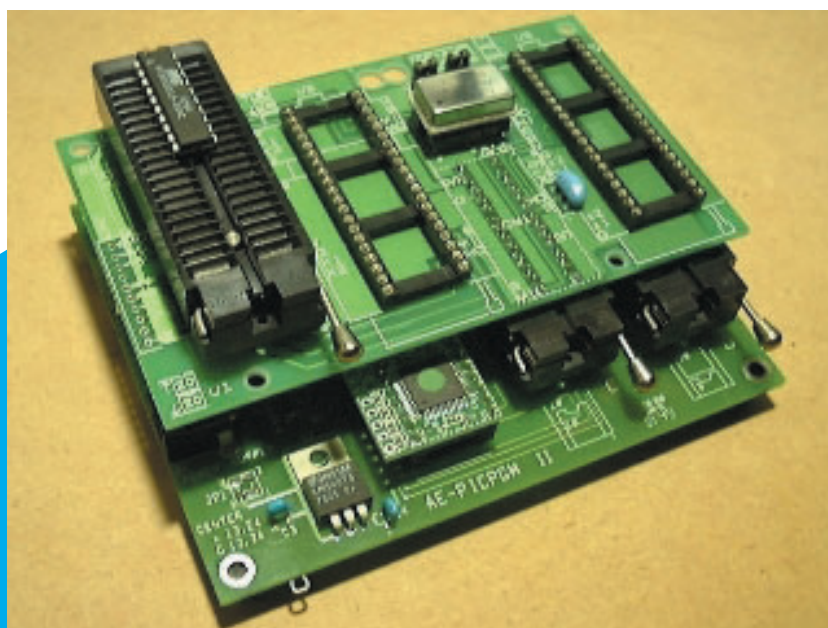


# PICプログラマーVer.4用 AVRアタッチメントボード・キット

PICプログラマーVer.4にこのアタッチメントボードを差し込むだけで、ATMEL AVRプログラマーに変身します。

対応マイコン：AT90Sxxxシリーズ、ATtiny12L、15L、26L（2004.3月現在）

テスト済みOS：Windows98SE、ME、2000、XP



# PICプログラマー用 AVRアタッチメントボードキット

型番:K-00560

秋月オリジナルPICプログラマーVer4 (K-00038 & K-00200) をお持ちの方に朗報！！  
PICプログラマーに、このアタッチメントボードをセットするだけで、PICプログラマーが  
AVRプログラマーに変身します。  
ご注意：このキットを使用するには、秋月オリジナルPICプログラマーVer4が必要です。

for WindowsXP/ME/2000/98SE

## ■購入の前に、ご確認ください■

- ★ 秋月オリジナルPICプログラマーVer4が必要です。Ver3.Xファームウェアでは動作いたしません。
- ★ AT90S1200Aを外部クロック動作に、また、AT90S1200を内部クロックモードに変更することはできません。
- ★ プログラミング方法は、シリアルプログラミング方式を採用しています。

## ■サポートマイコン■ (2004.03月現在)

Tiny12L, Tiny15L, Tiny26L

AT90S2323, AT90S2343

AT90S1200, AT90S200A, AT90S2313

AT90S4433

AT90S4414, AT90S4434, AT90S8515, AT90S8535

## ■セット内容・部品表■

部品番号	数	部品名	備考 表示等
XT1	1個	発振器	4MHz (DIP8ピンまたは、DIP14ピンタイプ)
XT2	1個	セラミック発振子	4MHz
CN1	2本	ピンヘッダー	14ピン×1列 (40ピン×1列が同封されている場合は切断して使用します)
JP1, JP2	2本	ピンヘッダー	2ピン×1列 (40ピン×1列が同封されている場合は切断して使用します)
U1, U2, U3	3個	40ピン ICソケット	
	2個	ジャンパーソケット	JP1, JP2用
	1個	14ピン ICソケット	XT1取付け用
	1個	40ピン ゼロプレッシャーソケット	U1, U2, U3用
	1枚	AE-AVR/PIC基板	
	1枚	CD-R	Windows用AVRプログラマ専用ソフト

## ■AVRアタッチメントボード回路図■

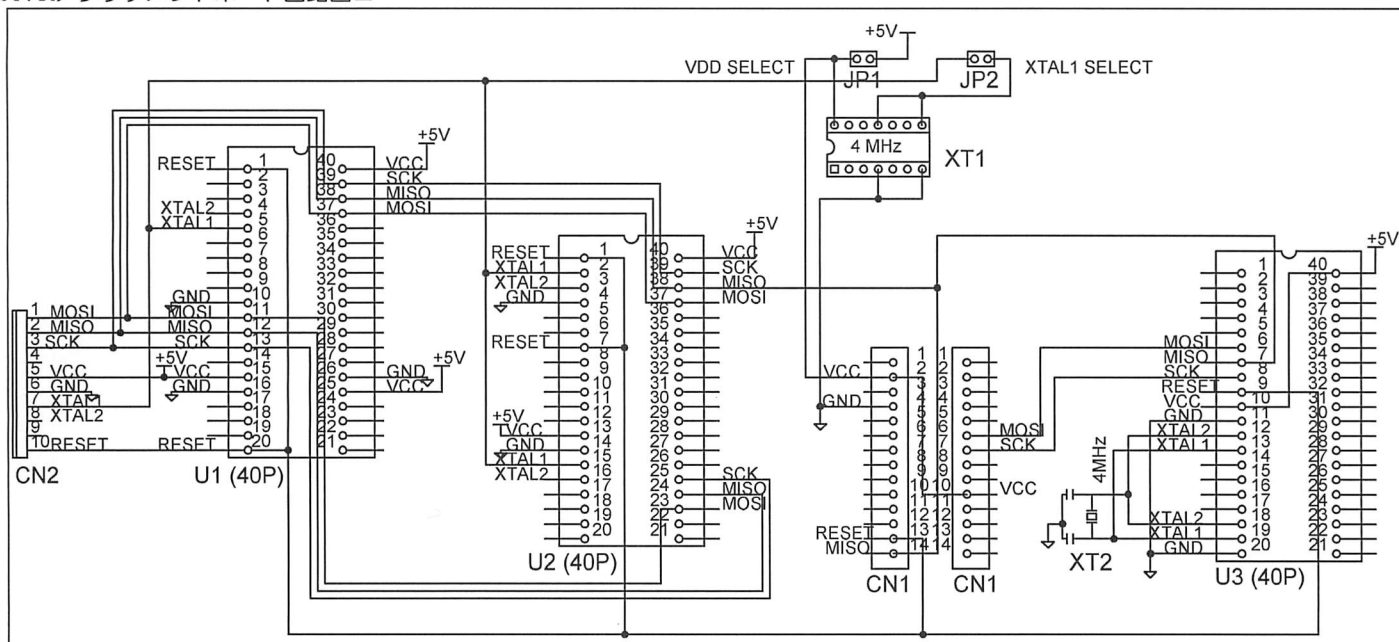


図 AVR アタッチメントボード回路 -1

## ■製作■

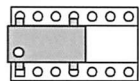
部品は、予告なく相当品・互換品に変更になることがあります。メーカーにより若干型番が異なることがあります。製作前に部品表と照らし合わせてご確認ください。部品表、回路図、部品配置図を参考にご覧ください。

特に難しい半田箇所はありませんので、一つ一つ確実に半田付けしてください。

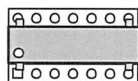
- 極性がある部品 発振器
- 極性がない部品 セラミック発振子

1. 部品面から部品を取付けます。
2. 40ピン、ICソケットをU1、U2、U3に取付けます。
3. 14ピン、ICソケットをXT1に取付けます。

8ピンタイプの発振器の  
取付け方法。



14ピンタイプの発振器の  
取付け方法。



★基板のシルク図のXT1は、誤りですのでご注意ください。★

4. ジャンパーピン、JP1,JP2を取付けます。  
通常は、両方のジャンパーピンをショートした状態で使用します。
5. セラミック発振子、XT2に取付けます。

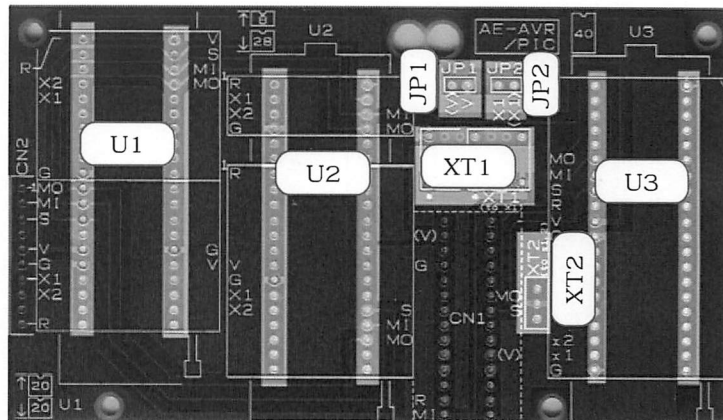
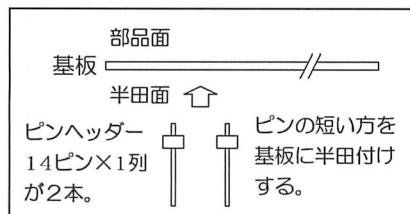


図 部品配置図 -1 部品面

6. 半田面の部品の取り付け
7. 14ピン1列ピンヘッダーを、CN1に、半田面に取付けます。  
取付け箇所は、2箇所あります。



8. 40ピン、ゼロプレッシャーソケットは、必要に応じてU1、U2、U3のいずれかのソケットに挿入します。

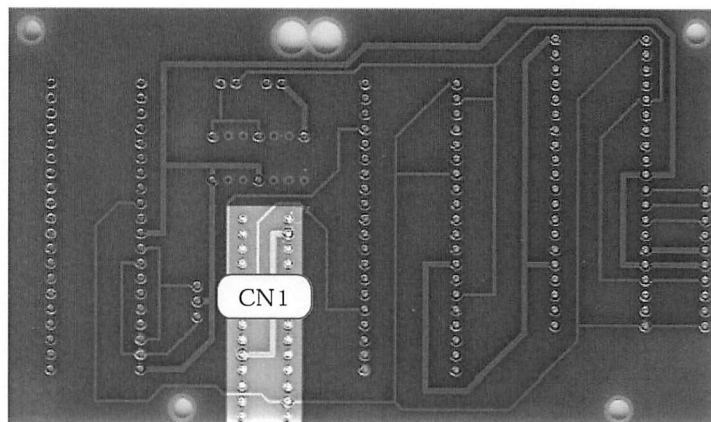
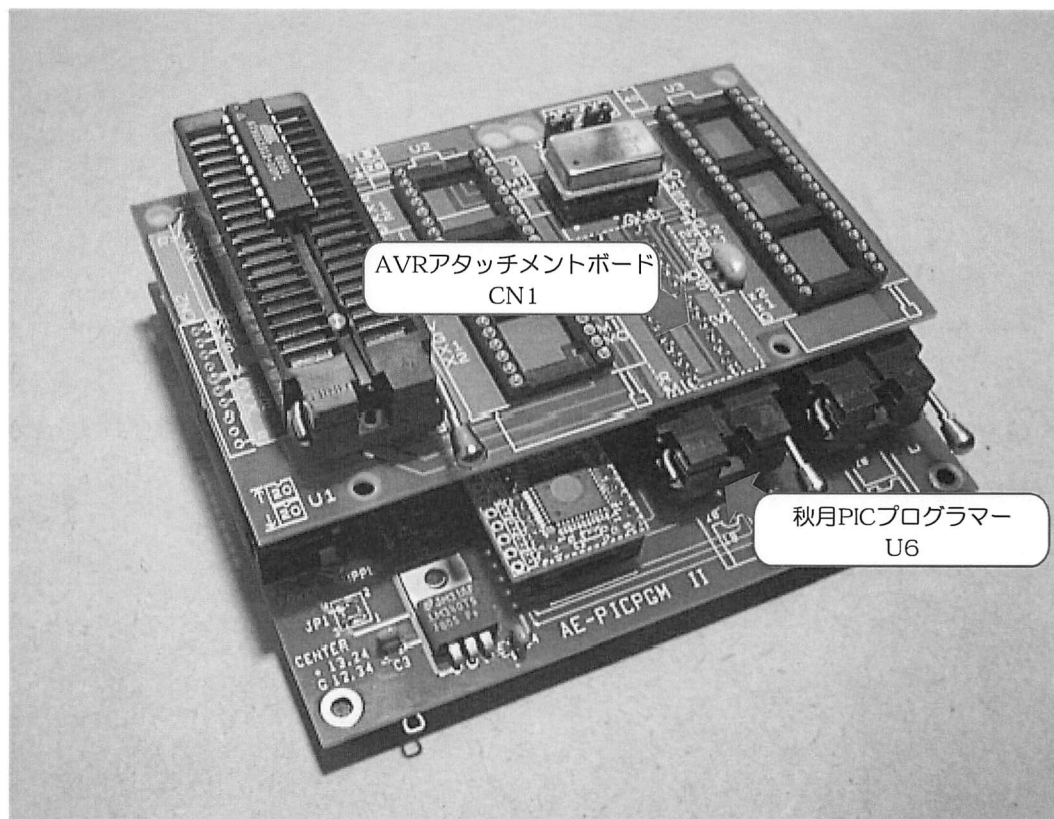


図 部品配置図 -2 半田面

■秋月PICプログラマーとAVRアタッチメントボードの接続■

秋月PICプログラマーの、28ピンゼロプレッシャーソケット-U6に、AVRアタッチメントボードの、CN1挿入し、ゼロプレッシャーソケットのレバーを倒して接続します。



秋月PICプログラマーとAVRアタッチメントボードの接続

■実際に書き込んで動作させてみよう■

ライターハードのチェックを兼ねてAT90S2313に簡単なサンプルプログラムを書き込んでみます。ナイトライター風にLEDが光るプログラムです。ATMELのAVRマイコンはフラッシュマイコンなので何度でも書替えができ便利です。

CD-Rの内容



1. ソフトの準備

AVR Studioを使い、サンプルのledtest.asmから、書き込み用のHEXファイルledtest.hexをつくります。AVR Studioは、ATMEL社がフリーで供給しているAVR開発ツールで、エディター、アセンブラ、シミュレータを統合したソフトです。ATMEL社純正の開発ハードウェアは、このAVR Studioから操作できるようになっていますが、弊社のプログラマキットはこの統合環境上から操作する構成にはなっていません。そのため、書き込みのみ専用ソフトavrpgm6.exeを使用します。

1-1. ソフトのインストールと、サンプルプログラムの準備

- ① CD-Rのavrstudioフォルダー内のsetup.exeをダブルクリックし、AVR Studioをインストールします。また、CD-Rのwin内のsetup.exeをダブルクリックし、専用書き込みソフトavrpgm6をインストールします。
- ②ルートドライブ（ここでは、Cドライブを使用します。）にC:\¥AVRを作成し、ここにCD-Rのsampleフォルダー内のledtest.asmと、2313def.incをコピーしてください。

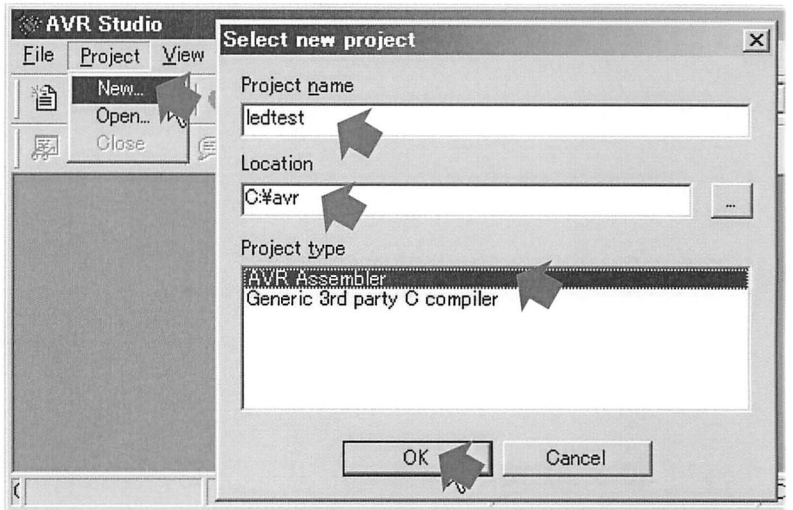


図 AVR Studio -1 新規プロジェクトの設定

1-2. AVR Studioの操作とHEXファイルの作成

- ①Windows画面左下の【スタート】ボタンから『プログラム』の『AT MEL AVR TOOLS』の『AVR Studio 3.56』をクリックし AVR Studioを起動してください。
- ②メニューバー『Project』の『New』で、「図 AVR Studio -1 新規プロジェクトの設定」のように設定します。
- ③メニューバー『Project』の『Project Settings...』で、「図 AVR Studio -2 プロジェクトにファイルを追加」のように設定します。

- ④「図 AVR Studio -3 追加ファイルの確認」のように、Assembler Filesの下に、ledtest.asmが追加されていることを確認してください。
- ⑤「図 AVR Studio -4 プロジェクト設定」のように” Project Settings...”をクリックすると、[AVR Assembler Options]ウィンドウが現れますので、各項目の設定を行います。

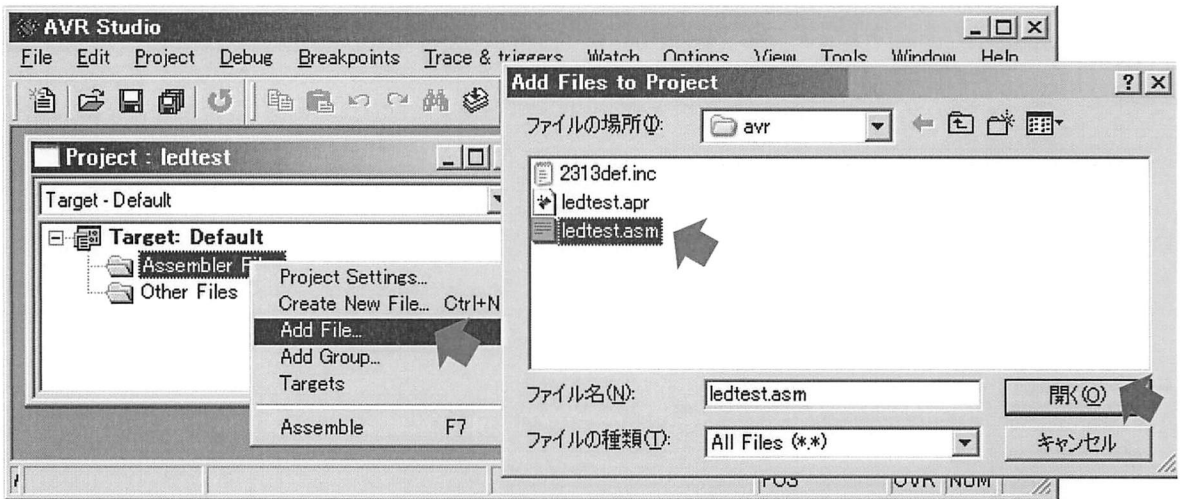


図 AVR Studio -2 プロジェクトにファイルを追加

- ⑥「図 AVR Studio -5 ビルドと実行」のように、” Build and run ”をクリックすると、[Simulator Options]ウィンドウが現れますので、各項目の設定を行います。このウィンドウは、新たらしいプロジェクトを作成した時のみ現れます。
- ⑦「図 AVR Studio -6 フォントの変更」のように、” Fonts...” をクリックするとソースコードが表示されているウィンドウのフォントを日本語に変更できます。

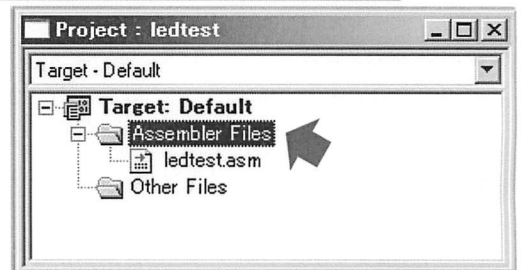


図 AVR Studio -3 追加ファイルの確認

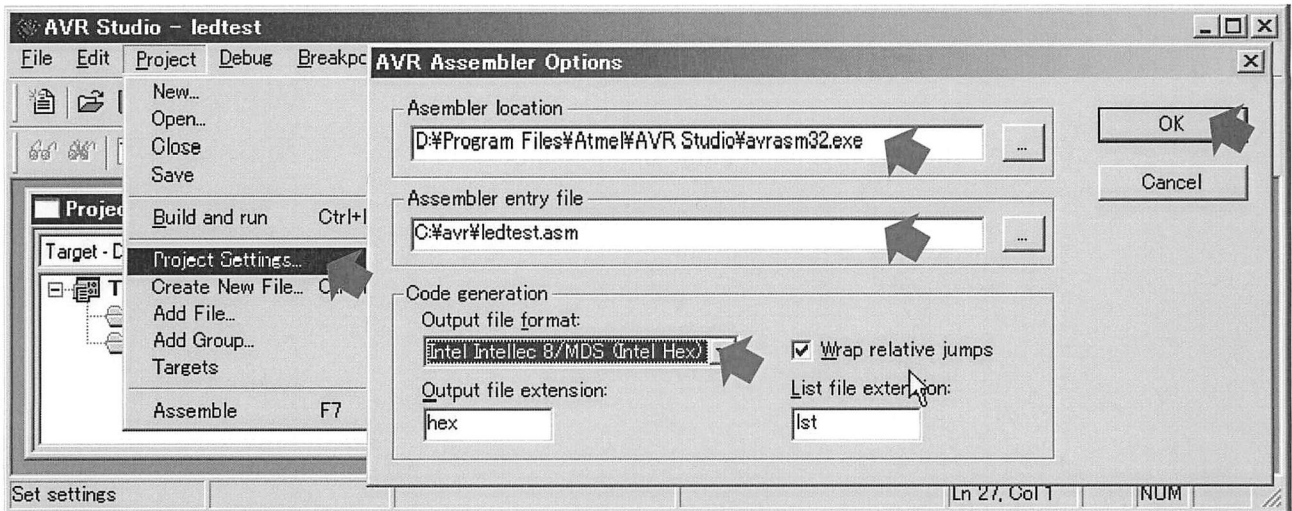


図 AVR Studio -4 プロジェクト設定

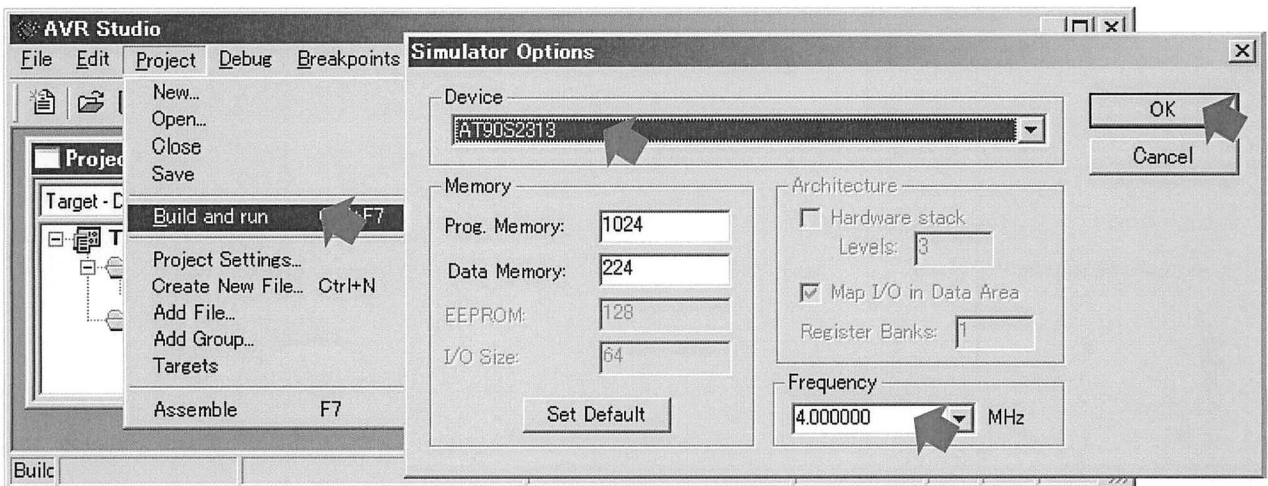


図 AVR Studio -5 ビルドと実行

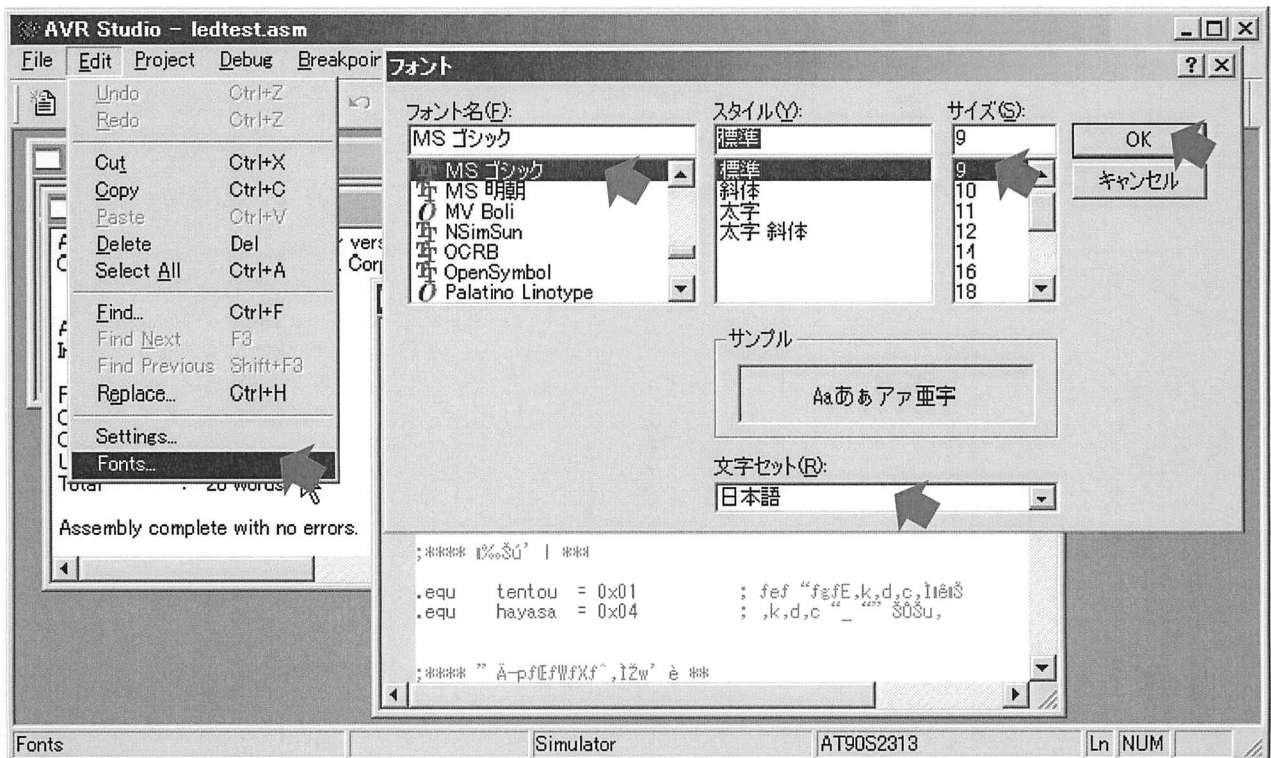


図 AVR Studio -6 フォントの変更

2. 付属のシミュレータで動作を確認してください。

「図 AVR Studio -7 シミュレーション」で、[ledtest.asm]ウインドのラベルmain: ldi count,hayasaの行をブレークポイントに設定します。ブレークポイントは、設定したい行にマウスの矢印を移動させ、マウスの右ボタンをクリックすると、プルダウンメニューが現れますので、「Toggle Breakpoint」を選ぶことで設定することができます。メニューボタンの「RUN」をクリックすることにより、[New I/O View (IO:1 Standard)]ウインドウの、「PoerB Data」項目がのチェック印が左シフトすることを確認できます。

[ledtest.asm]ウインドウは、エディタ画面となっているので、この画面を使用し、ソフトの修正ができます。ソースコードの修正後、再びアセンブルを行うには、メニューバーから、「Project」の「Build and Run」を指定するか、メニューバーの[Build and Run]ボタンを押します。このとき、ledtest.asmが存在するフォルダーに、ledtest.hex(マイコンへプログラムを書込むためのヘキサファイル)が生成されます。

シミュレータの詳しい操作については、AVR Studioに添付されているヘルプファイルをご覧ください。

```

;***** 割り込みのイニシャライズ *****
; サンプルム ledtest.asm
; cpu AT90S2313
; タイマー割り込みを使用しそれを4回数える毎に点灯の
; 位置をずらし、ナイトライダー風にする。
;*****
;**** インクルードファイルの指定 ****
include "2313def.inc"

;**** 初期値 ****
equ tentou = 0x01 ; テントウLEDの場所
equ hayasa = 0x04 ; LED点灯間隔の値

;**** 汎用レジスタの指定 ****
def count = r16 ; 点等間隔時間のカウンタ用
def led1 = r17 ; ポートbへの出力データ格納用
def temp = r19 ; 一時使用用
def stack = r20 ; 割込時のデータレジスタ待避場所

;****リセットベクトルと割り込みベクトルの指定 ****
ORG 0x00
rjmp RESET ; Reset handle
rjmp RESET ; INTO
rjmp RESET ; INT1
rjmp RESET ; TIMER1_CAPT1
rjmp RESET ; TIMER1_COMP1
rjmp RESET ; TIMER1_OVF1
rjmp TIMO_OVF ; Timer 0 overflow handle
rjmp RESET ; UART_RX
rjmp RESET ; UART_UDRE
rjmp RESET ; UART_TX
rjmp RESET ; ANA_COMP

RESET:
;**** PORTのイニシャライズ ****
; ポートBを出力に設定
ldi temp,0xFF
out DDRB,temp

; ポートDを出力に設定
ldi temp,0xFF
out DDRD,temp

;**** 割り込みのイニシャライズ ****
; タイマーオーバーフロー割り込みを指定
ldi temp,(1<<TOIE0)
out TIMSK,temp

; 内部クロック 1/256を指定
ldi temp,(1<<CS02)
out TCCR0,temp

;**** 初期値の格納 ****
; LED 1を点灯
ldi led1,tentou
out PORTB,led1

;**** MAINプログラム ****
main: ldi count,hayasa ;点減間隔を指定
sei ;割り込み許可

kakininn:
cpi count,$00 ;時間がくるまで待つ
brne kakininn

clc ;キャリーをクリア
rol led1 ;点灯箇所を左シフト
brcc ledout ;8番目まで行ったら1番目に戻る
ldi led1,tentou

ledout: out PORTB,led1 ;LED 1に出力する。

rjmp main

;*****
;* タイマー割り込み TIMO_OVF
;*****
TIMO_OVF: ; (Timer 0 overflow handle)
in stack,SREG ; SREGレジスタを待避する
dec count ;点減レジスタを減算
out SREG,stack ; SREGレジスタを復帰する。
reti ; 割り込みから戻る

;**** End of File ****

```

プログラムリスト - ledtest.asm



図 AVR Studio -7 シミュレーション

### 3. 書き込み

プログラマーキットを使いAT90S2313にサンプルソフトを書込みます。

専用書き込みソフトavrp6.exeのインストールを行っていないときは、CD-Rのwin内のsetup.exeをダブルクリックしインストールしてください。

①AT90S2313を1個、用意します。（まだAT90S2313はソケットに挿入しないでください。）

②秋月PICプログラマーボードに、AVRアタッチメントボードを取付けます。

AVRアタッチメントボード上の2箇所のジャンパー(JP1,JP2)はショートしてください。

40ピンゼロインプレッシャーソケットはU1の位置に取り付けます。

秋月PICプログラマーボードをパソコンのRS232ポートに接続し、電源を入ます。（供給電源電圧は1.5V以上が必要です。）

③Windows画面左下の【スタート】ボタンから『プログラム』の『Akizuki』の『AVR Programmer V6』をクリックしavrp6.exeを起動すると「図 avrp6 -2 メイン画面」のような画面が現れます。

④メイン画面の【デバイス選択】でAT90S2313を選びます。

選択後、AT90S2313を「図 avrp6-1 マイコンの挿入位置」で指定されている部分に挿入してください。

（今回の場合は、U1にゼロインプレッシャーソケットを取り付け、そこにAT90S2313を上揃えて挿入します。）

⑤『ファイル』の『開く』または、[開く]ボタンで、前の章で作成したledtest.hexを選んでください。

ledtest.hexは、ledtest.asmと同じフォルダーに生成されています。今回の場合は、C:\avr\ledtest.hexとなります。

⑥[プログラム]ボタンを押し、書き込みます。秋月PICプログラマーボード上のBUSY LED(D2)が点滅し、書き込みが行われます。

メイン画面の一番下の部分に、[プログラミング正常終了]のメッセージが表示されると書き込み成功です。

#### ★注意★

BUSY LED(D2)が光っている間は、プログラマーが稼働している事を示しています。その間はマイコンの抜き差しを行わないでください。

ISPコネクタを使用した書き込みは、動作保障していませんが、もし、使用される場合は、JP1とJP2を必要に応じてオープンにしてください。

JP1をオープンで使用するときは：

ターゲットボードで側で、マイコンに電源を供給し書き込む場合。

JP2をオープンで使用するときは：

ターゲットボード側で、マイコンに発振子を取付けて書き込む場合。

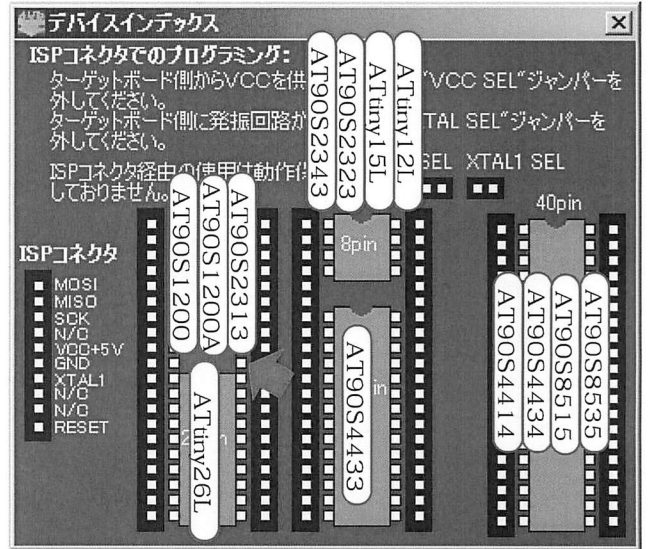


図 avrp6-1 マイコンの挿入位置

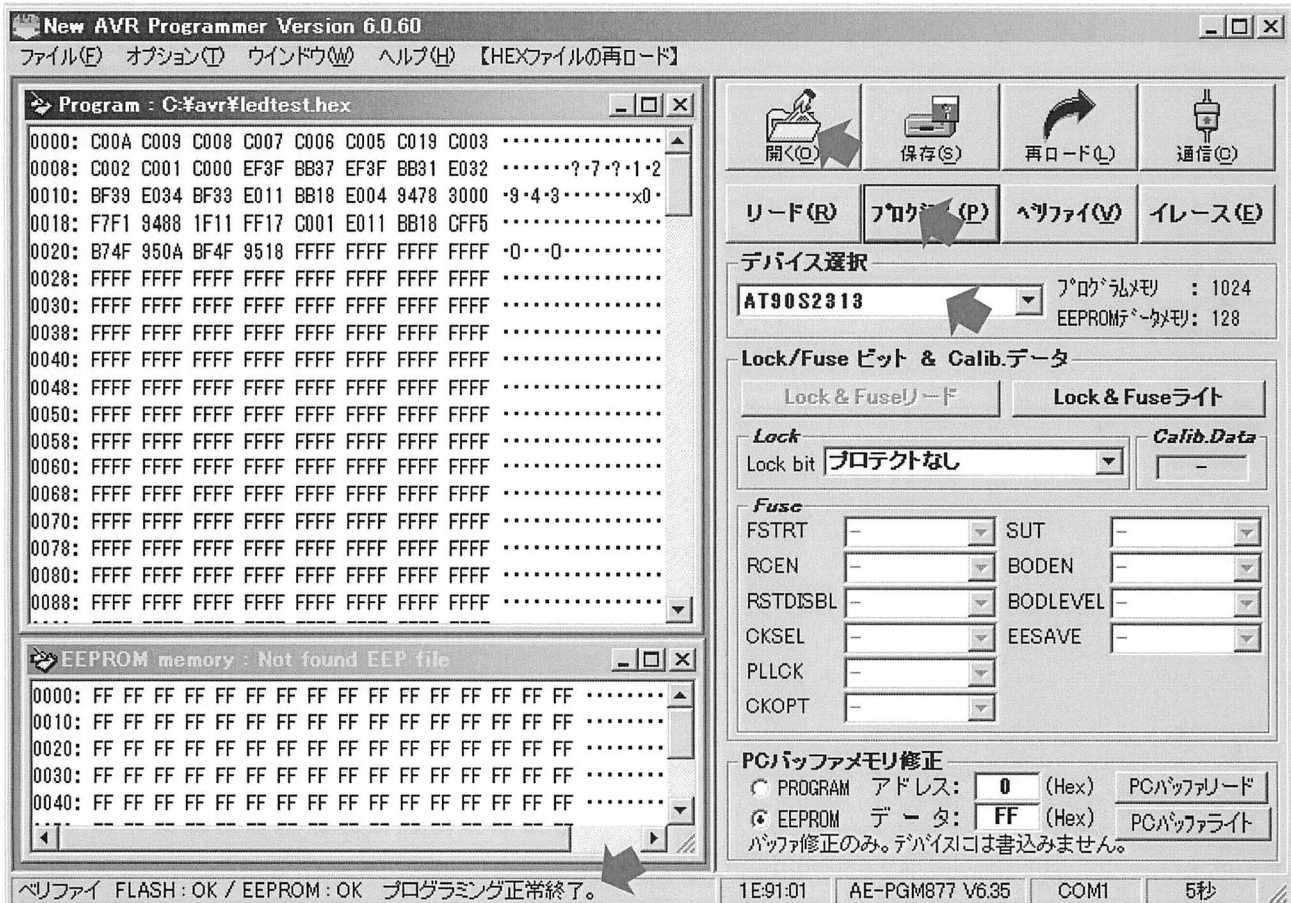


図 avrp6 -2 メイン画面

### 3-1. avrpgm6.exe画面の説明

#### メニューバーの説明

**ファイル** コード/データファイルを開く — [開く]ボタンと同じ動作。

このコマンドは、hexとeepの両方のファイルの内容を、PCバッファメモリに読み込みます。

アセンブル後に生成されるHEXファイルを、PCバッファメモリに読み込みます。拡張子がhexのファイルを、ここでは仮にコードファイルと呼んでおりマイコンのプログラムメモリに書込むデータであると認識します。また、拡張子が、eepのファイルを、ここでは仮にデータファイルと呼んでおり、マイコンのEEPROMメモリに書込むデータであると認識します。EEPROMメモリの初期データが、ソースコード内で指定されているとき、アセンブルするとeepファイルが生成されます。

**コードファイルを開く**

このコマンドは、hexファイルの内容のみ、PCバッファメモリに読み込みます。

**データファイルを開く**

このコマンドは、eepファイルの内容のみ、PCバッファメモリに読み込みます。

**上書き保存**

PCバッファメモリに読み込まれているhexおよびeepファイルを、読み込んだときと同じ名前で上書き保存します。

avrpgm6.exeには、PCバッファメモリを修正する機能があります。ファイルから読み込んだデータを修正し、保存するときや、マイコンから読み出したデータを保存するときに、このコマンドを使用します。

**コード/データファイルを別名で保存** — [保存]ボタンと同じ動作。

PCバッファメモリに読み込まれているhexおよびeepデータを、別名でファイルに保存します。

ファイルから読み込んだデータを修正し、保存するときや、マイコンから読み出したデータを保存するときに、このコマンドを使用します。

**コードファイルを別名で保存**

PCバッファメモリに読み込まれているhexデータを、別名でファイルに保存します。

**コードファイルを別名で保存**

PCバッファメモリに読み込まれているeepデータを、別名でファイルに保存します。

**終了**

avrpgm6.exeを終了します。

**オプション** COMポート選択 — [通信]ボタンと同じ動作。

PCと通信するための通信ポート番号を指定します。

**PCバッファのクリア**

PCバッファメモリの内容を、"1"でクリアします。

(マイコン内部のメモリを消去すると、メモリのデータは"1"となります。)

**ウインドウ** デバイスインデックスウインドウを開く

一度閉じてしまった、デバイスインデックスウインドウを、再び開きます。

**プログラムメモリウインドウを開く**

一度閉じてしまった、プログラムメモリウインドウを、再び開きます。

**EEPROMデータメモリウインドウを開く**

一度閉じてしまった、EEPROMデータウインドウを、再び開きます

**ヘルプ**

**デバイス挿入位置について**

[デバイスインデックスウインドウを開く]と同じ動作をします。

**ファームウェアのアップデート**

新しいマイコンがサポートされる時、avrpgm6.exeおよび、ファームウェアが、<http://akizukidenshi.com>に公開されます。ファームウェアマイコンは、フラッシュメモリタイプなので、内部のソフトウェアを更新することができます。この更新作業を行うとき、このコマンドを使用します。

**ヘルプ**

このファイルが開きます。

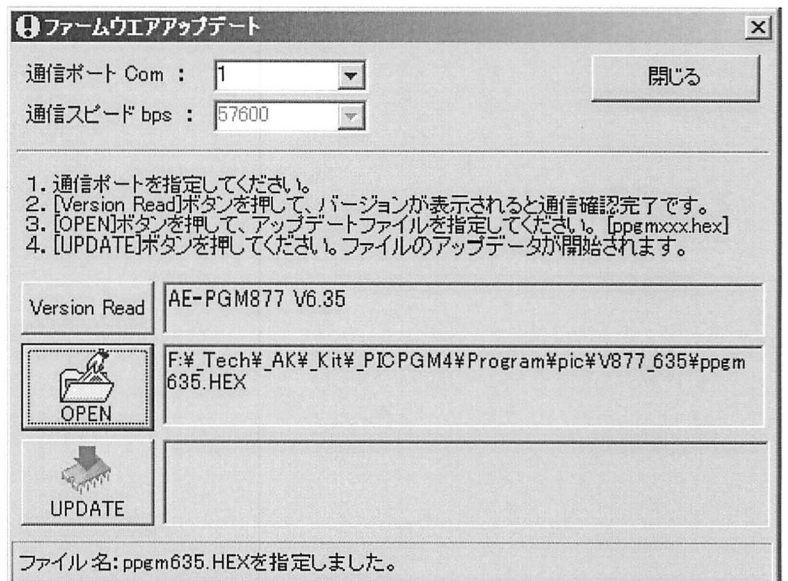
**バージョン情報**

avrpgm6.exeのバージョンを表示します。

**【HEXファイルの再ロード】**

ソースコードの修正を行い、もう一度同じファイル名のHEXを、PCバッファメモリに読み込むとき、[開く]ボタンを使用しなくても、【HEXファイルの再ロード】を実行すると、一回でHEXファイルを読み込むことができます。

```
***** EEPROM initial data の設定例 *****
.ESEG
.DB 0x01, 0x02, 0x03, 0x04
```





[リード]ボタン マイコン内部に書込まれているデータ  
読み出します。

リード(R)

[プログラム]ボタン マイコンにコードを書込みます。

プログラム(P)

[バリファイ]ボタン PCバッファメモリの内容と、マイコン  
に書込まれているデータの比較しま  
す。

バリファイ(V)

[イレース]ボタン マイコンに書込まれているデータ  
を消去します。消去後に読み出したデータ  
は、ALL"1"となります。

イレース(E)

#### デバイス選択ウィンドウ

使用するマイコンの種類を選びます。全ての操作は、使用するマイコンを選択する事から始まります。

デバイス選択

AT90S2313

プログラムメモリ : 1024  
EEPROMデータメモリ : 128

#### PCバッファメモリ修正ウィンドウ

ファイルから読み込んだデータや、マイコンから読み出したデータを修正するときに使用します。

PCバッファメモリ修正

PROGRAM アドレス: 0 (Hex) PCバッファリード

EEPROM データ: FF (Hex) PCバッファライト

バッファ修正のみ。デバイスには書き込みません。

#### ステータスバー (メイン画面の左下)

操作を行った結果がどのようなようになったかを表示します。書き込みが、正常終了したかどうかは、このステータスバーに表示されます。

バリファイ FLASH: OK / EEPROM: OK プログラミング正常終了。

[開く]ボタン



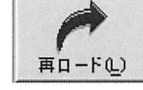
hex、eepファイルをPCバッファメモリに読み込みます。メニューバーの『ファイル』-『コード/データファイルを開く』と同じ機能です。

[保存]ボタン



PCバッファメモリの内容をhex、eepファイルに名前をつけて保存します。メニューバーの『ファイル』-『コード/データファイルを別名で保存』と同じ機能です。

[再ロード]ボタン



hex、eepファイルをPCバッファメモリに読み込みます。最初に、[開く]ボタンを使用し、ファイル名をavrpgm6.exeに記憶させる必要があります。メニューバーの【HEXファイルの再ロード】と同じ機能です。

[通信]ボタン



PCと通信するための通信ポート番号を選びます。メニューバーの『オプション』-『COMポート選択』と同じ機能です。

#### Lock/Fuseビット & Calib.データウィンドウ

Lock/Fuseビットの情報をマイコンから読み出すときや、書き込むときに使用します。マイコンにコードを書込んだ後に、このウィンドウの操作を行います。使用するマイコンの種類によって、Fuseビットは設定する内容が異なりますのでご注意ください。

Lock/Fuseビット & Calib.データ

Lock & Fuseリード Lock & Fuseライト

Lock bit: プロテクトなし

Calib.Data: -

Fuse

FSTRT	-	SUT	-
RCEN	-	BODEN	-
RSTDISBL	-	BODLEVEL	-
CKSEL	-	EESAVE	-
PLLCK	-		
CKOPT	-		

#### 4. 実際に動かしてみる

ledtest.hexを書き込んだAT90S2313を使って、次の回路を作り、電源を入るとLEDが順に灯します。

AT90S2313、LED8個、抵抗1本、セラミック1個だけの簡単な回路です。右図の回路図では、クリスタルを使用していますが、セラミックで代用することができます。

このマイコンの動作電圧範囲は、2.7~6.0Vなので、電池2本(3V)で動作し、レギュレータも不要です。

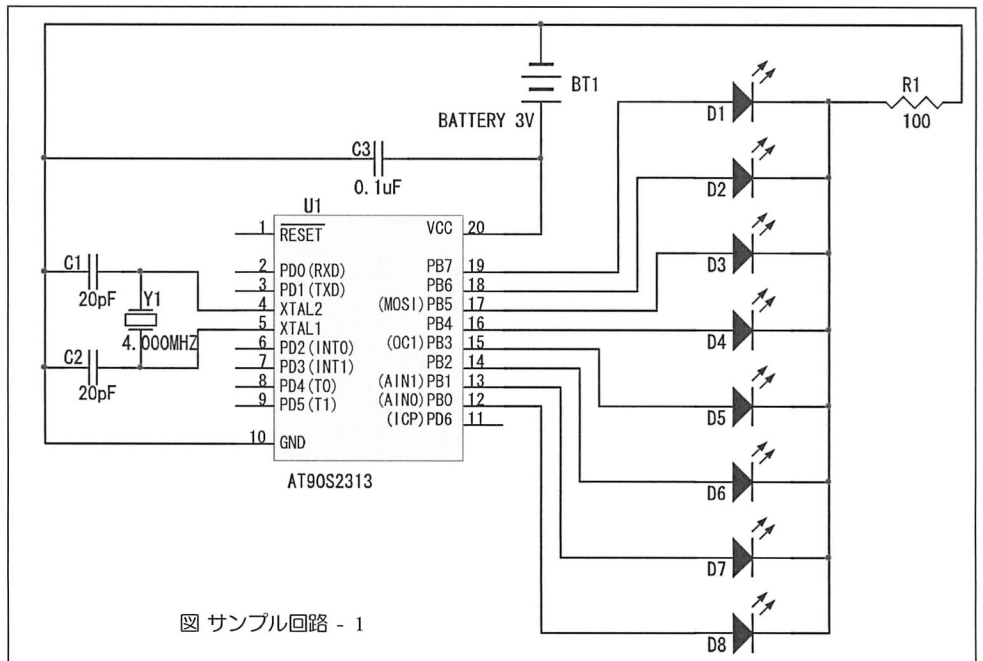


図 サンプル回路 - 1

#### AVRマイコン用アタッチメントキット

お問い合わせは、往復はがき または、返信用切手同封の封書にてお願いいたします。

電話、FAXでのお問い合わせは、受け付けておりません。

〒158-0095 東京都世田谷区瀬田5-35-6 (株)秋月電子通商 質問係

2004/03/10