

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)

見本

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)。

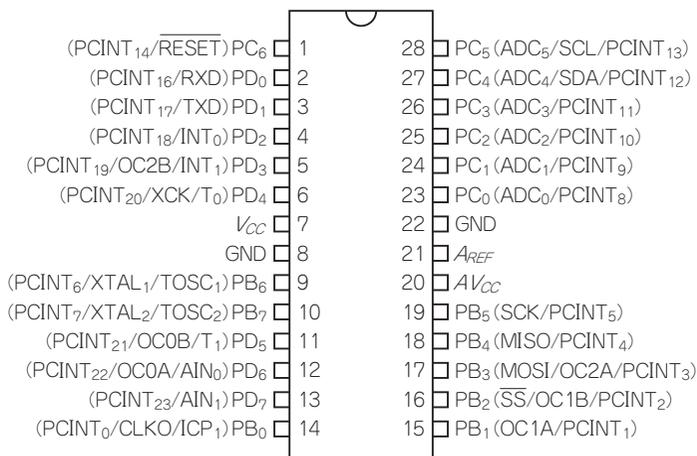


図1-1 ATmega48 / ATmega88 / ATmega168 (28 ピンDIP)のピン配置

見本

表1-2 ATmega48 / ATmega88 / ATmega168 (28 ピンDIP)のピンNo.別機能表

ピン No.	ポート	電源	外部割り込み	ピン変化割り込み	通信	A-D 変換	アナログ・コンパレータ	PWM 出力	その他
1	PC ₆			PCINT ₁₄					RESET
2	PD ₀			PCINT ₁₆	RXD				
3	PD ₁			PCINT ₁₇	TXD				
4	PD ₂		INT ₀	PCINT ₁₈					
5	PD ₃		INT ₁	PCINT ₁₉				OC2B	
6	PD ₄			PCINT ₂₀					
7		V _{CC}							
8		GND							
9	PB ₆			PCINT ₆					XTAL ₁
10	PB ₇			PCINT ₇					XTAL ₂
11	PD ₅			PCINT ₂₁				OC0B	
12	PD ₆			PCINT ₂₂			AIN ₀	OC0A	
13	PD ₇			PCINT ₂₃			AIN ₁		
14	PB ₀			PCINT ₀					
15	PB ₁			PCINT ₁				OC1A	
16	PB ₂			PCINT ₂				OC1B	
17	PB ₃			PCINT ₃	MOSI			OC2A	
18	PB ₄			PCINT ₄	MISO				
19	PB ₅			PCINT ₅	SCK				
20		AV _{CC}							
21		A _{REF}							
22		GND							
23	PC ₀			PCINT ₈		ADC ₀			
24	PC ₁			PCINT ₉		ADC ₁			
25	PC ₂			PCINT ₁₀		ADC ₂			
26	PC ₃			PCINT ₁₁		ADC ₃			
27	PC ₄			PCINT ₁₂	SDA	ADC ₄			
28	PC ₅			PCINT ₁₃	SCL	ADC ₅			

各ピンのピン別機能を表1-2に示します。それぞれの機能の概要は次のようになります。

ポート

信号入出力用として用意された端子です。

ポートB(PB₀~PB₇)、ポートC(PC₀~PC₆)、ポートD(PD₀~PD₇)の計23本です。

内部に入力用のプルアップ抵抗をもっており、利用するか否かを設定することができます。

電源

V_{CC}(+電圧)とGND(-電圧)間には電源電圧を加えます。

AV_{CC}は、A-D変換機能を利用する場合に必要な+電圧でV_{CC}±0.3V以内とする必要があります、メーカーのデータシートによるとA-D変換を使用しない場合でもAV_{CC}への供給を推奨しているの、通常はV_{CC}とAV_{CC}を接続するようにしてください。

外部割り込み

見本 INT₀端子(外部割り込み0)に加わる信号が変化すると、システムはプログラムの割り込みベクタ2(プログラム・アドレス0x0001)をアクセスし、外部割り込み0の処理が行われます。

INT₁端子(外部割り込み1)に加わる信号が変化すると、システムはプログラムの割り込みベクタ3(プログラム・アドレス0x0002)をアクセスし、外部割り込み1の処理が行われます(「8-3 外部割り込み」参照)。

ピン変化割り込み

すべてのポートの端子は、入出力値の変化で割り込みを発生することが可能です(「8-4 ピン変化割り込み」参照)。

通信

通信機能として下記をサポートしています。

■ USART (「8-8 USART」参照)

汎用同期・非同期通信方式です。RS-232Cなどのシリアル通信を行う場合に利用されます。

■ TWI (「8-9 TWI」参照)

2線式シリアル・インターフェースあるいはI²Cインターフェースとも呼ばれます。

通信速度の上限が400 kHzとSPIと比べて低速ですが、SPIより前に規格化されたため、多くのデバイスで採用されています。

■ SPI (「8-10 SPI」参照)

3線式シリアル・インターフェースとも呼ばれます。

TWIよりも高速通信が可能で、通信速度の上限はマスタのシステム・クロックの1/2となります。

A-D変換

アナログ・データを取り込む部分です(「8-5 A-D変換」参照)。

アナログ・コンパレータ

アナログのデータを比較する機能です(「8-6 アナログ・コンパレータ」参照)。

PWM出力

モータなどを駆動する出力です(「8-7 PWM」参照)。

その他

XTAL₁, XTAL₂は、外部からクロックを供給するための外部クリスタルやセラミック発振子などを接続する端子です(「11-4 クロック設定」参照)。

1-2 マイコンの動作

クロック

マイコンは、ほとんどの機能がクロックと呼ばれるパルス信号に同期して動作します。クロックは外部から供給するほかに、マイコン内部で生成することが可能です。

マイコン内部で生成するクロックは内蔵RCオシレータと呼ばれ、周波数は8 MHzです。ただし実際は、ヒューズ・ビットの設定値で分周された周波数でマイコンは動作します(「11-1 ヒューズ・ビット」参照)。

通常の工作では、8 MHzを超えるクロックが必要になることはあまりないので、部品点数削減の

見本のためにも内蔵RCオシレータの使用が便利です。本書におけるすべての使用例(サンプル・プログラム)は、内蔵RCオシレータを用いています。

リセット

マイコンは、電源ONから一定時間経過後にリセットがかかります。リセットとは、マイコンを初期状態から起動させることをいいます。

リセットがかかると、マイコンのハードウェアはプログラムの割り込みベクタ1(プログラム・アドレス0x0000*)をアクセスします。プログラム・アドレス0x0000にはメイン・ルーチンへのジャンプ命令が書かれているので、メイン・ルーチンの最初から実行されることになります。

電源ONによるリセットをパワーオン・リセットといいます。その他にも下記の要因によりリセットが発生します。

■ 外部リセット

RESET端子が2.5 μ s以上Lowになったとき。

■ ウォッチドッグ・リセット

ウォッチドッグ・タイマがタイムアウトしたとき(「11-2 ウォッチドッグ・タイマ」参照)。

■ ブラウンアウト・リセット

マイコンの電源電圧がある値より低下したとき。この値はヒューズ・ビットで設定されます(「11-1 ヒューズ・ビット」参照)。

プログラムの実行

マイコンは電源ONによるリセットの後、プログラムのメイン・ルーチンに処理が移ります。

メイン・ルーチンでは、初期処理として各ポートの入出力設定や、利用する機能の関連レジスタの初期設定を行います。

次に、主処理としてスイッチなどの入力やLED点灯などの出力、タイマやA-D変換などの機能の利用、判断や計算などを繰り返し行います。ただし、(リセット)のところでも述べた要因でリセットがかかった場合は、メイン・ルーチンの最初から実行されます。

1-3 メモリ構造

AVRマイコンのメモリは、プログラム領域、データ領域、EEPROM領域に分かれます。

これらの構造について図示したものをメモリ構造またはメモリ・マップと呼びます。アドレスの表現には16進表記が用いられ、先頭に0xまたは末尾にhを付けて表します(例 0x33FA, 33FAh)。

プログラム領域

プログラム領域はフラッシュ・プログラム・メモリと呼ばれ、電源がOFFになっても内容は消えません。1アドレスが2バイトで構成され、ATmega88における容量は8Kバイト、アドレスは0x0000~0x0FFFです(図1-2)。

データ領域

データ領域はSRAM(Static Random Access Memory)と呼ばれ、電源がOFFになると内容が初期状態(値は不定)に戻ります。1アドレスが1バイトで構成されます(図1-3)。

データ領域内部はさらに、32個の汎用レジスタ、64個の標準I/Oレジスタ、160個の拡張I/Oレジ

見本
*) 0x0000 16進の表記の0。

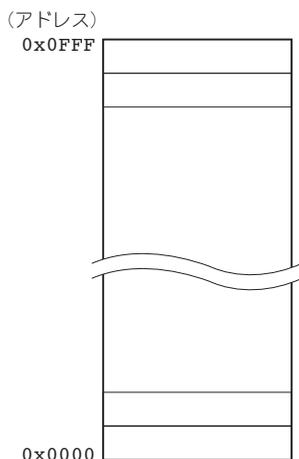


図1-2 プログラム領域のメモリ・マップ

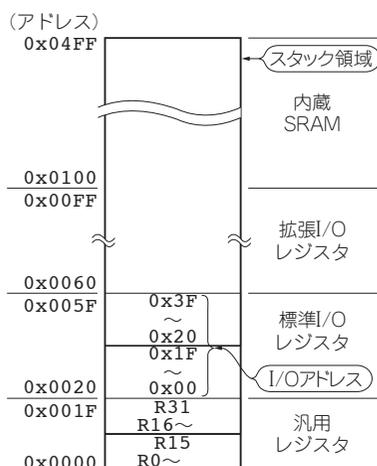


図1-3 データ領域のメモリ・マップ

スタ、および1 Kバイトの内蔵SRAMに分けられます。レジスタ・グループで256バイトを占め、アドレスは0x0000～0x00FFです。

アドレス0x0100～0x04FFの1 Kバイトは、内蔵SRAMです。

レジスタというのは、CPUとの高速なやりとりを目的とした特別なメモリです。

■ 汎用レジスタ

ユーザがプログラム中で一時的にデータを保存するために使用します。データ・メモリ・アドレスとは別に、R0～R31の名称が付けられています。

■ 標準I/Oレジスタ

標準的に使用される機能の設定値や、CPUの処理結果などが格納されます。データ・メモリ・アドレスとは別に、0x00～0x3FのI/Oアドレスが付けられています。

標準I/Oアドレスの詳細および使用上の注意については、「第6章 メモリ・アクセス」を参照してください。

■ 拡張I/Oレジスタ

標準I/Oレジスタ以外の機能の設定値や、CPUの処理結果などが格納されます。拡張I/Oレジスタの詳細および使用上の注意については、「第6章 メモリ・アクセス」を参照してください。

■ 内蔵SRAM

電源がOFFになると内容が消えるので、演算結果や計測データなどの一時保管の用途に使用されます。読み書きは汎用レジスタを介して行います。

内蔵SRAMの最終アドレスから下へ向かってスタック領域として使用されるので、プログラムで使用する際はスタック領域と反対方向、すなわち内蔵SRAMの開始アドレスから上へ向かって使用するにします(「6-5 スタック・ポインタのアクセス」参照)。

EEPROM領域

EEPROM領域は1アドレスが1バイトで構成され、データの保存に使用する点ではデータ領域と同

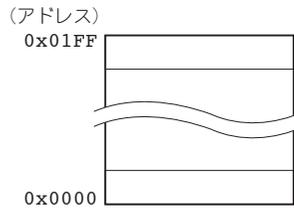


図1-4 EEPROM領域のメモリ・マップ

様ですが、アクセス方法がSRAMと異なります。電源を切っても書き込まれた内容が保持されるので、可変データの保存などに適します(図1-4)。

書き換え回数に限度がある(本書で使用するマイコンでは10万回程度)ので、頻繁に内容が変わるような用途には不向きです。また書き込みにはある程度の時間が必要ですので、プログラムの際に考慮する必要があります。

1-4 ISPプログラミング

マイコンを動作させるためのプログラムをパソコンで作成した後、マイコンへ書き込む必要があります。現在のマイコンはプログラムを何度でも書き込めるものが主流で、非常に短時間で内容を書き換えることが可能です。

AVRマイコンは、マイコンを基板に実装した状態でプログラムを書き込む(書き換える)ことができ、この方式をISP(In System Programming)といいます。書き換え可能な回数も1万回以上と、通常の使用であれば回数を気にせず書き込みます。

1-5 ATmega48/88/168の相違点(注意点)

「概要」のところで、ATmega48/88/168はメモリ容量が異なると書きました。その違いは下記の点で影響するため、プログラムにおいては注意が必要です。

割り込み定義に関するもの

割り込み発生時のプログラム・アドレスがATmega48/88/とATmega168とで異なります。

割り込みベクタ・テーブルの記述において、ATmega48および88は割り込みプログラム・アドレスが1番地ごとに割り振られるのに対し、ATmega168は2番地ごとに割り振ります。詳しくは「第7章 割り込みについて」を参照してください。

飛び先指定に関するもの

ATmega48および88は、無条件ジャンプに1ワード(2バイト)命令であるRJMP命令を使用します。

見本 それに対してATmega168は、RJMP命令に加えて2ワード(4バイト)命令であるJMP命令を使用することが可能です。詳しくは「5-5 無条件ジャンプ」を参照してください。

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。

ライター

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ー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。

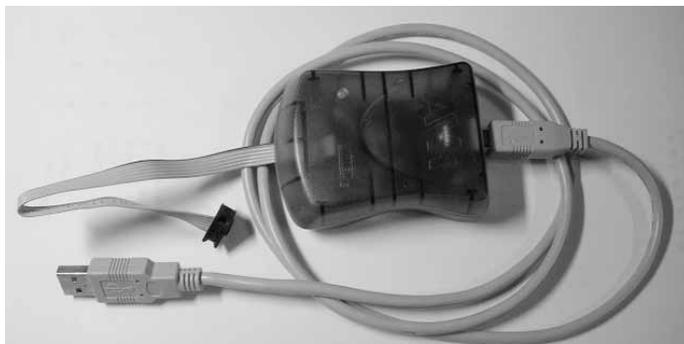


写真2-1 アトメル社純正のライターAVRISP-mkII

見本

なお、クロックの下限についてはメーカー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leshop……………<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ーである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。

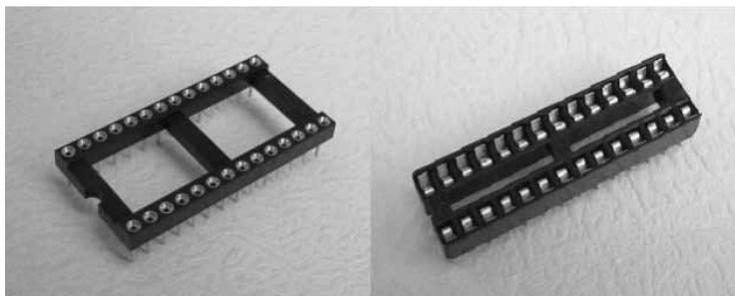
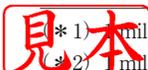


写真2-2 ICソケット(左600 mil, 右300 mil)



*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

AVR Studioの入手

図2-1～図2-9の手順でパソコンにダウンロードします。

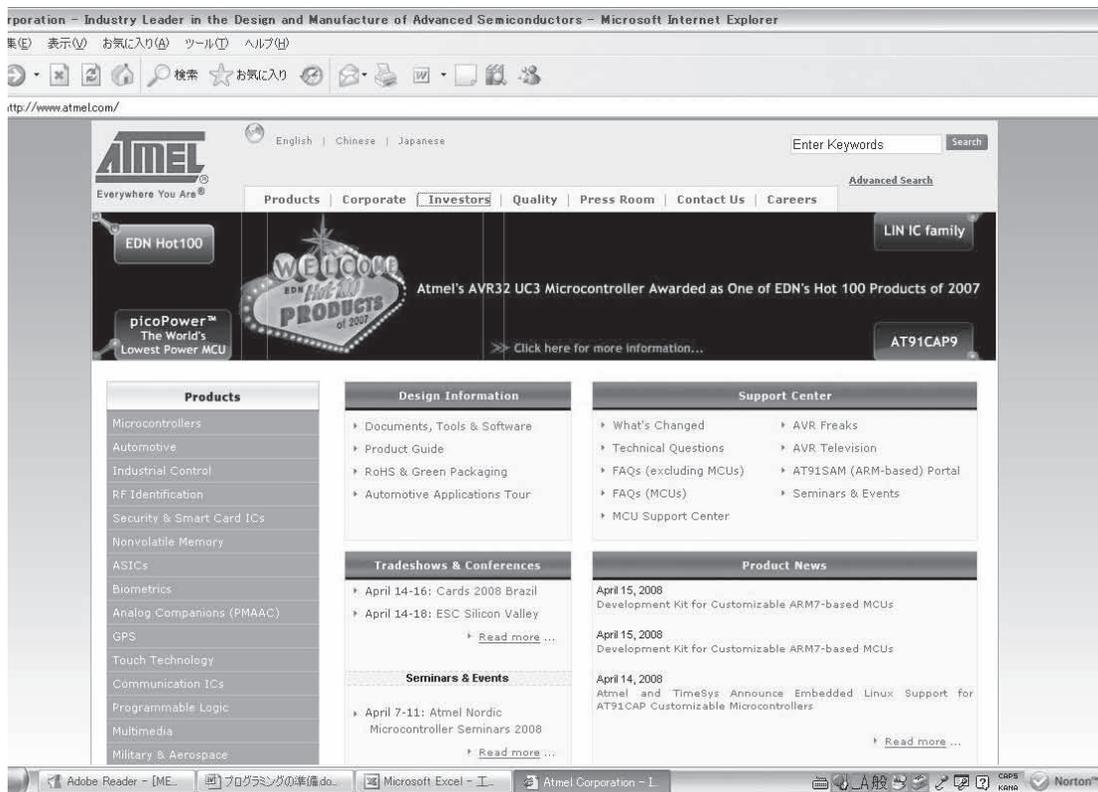
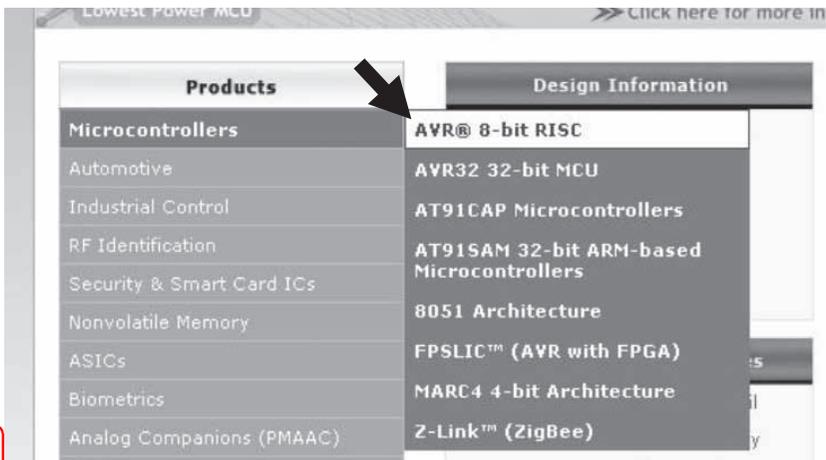


図2-1 アトメル社のWebサイト <http://www.atmel.com/> にアクセスする



見本

図2-2 左側メニューからMicrocontrollers→AVR[®] 8-bit RISCを選択する

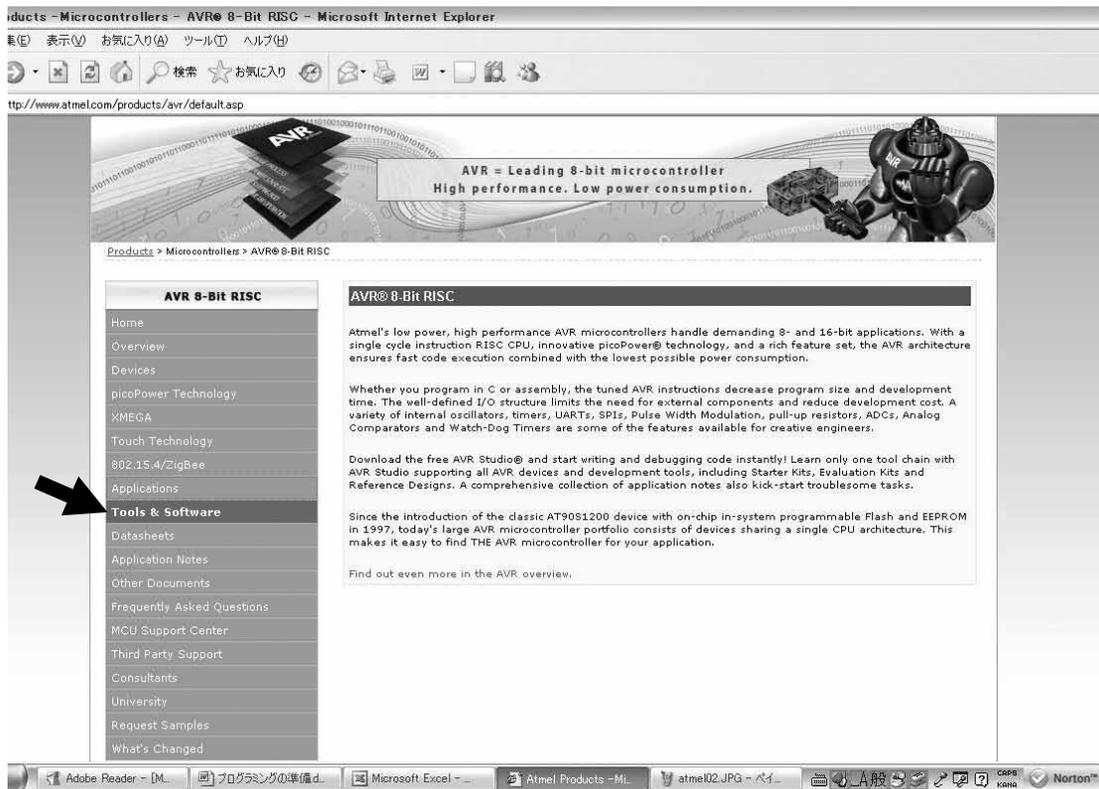


図2-3 表示された画面の左側メニューからTools & Softwareを選択する

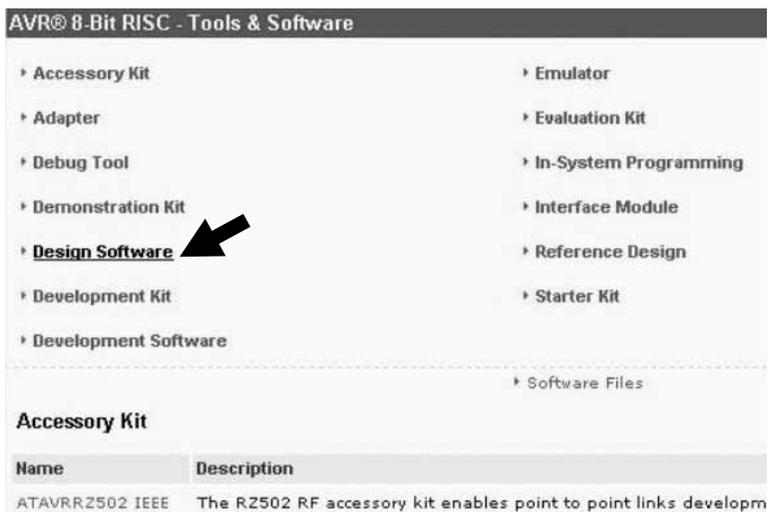


図2-4 画面の中央メニューからDesign Softwareを選択する

見本

Design Software	
Name	Description
ATAVRTOS	<p>ATAVRTOS contains the complete source code of Micrium uC/OS-II Real Time Operating System. It also includes the Kernel Aware plug-ins for AVR Studio and Micrium supported versions for ATmega128/2561. With this kit you can write complex uC/OS-II based code and debug within the AVR Studio framework. Support is included for STK500/501/503 and JTAGICE mkII.</p> <p>ATAVRTOS contains a book and CD-ROM with software. .</p> <p>The AVR Studio online-help contains the most current information and a complete list of supported devices.</p> <p>Find more information here.</p> <p>Ordering code: ATAVRRTOS</p>
AVR Studio 4	<p>AVR Studio® 4 is the new professional Integrated Development Environment (IDE) for writing and debugging AVR® applications in Windows® 9x/NT/2000/XP environments.</p> <p>AVR Studio 4 includes an assembler and a simulator. The following AVR development tools are also supported: ICE50, ICE40, JTAGICE mkII, JTAGICE, ICE200, STK500/501/502/503/504/505/520, AVRISP mkII, and AVRISP.</p> <p>Release Notes</p>
AVR Studio 4 SDK	The AVR Studio® Software Development Kit makes it possible to develop seamless extensions to AVR Studio 4. To receive the free SDK users must:

図2-5
表示された画面からAVR Studio 4を選択する

AVR Studio 4.14 (build 589) (89 MB, updated 4/08)
AVR Studio 4.14 includes new device support and numerous overall enhancements; Support for teh new XMEGA device family, full support for 32-bit and 64 bit versions of Vista, AVR Wireless Services application supporting the AVR Raven Kit See release notes for more details.
AVR Studio 4.13 SP2 (build 571) (45 MB, updated 12/07)
AVR Studio 4.13 SP1 (build 557) (37 MB, updated 9/07)
AVR Studio 4.13 (build 528) (73 MB, updated 03/07)
AVR Studio 4.13 includes new device support and numerous overall enhancements; Complete device support for AVR Dragon, 2 new I/O- and Processor views, WinAVR and 32- and 64-bit Windows(R) support! See release notes for more details.
AVR LCD Visualizer (1 MB, revision 1.0, updated 02/04)
Create and modify your own LCD displays with the editor, debug and visualize inside AVR Studio 4. Run-time update with ICE50 and Simulator. Supports ATmega169. Minimum requirement: AVR Studio 4.07. Remove old LCD plug-ins before installing.

図2-6
最もバージョンが大きいAVR Studio 4 (画面では4.14)を選択する
これがAVR Studioの本体だが、本体のバージョンの後ろにSPと付くファイル(サービス・パック)がある場合はそれもダウンロードしておく(画面では4.14用のサービス・パックは存在しない)。

Fields marked with an asterisk (*) are required.

First Name*	<input type="text"/>
Last Name*	<input type="text"/>
Title	<input type="text"/>
Job Function*	Select <input type="button" value="v"/>
Company*	<input type="text"/>
Address 1*	<input type="text"/>
Address 2	<input type="text"/>
City*	<input type="text"/>
State/Province* (US and Canada only)	Select <input type="button" value="v"/>
Province (only if outside US and Canada)	<input type="text"/>
Postal Code*	<input type="text"/>

図2-7
ユーザ情報を入力する
本体をダウンロードするには、ユーザ情報を入力する。*印の項目は必須。最後に画面下方にある“Click to Download”をクリックすると情報が転送される。



図2-8
 “ Thank you for registering.
 Please click here to begin the download.” の “here” のところ
 をクリックする



図2-9
 保存(S)をクリックし、インストール先を指定する
 ダウンロードが開始される。サービス・パックのファイル
 がある場合は、併せてダウンロードする。



AVR Studioのインストール

図2-10と図2-11の手順でパソコンにインストールします。



見本

図2-10 ダウンロードしたファイルを開くとこの画面が表示される
 Next> をクリックする。

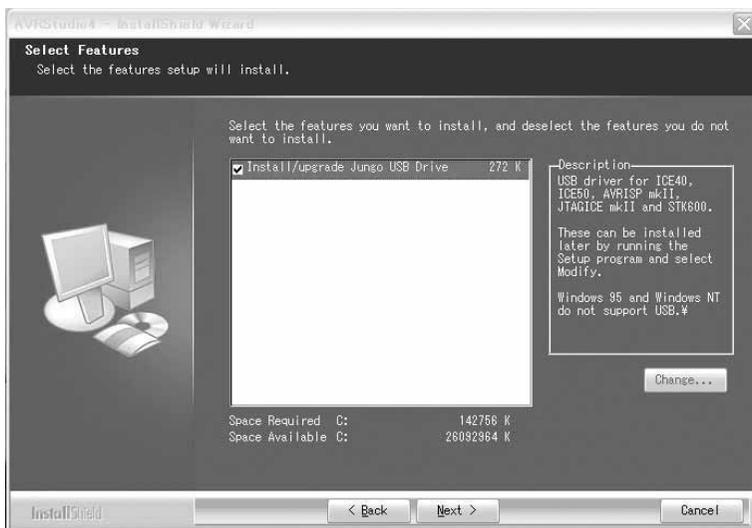


図2-11 Install/upgrade Jungo USB Driveにチェックを入れてNext>ボタンをクリックする



図2-12 プログラム一覧からAtmel AVR Tools→AVR Studio 4を選択する

AVR Studioの起動

図2-12～図2-16の手順でAVR Studioを起動します。

続いて、Project nameの欄に半角英数字でプロジェクト名を入力します。

見本 creative initial fileにチェックを入れると、.asmの拡張子をもつファイルが作成されます。
 Creative folderにチェックを入れると、Project nameの欄に入力した名前フォルダが作成されます。

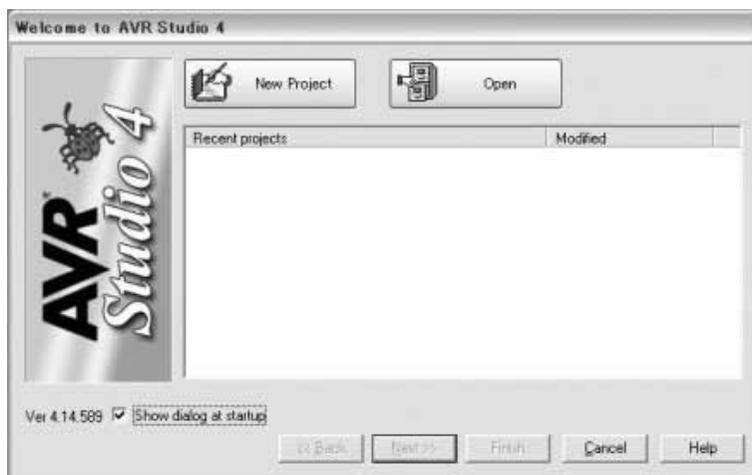


図2-13 New Projectをクリックする

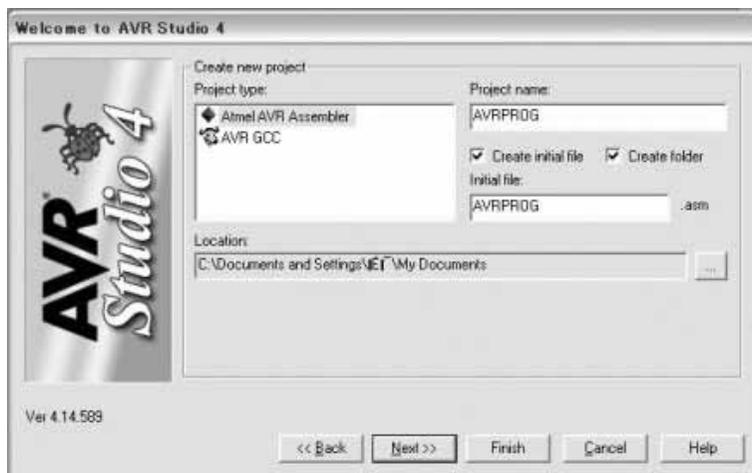


図2-14 Atmel AVR Assemblerをクリックする

プロジェクトが作成される場所はLocationで示される場所になりますが、入力欄右横の...をクリックして変更することができます。

次に、使用するAVRを選びます(図2-15)。終了は図2-16のように[]をクリックします。

このように、プログラムを書き始めるまでに準備することが多くありますが、このプロジェクト管理のメリットは、単にアセンブルをするだけでなく、統合的な開発および管理ができるという点にあります。

AVR Studioの中では、プログラム記述→アセンブル→シミュレーション・デバッグ→書き込みの一連の作業が実行できます。

見本 シミュレーション・デバッグは、マイコンにプログラムを書き込む前にプログラム動作をシミュレートすることが可能で、処理の流れによりマイコンのレジスタ値やメモリ値がどのように変化するか