

## AVR1900 : STK600でのATxmega128A1開始に際して

### 1. 序説

本応用記述はSTK<sup>®</sup>600でATxmega128A1をどう始めるかについての情報を含みます。

最初の3項はATxmega128A1に対するコードのコンパイルとデバッグに必要なソフトウェア(AVR Studio<sup>®</sup>, WinAVR, AVR<sup>®</sup>用IAR Embedded Workbench<sup>®</sup>)についての情報を含みます。次の項はATxmega128A1に対するSTK600の初期設定方法の概要説明を含みます。最終項はWinAVRまたはIAR Embedded Workbenchを使ってプロジェクトの作成とATxmega128A1に対するコードのコンパイルの方法と、AVR StudioとJTAGICEmk IIを使ってSTK600上でコードのデバッグを開始する方法を示します。

ソフトウェア ツールを何処で探すかについてのより多くの情報は7章で得られます。

### 2. ATxmega128A1を支援するAVR Studio

AVR Studioはコードのデバッグに使われます。これはシミュレータとAVR XMEGAでのチップ上デバッグを行うためのJTAGICEmk II用の前処理部を含みます。STK600とATxmega128A1に対する支援を得るにはAVR Studio 4.14またはそれ以降の版がインストールされていなければなりません。

**注:** AVR Studioのインストール中にUSBドライバをインストールすることを覚えて置いてください。

### 3. IAR Embedded Workbench 5のインストール

AVR用のIAR Embedded Workbench 5.10(またはそれ以降) (IAR EW評価版と完全版)はATxmega128A1に対するコンパイル支援を持っています。

応用自動実行を使ってインストールを開始するには“Install IAR Embedded workbench”を選択してください。

図3-1. AVR用IAR EWインストール



インストール中にIAR Embedded Workbenchの以前版に対してと同じ承諾番号と承諾鍵を使うことが可能です。

IARのインストール中の既定設定を使うことが推奨されます。IAR EWでのプロジェクトの開始方法は6.1.項で示されます。

### 4. WinAVRのインストール

SourceforgeからWinAVR公開20080411(またはそれ以降)をダウンロードしてください。avr-gccツールチェーンはAVR Studioと自動的に統合します。WinAVRでのプロジェクトの開始方法は6.2.項で示されます。



8ビット **AVR<sup>®</sup>**  
マイクロコントローラ

応用記述

本書は一般の方々の便宜のため有志により作成されたもので、Atmel社とは無関係であることを御承知ください。しおりのはじめにでの内容にご注意ください。

Rev. 8107A-04/08, 8107AJ2-06/21

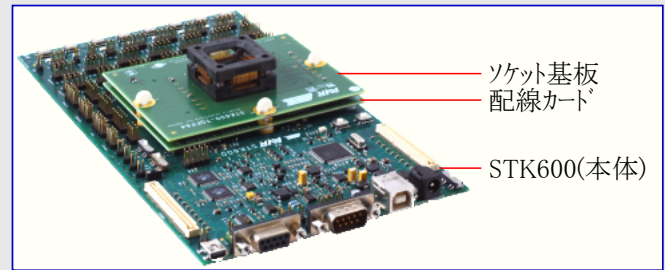
## 5. ATxmega128A1用のSTK600初期設定

STK600はフラッシュメモリを持つ全ての8ビットAVRとAVR32を支援するように設計されています。ソケットと配線カードに基づくシステムはSTK600基板上で各種の外圍器形式とピン配置を支援するのに用いられます。

STK600を使う前とソケットと配線カードを載せる前にSTK600が最新ファームウェアを含み、正しいVTarget電圧を持つのを調べるのが推奨されます。

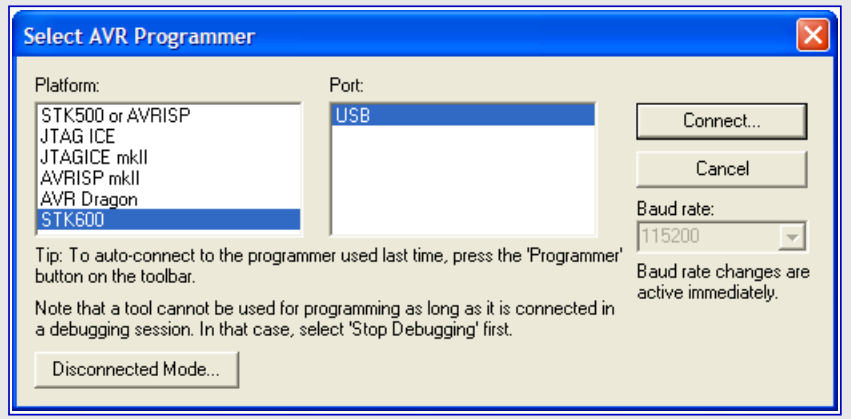
**注:** ドライバがインストールされているのを保証するためにSTK600への接続前にAVR Studioがインストールされるべきです。

図5-1. STK600のソケットと配線システム



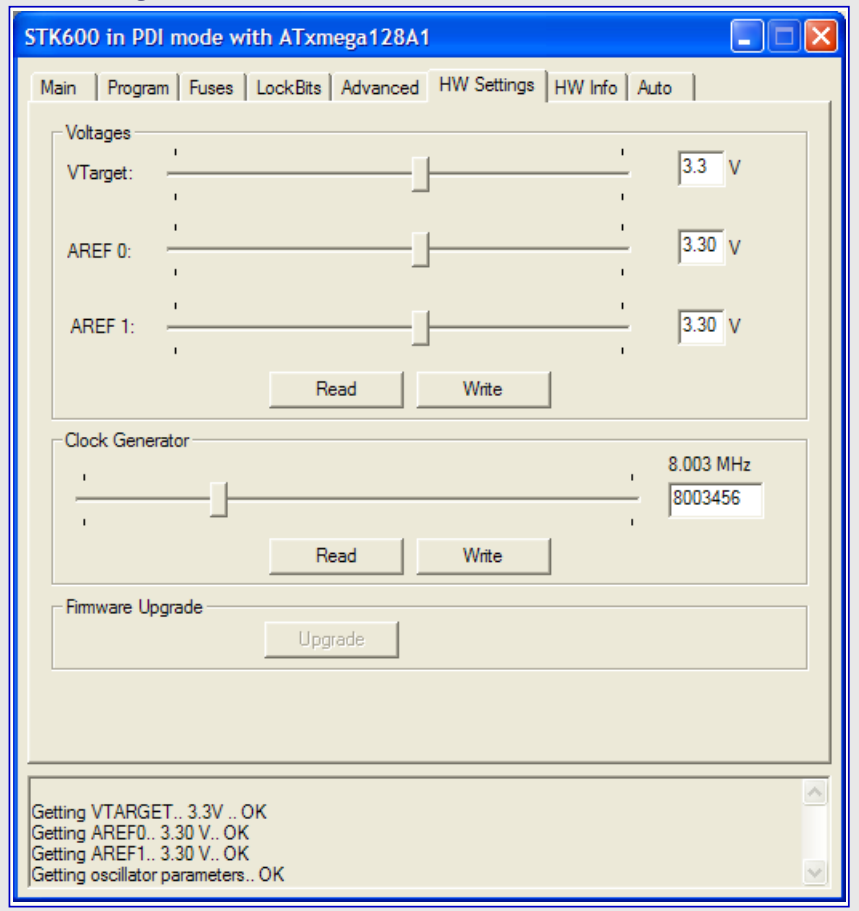
1. STK600にUSBケーブルを接続して、それを電源ONにしてください。
2. Windows®は新しいハードウェアが見つかったことを知らせます。STK600用のドライバをインストールしてください。
3. AVR Studioを開いて始動ダイアログを取り消して(Cancel)ください。
4. メニューバーから「Tools(ツール)⇒Program AVR(AVR書き込み)⇒Connect...(接続...)」を選び、図5-2.で図解されるようにSTK600とUSBを選択することによってSTK600との接続を始めてください。

図5-2. STK600への接続



1. STK600に対してより新しいファームウェアが利用可能な場合、AVR Studioは更新を行うことを希望するかの問い合わせを行うでしょう。ファームウェアの更新を行うにはAVR Studioによって与えられる指示に従ってください。
2. 更新後、(上の手順4.と同じ方法で)STK600へ再び接続し、HW Settings(ハードウェア設定)頁を開いてください。1.8~3.6Vの間にVTarget電圧を設定し、STK600にその値を書いてください。正しい値が書かれたことを検証するために読み戻すことが推奨されます。

図5-3. VTarget電圧設定

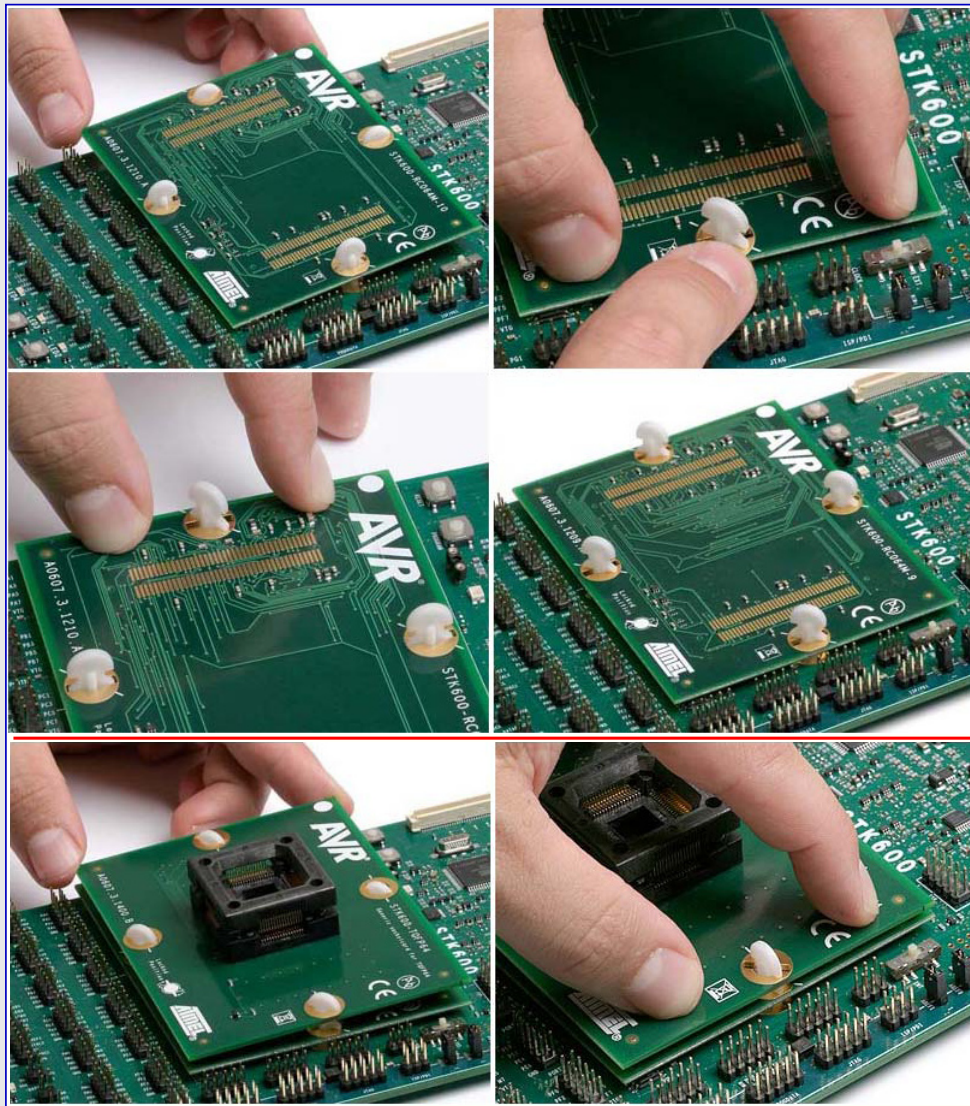


最新ファームウェアと正しい目的電圧が設定されたなら、配線カードとソケット基板を載せる前にSTK600を電源OFFにしてください。

ATxmega128A1に用いられる配線カードは“STK600-RC100X-13”、100ピンデバイスに用いられるソケット基板は“STK600-TQFP100”です。

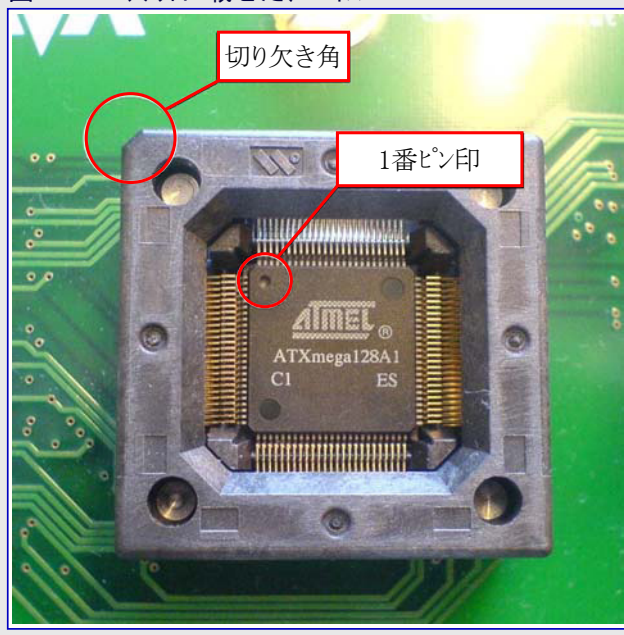
STK600への配線カードとソケット基板の載せ方は図5-4.で示されます。誤った方向で配線カードとソケット基板を載せることは不可能です。

図5-4. STK600への配線カードとソケット基板の載せ方



ZIFソケットにデバイスを載せる時は正しい向きでデバイスが載せられることを保証してください。デバイスの1番ピンは円点で印されています。図5-5.で見えるように、この角はソケットの切り欠き角と同じ方向に向かなければなりません。

図5-5. ZIFソケットに載せたデバイス

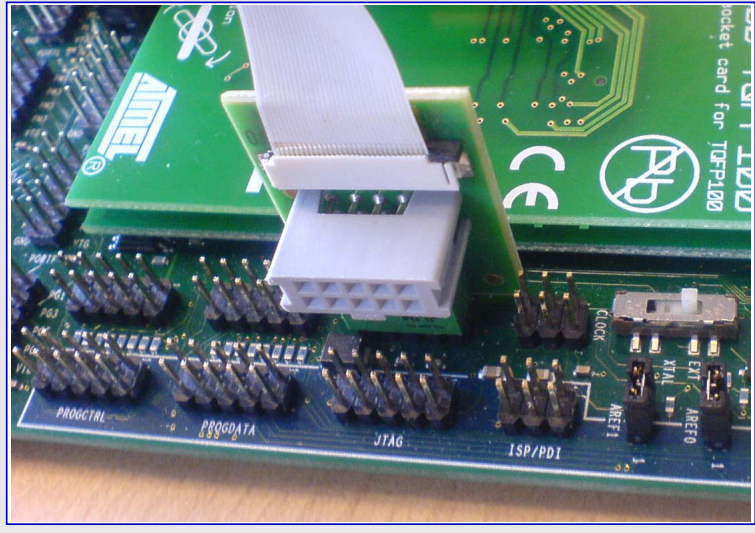


STK600はUSBケーブルを通して電力を供給することができます。USBケーブルを通して利用可能な電力が制限されていることを覚えて置いてください。STK600が300mAよりもっと消費する外部ハードウェアに接続される場合、STK600上のDC入力ソケットに接続される外部電源を使うべきです。この外部電源は中心が正(+)のコネクタで、10～15Vであるべきです。

コードのデバッグはAVR StudioとJTAGICEmk IIとで行われます。STK600にJTAGICEmk IIを使ってチップ上デバッグを行う方法は以下で記述されます。

1. 下で示されるようにJTAGICEmk IIをSTK600に接続してください。

図5-6. STK600へのJTAGICEmk II 接続



2. STK600とJTAGICEmk IIを電源ONにしてください(JTAGICEmk IIが使われるのが初回なら、Windowsはドライバのインストールを問い合わせるでしょう)。

STK600とJTAGICEmk IIについてのより多くの情報はAVR Studioで利用可能です(Help(ヘルプ)⇒AVR Tools User Guide(AVRツール使用者の手引き))。

## 6. コードのコンパイルとチップ上でのコード デバッグ

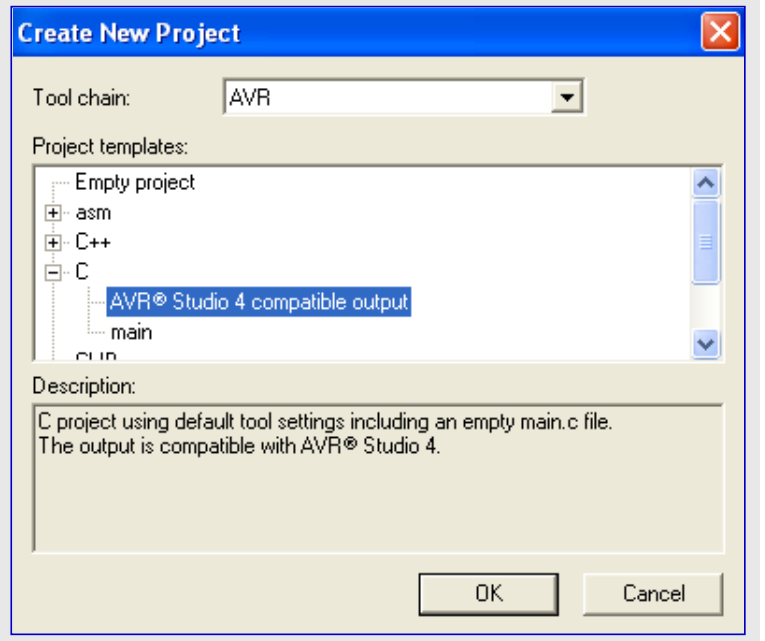
本項はIAR EWとWinAVRの両方でのプロジェクト作成、コンパイル、チップ上デバッグの方法を記述します。IAR EWでの開発については6.1. 節をお読みください。WinAVRでの開発については6.2. 項をお読みください。

### 6.1. IARとAVR Studioでのコンパイルとデバッグ

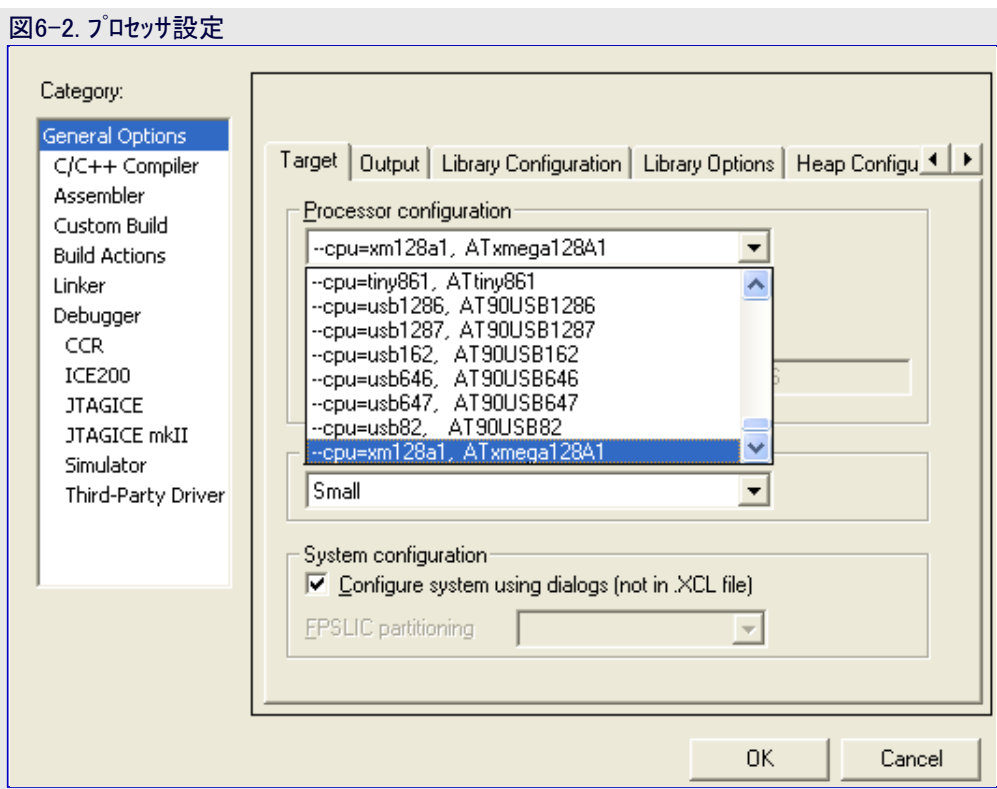
IAR EWでATxmega128A1に対するプロジェクトを始めるには、下で記述されるように行ってください。

- IAR Embedded Workbenchを開いてください。
- メニューバーで"Project(プロジェクト)⇒Create New Project(新規プロジェクト作成)"を選択してください。
- AVR Studio互換出力で希望するコーディング言語を選択してください。

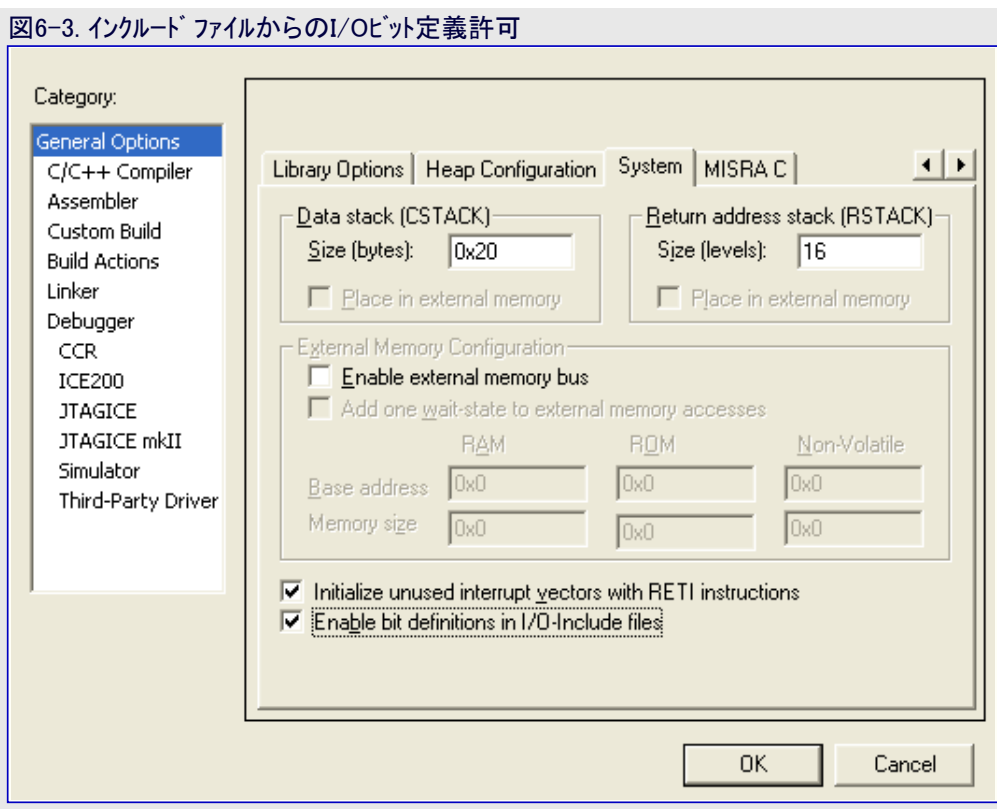
図6-1. AVR用IAR EWでの新規プロジェクト作成



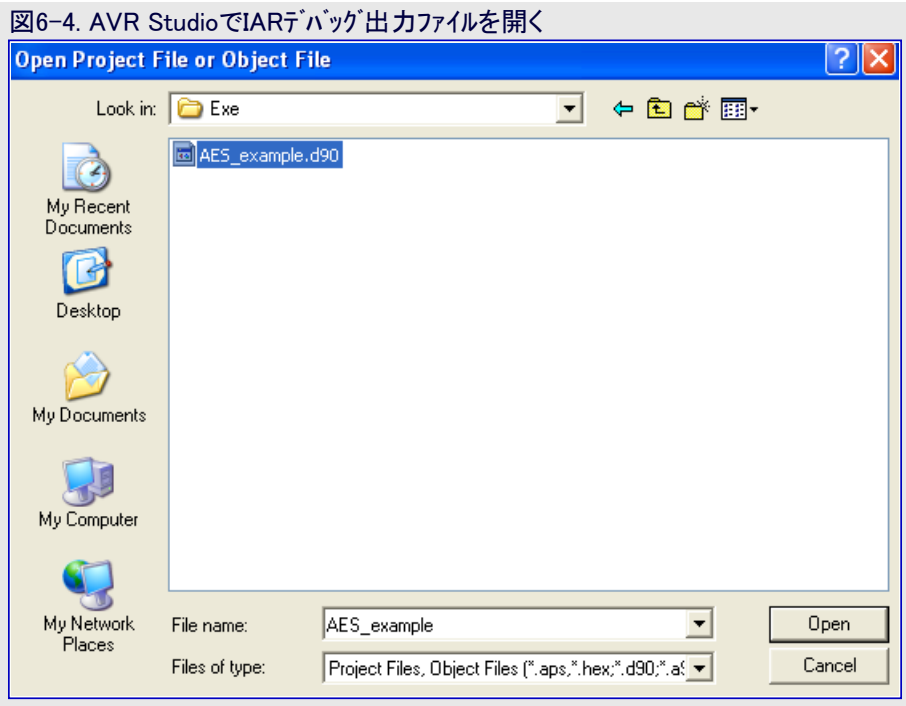
- 希望する位置<project location>にプロジェクトを保存してください。
- プロジェクト任意選択を開いてください(Project(プロジェクト)⇒Options(任意選択)またはAlt+F7)。
- **General Options**(全般任意選択)Category:(分野)で**Target**(目的対象)タブを選択して、下図で示されるように**Processor configuration**(プロセッサ構成設定)を**ATxmega128A1**に設定してください。



- I/O空間に於けるビット定義を許可するために、**System**(システム)タブを開いて“**Enable bit definitions in I/O-Include files**(インクルードファイルからのI/Oビット定義許可)”チェック枠をチェックしてください。



- I/Oビット宣言を使うであろうユーザーのファイルで`ioavr.h`をインクルードしてください。
- “Compile(コンパイル)” 釦押下、またはProject(プロジェクト)メニューからCompile(コンパイル)を選択、または単にCtrl+F7押下でコードをコンパイルしてください。
- “Make(作成)” 釦押下、またはProjectメニューからMake(作成)を選択、または単にF7を押下することによってAVR Studio用の出力ファイルを作成してください。
- プログラムのデバッグに関しては5章で記述されるような接続を行ってください。AVR Studioを開いて始動ダイログで“Open(開く)” 釦をクリックしてください。IARでプロジェクトを保存した位置へ行ってください。プロジェクトフォルダに於いて、デバッグフォルダを“Exe”フォルダ内に位置を定めてください。この“Exe”フォルダはAVR Studioに対するIARのデバッグ出力ファイルであるd90ファイルを含んでいます。このd90ファイルをAVR Studioで開いてください。



- そしてAVR Studioは命名されて保存される必要があるAVR Studioプロジェクトファイルを生成します。
- 下図で示されるようにデバッグ基盤としてJTAGICEmk IIを、デバイスとしてATxmega128A1を選んでください。



これが行われると、デバッグ作業が開始されます。AVR StudioとAVR Studioでのデバッグについてのより多くの情報は“AVR Studio使用者の手引き”(Help(ヘルプ)⇒AVR Studio User Guide(AVR Studio使用者の手引き))で得られます。

## 6.2. WinAVRとAVR Studioでのコンパイルとデバッグ

WinAVRとAVR StudioでATxmega128A1に対するプロジェクトを始めるには以下で記述されるように行ってください。

- AVR Studioを開いてください。
- メニューバーから**Project**(プロジェクト)⇒**New Project**(新規プロジェクト)を選択してください
- 下図で示されるようにポップアップ ウィンドウで**AVR GCC**を選択し、**Project name:**(プロジェクト名)と**Initial file:**(初期ファイル名)を入力してください。



- **Next>>**(次へ)>>)を押下し、**Debug platform:**(デバッグ基盤)で**JTAGICEmk II**を、**Device:**(デバイス)として**ATxmega128A1**を選択してください。



- **Finish**(終了)を押下してください。するとプロジェクトが作成されます。

- ・メニューバーから“Project(プロジェクト)⇒Configuration(構成設定)”を選択してください。
- ・General(全般)構成設定ウィンドウに於いて、Device:(デバイス)任意選択で“atxmega128a1”を選び、望むOptimization:(最適化レベル)を選んで“OK”鈕を押下してください。



今やプロジェクトが初期設定され、Cファイルでプログラムを書くことができます。プログラムをコンパイルしてデバイスでそれをデバッグするには、5.章で記述されるのと同じ接続にしてください。プログラムのコンパイルは以下のように行われ得ます。

- ・活性(有効)なファイルだけをコンパイルするには、メニューバーから“Build(構築)⇒Compile(コンパイル)”を選択するか、または“Compile(コンパイル)”鈕押下、または単に“Alt+F7”を押下してください。
- ・プロジェクト全体を作成するには、メニューバーから“Build(構築)⇒Build(構築)”を選択するか、または“Build(構築)”鈕押下、または単に“F7”を押下してください。
- ・今やプロジェクトが構築され、メニューバーから“Debug(デバッグ)⇒Run(走行)”を選択するか、または“Run(走行)”鈕押下、または単に“F5”を押下することによってチップ上でコードをデバッグすることができます。
- ・直前の2つの段階はメニューバーから“Build(構築)⇒Build and Run(構築して走行)”を選択するか、または“Build and Run(構築して走行)”鈕押下、または単に“Ctrl+F7”を押下することによって同時に行うことができます。

デバッグ作業が今や開始され、チップ上でプログラムをデバッグすることができます。デバッグについてのより多くの情報に関してはAVR Studioで“Help(ヘルプ)⇒AVR Tools User Guide(AVRツール使用者の手引き)”を読んでください。

## 7. 更なる情報

ATxmega128A1デバイスについてのもっと詳細な情報に関してはAtmel®のウェブサイトで利用可能なデータシートとXMEGA A手引書を読んでください。

- ATxmega128A1データシート
  - [http://www.atmel.com/dyn/resources/prod\\_documents/doc8067.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc8067.pdf)
- ATxmega A手引書
  - [http://www.atmel.com/dyn/resources/prod\\_documents/doc8077.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc8077.pdf)
- XMEGA応用記述
  - [http://www.atmel.com/dyn/products/app\\_notes.asp?family\\_id=607#XMEGA%20Microcontrollers](http://www.atmel.com/dyn/products/app_notes.asp?family_id=607#XMEGA%20Microcontrollers)

開発に必要なツールはここで得られます。

- AVR Studio
  - [http://www.atmel.com/dyn/products/tools\\_card.asp?tool\\_id=2725](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725)
- IAR Embedded Workbench(評価版または完全版)
  - <http://www.iar.se/>
  - <http://supp.iar.com/Download/SW/?item=EWAVR-EVAL>
- WinAVR
  - <http://winavr.sourceforge.net/>
  - [http://sourceforge.net/project/showfiles.php?group\\_id=68108](http://sourceforge.net/project/showfiles.php?group_id=68108)



## 本社

### *Atmel Corporation*

2325 Orchard Parkway  
San Jose, CA 95131  
USA  
TEL 1(408) 441-0311  
FAX 1(408) 487-2600

## 国外営業拠点

### *Atmel Asia*

Unit 1-5 & 16, 19/F  
BEA Tower, Millennium City 5  
418 Kwun Tong Road  
Kwun Tong, Kowloon  
Hong Kong  
TEL (852) 2245-6100  
FAX (852) 2722-1369

### *Atmel Europe*

Le Krebs  
8, Rue Jean-Pierre Timbaud  
BP 309  
78054 Saint-Quentin-en-  
Yvelines Cedex  
France  
TEL (33) 1-30-60-70-00  
FAX (33) 1-30-60-71-11

### *Atmel Japan*

104-0033 東京都中央区  
新川1-24-8  
東熱新川ビル 9F  
アトメル ジャパン株式会社  
TEL (81) 03-3523-3551  
FAX (81) 03-3523-7581

## 製品窓口

### ウェブサイト

[www.atmel.com](http://www.atmel.com)

### 技術支援

[avr@atmel.com](mailto:avr@atmel.com)

### 販売窓口

[www.atmel.com/contacts](http://www.atmel.com/contacts)

### 文献請求

[www.atmel.com/literature](http://www.atmel.com/literature)

お断り: 本資料内の情報はAtmel製品と関連して提供されています。本資料またはAtmel製品の販売と関連して承諾される何れの知的所有権も禁反言あるいはその逆によって明示的または暗示的に承諾されるものではありません。Atmelのウェブサイトに位置する販売の条件とAtmelの定義での詳しい説明を除いて、商品性、特定目的に関する適合性、または適法性の暗黙保証に制限せず、Atmelはそれらを含むその製品に関連する暗示的、明示的または法令による如何なる保証も否認し、何ら責任がないと認識します。たとえばAtmelがそのような損害賠償の可能性を進言されたとしても、本資料を使用できない、または使用以外で発生する(情報の損失、事業中断、または利益の損失に関する制限なしの損害賠償を含み)直接、間接、必然、偶然、特別、または付随して起こる如何なる損害賠償に対しても決してAtmelに責任がないでしょう。Atmelは本資料の内容の正確さまたは完全性に関して断言または保証を行わず、予告なしでいつでも製品内容と仕様の変更を行う権利を保留します。Atmelはここに含まれた情報を更新することに対してどんな公約も行いません。特に別の方法で提供されなければ、Atmel製品は車載応用に対して適当ではなく、使用されるべきではありません。Atmel製品は延命または生命維持を意図した応用での部品としての使用に対して意図、認定、または保証されません。

© Atmel Corporation 2008. 不許複製 Atmel<sup>®</sup>、ロコとそれらの組み合わせ、AVR<sup>®</sup>とその他はAtmel Corporationの登録商標、XMEGA<sup>®</sup>とその他は商標またはその付属物です。他の用語と製品名は一般的に他の商標です。

© HERO 2021.

本応用記述はAtmelのAVR1900応用記述(doc8107.pdf Rev.8107A-04/08)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意識されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には( )内に英語表記や略称などを残す形で表記しています。

青字の部分はリンクとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。