応用記述

AT03160：ATxmega128A1から他のAtmel XEMGAﾃﾞﾊﾞｲｽへのﾌﾞｰﾄﾛｰﾀﾞ移植

Atmel AVR XMEGA

要点

- Atmel® AVR® XMEGA®ﾌﾞｰﾄﾛｰﾀﾞ
- 手順応用
- 自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ用Cｺｰﾄﾞ試供応用
- ﾌﾗｯｼｭ ﾒﾓﾘとEEPROMの両ﾒﾓﾘの読み書き
- 施錠ﾋﾞｯﾄ読み書き
- ﾋｭｰｽﾞ ﾋﾞｯﾄ読み込み

説明

多くの電機設計が急速に進化するため、既に出荷または販売された製品を更新できることが必要になります。ﾏｲｸﾛ ｺﾝﾄﾛｰﾗに組み込まれたﾌﾞｰﾄﾛｰﾀﾞ ﾌｧｰﾑｳｪｱは書き込み器の必要なしに応用ﾌﾗｯｼｭ更新を容易にすることができます。

この応用記述はAtmel ATxmega128A1から他のAVR XMEGAﾃﾞﾊﾞｲｽへﾌﾞｰﾄﾛｰﾀﾞを段階的に移植する方法を記述します。この応用記述がAVR1605用に拡張された読み物のため、この応用記述を読む前に、AVR1605:XMEGAﾌﾞｰﾄﾛｰﾀﾞ即時開始の手引きを読むべきです。AVR1605のﾌﾟﾛｼﾞｪｸﾄはATxmega128A1に基づき、ﾌﾟﾛｼﾞｪｸﾄ一括はAtmelｳｪﾌﾞｻｲﾄからﾀﾞｳﾝﾛｰﾄﾞすることができます。

自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞについてのより多くの情報に関してはAVR109:自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ応用記述を参照してください。

Rev. 42143A-07/2013, 42143AJ2-05/2021

目次

1. 手順

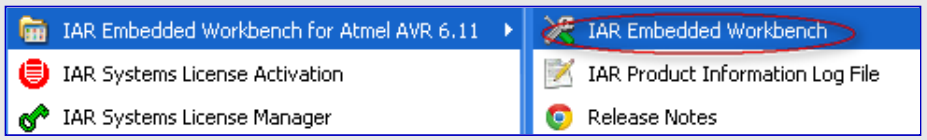
ここはATxmega128A1から他のXMEGAﾃﾞﾊﾞｲｽへﾌﾞｰﾄﾛｰﾀﾞを移植するための主な手順です。手続きに従って移植を行う前にATxmega128A1に基づく元のﾌﾟﾛｼﾞｪｸﾄがﾘﾝｸのhttp://www.atmel.com/Images/AVR1605.zipを通してﾀﾞｳﾝﾛｰﾄﾞされるべきです。

移植手続きを開始する前にIAR Embedded Workbench 6.11版またはそれ以降が最初にｲﾝｽﾄｰﾙされるべきです。ﾌﾞｰﾄﾛｰﾀﾞがAtmel Studioでﾌﾟﾛｸﾞﾗﾐﾝｸﾞされる場合、任意選択のAtmel Studio 6.0版またはそれ以降がｲﾝｽﾄｰﾙされます。

1.1. 手順1 – IAR開始

Windowsのｽﾀｰﾄ⇒ﾌﾟﾛｸﾞﾗﾑ⇒IAR Systems⇒IAR Embedded Workbench for Atmel AVR⇒IAR Embedded Workbenchをｸﾘｯｸしてください。

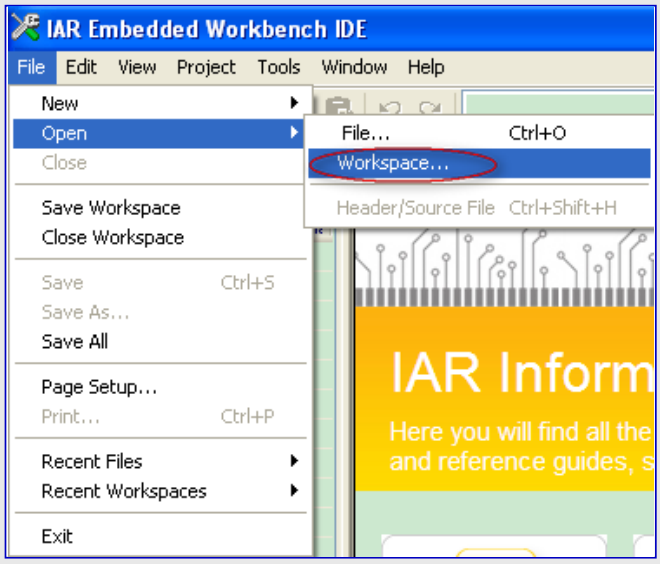
図1-1. IAR開始



1.2. 手順2 – ﾌﾟﾛｼﾞｪｸﾄを開く

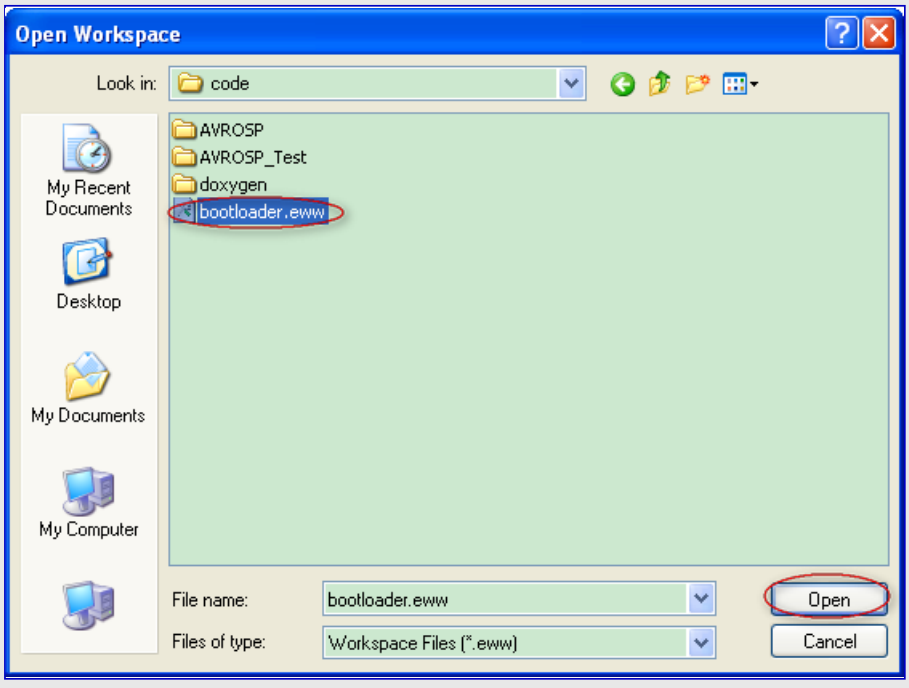
File(ﾌｧｲﾙ)⇒Open(開く)⇒Workspace(作業空間)をｸﾘｯｸしてください。

図1-2. 作業空間を開く



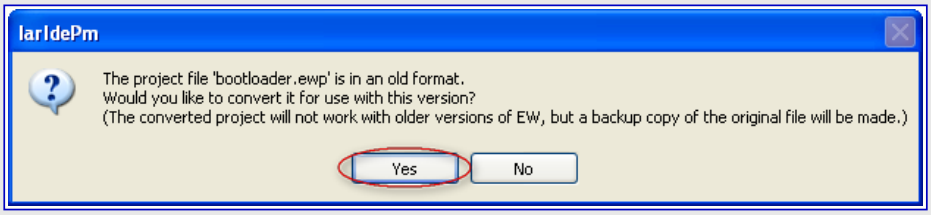
ﾌﾟﾛｼﾞｪｸﾄ ﾌｫﾙﾀﾞを閲覧して”bootloader.eww” ﾌﾟﾛｼﾞｪｸﾄ ﾌｧｲﾙを選択し、そして”Open(開く)” 釦をｸﾘｯｸしてください。

図1-3. bootloaderを開く



使ったIARソフトウェアの版がより新しい場合、変換指示が発生するでしょう。続けるには”Yes”をクリックしてください。

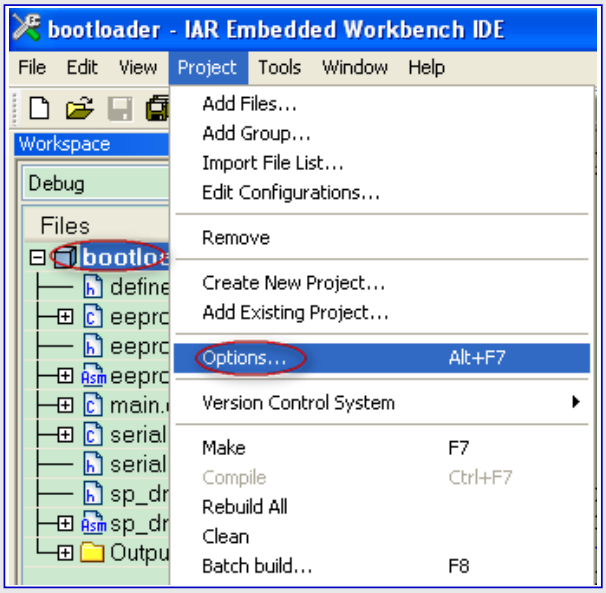
図1-4. 変換指示



1.3. 手順3 – プロセッサとxclファイルの変更

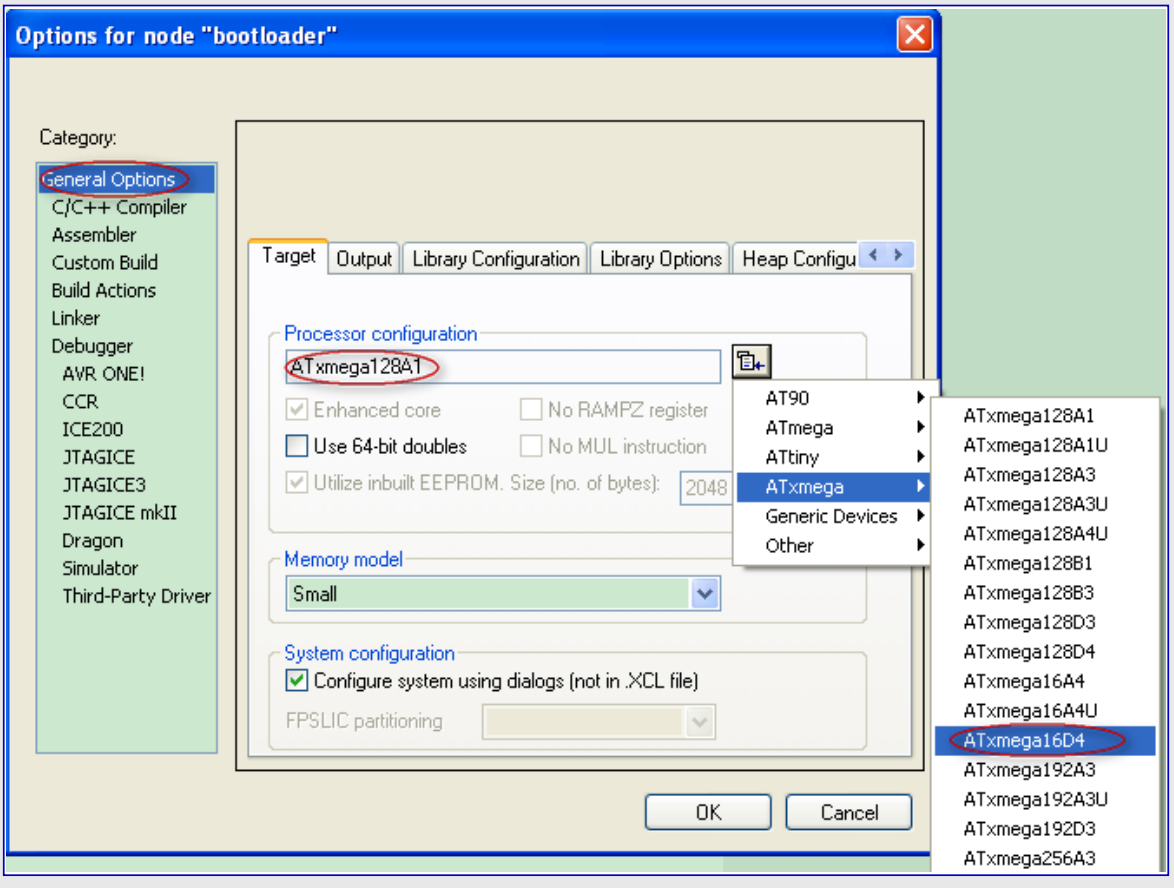
作業空間内でbootloader – Debugを選択し、そしてProject(プロジェクト)⇒Options(任意選択)をクリックしてください。

図1-5. 任意選択(Options)を開く



Category(区分)でGeneral Optionsそしてその後にTarget(目的対象)タブを選択し、ATxmega128A1を目的対象デバイスに変更してください。

図1-6. プロセッサ変更



¥codeフォルダ内で"link_bootloader_ATxmega128A1.xcl"リンカ構成設定ファイルを見つけてください。そして目的対象のリンカ構成設定ファイルを得るためにそれから複写してそれを目的対象デバイス名で改名してください。この目的対象デバイス.xclファイルに於いて、'Segments in program address space (internal Flash memory)「プログラム アドレス空間内のセグメント(内部フラッシュ メモリ)」'、'Data (SRAM, external ROM or external NV RAM) memory「データ(SRAM、外部ROMまたは外部NV RAM)メモリ」'、'Internal EEPROM「内部EEPROM」'部分を目的対象デバイスのデータシートに従って適切な値に変更してください。これらの詳細について対応デバイスに対するIARからのリンカファイルを参照することもできます。IARのリンカ ファイルは¥IAR Systems¥Embedded Workbench 6.0¥avr¥src¥templateフォルダで利用可能です。

図1-7. xclファイル変更

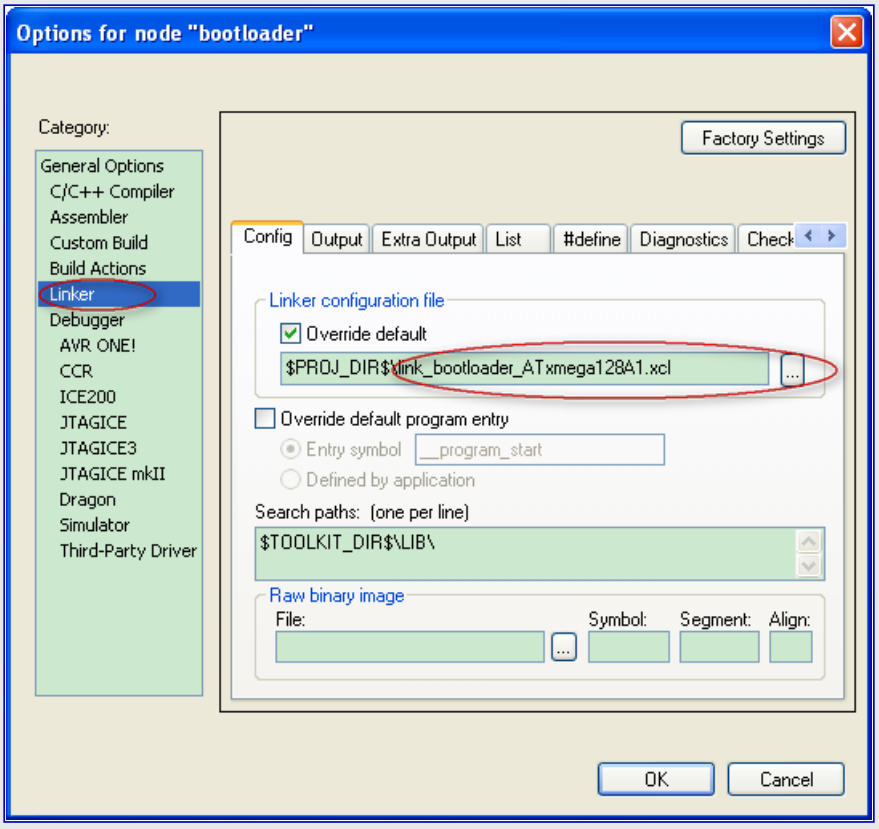
```
/*************************************************************/
/* Segments in program address space (internal Flash memory) */
/*************************************************************/
-D_..X_INTVEC_SIZE=1F4      // 4バイト×125ベクタ
-D_..X_APPLICATION_SECTION_START=20000
-D_..X_APPLICATION_SECTION_SIZE=2000
-D_..X_APPLICATION_START=(_..X_APPLICATION_SECTION_START+_..X_INTVEC_SIZE)
-D_..X_APPLICATION_END=(_..X_APPLICATION_SECTION_START+_..X_APPLICATION_SECTION_SIZE-1)

/////////////////////////////////////////////////////////////
// Data (SRAM, external ROM or external NV RAM) memory
-D..X_SRAM_BASE=2000        // RAMメモリの開始
-D..X_SRAM_TBASE=0          // Tiny RAMメモリの開始
-D..X_SRAM_TSIZE=0          // Tiny RAMメモリの量
-D..X_SRAM_END=3FFF         // RAMメモリの最後
//-D..X_CSTACK_BASE=_..X_SRAM_BASE
//-D..X_CSTACK_END=_..X_SRAM_END
//-D..X_RSTACK_BASE=_..X_SRAM_BASE
//-D..X_RSTACK_END=_..X_SRAM_END

/////////////////////////////////////////////////////////////
// Internal EEPROM
-D..X_EEPROM_END=7FF        // EEPROMメモリの最後
-D..X_EEPROM_START=0
```

Category(区分)でLinker(リンカ)そしてその後にConfig(構成設定)タブを選び、link_bootloader_ATxmega128A1.xclを変更した目的対象ファイルに変更してください。

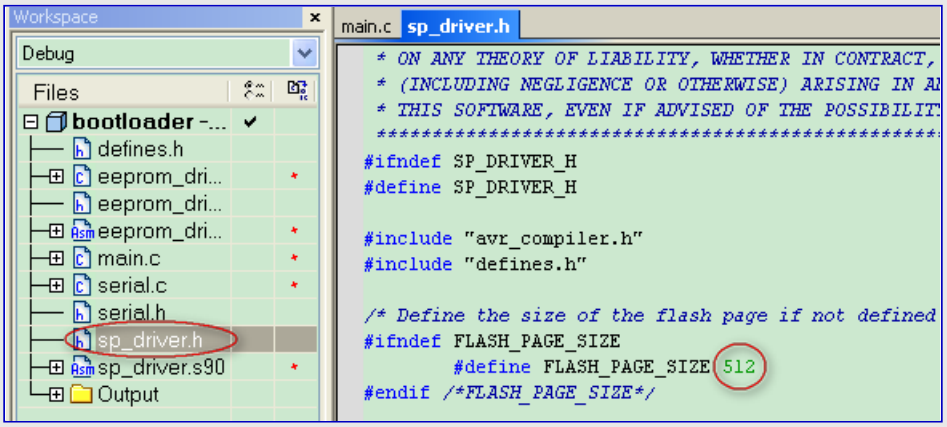
図1-8. xclファイル選択



1.4. 手順4 – フラッシュ ページ容量変更

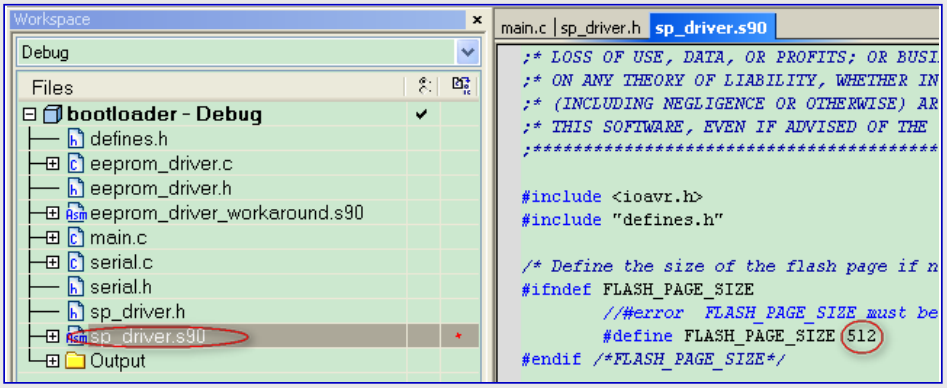
作業空間内でsp_driver.hファイルを選択してFLASH _PAGE_SIZE 512を目的対象デバイスに対応する正しい値に変更してください。フラッシュ ページ容量はデータシートで得ることができます。

図1-9. sp_driver.hでのフラッシュ ページ容量の変更



作業空間内でsp_driver.s90ファイルを選択してFLASH _PAGE_SIZE 512を目的対象デバイスに依存する正しい値に変更してください。

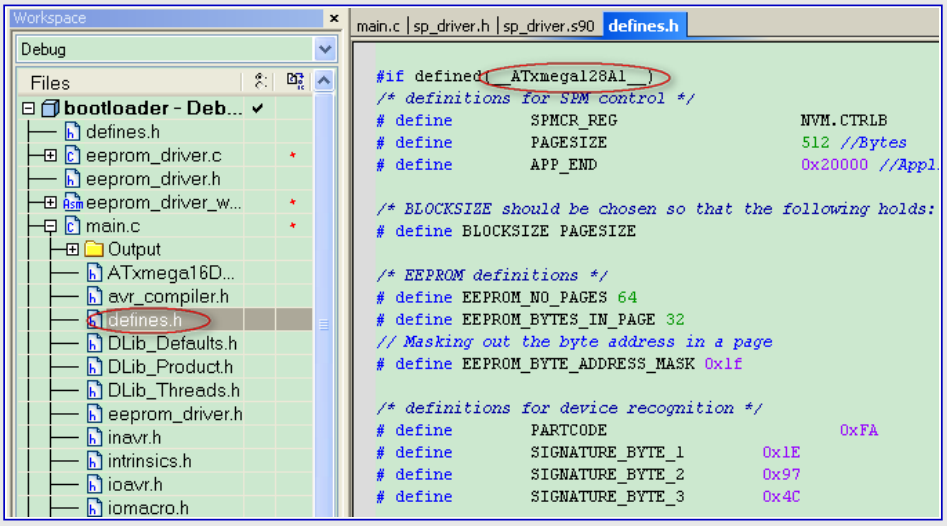
図1-10. sp_driver.s90でのフラッシュ ページ容量の変更



1.5. 手順5 – 定義情報変更

作業空間内でdefines.hファイルを選択してください。_ATxmega128A1_を目的対象デバイスに変更し、その後に予め定義されている全ての項目を目的対象デバイスと一致するように変更してください。満たされた全ての情報はデータシートで得ることができます。

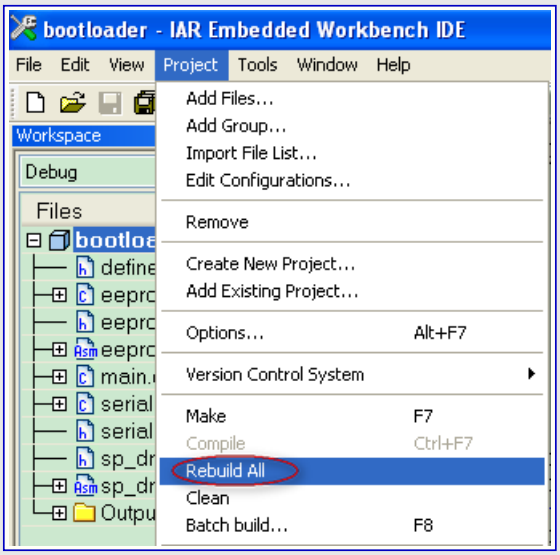
図1-11. 予め定義されている情報の変更



1.6. 手順6 – プロジェクトのコンパイル

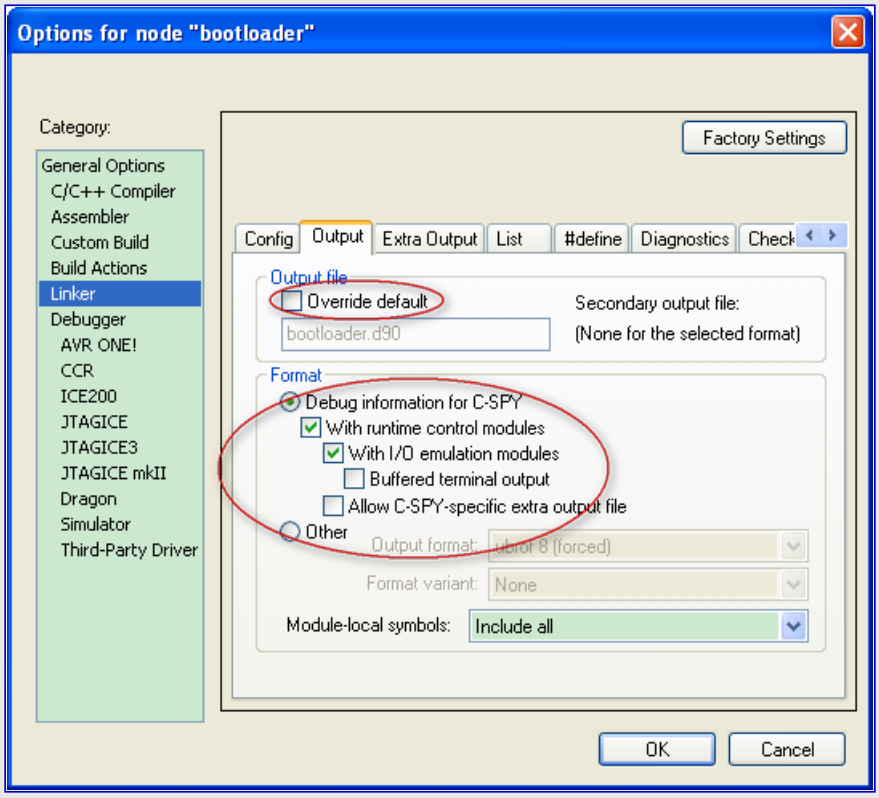
全てが正しく処理されたなら、プロジェクトは成功裏にコンパイルできます。Project(プロジェクト)⇒Rebuild All(全て再構築)をクリックしてください。すると¥Debug¥Exeフォルダ内に新しいデバッグファイルの"bootldr.dbg"が作成されるでしょう。このデバッグファイルはAtmel Studioでブートローダをデバッグするのに使われます。

図1-12. プロジェクトの再構築



ブートローダがIAR Embedded Workbenchでデバッグされる場合、出力形式は図1-13.で示されるように変更されるべきです。設定後、Project(プロジェクト)⇒Rebuild All(全て再構築)をクリックしてください。bootloader.d90が生成されるでしょう。このデバッグファイルはブートローダをIAR Embedded Workbenchでデバッグするのに使われます。

図1-13. 出力形式



1.7. 手順7 – ﾌﾞｰﾄﾛｰﾀﾞのﾃﾞﾊﾞｯｸﾞ

ﾌﾞｰﾄﾛｰﾀﾞはIAR Embedded Workbenchでﾃﾞﾊﾞｯｸﾞすることができます。図1-14.で示されるようにSetup(準備設定)ﾀﾌﾞで使うﾃﾞﾊﾞｯｶﾞを選んでください。

図1-14. ツール選択

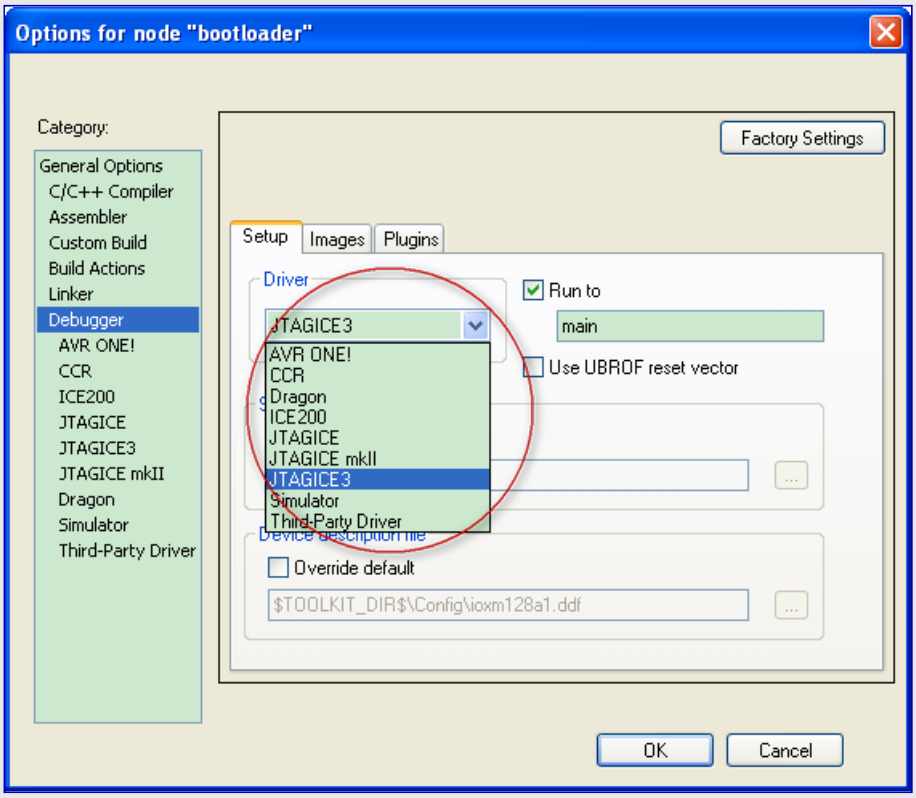
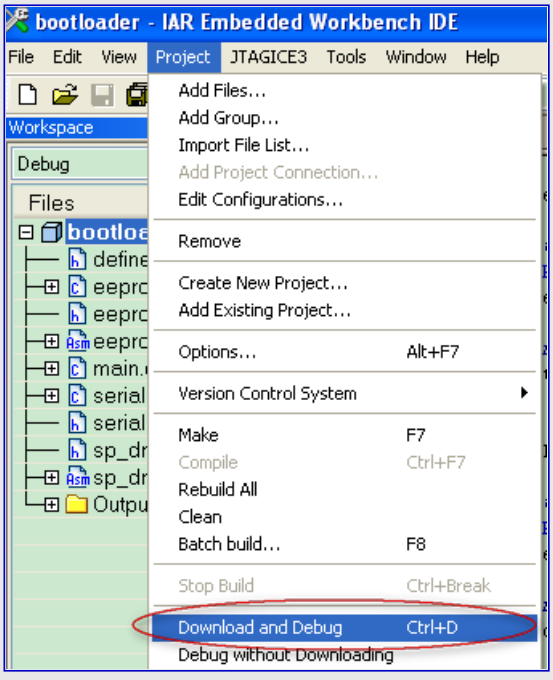


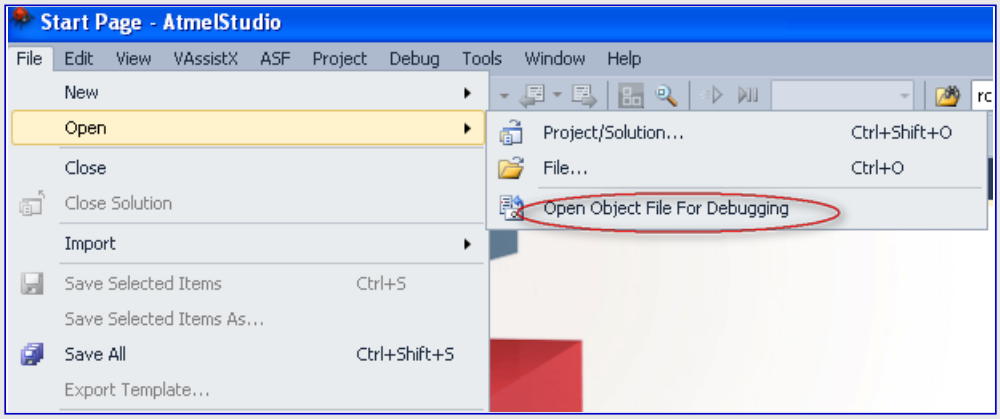
図1-15.で示されるようにProject(ﾌﾟﾛｼﾞｪｸﾄ)⇒Download and Debug(書き込んでﾃﾞﾊﾞｯｸﾞ)をｸﾘｯｸしてください。そしてこれは目的対象ﾃﾞﾊﾞｲｽに対するﾌﾞｰﾄﾛｰﾀﾞをﾃﾞﾊﾞｯｸﾞします。ﾀﾞｳﾝﾛｰﾄﾞ(書き込み)後、ﾌﾞｰﾄﾛｰﾀﾞのﾃﾞﾊﾞｯｸﾞを開始することができます。

図1-15. ﾀﾞｳﾝﾛｰﾄﾞ(書き込み)とﾃﾞﾊﾞｯｸﾞ



代わりに、bootldr.dbgファイルはAtmel Studio 6でデバッグされるブートローダを作ることができます。Atmel Studio 6を開いた後、File(ファイル)⇒Open(開く)⇒Open Object File For Debugging(デバッグ用にオブジェクト ファイルを開く)をクリックし、そしてbootldr.dbgファイルを開くために¥Debug¥Exeフォルダを閲覧してください。オブジェクト ファイルでデバッグする方法のより多くの情報についてはAtmel Studioヘルプ ファイルを参照してください。

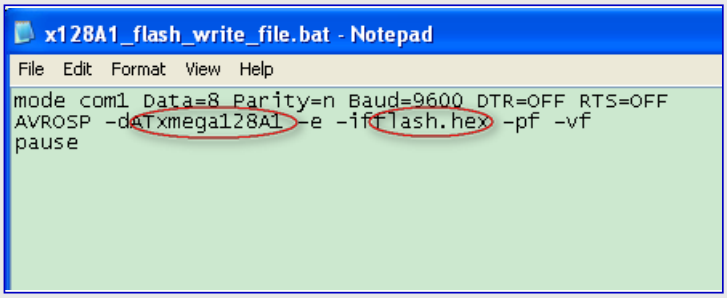
図1-16. オブジェクト ファイルを開く



1.8. 手順8 – bat(バッチ)ファイルの変更

¥AVROSP_Testフォルダを閲覧し、ノートパッド(Notepad)でbatファイルを開いてください。ATxmega128A1を目的対象デバイスに変更してください。写像ファイル(.hex)はbatファイルと同じフォルダに置かれるべきです。他のbatファイルはそれらが使われるならば同じ方法で変更されるべきです。

図1-17. batファイルの変更



2. 推奨読み物

自己プログラミングとブートローダについての全体的な知識を得るために以下の応用記述を読むことが推奨されます。

- AVR109: 自己プログラミング – この応用記述はSPM命令を持つデバイスが自己プログラミングに対してどう構成設定され得るかを説明します。これがAtmel tinyAVR®とAtmel megaAVR®のデバイスに対して与えられるとは言え、それは自己プログラミングについての全般的な情報を与えます。
- AVR1316: XMEGA自己プログラミング – この応用記述はAtmel AVR XMEGA自己プログラミングの基本的な機能を記述します。
- AVR1622: XMEGA用TWIブートローダ – この応用記述は応用領域を更新するためにXMEGA系統のデバイスのブートローダの使用法と自己プログラミングに対してXMEGAがどう構成設定され得るかを記述します。
- AVR1605: XMEGAブートローダ即時開始の手引き – この応用記述はXMEGA系統デバイスの1つ(換言するとATxmega128A1)でブートローダ応用を使う方法と自己プログラムに対してプログラム メモリ格納(SPM:Store Program Memory)命令を持つAVR®がどう構成設定され得るかを記述します。

3. 改訂履歴

文書改訂	日付	注釈
42153A	2013年7月	初版

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation
1600 Technology Drive
San Jose, CA 95110
USA
TEL (+1)(408) 441-0311
FAX (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
TEL (+852) 2245-6100
FAX (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parking 4
D-85748 Garching b. Munich
GERMANY
TEL (+49) 89-31970-0
FAX (+49) 89-3194621

Atmel Japan G.K.
141-0032 東京都品川区
大崎1-6-4
新大崎勧業ﾋﾞﾙ 16F
ｱﾄﾒﾙ ｼﾞｬﾊﾟﾝ合同会社
TEL (+81)(3)-6417-0300
FAX (+81)(3)-6417-0370

本応用記述はAtmelのAT03160応用記述(Rev.42153A-07/2013)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訳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。