メニューに戻ります。

「Device Drivers」メニューの「USB support」に入ります。

		root@tom:/	:/opt/FriendlyARM/mini2440/linux-2.6.29 📃 🗧 🗧	x
File	Edit Vie	w <u>T</u> erminal	il Ta <u>b</u> s <u>H</u> elp	
. CO	nfig - Lin	nux Kernel v	v2.6.29 Configuration	
	Arrow key Highlight <m> modul for Searc</m>	vs navigate ted letters larizes feat th. Legend:	USB support e the menu. <enter> selects submenus&gt;. s are hotkeys. Pressing <y> includes, <n> excludes, atures. Press <esc><esc> to exit. <? > for Help, d: [*] built-in [] excluded <m> module &lt; &gt;</m></esc></esc></n></y></enter>	
	< >	USB Wirele USB Test a *** NOTE: *** see US	less Device Management support and Measurement Class support : USB STORAGE depends on SCSI but BLK DEV SD may al USB STORAGE Help for more information ***	111
4	<pre></pre>	USB Mass S USB Mass Datafab Freecom ISD-200	Storage support as Storage verbose debug b Compact Flash Reader support m USB/ATAPI Bridge support D USB/ATA Bridge support	
_		<b>.</b>	<select> &lt; Exit &gt; &lt; Help &gt;</select>	J>

「USB Mass Storage support」を選択して、「Exit」で「Device Drivers」メニューに戻ります。



### 7.4.5 汎用 USB カメラ

「Device Drivers」メニューの「Multimedia devices」に入ります。



「video capture adapters」に入ります。





	root@t	om:/opt/FriendlyARM/mini2440/linux-2.6.29	÷ X
<u>File</u>	dit <u>V</u> iew Terr	nınal Ta <u>b</u> s <u>H</u> elp	
.conf	ig - Linux Ker	nel v2.6.29 Configuration	
A H t	rrow keys navi ighlighted let M> modularizes or Search. Le	Video capture adapters jate the menu. <fnter> selects submenus&gt;. ters are hotkeys. Pressing <y> includes, <n> excludes, features. Press <esc><esc> to exit, <? > for Help,  gend: [*] built-in [ ] excluded <m> module &lt; &gt;</m></esc></esc></n></y></fnter>	
	[] Enabl [] Enabl [] Autos Enc <> Virtu <> SAA52 <> SAA52 <> SAC c [*] V4L U	e advanced debug functionality e old-style fixed minor ranges for video devices elect pertinent encoders/decoders and other helper chi oders/decoders and other helper chips> al Video Driver 46A, SAA5281 Teletext processor 49 Teletext processor amera support SB devices>	
_		<pre><select> &lt; Exit &gt; &lt; Help &gt;</select></pre>	

「V4L USB devices」に入ります。

	root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29 🔤 👻 🕷
File	<u>E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp
. COI	nfig - Linux Kernel v2.6.29 Configuration
	V4L USB devices Arrow keys navigate the menu. <enter> selects submenus&gt;. Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes, <m> modularizes features. Press <esc><esc> to exit, <? > for Help,  tor Search. Legend: [*] built-in [ ] excluded <m> module &lt; &gt;</m></esc></esc></m></n></y></enter>
	<pre> V4L USB devices &lt;*&gt; USB Video Class (UVC) [*] USB video Class device support &lt;*&gt; GSPCA based webvalles&gt;</pre>
	<pre>&lt; &gt; Empia EM28xx USB video capture support &lt; &gt; USB video devices based on Nogatech NT1003/1004/1005 &lt; &gt; USB ET61X[12]51 PC Camera Controller support &lt; &gt; USB SN9C1xx PC Camera Controller support</pre>
	<pre><vre><vre><vre><vre><vre><vre><vre><v< td=""></v<></vre></vre></vre></vre></vre></vre></vre></pre>

「GSPCA based webcams」に入ります。GSPCAはあるフランス人によって作られた汎用USB カメラドライバです。たくさん種類のUSBカメラをサポートしますが、USBカメラは微妙な 差がありますので、アプリケーションは区別に処理しなければなりません。



	root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29	• X
<u>File E</u>	dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp	
.confi	g - Linux Kernel v2.6.29 Configuration	4
Ar Fi dM fo	GSPCA based webcams row keys navigate the menu. <enter> selects submenus ghlighted letters are hotkeys Pressing <y> includes, <n> excludes, &gt; modularizes features. Press <esc> to exit, <? > for Help,  r Search. Legend: [*] built-in [] excluded <m> module &lt; &gt;  GSPCA based webcams &lt;*&gt; ALI USB m5602 Camera Driver &lt;*&gt; STV06XX USB Camera Driver &lt;*&gt; Conexant Camera Driver &lt;*&gt; Conexant Camera Driver &lt;*&gt; Etoms USB Camera Driver &lt;*&gt; Mars USB Camera Driver &lt;*&gt; OV519 USB Camera Driver &lt;*&gt; OV519 USB Camera Driver &lt;*&gt; OV514 USB Camera Driver &lt;*&gt; OV534 USB Camera Driver &lt;*&gt; OV53</m></esc></n></y></enter>	

すべてのUSBカメラを選択して、「Exit」で「Device Drivers」メニューに戻ります。

# 7.4.6 CMOS イメージセンサー(OV9650)

「Device Drivers」メニューの「Multimedia devices」に入ります。

🔳 root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29 📃 🗧	×
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp	
.config - Linux Kernel v2.6.29 Configuration	
Device Drivers Arrow keys navigate the menu. <enter> selects submenus&gt;. Highlighted letters arc hotkeys. Pressing <y> includes, <n> excludes, <m> modularizes features. Press <esc> to exit, <? > for Help,  for Search. Legend: [*] built-in [] excluded <m> module &lt;&gt; </m></esc></m></n></y></enter>	at 100 and 100





「OV9650 on the S3C2440 driver」を選択して、「Exit」で「Device Drivers」メニューに戻ります。

※ このドライバはV4L/V4L2ドライバではありません。普通なキャラク タ・ドライバです。



## 7.4.7 イーサネット

「Device Drivers」メニューの「Networking support」に入ります。



TCP/IPプロトコルを選択して、「Exit」で「Device Drivers」メニューに戻ります。



「Device Drivers」メニューの「Network device support」に入ります。



「Ethernet(10 or 100Mbit)」に入ります。



	root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29 📃 -	ф X
<u>File E</u>	dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp	
.confi	g - Linux Kernel v2.6.29 Configuration	2
Ar Hi fo	Ethernet (10 or 100Mbit)         row keys navigate the menu. <enter> selects submenus&gt;.         ghlighted letters are hotkeys. Pressing <y> includes, <n> excludes,         &gt; modularizes features. Press <esc><to <?="" exit,=""> for Help, &gt; r Search. Legend: [*] built-in [] excluded <m> module &lt;&gt;        </m></to></esc></n></y></enter>	

<\*> Generic Media Independent Interface device support

### <\*> DM9000 support

「Exit」で「Device Drivers」メニューに戻ります。



#### 7.4.8 USB 無線 LAN

「Device Drivers」メニューの「Networking support」に入ります。





「wireless」に入ります。



画面のように「\*」を選択して、「Exit」で「Device Drivers」メニューに戻ります。



「Device Drivers」メニューの「Network device support」に入ります。



📕 root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29 🔤 🛨 🤉
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp
root@tom:/opt/FriendlyA 💥 root@tom:/opt/FriendlyA 💥 root@tom:/opt/FriendlyAR 🐒
.config - Linux Kernel v2.6.29 Configuration
Arrow keys navigate the menu. <enter> selects submenus&gt;.</enter>
Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes,</n></y>
<pre><m> modularizes features. Press <esc> to exit, <?> for Help,  for Search Legendy [*] huilt-in [] excluded <m> module &lt; &gt;</m></esc></m></pre>
(-)
<pre>&lt; &gt; PHY Device support and infrastructure&gt;</pre>
[*] Ethernet (10 or 100Mblt)> [] Ethernet (1000 Mbit)>
[] Ethernet (10000 Mbit)>
Wireless LAN>
USB Network Adapters>
[] Wan interfaces support>
<pre>&lt; &gt; PPP (point-to-point protocol) support <pre></pre></pre>
v(+)
<pre><select> &lt; Exit &gt; &lt; Help &gt;</select></pre>

「Wireless LAN」に入ります。

	root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29 📃 🔸 🕴	\$
<u>F</u> ile	<u>E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp	
root	etom:/opt/FriendlyA 💥 root@tom:/opt/FriendlyA 💥 root@tom:/opt/FriendlyAR ;	×
.00	ifig - Linux Kernel v2.6.29 Configuration	
	Wireless IAN	
	Arrow keys navigate the menu. <enter> selects submenus&gt;.</enter>	
	Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes,</n></y>	
	<pre><m> modularizes features. Press <esc> to exit, <?> for Help, </esc></m></pre>	
	Tor Search. Legend: [*] built-in [] excluded <m> module &lt; &gt;</m>	
	< USB ZD1201 based Wireless device support	
	< > Wireless RNDIS USB support	
	< > Realtek 8187 and 8187B USB support	
	< > Simulated radio testing tool for mac80211	11
	<pre>&lt; &gt; Softmac Prism54 support <pre>&lt; &gt; TEFE 802 11 for Host AD (Prism2/2 5/3 and WED/TKTD/CCMD)</pre></pre>	
	<pre>&lt; &gt; Broadcom 43xx wireless support (mac80211 stack)</pre>	
	< > Broadcom 43xx-legacy wireless support (mac80211 stack)	
	< > ZyDAS ZD1211/ZD1211B USB-wireless support	
	<pre>&lt;*&gt; Ralink driver support&gt;</pre>	
	<pre><select> &lt; Exit &gt; &lt; Help &gt;</select></pre>	
		2

「Ralink driver support」に入ります。



📕 root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29 🔤 🛧 🗶
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp
root@tom:/opt/FriendlyA 💥 root@tom:/opt/FriendlyA 💥 root@tom:/opt/FriendlyAR 💥
.config - Linux Kernel v2.6.29 Configuration
-Ralink driver support
Arrow keys navigate the menu. <enter> selects submenus&gt;.</enter>
Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes,</n></y>
for Search. Legend: [*] built-in [] excluded <m> module &lt; &gt;</m>
Polipk driver support
<pre>&lt; Ralink rt2500 (USB) support</pre>
A*> Ralink rt2501/rt73 (USB) support
[ ] Ralink debug output
<pre><select> &lt; Exit &gt; &lt; Help &gt;</select></pre>

「Exit」で「Device Drivers」メニューに戻ります。

# 7.4.9 オーディオ

「Device Drivers」メニューの「Sound card support」に入ります。

		ro	ot@tom:/	opt/Fr	iendly <i>l</i>	ARM/I	mini 2	2440	)/linu	1x-2.6	i.29			*	ж
File	<u>E</u> dit	View	Terminal	Ta <u>b</u> s	<u>H</u> elp										
. cor	nfig -	Linux	Kernel v	/2.6.29	) Confi	gurat	tion								
	Arrow Highl: <m> mo for Se &lt; { [<sup>3</sup></m>	keys ighted odular earch. (-)- > Har > Gen *] Wat Son Mul Gra *] HID (*)	navigale letters izes feat Legend: dware Mor eric Ther chdog Tin ics Silic tifunctic tifunctic timedia c phics sup nd card s Devices	the me are ho ures. [*] b itorin mal sy her Sup on Bac on devi evices port upport	Devic nu. < tkeys. Press uilt-i g supp sfs dr port kplane ce dri >	e Dri Enter Pre <esc n [ oort iver vers</esc 	<pre>ivers r&gt; se essin c&gt;<fs ]="" ex=""></fs></pre>	<pre>clect g <y c=""> t cclud &gt; &gt; </y></pre>	s su '> in o ex led	bmenu clude it, < <m> m</m>	s s, <n ?&gt; fo odule</n 	>. r Hell < >	Ludes,		



🔲 root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29 🔤 🕂 🗶
<u>File E</u> dit <u>V</u> iew <u>Terminal</u> Ta <u>b</u> s <u>H</u> elp
.config - Linux Kernel v2.6.29 Configuration
Sound card support Arrow keys navigate the menu. <enter> selects submenus&gt;. Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes, <m> modularizes features. Press <esc><esc> to exit, <? > for Help,  for Search. Legend: [*] built-in [] excluded <m> module &lt; &gt;</m></esc></esc></m></n></y></enter>
<pre> Sound card support </pre> Advanced Linux Sound Architecture> < > Open Sound System (DEPRECATED)>
<pre><select> &lt; Exit &gt; &lt; Help &gt;</select></pre>

		ro	ot@tom:/	opt/Fr	iendl	yARM/min	i2440/l	inux-2	.6.29		100	*	ж
File	Edit	View	Terminal	Ta <u>b</u> s	<u>H</u> elp	(							
. CO	nfig -	Linu	c Kernel v	/2.6.29	) Con	figuration	)						
	Arrow Highl <m> m tor S [ [ [ [ [ [ [ [</m>	<pre>/ keys ighted odular earch ] ' ] ' ] ' ] ' ] ' ] ' ] ' ] ' ] ' ] '</pre>	A navigate l letters rizes feat Legend ynamic de Support o erbose pr erbose pr ebug eneric so RM sound JSB sound LISA for	the me are ho tures. : [*] t evice f Id ALSA rocfs of rintk bund de device device SoC au	t Linn enu. Dtkeys Pres Duilt file r A APT conter evices	<pre>ux Sound / <enter> s s. Pressi ss <esc><e -in [] e minor numb nts s&gt; &gt; upport &lt; Exit &gt;</e </esc></enter></pre>	<pre>Architec selects ing <y> isc&gt; to excluded pers</y></pre>	submer incluc exit, i <m></m>	nus: les, <n <? > fo module</n 	<pre>&gt; exclu r Help, &lt; &gt;</pre>	des, 		148.) (148.)
	-			Select>		< <mark>E</mark> xit >	< <mark>H</mark> el	Lp >					
		] ] *>	eneric so RM sound JSB sound ALSA for	ound de device device SoC auc	evices es es lio su	s> > upport < Exit >	> < He1	Lp >				-	



🔳 root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29 🔽 🔫	×							
<u> H</u> ile <u>E</u> dit <u>V</u> iew <u>I</u> erminal Ia <u>b</u> s <u>H</u> elp								
.config - Linux Kernel v2.6.29 Configuration								
ALSA for SoC audio support Arrow keys navigate the menu. <enter> selects submenus&gt;. Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes, <m> modularizes features. Press <esc><esc> to exit, <? > for Help,  for Search. Legend: [*] built-in [] excluded <m> module &lt;&gt;  ALSA for SoC audio support &lt;*&gt; SoC Audio for the Samsung S3C24XX chips &lt;&gt; SoC AC97 Audio support for LN2440SBC - ALC650 &lt;&gt;&gt; SoC I2S Audio support UDA134X wired to a S3C24XX &lt;&gt; Build all ASoC CODEC drivers</m></esc></esc></m></n></y></enter>								
Select> < Exit > < Help >								
	N							

「Exit」で「Device Drivers」メニューに戻ります。

## 7.4.10 SD/MMC メモリカード

「Device Drivers」メニューの「MMC/SD/SDIO card support」に入ります。

🔳 root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29 📃 🔶 🗶
<u>File E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp
.config - Linux Kernel v2.6.29 Configuration
Device Drivers Arrow keys navigate the menu. <enter> selects submenus&gt;. Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes, <m> modularizes features. Press <esc> to exit, <? > for Help,  for Search. Legend: [*] built-in [] excluded <m> module &lt;&gt;</m></esc></m></n></y></enter>



📕 root@tom:	/opt/FriendlyARM/mini2440/linux-2.6.29 📃 🗧	+ X
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>T</u> ermina	Ta <u>b</u> s <u>H</u> elp	
.config - Linux Kernel	v2.6.29 Configuration	
Arrow keys navigate Highlighted letters <m> modularizes fea for Search. Legend [] MMC debug [] Allow uns *** MMC/S &lt;*&gt; MMC block [*] Use bou &lt;&gt; SDIO UART &lt;&gt; MMC host *** MMC/S &lt;&gt; Secure Di &lt;<b>*</b>&gt; Samsung S</m>	<pre>MMC/SD/SDIO card support the menu. <enter> selects submenus ···&gt;. are hotkeys. Pressing <y> includes, <n> excludes, tures. Press <esc><esc> to exit, <?> for Help,  : [*] built-in [] excluded <m> module &lt;&gt; ging afe resume (DANGEROUS) D/SDIO Card Drivers *** tevice driver nce buffer for simple hosts /GPS class support test driver D/SDIO Host Controller Drivers *** gital Host Controller Interface support GC SD/MMC Card Interface support </m></esc></esc></n></y></enter></pre>	

画面のように「\*」を選択して、「Exit」で「Device Drivers」メニューに戻ります。

# 7.4.11 Watchdog

「Device Drivers」メニューの「Watchdog Timer support」に入ります。

🔲 root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29 🔤 🚸 🖓	R.
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp	
.config - Linux Kernel v2.6.29 Configuration	
Device Drivers	
Highlighted letters are hotkeys. Pressing <y> includes. <n> excludes.</n></y>	
<m> modularizes features. Press <esc><esc> to exit, <? > for Help, </esc></esc></m>	
for Search. Legend: [*] built-in [ ] excluded <m> module &lt; &gt;</m>	
Co Hardware Menitoring support	
<pre>&lt; &gt; Generic Thermal sysfs driver&gt;</pre>	
[*] Watchdog Timer Support>	
Sonics Silicon Backplane>	12
Multitunction devices>	
Graphics support>	
<*> Sound card support>	
[*] HID Devices>	
[*] USB support>	
<pre><select> &lt; Exit &gt; &lt; Help &gt;</select></pre>	
	9





「S3C2410 Watchdog」を選択して、「Exit」で「Device Drivers」メニューに戻ります。

# 7.4.12 LED





## 7.4.13 ボタン

「Device Drivers」メニューの「Character devices - - ->」に入ります。



### 7.4.14 PWM ブザー

🔲 root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29 📃 🕂 🕷
<u>File E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp
.config - Linux Kernel v2.6.29 Configuration
Character devices
Highlighted letters are botkeys. Pressing <y> includes <n> excludes</n></y>
<pre><m>modularizes features. Press <esc><to <?="" exit,=""> for Help, </to></esc></m></pre>
for Search. Legend: [*] built-in [] excluded <m> module &lt; &gt;</m>
* Victual terminal
[] Support for binding and unbinding console drivers
[] /dev/kmem virtual device support
<*> LED Support for QQ2440/Mini2440 GPIO LEDs
<*> Buttons driver for FriendlyARM QQ2440/Mini2440 development bo dts Ruzzer driver for FriendlyARM 002440/Mini2440 development books dts Ruzzer dts Ruzzer d
[*] ADC driver for FriendlyARM 002440/Mini2440 development boards
[] Non-standard serial port support
Serial drivers>
-*- Unix98 PTY support
· · · · · · · · · · · · · · · · · · ·
<pre><select> &lt; Exit &gt; &lt; Help &gt;</select></pre>



### 7.4.15 AD

「Device Drivers」メニューの「Character devices - - ->」に入ります。

🔲 root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29 📃 🕂 🕷
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp
.config - Linux Kernel v2.6.29 Configuration
Character devices Arrow keys navigate the menu. <enter> selects submenus&gt;. Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes, <m> modularizes features. Press <esc><to <?="" exit,=""> for Help,  for Search. Legend: [*] built-in [] excluded <m> module &lt; &gt;</m></to></esc></m></n></y></enter>
<pre>-*- Virtual terminal [ ] Support for binding and unbinding console drivers [ ] /dev/kmem virtual device support &lt;*&gt; LED Support for Q02440/Mini2440 GPI0 LEDs &lt;*&gt; Buttons driver for FriendlyARM Q02440/Mini2440 development boa [*] ADC driver for FriendlyARM Q02440/Mini2440 development boards</pre>
<pre>[ ] Non-standard serial port support     Serial drivers&gt;     -*- Unix98 PTY support     v(+)</pre>
<pre><select> &lt; Exit &gt; &lt; Help &gt;</select></pre>

### 7.4.16 シリアルポート





「serial driver」に入ります。



# 7.4.17 リアルタイマーRTC





「Real Time Clock」に入ります。





### 7.4.18 I2C - EEPROM

「Device Drivers」メニューの「I2C support」に入ります。



	root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29 📃 🍝	×
<u>F</u> ile	<u>E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp	
. COI	nfig - Linux Kernel v2.6.29 Configuration	(
	I2C support Arrow keys navigate the menu. <enter> selects submenus&gt;. Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes, <m> modularizes features. Press <esc><esc> to exit, <? > for Help,  for Search. Legend: [*] built-in [] excluded <m> module &lt; &gt;</m></esc></esc></m></n></y></enter>	
	<pre> I2C support      I2C device interface [ ] Autoselect pertinent helper modules         I2C Algorithms&gt;         I2C Hardware Bus support&gt;         Miscellaneous I2C Chip support&gt; [ ] I2C Core debugging messages [ ] I2C Algorithm debugging messages [ ] I2C Bus debugging messages [ ] I2C Chip debugging messages [ ] I2C Chip debugging messages</pre>	194 194
	<pre><select> &lt; Exit &gt; &lt; Help &gt;</select></pre>	



	root@tom:/opt/FriendlyARM/mini2440/iinux-2.6.29 🔤 👘	ж
File	<u>E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp	
. CO	onfig - Linux Kernel v2.6.29 Configuration	1
	I2C Hardware Bus support Arrow keys navigate the menu. <enter> selects submenus&gt;. Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes. <m> modularizes features. Press <esc> to exit. <? > for Help,  for Search. Legend: [*] built-in [] excluded <m> module &lt; &gt;</m></esc></m></n></y></enter>	
	<pre>*** I2C system bus drivers (mostly embedded / system-on-chip) &lt; &gt; GPIO-based bitbanging I2C &lt; &gt; OpenCores I2C Controller &lt;** S3C2410 I2C Driver &lt; &gt; Simtec Generic I2C interface     *** External I2C/SMBus adapter drivers *** &lt; &gt; Parallel port adapter (light) &lt; &gt; TAOS evaluation module &lt; &gt; Tiny-USB adapter     *** Other I2C/SMBus bus drivers *** </pre>	
	<pre><select> &lt; Exit &gt; &lt; Help &gt;</select></pre>	

# 7.4.19 yaff2s ファイルシステム

「Device Drivers」メニューの「Memory Technology Device (MTD) support」に入ります。

		ro	ot@tom:	/opt/Fr	riendlyARM/mini2440/linux-2.6.29 📃 🚸 🗶
<u>F</u> ile	Edit	View	Terminal	Tabs	Help
. CO	nfig -	Linux	Kernel	V2.6.29	9 Configuration
	Arrow Highl <m> m tor S</m>	keys ighted odular earch.	navigate letters izes fea Legend	the me are ho tures. : [*] t	Device Drivers enu. <enter> selects submenus&gt;. otkeys. Pressing <y> includes, <n> excludes, Press <esc><esc> to exit, <? > for Help, built-in [ ] excluded <m> module &lt; &gt;</m></esc></esc></n></y></enter>
	V V (	Gen > Con *> Men > Par *] Blo ] Mis > ATA SCS > Ser ] Mul (+)	cric Dri nector - iory Tech allel po ck devic c device ./ATAPI/M I device ial ATA tiple de	ver Opt unifie nology rt supp es s	tions > ed userspace <-> kernelspace linker> Device (MTD) support> port> -> support> rt> and Parallel ATA (experimental) drivers driver support (RAID and LVM)>
			<	select	<pre>&lt; Exit &gt; &lt; Help &gt;</pre>



root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29 🔤	+ X
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp	
.config - Linux Kernel v2.6.29 Configuration	
Memory Technology Device (MTD) support Arrow keys navigate the menu. <enter> selects submenus&gt;. Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes, <m> modularizes features. Press <esc><esc> to exit, <? > for Help,  for Search. Legend: [*] built-in [] excluded <m> module &lt; &gt;</m></esc></esc></m></n></y></enter>	
<pre>&lt; MTD concatenating support [*] MTD partitioning support &lt; &gt; MTD tests support &lt; &gt; RedBoot partition table parsing [] Command line partition table parsing &lt; &gt; ARM Firmware Suite partition parsing &lt; &gt; TI AR7 partitioning support     **** User Modules And Translation Layers ***</pre>	
<pre>&lt;*&gt; Direct char device access to MTD devices     -*- Common interface to block layer for MTD 'translation layers     v(+)</pre>	
<pre><select> &lt; Exit &gt; &lt; Help &gt;</select></pre>	

画面のように「\*」を選択します。「NAND Device Support」を探して、入ります。

🔲 root@tom:/opt/FriendlyARM/minl2440/linux-2.6.29 두 🗧 🕈	×
<u>File E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp	
.config - Linux Kernel v2.6.29 Configuration	
Memory Technology Device (MTD) support Arrow keys navigate the menu. <enter> selects submenus&gt;. Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes, <m> modularizes features. Press <esc><esc> to exit, <? > for Help,  for Search. Legend: [*] built-in [] excluded <m> module &lt;&gt;</m></esc></esc></m></n></y></enter>	
<pre>&lt;    NFTL (NAND Flash Translation Layer) support &lt;    INFTL (Inverse NAND Flash Translation Layer) support &lt;    Resident Flash Disk (Flash Translation Layer) support &lt;    NAND SSFDC (SmartMedia) read only translation layer &lt;    Log panic/oops to an MTD buffer HAM/ROM/Flash chip drivers&gt; Mapping drivers for chip access&gt; Self-contained MTD device drivers&gt;</pre>	112.1
<pre>&lt; NAND Device Support&gt; &lt; &gt; OneNAND Device Support&gt; v(+)</pre>	
Select> < Exit > < Help >	



	root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29 📃 🛨 🗶
<u>File Edit Vie</u>	ew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp
.config - Li	nux Kernel v2.6.29 Configuration
Arrow ke Highligh <m> modu for Sear [] &lt;&gt; [] [] [] [] [] [] [] ] &lt;&gt; &lt;&gt; &lt;&gt; &lt;&gt; &lt;&gt;</m>	NAND Device Support ys navigate the menu. <enter> selects submenus&gt;. ted letters are hotkeys. Pressing <y> includes, <n> excludes, larizes features. Press <esc> to exit, <? > for Help,  for Help,  ch. Legend: [*] built-in [] excluded <m> module &lt; &gt; Enable chip ids for obsolete ancient NAND devices GPIO NAND Flash driver NAND Flash support for S3C2410/S3C2440 SoC S3C2410 NAND driver debug S3C2410 NAND IDLE clock stop DiskOnChip 2000, Millennium and Millennium Plus (NAND reimp Support for NAND Flash Simulator Support for generic platform NAND driver MTD driver for Olympus MAUSB-10 and Fujifilm DPC-R1 <select> &lt; Exit &gt; &lt; Help &gt;</select></m></esc></n></y></enter>

選択して、メインメニューに戻ります。メインメニューで「File systems」を探して、入ります。

		ro	ot@tom:	/opt/Fri	endly/	ARM/m	nini24	40/lii	nux-2.6	.29		- 10 - 10 - 10 - 10 - 10 - 10 - 10 - 10	4	я
<u>F</u> ile	Edit	View	Terminal	Ta <u>b</u> s	Help									
. COI	nfig -	Linux	Kernel	v2.6.29	Confl	gurati	Lon							0
	Arrow Highl <m> m tor S</m>	keys ighted odular earch.	navigate letters izes fea Legend	the me are ho tures. : [*] b	Kerne nu. < tkeys. Press uilt-i	l Conf Enter> Pres <esc> n [ ]</esc>	figur > sel ssing > <esc ] exc</esc 	ects s <y> i &gt; to e Luded</y>	submenu Include exit, < <m> m</m>	s> s, <n> ?&gt; for odule</n>	exclu Help < >	udes, ,		
	T I	Usc Pow *1 Net Dev Fil Ker Sec	rspace b er manag working tice Driv e system nel hack urity op	inary f ement o support ers S> ing tions	ormats ptions > ->	3	*							Ш
	- V	> Cry Lib	ptograph prary rou	ic API tines	>	Fxit >	>	< Helr	) >				-	
-						LALL ?		- nech		-	-	-		



「Miscellaneous filesystems」に入ります。



「YAFFS2 file system support」を選択します。「Exit」で「Device Drivers」メニューに戻ります。



# 7.4.20 EXT2/VFAT/ NFS ファイルシステム

#### NFS :

「File systems」メニューのNetwork File Systemsに入ります。





### FAT :

「File systems」メニューの「DOS/FAT/NT Filesystems」に入ります。







### 7.4.21 USB-RS232 シリアルポート



弊社が販売している USB-RS232 変換ケーブル

「Device Drivers」メニューの「USB support」に入ります。







「USB Serial Converter support」に入ります。

zqing@localhost:~/linux-2.6.29	÷	×
ファイル( <u>F</u> ) 編集( <u>E</u> ) 表示( <u>V</u> ) 端末( <u>T</u> ) ヘルプ( <u>H</u> )		
.config - Linux Kernel v2.6.29.4 Configuration		$\hat{}$
USB Serial Converter support Arrow keys navigate the menu. <enter> selects submenus&gt;. Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes <m> modularizes features. Press <esc><esc> to exit, <? > for Help, <!--<br-->for Search. Legend: [*] built-in [] excluded <m> module &lt; &gt;</m></esc></esc></m></n></y></enter>	,   ,   ,	
<pre>C USB KL5KUSB105 (Palmconnect) Driver</pre> < > USB KOBIL chipcard reader< > USB MCT Single Port Serial Driver< > USB Moschip 7720 Serial Driver< > USB Moschip 7840/7820 USB Serial Driver< > USB Motorola Phone modem driver< > USB Navman GPS deviceC USB Prolific 2303 Single Port Serial Driver< > USB Ours Technology Inc. OTi-6858 USB To RS 32 Bridge Contr		
<pre><sobb 05b="" <="" driver="" l="" pre="" serial="" spcp8x5="" to="" v(+)=""> <pre><select> &lt; Exit &gt; &lt; Help &gt;</select></pre></sobb></pre>		
		$\sim$

Prolific 社の PL2303 のドライバを選択します。

™┉┉ 株式会社日昇テクノロジー

#### 7.4.22 ARM7TDMI/LPC2148 との通信

ARM9 は標準 OS に Linux を採用します。Linux には、信頼性が高いネットワークスタッ クが実装され、利用できます。従って、ネットワークに接続する信頼性の高い遠隔制御機 器が、容易に作成できる利点があります。Linux にも USB スタックが実装され、多種類の USB デバイスを利用できます。例えば、USB プリンター、USB 無線 LAN、USB メモリ、 SD カードなど。パソコンの Linux 上のアプリケーションが ARM9 上で利用できます。ゼ ロから開発せずに、例えば Web サーバーなどが組み込み用機器で利用できるわけで、これ は非常に大きな利点といえます。

Linuxの便利さの反面、複雑、重い、反応速度が遅いです。反応速度は大体数十ms ぐらいです。この反応速度は人間との会話に満足できますが、機械制御のリアルタイム性に足りないかもしれません。

ARM7 シリーズはリアルタイム制御に向けのマイコンです。OS なしあるいは簡単な RTOS を搭載します。1us~1ms 以上の反応速度が実現できます。LPC2148 は NXP 社によって開 発された ARM7 シリーズのマイコンです。CPU の周波数 60MHz、512KB Flash、42KB RAM。14 チャネル 10 ビット AD、1 チャネル 10 ビット DA、6 チャネルの PWM。

その上、LPC2148にはUSBターゲットポートを持ちます。最大通信速度12Mbps。LPC2148 はUSBデバイスとして使えます。ARM9はUSBハブを経由すれば、何台分のLPC2148 にも接続できます。システムはARM9/MINI2440とARM7/LPC2148を同時に採用すれば、 Linuxの便利な機能とARM7のリアルタイム性を組み合わせ、高度複雑なアプリケーショ ンとリアルタイム制御が両立できるシステムを作れます。



弊社が販売している LPC2148 開発キットとモジュール







ARM9 が ARM7TDMI/LPC2148 を USB で通信する様子



「Device Drivers」メニューの「USB support」に入ります。



「USB Modem(CDC ACM) support」を選択します。"Exit" & "Save" します。





# 7.5 Linux 起動ロゴを作る



Linux が起動の時、このようなロゴが出てきます。自分で好きな画像に変換することができます。

ロゴを作るツールを解凍します。 \$ su スーパーユーザーに切り替え # tar zxvf logomaker.tgz -C /

コンソールで

### \$ logomaker



Logomaker の初の画面です。

File → Open a picture file...で好きなピクチャを選択します。



	the picture file for the loge	0
< 🖹 root		
<u>P</u> laces	Name	~ Modified
🏥 Search	🖾 Desktop	Today at 01:15
Recently Used	🛅 Documents	03/26/2009
🗟 root	💼 Download	03/26/2009
🔯 Desktop	🗄 🚞 Music	03/26/2009
🔜 File System	📋 Pictures	03/26/2009
share_f9 on 192.168.1.	🚞 Public	03/26/2009
🔙 Floppy Drive	🛅 Templates	03/26/2009
Contract Con	🖹 Videos	03/26/2009
Music	📄 girl.jpg	Monday
Pictures		
♣ <u>A</u> dd — <u>R</u> emov	e	Images 🗸
		Cancel
👗 Fr	iendly Linux Logo Maker	- + X

ホームページ:<u>http://www.csun.co.jp</u>

cations Places System 国 🙆 🖳

🗴 Friendly Linux Logo ...

root@tom:~

root 📝 🎂 📰 Mon Apr 6.

Δ



選択したピクチャが表示されます。File → Convert the picture to a Linux Logo File で Linux logo に変換します。linux\_logo\_clut224.ppm というロゴファイルを生成します。このファイルを linux-2.6.29/drivers/video/logo にコピーすれば、新ロゴを生成しまします。

# 7.6 yaffs ルートファイルシステムのイメージを生成

1. yaffsイメージを生成するツールを解凍します。 \$ su スーパーユーザーに切り替え

# tar xvzf mkyaffsimage.tgz -C /

#### # mkyaffsimage root\_qtopia root\_qtopia.img

🔲 root@tom:/opt/FriendlyARM/mini2440 📃 🚸	ж
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp	
Object 3094, root_qtopia/etc/init.d/rcS is a file, 3 data chunks written Object 3095, root_qtopia/etc/init.d/ifconfig-eth0 is a file, 2 data chunks wri	tt
en Object 3096 root gtopia/etc/issue pet is a file 1 data chunks written	
Object 3097. root gtopia/etc/boa is a directory	
Object 3098, root gtopia/etc/boa/boa.conf is a file, 1 data chunks written	
Object 3099, root_qtopia/etc/ftpchroot is a file, 1 data chunks written	
Object 3100, root_qlopia/etc/profile is a file, 1 data chunks written	
Object 3101, root_qtopia/ctc/cth0 setting is a file, 1 data chunks written	
Object 3102, root_qtopia/etc/services is a file, 2 data chunks written	
Object 3103, root_qtopia/etc/localtime is a file, 1 data chunks written	
Object 3104, root dtopia/etc/mtab is a symlink to "/proc/mounts"	
Object 3105, Foot gtopia/etc/passwo is a file, I data chunks written	
Object 3100, root glopia/etc/mime.types is a file, 25 data chunks written	
Object 2107, Foot dtopia/etc/mdev.com is a file, 2 data chunks written	
Object 3100, root dopia/etc/legity.com is a file, i data chunks written	
Object 3110, root donia/etc/hosts is a file 1 data chunks written	
Object 3111, root dtopia/proc is a directory	
Operation complete.	
2855 objects in 247 directories	
86739 NAND pages	
FriendlyARM Computer Technology Inc.	10
[root@tom mini2440]#	1



# 7.7 Linux ドライバの開発入門

Linuxなどの現代的なOSでは、デバイスに対する入出力はデバイスドライバを通じて行うの が常識です。Linuxは「特権モード」を使い、カーネルモードとユーザーモードを厳密に分 離しています。ユーザーモードからは、物理メモリアドレスやI/Oポートなどへのアクセス はできません。したがって、デバイスに対する入出力は、カーネルモードで動作するドラ イバを通じて行うしかないです。

ある例を通じて、カーネルモードで動作するドライバの設計を紹介します。

# 7.7.1 簡単なドライバのソースコード

ソースコード: linux-2.6.29/drivers/char/mini2440\_hello\_module.c

<pre>#include <linux kernel.h=""></linux></pre>				
#include <linux module.h=""></linux>				
static intinit mini2440_hello_module_init(void)				
{				
printk("Hello, Mini2440 module is installed !¥n");				
return 0;				
}				
static voidexit mini2440_hello_module_cleanup(void)				
{				
printk("Good-bye, Mini2440 module was removed!\n");				
}				
<pre>module_init(mini2440_hello_module_init);</pre>				
<pre>module_exit(mini2440_hello_module_cleanup);</pre>				
MODULE_LICENSE("GPL");				

# 7.7.2 コンフィグファイルを編集します

**linux-2.6.29/drivers/char/Kconfig**を開きます。下の内容を添加します(実は、添加完了し ました、確認してみます)。


▼▼■■への単■ 株式会社日昇テクノロジー

	root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29/drivers/char 📃 🔸	×
<u>File</u>	<u>E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp	
	/dev/kmem device is rarely used, but can be used for certain kind of kernel debugging operations. When in doubt, say "N".	() ×(
config	LEDS MINI2440	
	tristate "LED Support for Mini2440/QQ2440 GPI0 LEDs"	
	depends on ARCH_S3C2410	
	This option enables support for LEDs connected to GPIO lines on Mini2440/QQ2440 boards.	
config	MINI2440 HELLO MODULE	
	tristate "Mini2440/QQ2440 module sample"	44
	depends on ARCH_S3C2410	
	belp	
	Mini2440/QQ2440 module sample.	
confia	MINI2440 BUTTONS	
	tristate "Buttons driver for FriendlyARM Mini2440/QQ2440 development b	oa
rds"		
	depends on MACH_FRIENDLY_ARM_MINI2440	
	deraute y IT MACH_FILENDET_ANM_MINIZ440	2



Linux-2.6.29でmake menuconfigを実行して、メニューDevice Drivers  $\rightarrow$  Character devicesを選んで、

		ro	ot@tom:/	opt/Fr	riendiyARM/mini2440/linux-2.6.29 📃 🔶 🗶
File	<u>E</u> dit	<u>V</u> iew	Terminal	Ta <u>b</u> s	<u>H</u> elp
. CD	nfig -	Linux	Kernel v	/2.6.2	9 Configuration
	Arrow Highl ≺M> m for S	/ keys ighteo odular earch.	navigate 1 letters rizes feat . Legend	the me are ho tures. [*]	Character devices menu. <enter> selects submenus&gt;. otkeys. Pressing <y> includes, <n> excludes, Press <esc><esc> to exit, <? > for Help,  built-in [] excluded <m> module &lt;&gt;</m></esc></esc></n></y></enter>
	- [           	*- Vir ] /de *> LEU *> But *> But *> But *> But (*> But (*) (*) (*) (*) (*) (*) (*) (*) (*) (*)	tual tern Support fo ev/kmem vi ) Support ni2440/002 ttons driv zer driver driver driver standaro rial drive	for bind for bind for M 2440 m Ver for for Fr for Fr seria	ding and unbinding console drivers device support ini2440/QQ2440 GPIO LEDs odule sample r FriendlyARM Mini2440/QQ2440 development bo rriendlyARM Mini2440/QQ2440 development boa iendlyARM Mini2440/QQ2440 development bcards al port support >
				elect	> < Exit > < Help >
Ē	-				

添加されたものが見えます。spaceキーで「M」を選択します。

#### 7.7.3 Makefile を編集

linux-2.6.29/drivers/char/Makefile を開きます。下の内容を添加します(実は、添加完了しました、確認してみます)。

```
root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29
File Edit View Terminal Tabs Help
obj-$(CONFIG IPMI HANDLER)
                               += ipmi/
obj-$(CONFIG HANGCHECK TIMER) += hangcheck-timer.o
obj-$(CONFIG TCG TPM)
                               += tpm/
obj-$(CONFIG PS3 FLASH)
                              += ps3flash.p
obj-$(CONFIG JS RTC)
                              += js-rtc.o
js-rtc-y = rtc.o
obj-$(CONFIG LEDS MINI2440)
                               += mini2440 leds.o
obj-$(CONFIG_MINI2440_HELL0_MODULE) += mini2440_hello_module.o
obj-$(CONFIG_MINI2440_BUTTONS) += mini2440_buttons.o
obj-$(CONFIG_MINI2440_BUZZER) += mini2440_pwm.o
obj-$(CONFIG MINI2440 ADC)
                              += mini2440 adc.o
# Files generated that shall be removed upon make clean
clean-files := consolemap deftbl.c defkeymap.c
quiet cmd conmk = CONMK $@
     cmd conmk = scripts/conmakehash $< > $@
$(obj)/consolemap deftbl.c: $(src)/$(FONTMAPFILE)
```



### 7.7.4 ドライバをコンパイルします

linux-2.6.29 で make modules を実行します。linux-2.6.29/drivers/char/でオブジェクトフ アイル mini2440\_hello\_module.ko を生成させます。

🔲 root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29 📃	6 X
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>T</u> erminal Ta <u>b</u> s <u>H</u> elp	
<pre>make[1]: `include/asm-arm/mach-types.h' is up to date.</pre>	()
CHK include/linux/utsrelease.h	
SYMLINK include/asm -> include/asm-arm	
CALL SCIEPTS/CHECKSyScalls.sn	
<stdin>:1097:2: Warning: #Warning syscall radvised4 not implemented</stdin>	
<pre><stdin>.1205.2. Warning. #Warning syscall nselect6 not implemented</stdin></pre>	
<pre><stdim>:1321:2: warning: #warning syscall pooll not implemented</stdim></pre>	
<pre><stdin>:1365:2: warning: #warning syscall cpoll pwait not implemented</stdin></pre>	
CC [M] drivers/char/mini2440 hello module.o	
CC [M] drivers/scsi/scsi_wait_scan.o	
Building modules, stage 2.	
MODPOST 2 modules	
CC drivers/char/mini2440_hello_module.mod.o	
LD [M] drivers/char/mini2440_hello_module.ko	
CC drivers/scsi/scsi wait scan.mod.o	
LD [M] OFIVEFS/SCS1/SCS1 WAIT SCAD.KO	
[root@com Linux-2.6.29]# is grivers/char/mini2440_netto_module.*	
drivers/char/mini/2440_netto_module.c	
drivers/char/mini2440_hello_module.mod.c	
drivers/char/mini2440 hello module.mod.o	
drivers/char/mini2440 hello module.o	110
[root@tom linux-2.6.29]#	) y

### 7.7.5 ARM9 ボードでドライバをインストールします

mini2440\_hello\_module.koをARM9にダウンロードロードします。 #insmod mini2440 hello module.ko

ドライバを削除します。

#rmmod mini2440\_hello\_module.ko



# 第八章 生成されたファイルを書き込む

### 8.1 NOR Flash から起動

ARM9ボードのS2スイッチをNor Flashに設定して、電源を入れて、ARM9ボードはNor Flashから起動します。

Embedded - HyperTerminal	
Ele Edit View Coll Transfer Help	- 『 🧐 A 寂 😂 🤗 🖤 tanai 🤉
##### FriendlyARM BIOS 2.0 for 2440 #####	^ _
[x] format NAND FLASH for Linux	
[v] Download vivi	
[k] Download linux kernel	
[y] Download root_yaffs image	
[a] Absolute User Application	
[n] Download Nboot for WinCE	
[1] Download WinCE boot-logo	
[w] Download WinCE NK.bin	
[d] Download & Run	
[z] Download zImage into RAM	1
[g] Boot linux from RAM	e .
[f] Format the nand flash	
[b] Boot the system	
[s] Set the boot parameters	
[u] Backup NAND Flash to HOST through USB(upload)	
[r] Restore NAND Flash from HOST through USB	
[q] Goto shell of vivi	
[i] Version: 0945-2K	
Enter your selection:	

## 8.2 USB ドライバのインストール

開発されたOSとプログラムをUSB通じてmini2240にダウンロードします。その為、USB ケーブルでmini2240のUSBスレーブポートとパ ソコンのUSBポートを繋ぐことが必要で す。繋ぐと、パソコンは新しいデバイスを発見して、USBドライバをインストールします。









新しいハードウェアの検出ウィザード
検索とインストールのオブションを選んでください。
● 次の場所で最適のドライバを検索する(5) 下のチャック ボックスを使って、リノードゴル メディアやローヤル パスから検索できます。検索された最適のドラ
イバがインストールされます。
□リムーバブル メディア (フロッピー、CD-ROM など)を検索(M)
✓ 次の場所を含める(Q):
C:¥qq2440v3¥windows-tools¥usb-driver 🛛 参照(R)
○検索しないで、インストールするドライバを選択する(D)
→ 暫からドライバを選択するには、このオプションを選びます。選択されたドライバは、ハードウェアに最適のものとは限りません。
< 戻る(B) 次へ(N) > キャンセル





新しいハードウェアの検出ウィザード				
新しいハードウェアの検索ウィザードの完了 次のハードウェアのソフトウェアのインストールが完了しました: SEC SOC Test Board				
	[完了]をクリックするとウィザードを閉じます。 < 戻る(B) 完了 キャンセル			

USBドライバをインストール完了あと、パソコンのダウンロード・ツールDNW.exeを実行 して、mini2240とパソコンを繋ぐことが確認できます。

DET	<b>v</b> 0.5	DA	[CO	[:x][USB:OK]	
Serial	Port	USB	Port	Configuration	Help
					•

※ USBドライバはバッグがあります。ARM9ボードが再起動、又はリ セットの時、ホスト側は死んだかもしれません。その原因で、ARM9 ボードが起動完了した後、USBケーブルでホストを繋ぎます。

### 8.3 NAND Flash のパーティション

メニューの中で、機能号[x]を選択して、NAND Flash のパーティション画面が出てきます。 ※ NAND Flash の中にエラーアリアがあるかもしれません。使用の影 響がありません。



□ COT1 (1) - CRT	
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>O</u> ptions <u>T</u> ransfer <u>S</u> cript <u>W</u> indow <u>H</u> elp	
17 X7 X7 X7 In Co Q Is 5 5 5 6 12 X7 9 4	
<pre>[q] Quit Enter your selection: s Enter the parameter's name(mach_type, media_ty Enter the parameter's value(if the value conta root=/dev/mtdblock2 init=/linuxrc console=ty Change linux command line to "noinitrd root=/o C1,115200"</pre>	<pre>pe, linux_cmd_line, etc): linux_cmd_line ins space, enclose it with "): "noinitrd SAC1,115200" ev/mtdblock2 init=/linuxrc console=ttySA</pre>
<pre>###### Parameter Menu ###### [r] Reset parameter table to default table [s] Set parameter [v] View the parameter table [w] Write the parameter table to flash memeory [q] Quit Enter your selection: w Found block size = 0x0000c000 Erasing done Writing done Writing done Written 49152 bytes Saved vivi private data</pre>	
##### Parameter Menu ##### [r] Reset parameter table to default table [s] Set parameter [v] View the parameter table [w] Write the parameter table to flash memeory [q] Quit Enter your selection:	
Ready Ser	ial: COM1 28, 23 28 Rows, 87 Cols Linux 🛛 🖊

### 8.4 ブートロードの書き込み

メニューの中で、機能号[x]を選択して、NAND Flash のパーティション画面が出てきます。 パソコンで DNW を実行します。



DNW のタイトルに[USB: OK]があれば、パソコンと ARM9 ボードを USB で繋ぎました。



メニューの中で、機能号[v]を選択して、

Embedded - HyperTerminal	
Ele Edit View Call Transfer Help	🐨 🥹 A RX 🚽 🌳 🔮 sans +
##### FriendlyARM BIOS 2.0 for 2440 #####	
[x] format NAND FLASH for Linux	
[v] Download vivi	
[k] Download linux kernel	
[y] Download root_yaffs image	
[a] Absolute User Application	
[n] Download Nboot for WinCE	
[1] Download WinCE boot-logo	
[w] Download WinCE NK.bin	
[d] Download & Run	
[z] Download zImage into RAM	1
[g] Boot linux from RAM	
[f] Format the nand flash	
[b] Boot the system	
[s] Set the boot parameters	
[u] Backup NAND Flash to HOST through USB(upload)	
[r] Restore NAND Flash from HOST through USB	
[q] Goto shell of vivi	
[i] Version: 0945-2K	
Enter your selection:	

#### DNW を待っています。DNW のメニュー「USB Port」→「Transmit」を選択して、





ブートロード supervivi\_mini2440 を選択して、「開く」を押します。
\* Mini2440 セットの場合: LCD 出荷状態: *Toppoly* 3.5" LCD
\* Micro2440 セットの場合: LCD 出荷状態: *INNOLUX* 7" LCD

🔁 ファイルを開く	×
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	<ul> <li>✓ 4→ linux-imagesの検索</li> </ul>
整理 ▼ 新しいフォルダー	64MNand Flash ボード用のルートファイル
<ul> <li>↓ ダウンロード</li> <li>▲ デスクトップ</li> <li>※ 最近表示した場所</li> <li>▲ root_qtopia-64M.img</li> <li>※ root_gtopia-128M img</li> </ul>	128M以上 Nand Flashボード用のルート 2009/04/29 21:15 Image Files 2009/07/19 0:24 Image Files
64M Nand Flash ボード用のブートローダ Supervivi-64M 128M Nand Flash ボード用のブートローダ Supervivi-128M	INNOLUX 7" LCD用のイメージファイル(出荷)
ZImage_A70 ZImage_L80 NEC 3.5° LCD用のイメージファイル ZImage_N35	Sharp 7* LCD用のイメージファイル 2009/07/09 3:22 ファイル VGA(1024×768)町のイメージファイル
Toppady 3.5" LCD用のイメージファイル(出荷) ZImage_T35 MP (C:) FACTORY IMAGE (D:)	2009/07/09 3:23 ファイル
	4
ファイル名( <u>N</u> ): zImage_N35_CDCACM	<ul> <li>すべてのファイル (*.*)</li> <li>ツール(L)</li> <li>聞く(0)</li> <li>キャンセル</li> </ul>

ブートロードを書き込み完了すると、自動的にメニューに戻ります。



### 8.5 Linux のカーネルの書き込み

a. メニューの中で、機能号[k]を選択して、

Embedded - HyperTerminal		- 0
Fie For New Coll Iteaster, Helb	1 /06 🔍 🖓 izeri	<u>a</u>
##### FriendlyARM BIOS 2.0 for 2440 #####		
[x] format NAND FLASH for Linux		
[v] Download vivi		
[k] Download linux kernel		
[y] Download root_yaffs image		
[a] Absolute User Application		
[n] Download Nboot for WinCE		
[1] Download WinCE boot-logo		
[w] Download WinCE NK.bin		
[d] Download & Run		
[z] Download zImage into RAM	I	
[g] Boot linux from RAM		
[f] Format the nand flash		
[b] Boot the system		
[s] Set the boot parameters		
[u] Backup NAND Flash to HOST through USB(upload)		
[r] Restore NAND Flash from HOST through USB		
[q] Goto shell of vivi		
[i] Version: 0945-2K		
Enter your selection:		

カーネルをダウンロードすることを待っています。



b. DNW のメニュー"USB Port → Transmit"を選択して、Linux カーネルファイル zImage\_XXXを選択して、「開く」を押します。

尾 ファイルを開く	
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	<ul> <li>← fy linux-imagesの検索</li> </ul>
整理 ▼ 新しいフォルダー	64MN and Flash ボード用のルートファイル
ダウンロード     名前       ■ デスクトップ     root_qtopia-64M.img       図 最近表示した場所     root_qtopia-128M.img	128M以上 Nand Flashボード用のルート 2009/04/29 21:15 Image Files 2009/07/19 0:24 Image Files
64M Nand Flash ボード用のブートローダ 128M Nand Flash ボード用のブートローダ 128M Nand Flash ボード用のブートローダ	INNOLUX 7" LCD用のイメージファイル(出荷)
ZImage_A70 ZImage_L80 NEC 3.5" LCD用のイメージファイル ZImage_N35	Sharp + 200所のイメ・ファイル 2009/07/09 3:22 ファイル VGA(1024×768)用のイメージファイル
Toppaly     3.5"     LCD用のイメージファイル(出荷)        ZIMade_133        ZImage_VGA1024x768        FACTORY_IMAGE (D:)        DVD RW ドライブ (E:) AIRNAVI12 + イ	2009/07/09 3:23 ファイル
ファイル名( <u>N</u> ): zImage_N35_CDCACM	<ul> <li>すべてのファイル (*.*)</li> <li>ツール(L)</li> <li>開く(0)</li> <li>年ャンセル</li> </ul>
Image_A70 7インチ液晶カーネル	

zImage\_T35 3.5 インチ液晶カーネル zImage\_VGA1024X768 VGA カーネル

c. 転送完了したら、自動的にメニューに戻ります。



## 8.6 ルート・ファイルシステムの書き込み

a. メニューの中で、機能号[y]を選択して、

Embedded - HyperTerminal		- 6 - *
Ele gat Yew Gall Irensfer Help		A
##### FriendlyARM BIOS 2.0 for 2440 #####		
[x] format NAND FLASH for Linux		
[v] Download vivi		
[k] Download linux kernel		
[y] Download root_yaffs image		
[a] Absolute User Application		
[n] Download Nboot for WinCE		
[1] Download WinCE boot-logo		
[w] Download WinCE NK.bin		
[d] Download & Run		
[z] Download zImage into RAM	I	1
[g] Boot linux from RAM		*
[f] Format the nand flash		
[b] Boot the system		
[s] Set the boot parameters		
[u] Backup NAND Flash to HOST through USB(upload)		
[r] Restore NAND Flash from HOST through USB		
[q] Goto shell of vivi		
[i] Version: 0945-2K		
Enter your selection:		

ルート・ファイルシステムをダウンロードすることを待っています。



b. DNW のメニュー"USB Port → Transmit"を選択して、ルート・ファイルシステム root\_qtopia.imgを選択して、「開く」を押します。

区 ファイルを開く	×
Weight and American Action	<ul> <li>✓ 4→ linux-imagesの検索</li> </ul>
整理 ▼ 新しいフ: フレター	64MN and Flash ボード用のルートファイル
	128M以上 Nand Flashボード用のルート
図 root_qtopia-64M.img 図 最近表示した場所 図 root_qtopia-128M.img	2009/04/29 21:15 Image Files 2009/07/19 0:24 Image Files
64M Nand Flash ボード用のブートローダ Supervivi-64M	INNOLUX 7" LCD用のイメージファイル(出荷)
128M Nand Flash ボード用のブートローダ ZImage_A70	Sharp 7" LCD用のイメージファイル
ZImage_L80 NEC 3.5" LCD用のイメージファイル ZImage_N25	2009/07/09 3:22 ファイル
Toppoly 3.5" LCD用のイメージファイル(出荷) ZImage_T35	VGA(1024×768)用のイメージファイル
A HP (C:)	2009/07/09 3:23 ファイル
➡ FACTORY_IMAGE (D:)	
See DVD RW ドライブ (E:) AIRNAVI12 ▼  イ	m. •
ファイル名( <u>N</u> ): zImage_N35_CDCACM	▼ すべてのファイル (*.*) ▼
	ツール(L) ▼ 【聞く(0) ▼ キャンセル

c. 転送完了したら、自動的にメニューに戻ります。

電源を切って、mini2440の起動 S2 を NAND Flash で起動に設定してください。再び電源 を入れて、NAND Flash で書き込み済みの Linux は起動します。

## 8.7 NAND Flash のバックアップ

※ 新ブートロード supervivi のみ メニューの中で、機能号[u]を選択して、



低価格、高品質が不可能?

日昇テクノロジーなら可能にする

Embedded - HyperTerminal     Ene Fdit View Call Transfer Help		- 0 - ×
		*
WWWW Friendlyake Blos 2.0 for 2440 WWWW		
LxJ format NAND FLASH for Linux		
[v] Download vivi		
[k] Download linux kernel		
[y] Download root_yaffs image		
[a] Absolute User Application		
[n] Download Nboot for WinCE		
[1] Download WinCE boot-logo		
[w] Download WinCE NK.bin		
[d] Download & Run		
[z] Download zImage into RAM	1	n
[g] Boot linux from RAM		
[f] Format the nand flash		
[b] Boot the system		
[s] Set the boot parameters		
[u] Backup NAND Flash to HOST through USB(upload)		
[r] Restore NAND Flash from HOST through USB		
[q] Goto shell of vivi		
[i] Version: 0945-2K		
Enter your selection:		

### DNW のメニュー「Usb Port」→「Backup NandFlash to File」を選択します。



バックアップのファイルの名前「backup.bin」を入力して



◆     ★ttyS0 - 超级终端     □	
Serial Port USB Port Configuration Help Flags : 0xc5 size : 64MB erasesize : 16KB oobblock : 512 oobsize : 16 [] Boot linux from [] [] Boot linux from [] [] Partition for Lin [] Boot the system [] Set the boot para [] Set the boot para [] Print the IOC ctr [] Backup Information: [] Print the IOC ctr [] Backuping to C:\Documents and Settings\plg\ Backup Information: [] Got Enter y Backup Information: times	
Start Addr End Addr       bBackup00B     : 1       bCheckBad     : 1       dwBackupTotalLen     : 0x4200000       dwReservedBlks     : 20       dwEPInPktSize     : 32       Use dnw.exe to receive backup.     Send Flash Backup: [======       Send Flash Backup: [======     62%]	

バックアップ完了したら、次の画面:



DNW ▼0.50A 友善之臂改进版(増加备份功能) [COM:x][ 🔳 🗖	×
Serial Port USB Port Configuration Help	
USB Backup Start	^
Nand Flash Information:	
type : 0x4	
flags : 0xc5	
size : 64MB	
erasesize : 16KB	
oobblock : 512	
oobsize : 16	
ecctype : 0x2	
eccsize : 250 Backup Information:	
Start Addr • 0x0	
Fod Addr : 0x000000	=
hBackun00B : 1	
bCheckBad : 1	
dwBackupTotalLen : 0x4200000	
dwReservedB1ks : 20	
dwEPInPktSize : 32	
dnw.exe read data 65536 bytes a time, has to read 1056	
times	
======================================	
	~

生成されたバックアップファイルの大きさは 66MB ぐらいです。

## 8.8 NAND Flash のリストア

※ 新ブートロード supervivi のみ

メニューの中で、機能号[r]を選択して、



低価格、高品質が不可能?

日昇テクノロジーなら可能にする

Embedded - HyperTerminal		- 0 <b>- × -</b>
Ble Edit View Call Transfer Help	✓ ● A 按 ● ● ✓ savi ▼	A
##### FriendlyARM BIOS 2.0 for 2440 #####		
[x] format NAND FLASH for Linux		
[v] Download vivi		
[k] Download linux kernel		
[y] Download root_yaffs image		
[a] Absolute User Application		
[n] Download Nboot for WinCE		
[1] Download WinCE boot-logo		
[w] Download WinCE NK.bin		
[d] Download & Run		
[z] Download zImage into RAM	I	1
[g] Boot linux from RAM		=
[f] Format the nand flash		
[b] Boot the system		
[s] Set the boot parameters		
[u] Backup NAND Flash to HOST through USB(upload)		
[r] Restore NAND Flash from HOST through USB		
[q] Goto shell of vivi		
[i] Version: 0945-2K		
Enter your selection:		

### DNW のメニュー「Usb Port」→「Transmit/Restore」を選択します。

I DNV v0.	50▲ 友善之臂改进舨 (増加备份功能) [CO∎:x][	
Serial Port	USB Port Configuration Help	
	Transmit/Restore =========	^
Nand Flast	Backup NandFlash to File	
	Status	
	size : 64MB	
	erasesize : 16KB	
	oobblock : 512	
	ecctupe : 0x2	
	eccsize : 256	
Backup Inf	Formation:	
	Start Addr : 0x0	Ξ
	End Addr : 0x4000000	
	DBackupuus : 1	
	dwBackunTotallen · Øyb200000	
	dwReservedB1ks : 20	
	dwEPInPktSize : 32	
dnw.exe re	ad data 65536 bytes a time, has to read 1056	
times		
	===== USB Backup End =============	
		~

バックアップされたファイル「backup.bin」を選択します。



🔷 ttyS0 - 超级终端	
文件(2) 编辑(2) 查看(2)	📟 DNW ▼0.50A 友善之臂改进舨(增加备份功能) [COM:x][ 📘 🗖 🗙
06 🖉 🗑 着 0	Serial Port USB Port Configuration Help
<pre>[i] Download WinCl [w] Download WinCl [d] Download &amp; Run [z] Download &amp; Run [z] Download zImag [g] Boot linux fro [f] Format the nam [p] Partition for [b] Boot the syste [s] Set the boot p [t] Print the TOC [u] Backup NAND F [t] Restore NAND F [t] Restore NAND F [t] Goto shell of Enter your select: USB host is conner Now, Downloading Flash params: oobs 67108864 Erasing and program Block erasing(add)</pre>	<pre>Nand Flash Information: type : 0x4 flags : 0xc5 size : 64MB erasesize : 16KB oobblock : 512 oobsize : 16 ecctype : 0x2 Backup Inf Downloading C:\Documents and Settings\plg\ X Backup Inf Downloading C:\Documents and Settings\plg\ X dwBackupTotalLen : 0x4200000 dwReservedBlks : 20 dwEPInPktSize : 32 dnw.exe read data 65536 bytes a time, has to read 1056 times </pre>
0x 003a 0000/ 0	3233 _ 0×00000000/00000 00232/04096=05%
<	
已连接 0:47:31 自动检测	115200 8-N-1 SCROLL CAPS NUM 捕打印

### 8.9 メモリで Linux カーネルを直接に実行

一般的に Linux のカーネルを NAND Flash に書き込み、実行させます。毎回 Linux カー ネルを更新すれば、NAND Flash も更新することが必要です。デバッグの時、不便です。 ブートロード Suppervivi は Linux カーネルをメモリにロードして、直接に実行します。

- 1. 電源を切って、mini2440の起動 S2 を Nor Flash で起動に設定してください。再び電源を入れて、Nor Flash で起動します。
- 2. Superviviのメニューの中で、機能号[z]を選択して、



低価格、高品質が不可能?

日昇テクノロジーなら可能にする

Embedded - HyperTerminal		
Eie Ear Ziem Zail Itauste. Helb	1 HK V Inni V	*
##### FriendlyARM BIOS 2.0 for 2440 #####		
[x] format NAND FLASH for Linux		
[v] Download vivi		
[k] Download linux kernel		
[y] Download root_yaffs image		
[a] Absolute User Application		
[n] Download Nboot for WinCE		
[1] Download WinCE boot-logo		
[w] Download WinCE NK.bin		
[d] Download & Run		
[z] Download zImage into RAM	1	n.
[g] Boot linux from RAM		r
[f] Format the nand flash		
[b] Boot the system		
[s] Set the boot parameters		
[u] Backup NAND Flash to HOST through USB(upload)		
[r] Restore NAND Flash from HOST through USB		
[q] Goto shell of vivi		
[i] Version: 0945-2K		
Enter your selection:		

- 3. DNW のメニュー"USB Port → Transmit"を選択して、カーネルファイル zImage を転送します
- 4. 転送完了したら、自動的に Supervivi のメニューに戻ります。機能号[g]を選択して、 linux カーネルを実行させます。この画面が出たら:

🏶 ttyS0 - 超级终端	×
文件(g) 编辑(g) 查看(y) 呼叫(g) 传送(g) 帮助(y)	
D 🖆 🍘 🕉 🗈 🗃	
<pre>s3c2410-sdi s3c2410-sdi: running at 198kHz (requested: 197kHz). s3c2410-sdi s3c2410-sdi: s3cmci_request: no card s3c2410-sdi s3c2410-sdi: s0cmci_request: no card s3c2410-sdi s3c2410-sdi: s10-sdi: s10-sdi s3c2410-sdi s3c2410-sdi: s10-sdi s3c2410-sdi s3c2410-sdi: s10-sdi s3c2410-sdi s3c2410-sdi: s10-sdi s3c2410-sdi s3c41</pre>	
已连接 0:18:25 ANSIW 115200 8-N-1 SCROLL CAPS NMM 描 打印	.:



ルートファイルシステムが見つかりませんでした!

Superviviのメニューの中で、機能号[y]を選択して、NAND Flash に root\_default.img を 書き込みます。ルートファイルシステムを作ります。又は、NFS をルートファイルシス テムとして指定します。

Linux カーネルを実行させる前に、NFS を指定します。Supervivi のメニューの中で、機能号[q]を選択して、次のコマンドを入力してください。

Supervivi>param set linux\_cmd\_line ''console=ttySAC0 root=/dev/nfs nfsroot=192.168.1.111:/root\_nfs

ip=192.168.1.70:192.168.1.111:192.168.1.111:255.255.255.0:MINI2440.arm9.net:et h0:off'' (NFSの設定)

Supervivi> boot ram (メモリでカーネルを起動させます)



## 第九章 NOR Flash のブートロードを更新

※ 一般的に NOR Flash のブートロードを更新することが必要ないです。

### 9.1 簡易 JTAG で書き込み

NOR Flash は H-JTAG というツールで更新されます。

H-JTAG は ARM の為の JTAG エミュレータです。AXD 又は keil をサポートします。デバッグのスピードも速いです。詳しい情報はこちらです。

http://www.hjtag.com

弊社は H-JTAG のハードウェアを提供しております。パソコンは LTP が必要です。

#### 9.1.1 H-JTAG をダウンロードとインストールします

ホームページ http://www.hjtag.com から最新版をダウンロードできます。

H-JTAG の特性:

- a. RDI 1.5.0 & 1.5.1 をサポートします;
- b. ARM7 & ARM9(ARM9E-S と ARM9EJ-S を含む);
- c. thumb & arm 命令;
- d. little-endian & big-endian;
- e. semihosting;
- f. 実行環境 WINDOWS 9.X/NT/2000/XP;
- g. flash の書き込み



弊社は H-JTAG のハードウェアを提供しております。パソコンは LTP が必要です。

インストール完了すれば、デスクトップで H-JTAG と H-Flasher を生成します。H-JTAG を実行すると、このエラーメッセージが出てきます。





設定しないから。"Ok"ボタンを押すと、初の画面が出てきます。



H-JTAG  $\mathcal{O} \neq \exists \exists \neg \neg$ : Setting  $\rightarrow$  Jtag Settings



itag Settings		
Jtag Selection	User Def	ined Pin Assignment –
C Wiggler (Predefined)	TMS	Pin4 D2 💌
C Sdt Jtag (Predefined)	TCK	Pin2 D0 💌
User Defined	TDI	Pin3 D1 💌
Reset Signal Output	TDO	Pin11 Busy 💌
nTRST output inverted	nTRST	NO TAP RST 💌
r nSRST output inverted	nSRST	NO SYS RST 👻

この様な設定して、"Ok"ボタンを押します。

H-JTAG O × =  $\neg$  − : Script → Init Script

ldx	Cmd	Width	Address	Value	
					•

"Load"ボタンを押します。



ファイルを聞く		? 🛛
ファイルの場所型:	H-JTAG	- ← 🗈 👉 📰
E Friendly ARM24	440.his	
ファイル名(N):	FriendlyARM2440.his	
ファイルの種類(工):	H-Jtag Init Script (*.his)	• キャンセル

FriendlyARM2440.his というファイルを選択します。

次の画面が出て	き	ま	す	-
---------	---	---	---	---

ldx	Cmd	Width	Address	Value	•
1	Setmem	32-Bit	0x53000000	0x00000000	
2	Setmem	32-Bit	0x4A000008	<b>OxFFFFFFF</b>	
3	Setmem	32-Bit	0x4A00001C	0x000007FF	
4	Setmem	32-Bit	0x53000000	0x00000000	
5	Setmem	32-Bit	0x56000050	0x000055AA	
6	Setmem	32-Bit	0x4C000014	0x0000007	
7	Setmem	32-Bit	0x4C000000	0x00FFFFFF	
8	Setmem	32-Bit	0x4C000004	0x00061012	
9	Setmem	32-Bit	0x4C000008	0x00040042	
10	Setmem	32-Bit	0x48000000	0x22111120	
11	Setmem	32-Bit	0x48000004	0x00002F50	-

"Ok"ボタンを押します。ご注意: "Enable Auto Init"をチェックしない。

パソコンと ARM9 ボードを H-JTAG で繋ぎます。ARM9 ボードの電源を入れます。 H-JTAG のメニュー: Operations → Detect Target を選択すると





H-JTAG はターゲット ARM ボードを認識しました。



#### 9.1.2 NOR Flash を書き込む

※ ARM9 ボードが NOR Flash から起動することを確認してください。

H-JTAG のメインメニュー「Flasher」→「Start H-Flasher」で H-Flasher を実行します。



H-Flasherのメインメニュー「Load」、H-Flasher\_mini2440.hfcというファイルを開きます。

🔁 H-Flasher - H-Fl	asher_QQ2440.hfc	
New Load Save Save	As Options Exit About	
Program Wizard	>> Flash Selection	

H-Flasherの左側の「4 Programming」を選択します。



🖬 H-Flasher - H-Flasher_QQ2440.hfc					
New Load Save Save	s Options Exit About				
Program Wizard	>> Programming - AM29LV160DE	3			
1 Flash Selection	Flash: Unchecked Target: Unchecked	Check			
3 Init Script					
4 Programming	Type: Auto Flash Download	✓ Program			
👎 H-Flasher Help	Src File:				
	Dst Addr:				
	From: Entire Chip	✓ Erase			
	To: Entire Chip	▼ Blank			

「Check」ボタンを押すと、mini2440が使用したNor Flash(AM29LV160DB)を発見 します。

🖬 H-Flasher - H-Flasher_QQ2440.hfc							
New Load Save Save As Options Exit About							
Program Wizard	>> Progra	amming - AM29LV160DB					
1 Flash Selection 2 Memory Config 3 Init Script	Flash: Target:	AM29LV160DB 0x22490001 ARM920T Little-Endian		Check			
Programming	Туре:	Auto Flash Download	•	Program			
₹ H-Flasher Help	Src File: Dst Addr:						
	From: To:	Entire Chip Entire Chip	•	Erase Blank			



「Type」の「Plain Binary Format」を選択します。

🖬 H-Flasher - H-Flasher_QQ2440.hfc					
New Load Save Save	As Options Exit About				
Program Wizard	>> Programming - AM29LV160DB				
1 Flash Selection	Flash: Unchecked Check				
2 Memory Config	Target: Unchecked				
3 Init Script	· · · · · · · · · · · · · · · · · · ·				
4 Programming	Type: Auto Flash Download    Program				
👎 H-Flasher Help	Src File: Auto Flash Download Intel Hex Format Plain Binary Format Dst Addr:				
	From: Entire Chip				
	To: Entire Chip   Blank				

書き込みのファイルsuperviviを選択します。「Dst Addr」で0を入力します。「Program」 ボタンを押すと、Nor Flashに書き込みます。



🖻 H-Flasher - H-F	lasher_QQ2440.hfc	
New Load Save Sav	e As Options Exit About	
Program Wizard	>> Programming - AM29LV160DB	
1 Flash Selection	Flash: Unchecked	Check
2 Memory Cor H-Fla	sher	
3 Init Scrip	Sent same	
4 Programmir	supervivi	Program
🤋 H-Flasher	00:01:10 19% 22 KB/s Size = 123.2 KB	
	Stop	
	From: Entire Chip	Erase
	To: Entire Chip 💌	Blank

書き込み完了の画面:

H-Flasher					
		Programmed and verified successfully.			
		00:05:40 100% 23 KB/s	Size = 123.2 KB		
			Close		



## 9.2 Open-JTAG で書き込み



弊社が販売している Open-JTAG は ARM 用 の USB-JTAG エミュレータです。ARM7、 ARM9、 Cortex-M3, XSCALE に対応、 OpenOCD をサポートします。USB-RS232 機能もあります。COM と LPT ポートがない ノートパソコンに最適。



Opne-JTAGで書き込み様子



#### 9.2.1 ドライバをインストールする

※Windows 2000、XP用ドライバダウンロード URL :

http://www.dragonwake.com/download/open-jtag/open-jtag-driver.zip

※Windows7 用ドライバーダウンロード URL:

http://www.dragonwake.com/download/open-jtag/open-jtag-driver\_win7.zip

「usb-drivers」フォルダーにある。

OpenJTAG をパソコンの USB ポートに挿入して、下の通りにドライバをインストールしてください。





新しいハードワェアの使出ワイザ	-  s
	このウィザードでは、)太のハードウェアに必要なソフトウェアをインストールします: USB<=>JTAG&RS232
	〈戻る(四) 次へ(N) > キャンセル
おしひり、 たち、スの 於川古 が	
新しいパードウェアの検出ウイリ	- k
新しいハードウエアの後出ウイリ 検索とインストールのオブション	ード を選んでください。
	►F を選んでください。 (パを検索する(5) さって、リムーパブル メディアやローカル パスから検索できます。検索された最適のドラ さす。 マイア (フロッピー、CD-ROM など)を検索(M) ふる(D): ¥usb-driver ● 参照(P) ールするドライバを選択する(D) ごするには、このオプションを選びます。選択されたドライバは、ハードウェアに最適のもの

₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩	- テクノロジー 日昇テクノロジーなら可能に
新しいハードウェアの検出ウィザード ソフトウェアをインストールしています。お待ちくま	きい
100ASK JTAG	
~~ >>	
	< 戻る(B) 次へ(N) > キャンセル

USB ドライバのインストールは3回があります。インストール完了すると、デバイスマネ ージャで三つのデバイスが見えます。



□□ デバイス マネージャ	
ファイル(E) 操作(A) 表示(V) ヘルプ(H)	
는 🙀 USB (Universal Serial Bus) 그가 🗆 - ㅋ	~
C IUUASK Serial Port	
The ICV OZOUT BA/ BAM USB Universal Host Controller - 2442	
NEC PCI to USB Enhanced Host Controller (B1)	
NEC PCI to USB Controller	
NEC PCI to USB Open Host Controller	
🖧 USB ルート ハブ	
ー 🥰 USB ルート ハブ	
ー 🥰 USB ルート ハブ	
… 🥰 USB ルート ハブ	
ー 🅰 USB ルート ハブ	
- 😋 USB 複合デバイス	=
日 🧼 キーボード	
日 100 サワンド、ビナオ、およりケーム コンドローフ	
田 🦉 ンステム ナノバス	
□ ● ● 11/2 1217	
ロー 電気 スットワーク アガブタ	
ローロー ビューマン インターフェイス デバイス	
田 🕞 フロッピー ディスク コントローラ	
🗉 🧸 フロッピー ディスク ドライブ	
🖕 💭 ポート (COM と LPT)	
%VID_1457&PID_5118.DeviceDesc% (COM4)	the second second
<u>夏 5CP プリンク ポート (LPT1)</u>	~

※ OpenJTAG は USB シリアルポートとして使えます。

### 9.2.2 書き込み

※書込みツールダウンロード URL:

上記 USB ドライバダウンロード URL と同じ

※Windows 2000、XP用ドライバダウンロード URL:

http://www.dragonwake.com/download/open-jtag/open-jtag-driver.zip

※Windows7 用ドライバーダウンロード URL :

http://www.dragonwake.com/download/open-jtag/open-jtag-driver win7.zip

「tools」フォルダーにある。

※ブートロードファイルダウンロード URL:

<u>http://www.dragonwake.com/download/arm9-download/linux-2.6.32.2/linux-images.tgz</u> 解凍後、色んな種類のボード用のブートロードファイルがありますが、ボードの種類によ り、正しいファイルを選べてください。(ファイルの説明は中身の「Readme.txt」を参照)



#### C:¥openJTAG¥open-jtag>sjf24x0\_ft2232.exe supervivi\_mini2440

+-----+

- | Flash Programmer for OpenJTAG of www.100ask.net
- OpenJTAG is a USB to JTAG & RS232 tool based FT2232
- This programmer supports both of S3C2410X & S3C2440
- Author: Email/MSN(thisway.diy@163.com), QQ(17653039)

+-----+

Usage: sjf24x0\_ft2232.exe [filename]

Select the CPU:

1. S3C2410X

2. S3C2440X

### Enter the number: **2**

S3C24X0 detected, cpuID = 0x0032409d

[Main Menu]

0:Nand Flash prog	1:Nor Flash prog	2:Memory Rd/Wr	3:Exit
1 0	1 8	5	

#### Select the function to test:1

Detect Nor Flash ... SST 39VF1601 Size: 2 MB

Image Size: 0x1f314

~略~

Erasing ..... done write ... 100% done


# 第十章 Web カメラストリーミング配信

### 10.1 MJPG-streamerのダウンロードとコンパイル

mjpg-streamerの最新バージョンをダウンロードします。

\$ svn co https://mjpg-streamer.svn.sourceforge.net/svnroot/mjpg-streamer mjpg-streamer

\$ cd mjpg-streamer/mjpg-streamer

**\$ make CC=arm-linux-gcc** 成功すれば実行ファイルmjpg\_streamerとライブラリ・ファイル\*.soを生成します。

### 10.2 MJPG-streamer を mini/micro2440 ボードにインストール

生成されたファイル(mjpg\_streamer、\*.so)とmjpg-streamerのwwwディレクトリをmini/micro2440ボ ードにダウンロードして、特定なディレクトリにをコピーしてください。

- # mkdir -p /usr/lib/
- # cp \*.so /usr/lib/
- # cp mjpg\_streamer /usr/bin/
- # mv www /www/uvc-www



弊社が販売している UVC(USB Device Class)に対応した Web カメラを mini/micro2440 の USB ホストに接続して、 mini/micro2440 のコンソールで次のコマ ンドで MJPG-streamer を起動させます。

# mjpg\_streamer ---input "input\_uvc.so ---device /dev/video0 ---fps 5 ---resolution 640x480 ---yuv" ---output "output\_http.so ---port 8080 ---www /www/uvc-www" MJPG Streamer Version.: 2.0

- i: Using V4L2 device.: /dev/video0
- i: Desired Resolution: 640 x 480
- i: Frames Per Second.: 5



i: Format.....: YUV

i: JPEG Quality.....: 80

format asked unavailable get width 352 height 288

o: www-folder-path...: /www/uvc-www/

o: HTTP TCP port.....: 8080

o: username:password.: disabled

o: commands.....: enabled

※ UVC カメラによって、"-yuv"というパラメータは不要の可能性があるかもしれません

※ UVC 以外のカメラは input\_gspcav1.so を使います。

### 10.3 Web ブラウザで Web カメラを見ましょう

Web ブラウザで、「http://mini/micro2440 ボードの IP アドレス:8080/」にアクセスすると、 MJPG-Streamer Demo Pages が表示されます。静止画、動画、および Pan/Tilt/LED の On/Off 等の制御をすることができます。(Internet Exploler 6 及び 7 では、MJPEG によるストリーム(動画) を閲覧することができません。しかし、Javascript を使用したストリーム(動画)は、 Internet Explorer でも閲覧することができます。)

Web ブラウザで見る様子:





# 第十一章 Eclipse + GCC + Open-JTAG

### 11.1 GCC ツールチェーン

各種ユーティリティ:

http://www.yagarto.de/download/yagarto/yagarto-tools-20070303-setup.exe

GCCツールチェーン

http://sourceforge.net/projects/yagarto/files/YAGARTO%20for%20Windows/yagarto-bu-2.19.1 gcc-4.3.3-c-c%2B%2B nl-1.17.0 gi-6.8.50 20090329.exe/download

インストールが出来たら make の確認をするためコマンドプロンプトを起動し、右記のコ マンドを入力します(make --version)。画面に下記のメッセージが出てくればOKです。





#### サンプルのコンパイル:

1. コマンドプロンプトでディレクトリを移動

2. 下記のコマンドを入力します

(make)



コンパイル成功したら、\*.binファイルを生成させます。



### **11.2 Integrated Development Environment(Eclipse)**

#### JRE バージョン確認:

確認コマンド: java -version



JRE がなければ、あるいは 1.4.2 以下なら、JRE のインストールが必要です。 <u>http://java.sun.com/javase/downloads/index.jsp</u>

### Eclipse のインストール:

1) 下記のリンクをクリック

http://www.eclipse.org/downloads/index.php

2) Eclipse IDE for C/C++ Debelopers(79MB)の Windows をクリック



🕘 Ec lipse	Jownloads - Microsoft Internet Explorer		
ファイル(圧)	編集(E) 表示(M) お気に入り(A) ツール(T) ヘルプ(H)		<b>.</b>
🔇 戻る 🔹	🕥 - 💌 😰 🏠 🔎 検索 🌟 お気に入り 🔗 😒 - 🗟 🔜 -	📕 🔏 🕹	28
アドレス(D) 纋	http://www.eclipse.org/downloads/index.php	🖌 🏹 移動	助 リンク <sup>≫</sup>
Norton	🛒 🕑 🔹 🕞 カードとログイン情報 🔹		
Google	🗸 🚰 検索・🖗 🍏・ 🛖・ 🧟 ・ 🏠 ブックマーク・	᠉ 🔌 • 🔵 айн:	2• 🛖 ·
	🔍 = 🔍 🔍 🔶 🏈 DAEMON Tools Lite 🔻 🎇 AstroBurn 🌄 Product	ts 🕶 <u>ត</u> ា News [30/30] 🕶	» 🥳
10015		LUNIO V IDA	<b>^</b>
đ	Eclipse IDE for Java Developers (92 MB) The essential tools for any Java developer, including a Java IDE, a CVS client, XML Editor and Mylyn. More Downloads: 120,776	Windows Mac OS X (Carbon) Mac OS X (Cocoa) Linux 32bit Linux 64bit	Popi Popi
Php	Eclipse for PHP Developers (139 MB) Tools for PHP developers creating Web applications, including PHP Development Tools (PDT), Web Tools Platform, Mylyn and others. More Downloads: 71,421	Windows Mac OS X (Carbon) Mac OS X (Cocoa) Linux 32bit Linux 64bit	= W = M = M
C	Eclipse IDE for C/C++ Developers (79 MB) An IDE for C/C++ developers with Mylyn integration. More Downloads: 56,017	Windows Mac OS X (Carbon) Mac OS X (Cocoa) Linux 32bit Linux 64bit	= C/ = Bi Ri = Si
	Eclipse for RCP/Plug-in Developers (183 MB) A complete set of tools for developers who want to create Eclipse plug-ins or Rich Client Applications. It includes a complete SDK, developer tools and source code, plus Mylyn, an XML editor and the Eclipse Communication Framework. More Downloads: 17,776	Windows Mac OS X (Carbon) Mac OS X (Cocoa) Linux 32bit Linux 64bit	■ M ■ Vi = Gi (G ❤
<		)	>
e		🔏 🔰 🥑 インターネット	

#### 3) 画面の下矢印をクリックしダウンロード





4) ダウンロードしたファイル"eclipse-cpp-galileo-win32.zip"を解凍し、そのなかの"eclipse"フォルダを適当な場所(C:¥eclipse)へ移動する。

5) Eclipse を起動する。



6) 最初に Workspace の場所を聞いてきます。適当なフォルダに変更してください。

🖶 Workspace Launcher	
Select a workspace Eclipse stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: C¥workspace	Browse
♥ this as the default and do not ask again	OK Cancel





画面の Workbench をクリックします。

E/C++ - Eclipse Platform					
Eile Edit Refactor Navigate Search Project	<u>R</u> un <u>W</u> indow <u>H</u> elp				
📬 • 🔚 📥   📸 • 🛍 • 💕 • 🞯 •	• ] 🗞 + 🔅 + ] 🏇 + 💽 + 💁 + ] 🤔	🔗   T 📮	$] \ge - \forall + \leftarrow +$	֥ _/	😭 🔂 C/C++
Project Explorer 🕱 🗧 🗖				- 6	🛛 🗄 Outlin 🛛 💿 Make 🖵 🗖
					An outline is not available.
	🖹 Problems 🛛 🧟 Tasks 📮 Console 🔲	Properties			⇒ ~ □ □
	0 errors, 0 warnings, 0 infos		1		
	Description A	Resource	Path	Location	
] ∎≎					] 🕋 🌻 🐼 💝



# 11.3 プロジェクトを作る

新規プロジェクトを作成するため"File"→"New"→"C Project"を選択します

<u>N</u> ew	Alt+Shift+N	🕨 📸 C++ Project	
Open File <u>.</u>		C Project	
<u>C</u> lose	Ctrl+W	Project	
Close All	Ctrl+Shift+W	E Convert to a C/C++ Project	
U Save	Ctrl+S	Source Folder	
Save <u>A</u> s			
👸 Sav <u>e</u> All	Ctrl+Shift+S	C Source File	
Rever <u>t</u>		► File from Template	
Mo <u>v</u> e		<b>G</b> Class	
Rena <u>m</u> e	F2		
🐑 Refresh	F5		-IN

プロジェクト名を聞かれるので適当な名前(LED)を入力し Finish ボタンを押します。

🖨 C Project	
<b>C Project</b> Create C project of selected type	
Project name: LED	
✓ Use default location	
Location: C:¥zero¥Example-2148GCC¥18.US	Btarget¥LED Browse
Project type:	Toolchains:
Empty Project	Other Toolchain
Show project types and toolchains only if	they are supported on the platform
	Next > Einish Cancel



**Project Explorer** にプロジェクト LED が追加されましたが中身が何もないので、"×"がついています。

ter c/C++ − Eclipse							
<u>F</u> ile	<u>E</u> dit	<u>S</u> ource	Refac <u>t</u> or	<u>N</u> avigate	Se <u>a</u> rch	<u>R</u> un	<u>P</u> roje
1 📬	• 🖫		) 🗄 🔂 -	🚳 • 🕻	• 🞯 •	: «	• 📎
F 🗗	Project	Explorer	×				
			Ē	1 🔄 🔤			
± į	🗳 LEC	)					

サンプル LEDs.zip のなかのファイルを"C:¥workspace¥LED"にコピーしてください。

Eclipse の"File"→"Refresh"を選択します。





**Project Explorer** の"LED"プロジェクトの左にある+をクリックするとファイルの一覧が 表示されます。



# 11.4 Eclipse プラグイン(Zylin Embedded CDT)インストール

Eclipse の"Help"→"Install New Software"を選択します

🖨 C/C++ - Eclipse					
<u>File E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate Sea	arch <u>R</u> un	<u>P</u> roject	<u>W</u> indow	Help	
📬 • 🔛 📥   📾   💣 • 😂 • 💽 • 🧿	9 - I 🔨	- 🛞 -	参• (	🚳 <u>W</u> elcome	
Project Explorer 🕱 📃 🗖				⑦ Help Contents	
				💯 S <u>e</u> arch	
				<u>D</u> ynamic Help	
ED			•	Key Assist Tips and Tricks Report Bug or Enhancement Oheat Sheets Check for Updates Install New Software About Eclipse	Ctrl+Shift+L



Add ボタンを押します。

🖨 Install		
Available Software Select a site or enter the location of a site	e.	
Work with <sup>®</sup> type or select a site	Find more software by working with the <u>Available Software Sites</u>	Add preferences.
type filter text		
Name Ve	ersion	
- Details		
Show only the latest versions of availab	ole software <u>H</u> ide items that are already installed	
Group items by category	What is <u>already installed</u> ?	
☑ <u>C</u> ontact all update sites during install to	o find required software	
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel

Add Site の"Location"に"http://opensource.zylin.com/zylincdt "と入力し OK ボタンを押す。

🖨 Add S	Site	
<u>N</u> ame:		Local
Location:	http://opensource.zylin.com/zylincdt	<u>Archive</u>
?	ОК	Cancel



Install に"http://opensource.zylin.com/zylincdt "が追加されるのでチェックボックスにク リックしチェックを入れて Next ボタンを押す。

🖨 Install	
Available Software Check the items that you wish to install.	
Work with: http://opensource.zylin.com/zylincdt Find more so	ftware by working with the <u>'Available Software Sites'</u> preferences.
type filter text	
Name Version	
Details	
Show only the latest versions of available software	<u>H</u> ide items that are already installed
☑ <u>G</u> roup items by category	What is <u>already installed</u> ?
☑ Contact all update sites during install to find required s	oftware
0	K Back Next > Einish Cancel



🖨 Install		
Install Details Review the items to be installed.		
Name	Ver Id	
🚯 Zylin Embedded CDT	4.8.1 com.zylin.cdt.feature.feature.group	
Size: Unknown - Details		×
•	< <u>B</u> ack ■ext > Einis	h Cancel



🖨 Install	
<b>Review Licenses</b> Licenses must be reviewed and accepted before the software can be installed.	
License text:	
ECLIPSE FOUNDATION SOFTWARE USER AGREEMENT March 17, 2005	<u>^</u>
Usage Of Content	
THE ECLIPSE FOUNDATION MAKES AVAILABLE SOFTWARE, DOCUMENTATION, INFORMATION AND/OR OTHER MATERIALS FOR OPEN SOURCE PROJECTS (COLLECTIVELY "CONTENT"). USE OF THE CONTENT IS GOVERNED BY THE TERMS AND CONDITIONS OF THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. BY USING THE CONTENT, YOU AGREE THAT YOUR USE OF THE CONTENT IS GOVERNED BY THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT SOR OF ANY APPLICABLE LICENSE AGREEMENTS OR OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW, THEN YOU MAY NOT USE THE CONTENT.	
Applicable Licenses	
Unless otherwise indicated, all Content made available by the Eclipse Foundation is provided to you under the terms and conditions of the Eclipse Public License Version 1.0 ("EPL"). A copy of the EPL is provided with this	~
<ul> <li>OI accept the terms of the license agreement</li> <li>OI do not accept the terms of the license agreement</li> </ul>	
	Cancel

🖶 Install 📃 🗖 🔀
Fetching com.zylin.embeddedcdt_4.8.1.jar/plugins/com.zylin.embeddedcdt_4.8.1.jar
S Install
Fetching com.zylin.embeddedcdt_4s/com.zylin.embeddedcdt_4.8.1.jar
Run in <u>B</u> ackground Cancel <u>&lt;&lt; D</u> etails





インストール完了したら、Yes ボタンを押して、Eclipse を再起動させます。

### 11.5 ビルドの設定

Eclipse の"Project"→"Preferences"を選択する。





Preferences の"Build"を選択し"Scanner Configuration Builder"のチェックマークを外して

🖨 Properties for LED		
type filter text	Builders	, , , , , , , , , , , , , , , , , , ,
Builders C/C++ General Project References Refactoring History Run/Debug Settings Task Repository WikiText	Configure the builders for the project:	New         Import         Edit         Remove         Up         Down
?	OK	Cancel



Preferences の"C/C++ Build"→"Settings"を選択し"Binary Parsers"タブの"GNU Elf Parser"にチェックを入れて OK ボタンを押します





### 11.6 ビルド

Eclipse  $\mathcal{O}$ "Project"→"Build Automatically"のチックを外してください。

🖨 C/C++ - LED/leds.c - Eclipse				
<u>F</u> ile <u>E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u>	rch <u>R</u> un	<u>Project Window H</u> elp		
	• • • «	Op <u>e</u> n Project Clo <u>s</u> e Project	2	
Project Explorer 🛛 🗖 🗖	leds.c	🔜 Build All	Ctrl+B	
E S crt0.S	#def #def	Build Configuration Build Project Build Working Set Clean Build Automatical	ıs ⊧.e ⊧.e	
jii crt0.o jii gdb.cmd jii leds_elf jii leds.bin jii leds.dis	#def #def #def	Make Target Properties	,	

"Project"→"Clean"を選択するクリアが行われます。





€	🖨 Clean	IX
) E	Clean will discard all build problems and built states. The next time a build occurs the projects will be rebuilt from scratch.	
(	Clean all projects     Clean projects selected below	V
	C C LED	
	Start a <u>b</u> uild immediately	
	Build the entire workspace	
	O Build only the selected projects	
	OK Cance	

「Start a build immediately」のチェックマークを外して、「Ok」を押します。

🖨 C/C++ - LED/leds.c - Eclipse <u>File Edit Source Refactor Navigate Search Run Project Window H</u>elp Open Project 🗄 🗂 T 🔚 📥 | 🚠 🗄 📸 T 🚳 T 🚱 T 🗟 🔨 🦓 Clo<u>s</u>e Project 🚹 Project Explorer 🙁 💽 leds.c Build <u>A</u>II Ctrl+B E \$  $\nabla$ Build Configurations 59 ۲ #def Build Project 🖃 📂 LED #def Build Working Set ۲ le 🗄 🔝 crt0.S Clea<u>n</u>... 🗄 🔂 leds.c Build Automatically #def 🐻 gdb.cmd #def --- 📄 leds.lds Make Target ۲ #def --- 🗋 Makefile **Properties** #def

"Project"→"Build All"を選択するとビルドが行われます。



🖨 Building Workspace	
Building all	
Always run in background	
Run in Background Cancel	etails >>

コンパイル中です。



コンパイルが成功すれば、実行ファイル led\_elf.elf と leds.bin を生成されます。

### 11.7 GDB の設定

Eclipse の"Run"→"Debug Configurations..."を選択します。



🖨 C/C++ – Eclipse	
<u>F</u> ile <u>E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rcl	n <u>R</u> un <u>P</u> roject <u>W</u> indow <u>H</u> elp
Project Explorer X   Project Explorer X     ED   Binaries   ED.c   ED.elf - [arm/le]   ED.o - [arm/le]   ED.dmp   ED.hex	Image: Second system       Ctrl+F11         Image: Second system       F11         Run History       F11         Run As       F11         Run As       F11         Run Configurations       F11         Debug History       F11         Debug History       F11         Debug As       F11         Image: Second system       F11         Run Configurations       F11         Debug As       F11         Image: Second system       F11         Image: Seco
ED.map Makefile	<ul> <li>Toggle <u>M</u>ethod Breakpoint</li> <li>Toggle Watchpoint</li> </ul>

Debug Configurations の"Zylin Enbedded debug(Native)"を右クリックし"New"を選択す

る。 Debug Configurations Create, manage, and run configurations 1 1 × 1 7 . Configure launch settings from this dialog: type filter text Press the 'New' button to create a configuration of the selected type. C/C++ Application Press the 'Duplicate' button to copy the selected configuration. C/C++ Attach to Applicat 💢 - Press the 'Delete' button to remove the selected configuration. C/C++ Postmortem Debu ► Launch Group 🔆 - Press the 'Filter' button to configure filtering options. -**≣c** Zylin Embedded de<u>hue (</u>C 🧊 🐨 🥵 🖉 🦉 🦉 🦉 or view an existing configuration by selecting it. Duplicate 🗶 Delete aunch perspective settings from the <u>Perspectives</u> preference page.



Name に適当な名前を入れる。例、"gdb"と入れます。Main タブの"Project"に"LED"、"C/C++ Appication:"に"C:¥workspace¥LED¥leds\_elf"と入力します。

🖨 Debug Configurations		×
Create, manage, and run cor	figurations	2
Image: Second state st	Nair	
Filter matched 7 of 7 items	Apply Revert	כ
?	Debug Close	



Debugger タブの"GDB debugger:"に"arm-elf-gdb"、"GDB command file:"に何も入力しま せん。

Debug Configurations	🔀
Create, manage, and run con	igurations to the second se
Image: Second system         Image: Second system <td>Name:       edb         Main       Debugger:       Commands         Debugger:       Embedded GDB       Advanced         Debugger:       C*Program Files¥yagarto¥bin¥arm=elf=gdb.exe       Browse         GDB debugger:       C*Program Files¥yagarto¥bin¥arm=elf=gdb.exe       Browse         GDB commands file:       Browse       Browse         Warning:       Some commands in this file may interfere with the startup operation of the debugger, for example "run".)         GDB command set:       Standard         Protocol:       mi         Image:       Verbose console mode         Use full file path to set breakpoints       Use full file path to set breakpoints</td>	Name:       edb         Main       Debugger:       Commands         Debugger:       Embedded GDB       Advanced         Debugger:       C*Program Files¥yagarto¥bin¥arm=elf=gdb.exe       Browse         GDB debugger:       C*Program Files¥yagarto¥bin¥arm=elf=gdb.exe       Browse         GDB commands file:       Browse       Browse         Warning:       Some commands in this file may interfere with the startup operation of the debugger, for example "run".)         GDB command set:       Standard         Protocol:       mi         Image:       Verbose console mode         Use full file path to set breakpoints       Use full file path to set breakpoints
Filter matched 7 of 7 items	Apply Revert
?	Debug Close



Commands タブの"'Initialize' command	ls"に下記の画面の様に入力します
-----------------------------------	-------------------

target remote	localhost:3333	
monitor halt		//ボードの

monitor halt	//ボードの実行を停止させる
monitor arm920t cp15 2 0	// MMU機能をクローズ
monitor step	//ステップで実行するように
load	//leds_elfをロード
break main	//「main」関数にブレークポイントを設定
continue	//プログラムを実行させて、「main」にストップ

Debug Configurations		×
Create, manage, and run con	nfigurations to the second sec	
Image: Second state st	Name:       gdb         Main	
Filter matched 7 of 7 items	Apply Revert	
?	Debug Close	



Debug Configurations	
Create, manage, and run cor	figurations
Image: Second state st	Name:       gdb         Main        Debugger         Save as       I Local file         Shared file:       Image: Im
Filter matched 7 of 7 items	Apply Revert
?	Debug Close

Common タブの"Display in favorites menu"の Debug にチェックを入れます。全てを入力 し終えたら"Apply"ボタンを押し、"Close"ボタンを押します。



## 11.8 OpenOCD の設定

Eclipse の"Run"→"External Tools."→"External Tools Configrurations..."を選択します。

🖨 C/C++ - LPC2388_LED/main.c - Eclipse		
<u>F</u> ile <u>E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rch	<u>Run</u> <u>P</u> roject <u>W</u> indow <u>H</u> elp	
: <mark>™・∷≥</mark> : : <u>∳</u> ・¦i-;>:>:	Q₀ Bun         Ctrl+F11           ‰ Debug         F11	• 0 • 9 • 1 😕 😂 🛷 • 1 🛃
Project Explorer 🛛 🖓 🕞 🖓	Run His <u>t</u> ory Run A <u>s</u> Ru <u>n</u> Configurations…	*
<ul> <li>■ <sup>25</sup> LPC2388_LED</li> <li>⊕ <sup>3</sup> Binaries</li> <li>⊕ <sup>3</sup> Includes</li> </ul>	Debug <u>H</u> istory Debug As De <u>b</u> ug Configurations	*/
⊞⊸S erts ⊞-LPC23xxh ⊞-L® main.c	● Toggle Breakpoint Ctrl+Shift+B ● Toggle Line Breakpoint	
⊞ _ In typedefsh ⊞ - In typedefsh ⊞ - In crto - [arm/le] ⊞ - IPC2388_LED_rom.elf - [arm/le] ⊞ - In main.o - [arm/le] - In crt.lst	<ul> <li>Toggle <u>M</u>ethod Breakpoint</li> <li><sup>6</sup> Toggle <u>W</u>atchpoint</li> <li><u>N</u> Skip All Breakpoints</li> <li><u>Remove</u> All Breakpoints</li> <li>Breakpoint Types</li> </ul>	ed LED */
PLPC2388_LED_rom.hex LPC2388_LED_rom.map DLPC2388 ram.id	External Tools	(no launch history)
ipc2388_rom.ld imain.lst imain.lst imain.lst	<pre>void LED_Off (char num) {      FIO2CLR = FIO2CLR   num; }</pre>	External Tools Configurations Organize Fayorites



External Tools Configrurations の"Program"を右クリックし、"New"を選択します。

듣 External Tools Configuratio	ns X
Create, manage, and ru Run a program	n configurations
Filter matched 1 of 1 items	Configure launch settings from this dialog:             • Press the 'New' button to create a configuration of the selected type.             • Press the 'Duplicate' button to copy the selected configuration.             • Press the 'Delete' button to remove the selected configuration.             • Press the 'Iter' button to configure filtering options.             • Edit or view an existing configuration by selecting it.             Configure launch perspective settings from the Perspectives preference page.
0	Run Close



Main タブの"Name"に適当な名前を入力してください。私は"OpenOCD"と入れました。 "Location:"に"C:¥Program Files¥OpenOCD¥0.1.0¥bin¥openocd.exe"、

"Working Directory:"に"C:¥openJTAG"、

"Arguments:"に「 "interface/open-jtag.cfg" 「 "target/samsung\_s3c2440.cfg" -c "arm7\_9 fast\_memory\_access enable"と入力します。

🖨 External Tools Configurat	ions 🛛 🗙
Create, manage, and run cor Run a program	nfigurations O
Program       OpenOCD	Name: apenOCD         Main       Refresh         Main       Refresh         OpenHert         C:#Program Files#OpenOCD#01.0#bin#openocd.exe         Browse Workspace       Browse File System         Working       Directory:         C:#openJTAG       Browse Workspace         Browse Workspace       Browse File System         Variables    Areuments: -1* "interface/open-jtag.cfg" -f " target/samsung_s3c2440.cfg" -c "arm7_9 fast_memory_access enable" Variables Variables Variables
Filter matched 2 of 2 items	Apply Revert
?	<u>R</u> un Close



🖨 External Tools Configurations 🛛 🗙		
Create, manage, and run cor Run a program	figurations	
type filter text	Name: OpenOCD  Main Refrest Build Environment Common  Build Beild Benvironment  Common  Specific projects  Include referenced projects	P <u>r</u> ojects
Filter matched 2 of 2 items		Apply Revert
?		<u>B</u> un Close

Build タブをクリックし"Build before launch"にチェックを外れます。



External Tools Configurat	ions	
Create, manage, and run cor Run a program	nfigurations	
Image: Second system         Image: Second system <th>Name:       openOCD         Main       Refresh         Save as          O Lgcal file          Display in favorites menu          Image: Console file          Standard Input and Output          Image: File:          Append          Image: Launch in background</th> <th>Browse     Console Encoding     O Default - inherited (MS932)     Other     ISO-8859-1     Workspace     File System     Variables</th>	Name:       openOCD         Main       Refresh         Save as          O Lgcal file          Display in favorites menu          Image: Console file          Standard Input and Output          Image: File:          Append          Image: Launch in background	Browse     Console Encoding     O Default - inherited (MS932)     Other     ISO-8859-1     Workspace     File System     Variables
Filter matched 2 of 2 items		
?		<u>R</u> un Close

Common タブをクリックし"Display in favarites menu"の"External Tools"にチェックを入 れます。全てを入力し終えたら"Apply"ボタンを押し、"Close"ボタンを押します。

## 11.9 デバッグ

電源投入

- 1. OpenJTAG をターゲット(MINI2440 ボード)とパソコンに接続
- 2. ターゲットに電源を入れます



External Tools の▼ボタンをクリックし、OpenOCD を選択



Console ウインドに下記のメッセージが出力

💽 Probler	ns 🙋 Tasks	📃 Console 🙁	Properties			
openOCD [	Program] C:¥Pr	ogram Files¥Open(	) 📕 🗶 🎽	🔓 🚮 🗐	🖉 🛃	- 📬 -
Info :	JTAG tap:	s3c2440.cpu	tap/device	found: 0:	x0032409d	(Ma 🐴
Info :	JTAG Tap/o	device matche	ed			
Warn :	no telnet	port specif:	ied, using	default po	ort 4444	
Warn :	no gdb por	t specified,	, using def	ault port	3333	
Warn :	no tel por	t specified,	, using def	ault port	6666	~
<						>

Debug の▼ボタンをクリックし、"gdb"を選択。





Con	firm Perspective Switch	×
?	This kind of launch is configured to open the Debug perspective when suspends.	it
	This Debug perspective is designed to support application debugging. incorporates views for displaying the debug stack, variables and breakpoint management.	It
	Do you want to open this perspective now?	
Ren	ember my decision	
	Yes No	

Yesボタンを押して、デバッグが開始します。





Eclipse に Debug 用のコマンドあるいはショットカット一覧 詳しくは Eclipse のドキュメントを参照



ステップ実行において良く使われる操作の一覧を以下に示します。

操作名	ショートカットキー
再開	F8
ステップイン	F5
ステップオーバー	F6
ステップ・リターン	F7

ステップ実行とは関係ありませんが、前回起動したクラスを再度実行したデバッグする場合は、 以下のショートカットキーが便利です。

操作名	ショートカットキー
前回の起動を実行	Ctrl + F11
前回の起動をデバッグ	F11

ブレークポイントでプログラムが中断した状態から、次のブレークポイントまで実行させたり、1行ずつ実行させたりできます。コード「GPBDAT = (~(i<<5))」を繰り返して実行することにより、LEDランプが1つずつ点滅



低価格、高品質が不可能? 日昇テクノロジーなら可能にする

📮 Debug – D:¥embeded¥eclipse¥workspace¥mini2440¥MINI2440_LED¥leds.c – Eclipse 📃 🗖 🗙			
Eile Edit Source Refactor Navigate Search Project Run Window Help			
📬 • 🖫 👜   🖬   : 🏇 • 🔕 • 🖓 •   🧶 😂 🔗 •   🍠   : 🖗 • 🖗 🗢	• ⇔ - 🖹 🐝 Debug 😼 C/C++		
🏇 Debug 🛛 🖓 🗖	🕪= Variables 🛛 🛛 🚻 Registers 🛋 Modules) 🐁 🏘 📄 💠 🌮 🗮 🗐		
	Name Value		
Que openocd [Program]	(x)= i 3		
Difembededfopenocafbinfopenoca.exe			
Section 2010 Control Cont			
🗍 🍿 Thread [0] (Suspended)			
■ 1 main 0 D.¥embeded¥eclipse¥workspace¥mini2440¥MINI2440_LED¥leds.c:23 0			
🛛 🚽 D.¥embeded¥yagarto¥bin¥arm-elf-gdb.exe (09/07/16 0:32)			
🖻 leds.c 🛛 🗖 leds.c 🕅	Disassembly 🛛 🖓 🖓		
	∧ 0x00000058 <main+20>: add r3, r3, #16 ; 0x10 ∧</main+20>		
int main(void)	0x0000005c <main+24>: mov r2, #87040 ; 0x15400</main+24>		
	0x00000060 <main+28>: str r2, [r3]</main+28>		
unsigned Long 1 = 0;	UXUUUUUU064 <main+32>: mo⊽ rU, #29952 ; UX7500</main+32>		
GPBCON = GPB5 out/GPB6 out/GPB7 out/GPB8 out: // (	PB 0x0000006c <main+30>: all 10, 10, #40 , 0x30</main+30>		
	0x00000070 <main+44>: mov r2, #1442840576 ; 0x560</main+44>		
while(1)(	0x00000074 <main+48>: add r2, r2, #20 ; 0x14</main+48>		
(10000),	0x00000078 <main+52>: ldr r3, [sp]</main+52>		
◆ < GPBDAT = (~(i<<5)); // iの値により、LED1-4を点減させる	0x0000007c <main+56>: lsl r3, r3, #5</main+56>		
	0x0000080 <main+60>: mvn r3, r3</main+60>		
i = 0;	0x00000084 <main+64>: str r3, [r2]</main+64>		
	> <		
Concele X P Tacke Richleme S Evenutables R Memory			
MNI2440Dabur [Zulin Embedded dabur (Currein)] D¥ambadad¥uaranta¥bin¥armaalf-rdb ava (00			
Millitz440Debug (Zylin Enibeddeb debug (Oygwin) D.+enibeded+yaga (0+bin+anin eni gdb.exe (05/	6//10/032/		
Breakpoint 1, main () at leds.c:17			
17 unsigned long i = 0;			
Warning: /cygdrive/D/embeded/eclipse/workspace/mini2440/MINI2440_LED: No such file or directory.			
mi_cmd_disassemble: Invalid filename.			
	×		
	<u>&gt;</u>		
	Writable Smart Insert 23:1		



Debug 途中の ARM9 ボードの様子の1(一番右の LED ランプが点灯)






Debug 途中の ARM9 ボードの様子の2(右から2番目の LED ランプが点灯)

## 11.10 デバッグ終了

1) gdb の停止

Debug ウインドウの gdb の Thread を選択し、停止ボタンと押します



🚝 Debug - Source not found Eclipse Platform
<u>F</u> ile <u>E</u> dit <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp
] 📸 ▪ 📰 💩   🚠   ॐ ▪ 🕥 ▪ 💁 ▪ ] 😕 🛷 ▪   ½ → ᅒ → ┶ - ⇔ → →
🎋 Debus 🕱 💫 🖄 🌾 🕪 🗉 📵 🗉 🕄 👁 🕐 📼 🖬
🖻 🗐 😴 gdb [Zylin Embedded debug (Native)]
🖨 🧽 Embedded GDB (09/03/25 17:31) (Suspended)
🖻 🔐 🏠 Thread [1] (Suspended)
1 <symbol available="" is="" not=""> 0x00000000</symbol>
arm-none-eabiredb (00/00/25 17:01)

2) OpenOCD の停止

Debug ウインドウの OpenOCD の Thread を選択し、停止ボタンと押します

ig – S	Source not	found E	Eclipse Pla	atform							
<u>E</u> dit	<u>N</u> avigate	Se <u>a</u> rch	<u>P</u> roject	<u>R</u> un	<u>W</u> indov	v <u>H</u> elp					
H	a   🗟	] 🏇 🔹	0 - 9	<b>-</b>	<u>0</u> 8	• ] ½	•	· م	← ¢	• •	÷ •
bug (	×		\$ I -	(				The second secon	i⇒ 3	v 🗸	
Оре	nOCD [Pro	ograml				_					
C:¥Program Files¥OpenOCD¥0.1.0¥bin¥openocd.exe											
⊟-fc gdb [Zylin Embedded debug (Native)]											
@ <sup>©</sup> <terminated>Embedded GDB (09/03/25 17:31) (Exited. Exit code = 0.)</terminated>											
-	<terminate< td=""><td>d, exit val</td><td>ue: O&gt;arm</td><td>-none-</td><td>-eabi-gdl</td><td>o (09/03.</td><td>/25 1</td><td>7:31)</td><td></td><td></td><td></td></terminate<>	d, exit val	ue: O>arm	-none-	-eabi-gdl	o (09/03.	/25 1	7:31)			
	ig - S idit Dug gdb gdb	ig - Source not dit <u>N</u> avigate ug A OpenOCD [Pro C¥Program gdb [Zylin Em o© <terminate< td=""><td>ig - Source not found E dit <u>N</u>avigate Search bug X OpenOCD [Program] O¥Program Files¥Op gdb [Zylin Embedded de S <terminated>Embedded</terminated></td><td>g - Source not found Eclipse Pla dit <u>Navigate</u> Search <u>Project</u> <u>Project</u> <u>Navigate</u> Search <u>Project</u> <u>Project</u> <u>Navigate</u> Search <u>Project</u> <u>OpenOCD [Program]</u> <u>OpenOCD [PropenOCD [Program]</u> <u>OpenOCD [Program]</u> <u>OpenOCD [Prop</u></td><td>g - Source not found Eclipse Platform dit <u>Navigate</u> Search <u>Project</u> <u>R</u>un B ■ ■ ■ ■ ★ • • • • • • • • • • • oug X • • • • • • • • • • OpenOCD [Program] • • • • • • • • • • • • • • • • • • •</td><td>g - Source not found Eclipse Platform dit Navigate Search Project Run Window R ■ I IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII</td><td>g - Source not found Eclipse Platform dit Navigate Search Project Run Window Help R ■ R I IIIIIIIIIIIIIIIIIIIIIIIIIIIIIII</td><td>g - Source not found Eclipse Platform dit Navigate Search Project Run Window Help B ■ ■ ■ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓</td><td>g - Source not found Eclipse Platform dit Navigate Search Project Run Window Help B ■ B   B   B + O + O + O + O + I = A + I = A + I = A oug S - I = II (III + A + III) OpenOCD [Program] OpenOCD [Program] OPenOCD</td><td>g - Source not found Eclipse Platform dit Navigate Search Project Run Window Help B ■ ■ ■ ■ ↓ ↓ • ● • ● • ● • ■ ↓ ⊕ ↓ • ↓ ↓ • ↓ • • • • • • • • • • • •</td><td>g - Source not found Eclipse Platform dit Navigate Search Project Run Window Help B ■ ■ ■ ■ ↓ ↓ • ● • ● • ● • ● • ● • ● • ● • ● • ●</td></terminate<>	ig - Source not found E dit <u>N</u> avigate Search bug X OpenOCD [Program] O¥Program Files¥Op gdb [Zylin Embedded de S <terminated>Embedded</terminated>	g - Source not found Eclipse Pla dit <u>Navigate</u> Search <u>Project</u> <u>Project</u> <u>Navigate</u> Search <u>Project</u> <u>Project</u> <u>Navigate</u> Search <u>Project</u> <u>OpenOCD [Program]</u> <u>OpenOCD [PropenOCD [Program]</u> <u>OpenOCD [Program]</u> <u>OpenOCD [Prop</u>	g - Source not found Eclipse Platform dit <u>Navigate</u> Search <u>Project</u> <u>R</u> un B ■ ■ ■ ■ ★ • • • • • • • • • • • oug X • • • • • • • • • • OpenOCD [Program] • • • • • • • • • • • • • • • • • • •	g - Source not found Eclipse Platform dit Navigate Search Project Run Window R ■ I IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	g - Source not found Eclipse Platform dit Navigate Search Project Run Window Help R ■ R I IIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	g - Source not found Eclipse Platform dit Navigate Search Project Run Window Help B ■ ■ ■ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	g - Source not found Eclipse Platform dit Navigate Search Project Run Window Help B ■ B   B   B + O + O + O + O + I = A + I = A + I = A oug S - I = II (III + A + III) OpenOCD [Program] OpenOCD	g - Source not found Eclipse Platform dit Navigate Search Project Run Window Help B ■ ■ ■ ■ ↓ ↓ • ● • ● • ● • ■ ↓ ⊕ ↓ • ↓ ↓ • ↓ • • • • • • • • • • • •	g - Source not found Eclipse Platform dit Navigate Search Project Run Window Help B ■ ■ ■ ■ ↓ ↓ • ● • ● • ● • ● • ● • ● • ● • ● • ●

3) 電源停止

ターゲットの電源を停止

4) OpenJTAG をターゲットから取り外す

5)上記が面倒であれば Eclipse を終了しターゲットの電源停止、open-JTAG を取り外しでも OK です。