[第1章]

AVRマイコンについて

● 概要

AVRマイコンはアトメル社が製造している8ビットRISCマイコンのことで，ATmegaおよびATtinyの名称が付けられています(旧型式および特殊なものを除く)．

ATmegaは多ピンで多プログラム容量であるのに対し，ATtinyは少ピンで少プログラム容量です(**写真1-1**)．

形状の違いや，ピン数，メモリ容量，機能数の大小からいくつもの型式がありますが，基本的な内部構造やアセンブラ命令はほとんど共通で，一つの型式を使えるようになれば，ほかのAVRマイコンも容易に理解できるようになります．

AVRマイコンの全型式および仕様については，アトメル社のホームページ(http://www.atmel.com)から知ることが可能です．

本書では安価で比較的入手しやすく，また工作に利用しやすい28ピンDIP形状のATmega88を使用し説明を行っています．

この1種類があれば工作において困ることはまずないと思いますが，もしATmega88が入手困難な場合はデータシートで相違点を確認してほかの型式を使用してください．

なお，ATmega88と機能的には同じでメモリ容量だけ異なるデバイスとしてATmega48およびATmega168がありますが，使用にあたっては，いくつか注意する点があります．詳しくは「1-5 ATmega48/88/168の相違点」を参照してください．**表1-1**に各デバイスの仕様比較を示します．

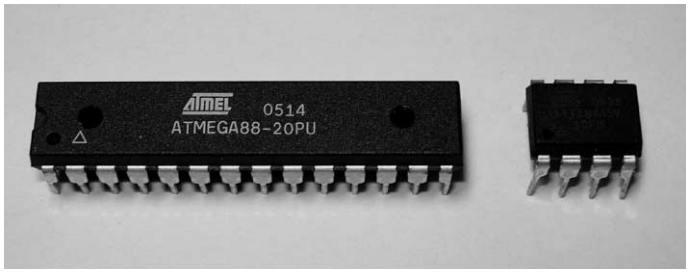

写真1-1　AVRマイコンの外観(左：ATmega88 28ピンDIP，右：ATtiny45 8ピンDIP)

表1-1　ATmega48 / ATmega88 / ATmega168(28 ピンDIP)**の仕様**

項　目		ATmega48	ATmega88	ATmega168	備　考
FLASH(K バイト)	フラッシュ・メモリ・サイズ	4	8	16	プログラム領域
SRAM(K バイト)	SRAM メモリ・サイズ	0.5	1	←	データ領域
EEPROM(K バイト)	EEPROM メモリ・サイズ	0.25	0.5	←	EEPROM 領域
Max I/O ピン	入出力端子数	23	←	←	PORTB(8) + PORTC(7) + PORTD(8)
F.max(MHz)	最大動作クロック	20	←	←	ATmega 48V/88V/168V は 10 MHz
V_{CC}(V)	電源電圧	2.7 ～ 5.5	←	←	ATmega 48V/88V/168V は 1.8 ～ 5.5V
10-bit A-D Channels	A-D 変換(10 ビット)数	8	←	←	アナログ・コンパレータを含む
Analog Comparator	アナログ・コンパレータ	Yes	←	←	
16-bit Timer	16 ビット・タイマ数	1	←	←	
8-bit Timer	8 ビット・タイマ数	2	←	←	
Brown Out Detector	電圧低下検出	Yes	←	←	
Ext Interrupts	外部割り込み数	26	←	←	ピン変化割り込みおよび RESET を含む
Interrupts	割り込み数	26	←	←	割り込みベクタ参照
On Chip Oscillator	内部オシレータ	Yes	←	←	
SPI		1 + USART	←	←	
TWI		Yes	←	←	
UART		1	←	←	
Watchdog	ウォッチドッグ・タイマ	Yes	←	←	

1-1　ピンの割り当て

AVRマイコンに限らず，昨今のマイコンは，一つのピンがいくつもの機能をもつマルチファンクションになっています．**図1-1**にピン配置を示します．( )内はマルチファンクションの機能です．マイコン起動時，各ポートの初期状態はプルアップなし入力となっています(4-5　ピンの初期設定 参照)．

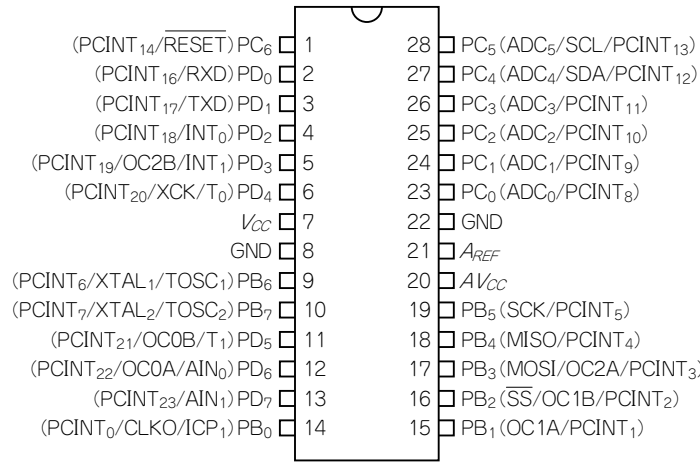

図1-1　ATmega48 / ATmega88 / ATmega168(28 ピンDIP)**のピン配置**

[第2章]

開発環境について

2-1　必要機材

AVR Studio

AVR Studioとは，アトメル社のWebサイトから無償でダウンロードできるAVRマイコンのプログラミング・ツールです．ダウンロードおよびインストール方法については「2-2　プログラム環境の準備」を参照してください．

ライタ

作成したプログラムをマイコンに書き込むための装置です．

ネットで検索するとライタの自作情報などが紹介されていますが，パソコン側のインターフェースがDsub9ピン・コネクタ（RS-232Cのシリアル）だったりするので，本書では，現在のパソコンの主流であるUSBコネクタ対応アトメル社純正のAVRISP-mkIIを使用します（**写真2-1**）．

入手先は下記「マイコン」のところを参照してください．

マイコン

本書で使用するマイコンはATmega88という型式で，定格電圧は2.7 V～5.5 V（最大クロック20 MHz）です．ただし，末尾にVがついたもの（ATmega88V）は低電圧仕様のもので，1.8 V～5.5 V（最大クロック10 MHz）の範囲で使用可能です．

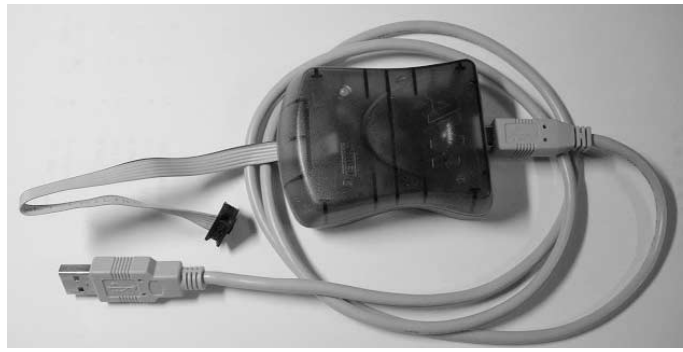

写真2-1　アトメル社純正のライタAVRISP-mkII

なお，クロックの下限についてはメーカのデータシートに明示されていないものの，電気的特性データが0.1 MHz～で表示されているので，0.1 MHz以上であれば問題なく使用できると思います．

AVRマイコン関連の取扱店はネットなどで検索してください．下記のお店は店頭または通販での販売を行っています．遠方の方は通販を利用するとよいでしょう．

- マルツパーツ館……http://www.marutsu.co.jp/user/index.php
- 共立電子エレショップ……http://eleshop.kyohritsu.com/
- 秋月電子通商……http://akizukidenshi.com/
- ストロベリーリナックス……http://strawberry-linux.com/（通販のみ）

電子部品など

トランジスタ，IC，抵抗，コンデンサ，LED，スイッチ，基板，配線材，ケースなどの部材については，電子部品を取り扱っているお店で購入してください．前述したお店でも入手可能です．

マイコンは，基板に直接はんだ付けしてしまうと交換が困難になるので，ICソケットを使用するようにしてください．

28ピンのAVRマイコン用ソケットは300 mil(＊1)幅のものを使用します．従来のマイコンでよく使われてきた600 mil(＊2)より幅が狭いため，購入にあたっては注意が必要です(**写真2-2**)．

その他に必要なもの

はんだゴテ，はんだ，ニッパー，ラジオ・ペンチなどの工具類は，必要に応じて用意してください．

2-2　プログラム環境の準備

アセンブラ言語による開発は，Windows上でプログラム作成やデバッグが可能なAVR Studioという統合開発環境で行うことが可能です．AVR Studioは，アトメル社純正のライタであるAVRISP-mkIIに付属するCD-Rに入っています．

また，アトメル社のWebサイトでは不定期に新しいバージョンがリリースされており，無料でダウンロードすることが可能です．本書ではWebサイトからAVR Studioを入手し，インストールする方法について述べます．

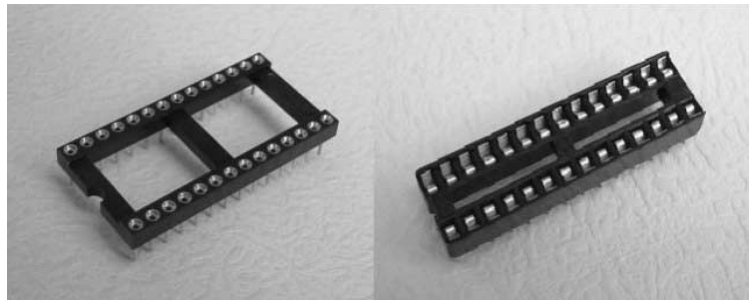

写真2-2　ICソケット(左600 mil，右300 mil)

(＊1) 1 mil=1/1000インチより，300 mil = 7.62 mm
(＊2) 1 mil=1/1000インチより，600 mil = 15.24 mm

[第3章]

ハードウェアの製作

この章では，マイコンを使用した工作を行ううえでのポイント，注意点について述べます．

3-1　部品選定および配置検討

ケースなどの部材や，必要な電子部品などを入手します．入手先については「2-1　必要機材」を参照してください．

配置については，下記に注意して決定します．

- マイコンは基板に直接はんだ付けせず，交換することを考えICソケットに実装する
- マイコンの向きはなるべく配線が短くて済む方向にする
- 書き込みコネクタは，書き込み器のソケットが差し込めるように周囲を空ける
- LEDやLCDなどの表示部品は，見やすい位置に配置する
- スイッチ類は操作しやすい位置に配置する

写真3-1～**写真3-3**は筆者の製作物で，GPSデータを取得しLCD(液晶表示器)に表示すると同時にSDカードに保存するものです．

一応，上記の注意点に基づいて部品を配置しています．ケースは100円ショップで購入しました．

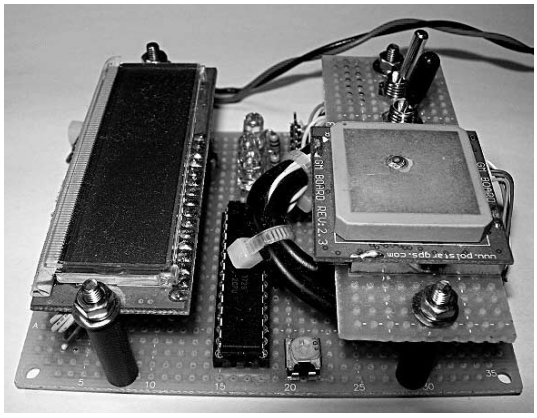

写真3-1　デバッグを意識した製作例

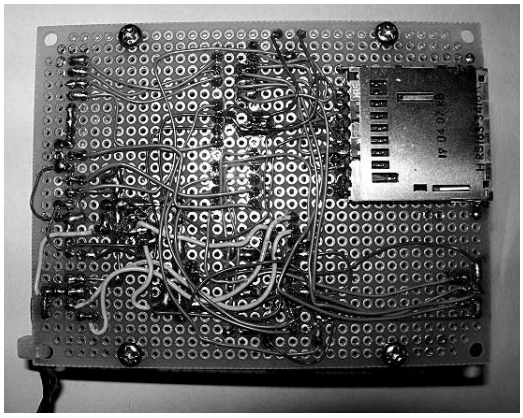

写真3-2　基板裏面(部品間配線)

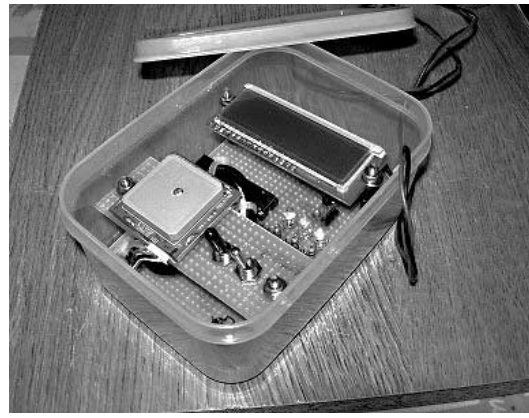

写真3-3　100円ショップにて購入したケースに収納

3-2　回路設計

回路設計にはある程度の知識と経験が必要ですが，マイコンの機能を理解しておけば，あとは接続するスイッチやLEDなどの機能を理解することで容易に設計できます．

下記に従い，マイコンのピン(端子)を割り当てます．

① 電源ピン V_{CC}，AV_{CC}，GND

どのような工作においても必ず配線されるピンです．

② プログラム書き込み用ピンMOSI，MISO，SCK，RESET

基板側のピン配置が**図3-1**になるように配線します．基板上に実装したようすを**写真3-5**に示します．

ピンは，筆者は秋月電子通商のピン・ヘッダ(オス)80P(2×40)，ピン・ヘッダ(オス)40P(2×20)などを使用しています(**写真3-4**)．ピン・ヘッダ(オス)のピン配列はいくつかあるので，適当なものを入手してください．ニッパやカッターで2×3にカットして使用します．

③ 外部クロック・ピンXTAL$_1$，XTAL$_2$

外部からクロックを供給する場合に，このピンを使用します．

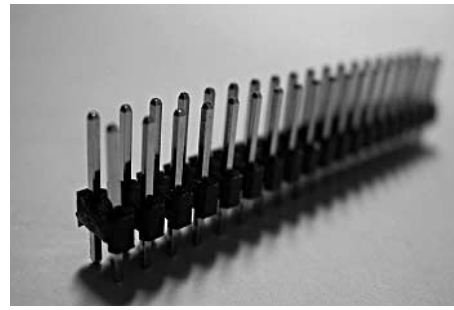

写真3-4　ピン・ヘッダ

[第4章]

ソフトウェアの製作

● 概要

ソフトウェアとはハードウェアの対義語で，コンピュータを動作させるための手順・命令を記述したものです．

ソフトウェアとプログラムはほぼ同じ意味ですが，プログラムで処理するためのデータを合わせてソフトウェアと呼ぶことがあります．

コンピュータはプログラムに忠実に従い動作するため，間違った動作をしないようにプログラムを記述する必要があります．この章では，プログラムを作成する際の構成，および考え方ついて解説します．

4-1　プログラムの構造

WindowsなどのOS上で動作するパソコンは，複数の異なるプログラムを並行して処理するため，プログラマがハードウェアに近い処理まで記述してしまうとほかのプログラムのメモリの内容を破壊するなどの悪影響が出てしまいます．そのため，パソコンのプログラムがハードウェアに近い処理を行おうとする場合，OSが提供するルーチンを使用するのが一般的です．

それに対してマイコンは，WindowsなどのOSは存在せず，一つのプログラムを順番に処理するため，プログラマが割り込み処理を含めたすべての動作を記述する必要があります．

プログラムの記述について決まった形というのはありませんが，本書では**図4-1**の構成で統一しています．

① ヘッダ

プログラムの見出しをコメント行として記述しておきます．

記述例

```
; プログラム名    ; タイマ例1
; ファイル名      ; timer1.asm
; 作成日          ;
; 作成者          ;
```

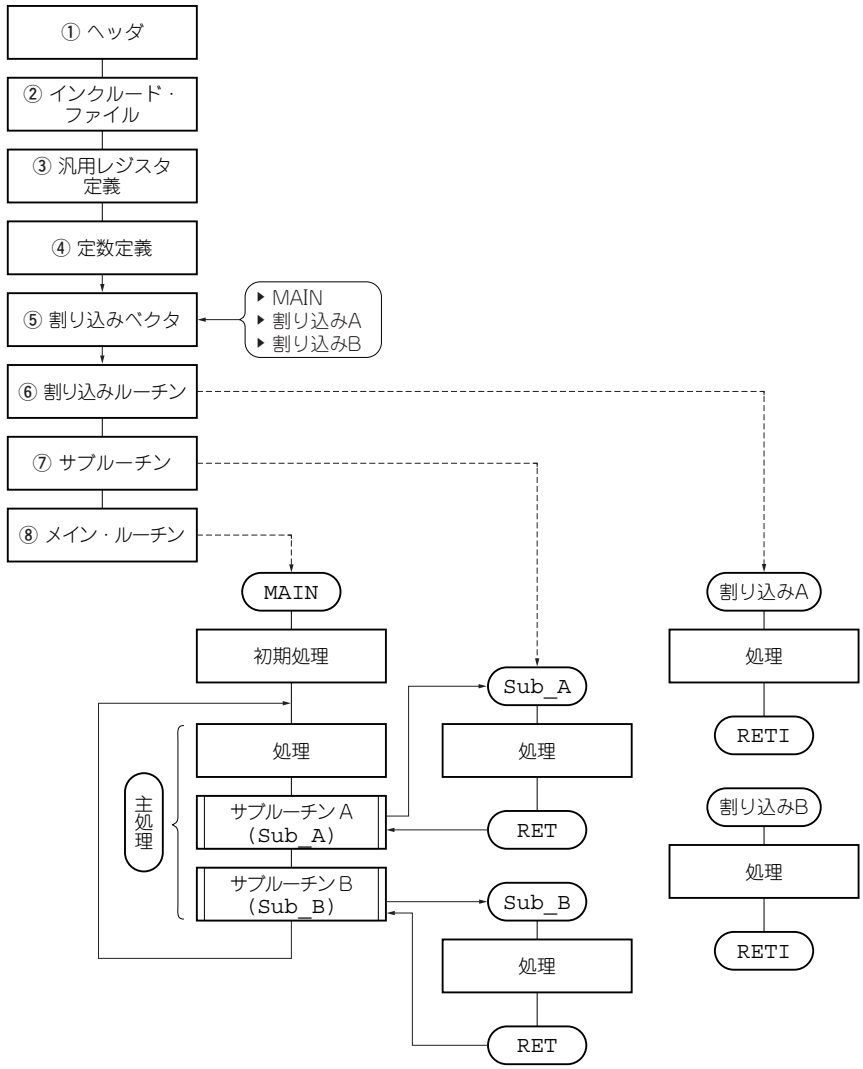

図4-1　プログラムの構造例

② インクルード・ファイル

プログラム中で使用する定数のうち，AVRマイコンのハードウェアに依存するものが定義されているファイルです．

マイコンの型式別に用意されており，たとえばATmega88では，`m88def.inc`，ATtiny45では`tn45def.inc`となります．

記述例

```
.include "m88def.inc"
.include "tn45def.inc"
```

[第5章]

アセンブラ命令の使い方

● 概要

AVRマイコンで使用するアセンブラ命令について説明します．**図5-1**に，さまざまな命令の記述例を示します．

AVRには非常に多くのアセンブラ命令が用意されており，頭で考えた処理の流れを素直にプログラムに反映することができます．

本書では，使用頻度の高い命令，扱いに注意を要する命令を中心に，使用例を交えて解説します．全命令については，「11-6　アセンブラ・コマンド一覧表」を参照してください．

なお，各命令に( )でアクセス範囲を明記しています．たとえば，SUB命令では汎用レジスタR0～R31をアクセスできますが，SUBI命令ではR16～R31になります．

アクセス範囲については，「6-1　レジスタのアクセス」を参照してください．

5-1　データ転送命令(1バイト)

アクセスするデータ・メモリの領域によって命令を使い分ける必要があります．

とくに標準I/Oレジスタをアクセスする際は，データ・メモリ・アドレスではなくI/Oアドレスを使用するので注意してください．

詳しくは，「第6章　メモリ・アクセス」を参照してください．

■ 汎用レジスタ(R0～R31)間

MOV命令 を使用します．

例 R16の内容をR0へ転送する．

```
MOV R0, R16
```

■ 標準I/Oレジスタ⇆汎用レジスタ(R0～R31)間

IN命令 標準I/Oレジスタ→汎用レジスタ

例 I/OアドレスのA0x0Aの内容をR0へ転送する．

```
IN R0, 0x0A
```

OUT命令 汎用レジスタ→標準I/Oレジスタ

例 R0の内容をI/OアドレスのA0x0Aへ転送する．

```
OUT 0x0A, R0
```

```
; プログラム名                    : タイマ0使用例2（ノーマルモード）
; ファイル名                      : timer02.asm

.include "m88def.inc"

;***************************************************************************
; 各種定義
;***************************************************************************
;-----------------------------------
; 汎用レジスタ
;-----------------------------------
.def STACK   = R16
.def R_TEMP1 = R17
.def R_TEMP2 = R18
.def R_FLAG1 = R19               ; ユーザ・フラグ・レジスタ
.def R_TMR0  = R20               ; タイマ0カウント領域（10ms単位）

;---- PORT ----
.EQU P_LED = PORTB               ; LED

;---- 入出力ビット -------------

.EQU B_LED = 0                   ; LED

;---- ユーザ・フラグ(R_FLAG1) -------------

.EQU B_IOVF0 = 0                 ; タイマ0オーバフロー

;-----------------------------------
; 割込みベクタ定義
;-----------------------------------
.CSEG                            ; コード・セグメント
 RJMP MAIN                       ; メイン・ルーチン
.ORG 0x0010
 RJMP IOVF0                      ; タイマ0

;***************************************************************************
; 割込みルーチン
;***************************************************************************
;*****************************
; タイマ0
;*****************************

IOVF0:
 IN STACK, SREG

 DEC R_TMR0
 BRNE IOVF0_1

 SBR R_FLAG1, (1<<B_IOVF0)
 RJMP IOVF0_9

IOVF0_1:
 LDI R_TEMP1, 0xD9               ; 割り込み時間10msを再セット
 OUT TCNT0, R_TEMP1

IOVF0_9:
 OUT SREG, STACK

 RETI
```


図5-1　アセンブラの記述例