
AT03160 : ATxmega128A1から他のAtmel XMEGAデバイスへの ブートローダ移植

Atmel AVR XMEGA

要点

- Atmel[®] AVR[®] XMEGA[®]ブートローダ
- 手順応用
- 自己プログラミング用コード試供応用
- フラッシュメモリとEEPROMの両メモリの読み書き
- 施錠ビット読み書き
- ヒューズビット読み込み

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ブートローダ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デバイスへブートローダ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ブートローダ即時開始の手引き](#)を読むべきです。AVR1605のプロジェクトはATxmega128A1に基づき、プロジェクト一括は[Atmelウェブサイト](#)からダウンロードすることができます。

自己プログラミングについてのより多くの情報に関しては[AVR109:自己プログラミング](#) 応用記述を参照してください。

目次

1. 手順	3
1.1. 段階1 - IAR開始	3
1.2. 段階2 - プロジェクトを開く	3
1.3. 段階3 - プロセッサとxclファイルの変更	4
1.4. 段階4 - フラッシュ ページ容量変更	6
1.5. 段階5 - 定義情報変更	6
1.6. 段階6 - プロジェクトのコンパイル	7
1.7. 段階7 - ブートローダのデバッグ	8
1.8. 段階8 - bat(バッチ)ファイルの変更	9
2. 推奨読み物	9
3. 改訂履歴	9

1. 手順

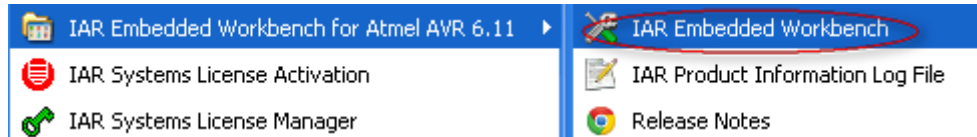
ここはATxmega128A1から他のXMEGAデバイスへブートローダを移植するための主な手順です。手続きに従って移植を行う前にATxmega128A1に基づく元のプロジェクトがリンクの<http://www.atmel.com/Images/AVR1605.zip>を通してダウンロードされるべきです。

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。ブートローダがAtmel Studioでプログラミングされる場合、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. 段階1 – IAR開始

Windowsの**スタート⇒プログラム⇒IAR Systems⇒IAR Embedded Workbench for Atmel AVR⇒IAR Embedded Workbench**をクリックしてください。

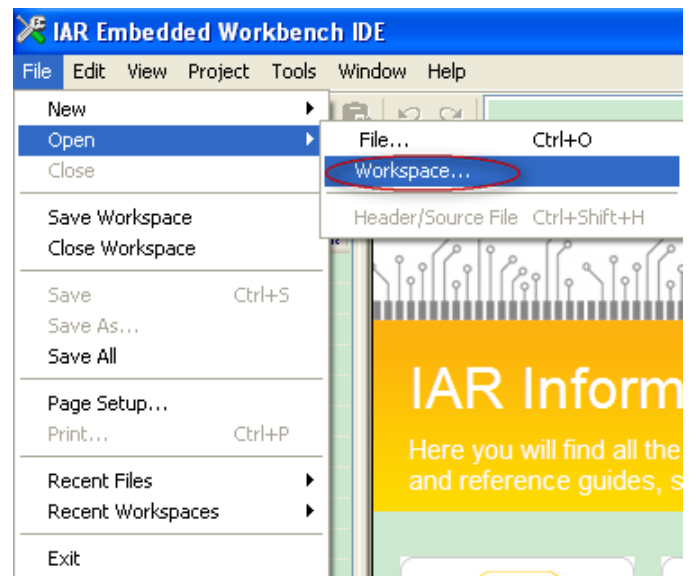
図1-1. IAR開始



1.2. 段階2 – プロジェクトを開く

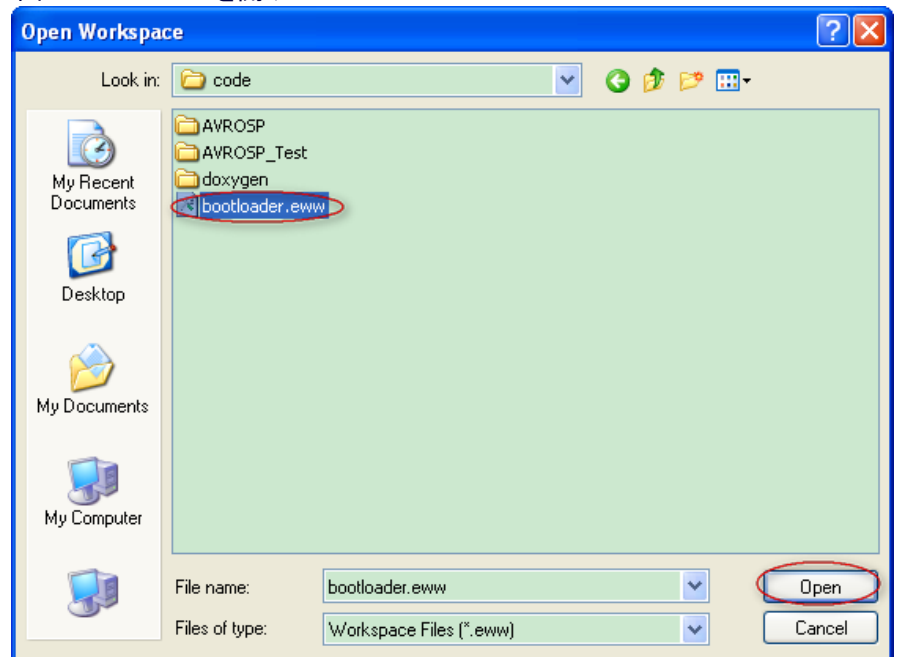
File⇒Open⇒Workspaceをクリックしてください。

図1-2. 作業空間を開く



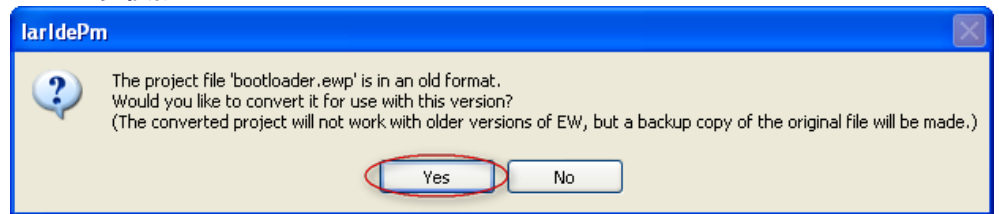
プロジェクトフォルダを閲覧して”bootloader.eww”プロジェクトファイルを選択し、そして”Open”鈕をクリックしてください。

図1-3. bootloaderを開く



使用した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。

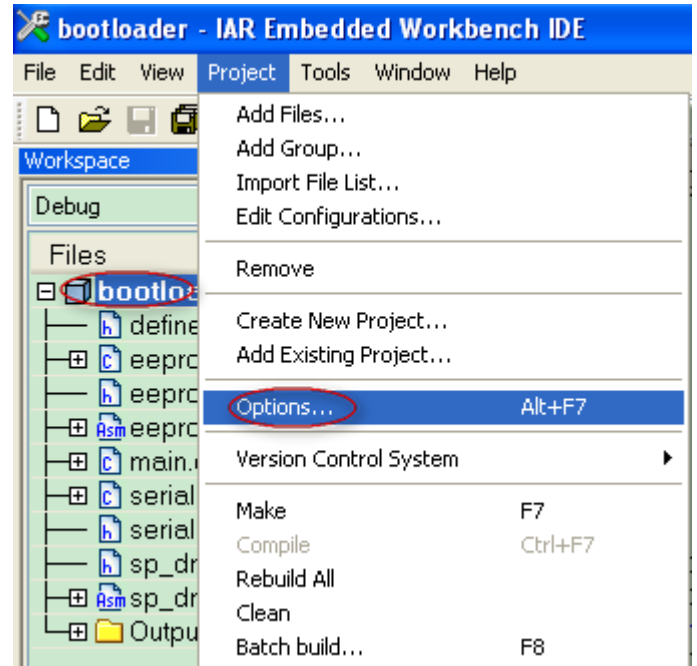
図1-4. 変換指示



1.3. 段階3 – プロセッサとxclファイルの変更

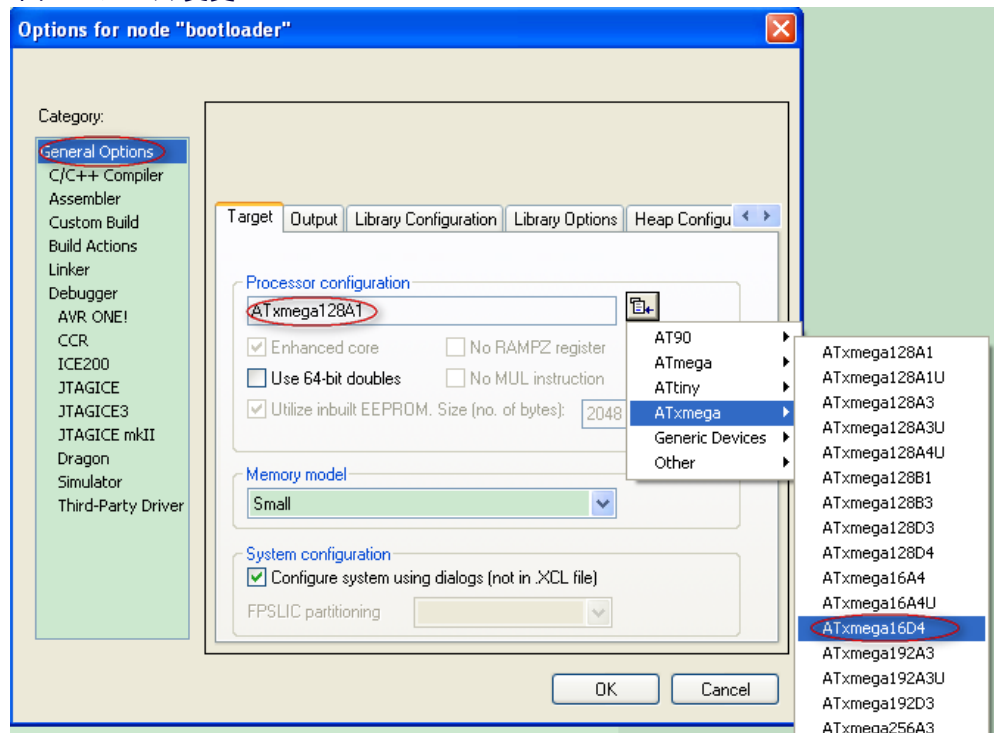
作業空間内でbootloader - Debugを選択し、そしてProject⇒Optionsをクリックしてください。

図1-5. 任意選択(Options)を開く



CategoryでGeneral Optionsそしてその後Target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”リンク形態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ク形態設定ファイルを得るためにそれから複製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からのリンクファイルを参照することもできます。IARのリンクファイルは¥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ヘクタ
-D. .X_APPLICATION_SECTION_START=2000
-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)

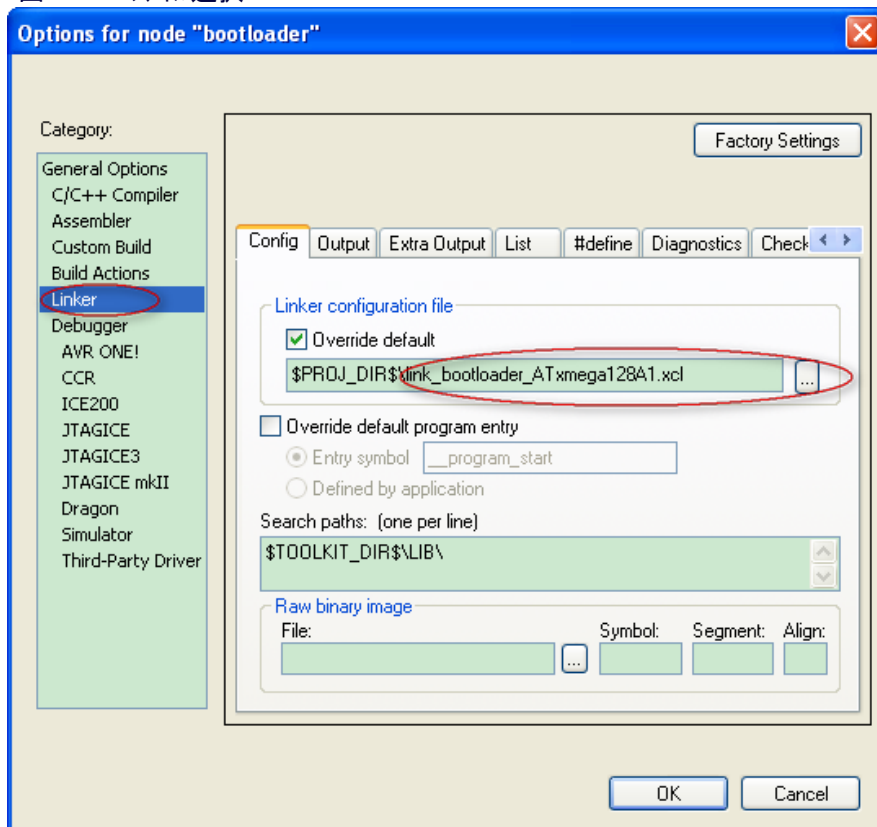
////////////////////////////////////
// Data (SRAM, external ROM or external NV RAM) memory
-D. .X_SRAM_BASE=2000 // RAMメモリの開始
-D. .X_SRAM_TBASE=0 // Tiny RAMメモリの開始
-D. .X_SRAM_TSIZE=0 // Tiny RAMメモリの量
-D. .X_SRAM_END=3FFF // RAMメモリの最後
//-D. .X_CSTACK_BASE=. .X_SRAM_BASE
//-D. .X_CSTACK_END=. .X_SRAM_END
//-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でLinkerそしてその後にConfigタブを選択し、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。

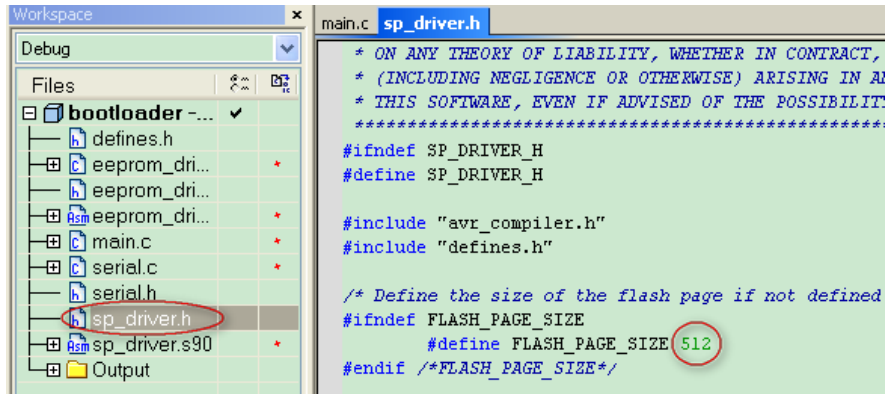
図1-8. xclファイル選択



1.4. 段階4 – フラッシュ ページ容量変更

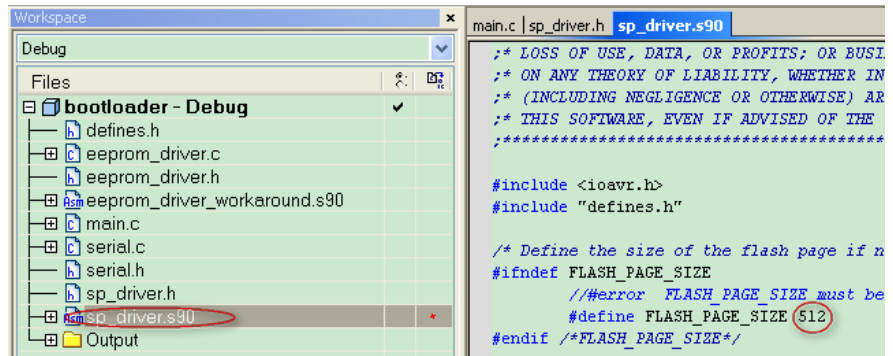
作業空間内でsp_driver.hファイルを選択してFLASH_PAGE_SIZE 512を目的対象デバイスに対応する正しい値に変更してください。フラッシュ ページ容量はデータシートで得ることができます。

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。

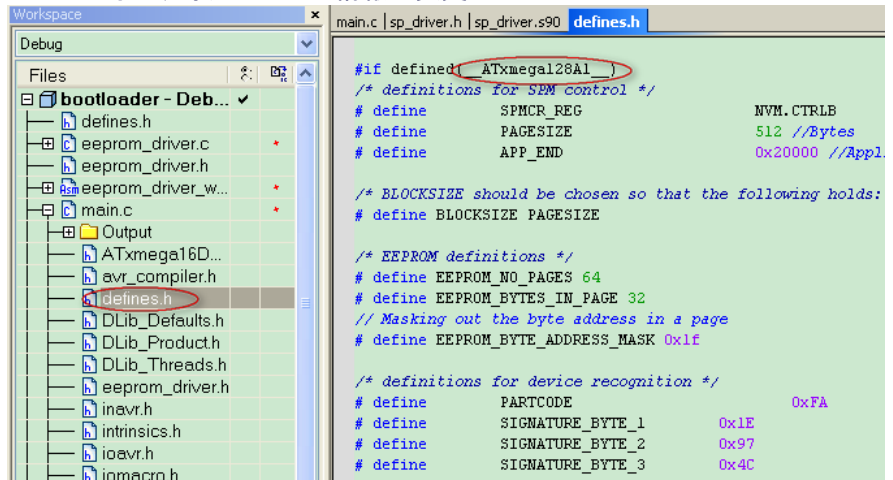
図 1-10. sp_driver.s90でのフラッシュ ページ容量の変更



1.5. 段階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。

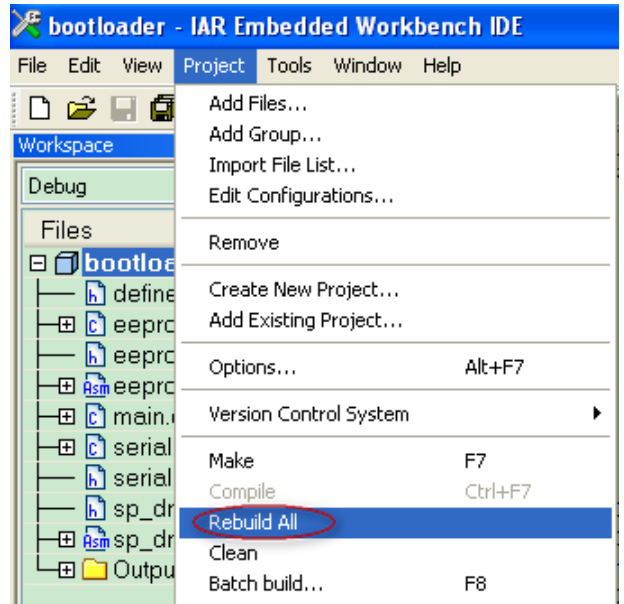
図 1-11. 予め定義されている情報の変更



1.6. 段階6 – プロジェクトのコンパイル

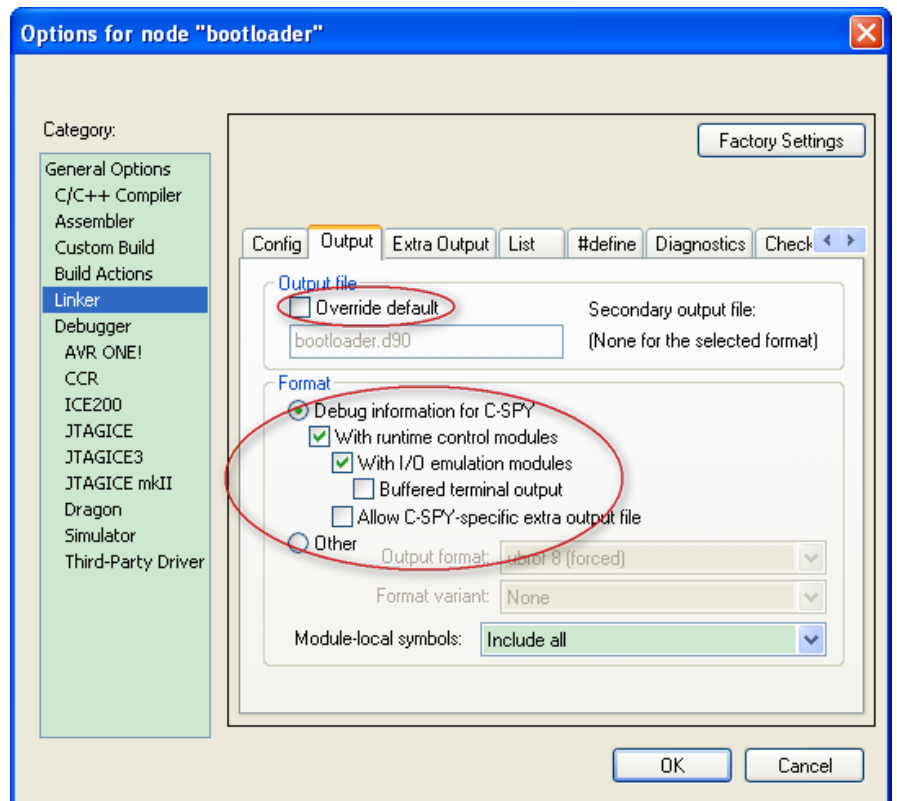
全てが正しく処理されたなら、プロジェクトは成功裏にコンパイルできます。Project⇒Rebuild Allをクリックしてください。すると¥Debug¥Exeフォルダ内に新しいデバッグファイルの”bootldr.dbg”が作成されるでしょう。このデバッグファイルはAtmel Studioでブートローダをデバッグするのに使用されます。

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⇒Rebuild All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用されます。

図1-13. 出力形式



1.7. 段階7 – ブートローダのデバッグ

ブートローダはIAR Embedded Workbenchでデバッグすることができます。図1-14.で示されるようにSetupタブで使用するデバッグを選択してください。

図1-14. ツール選択

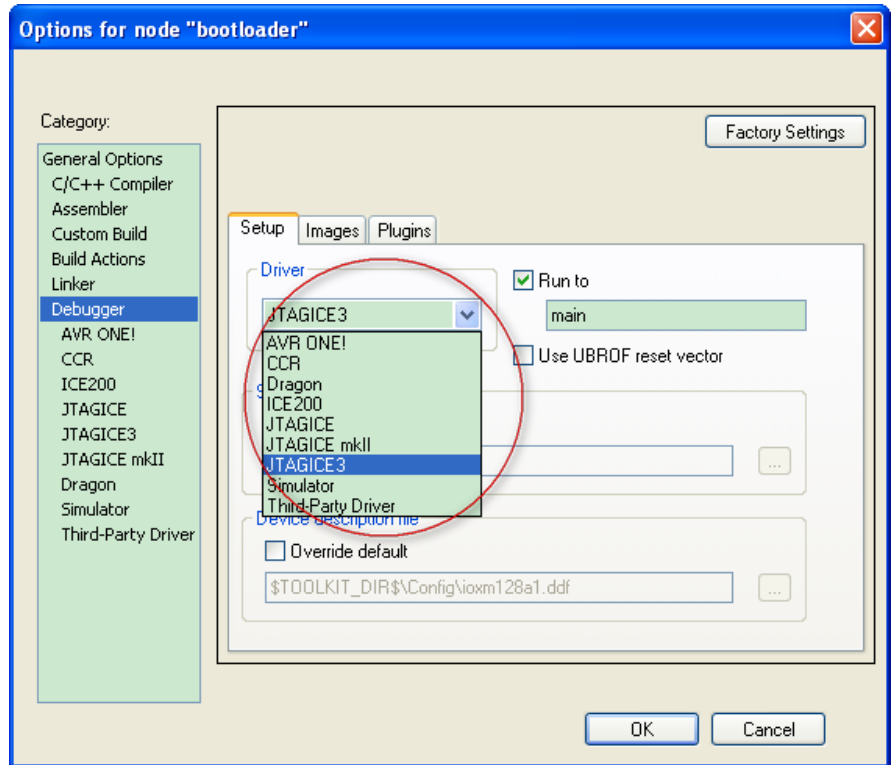
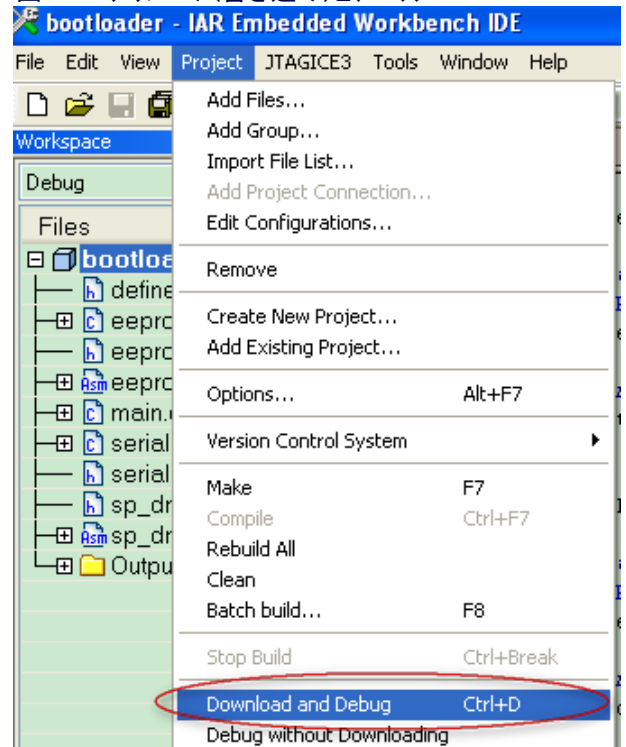


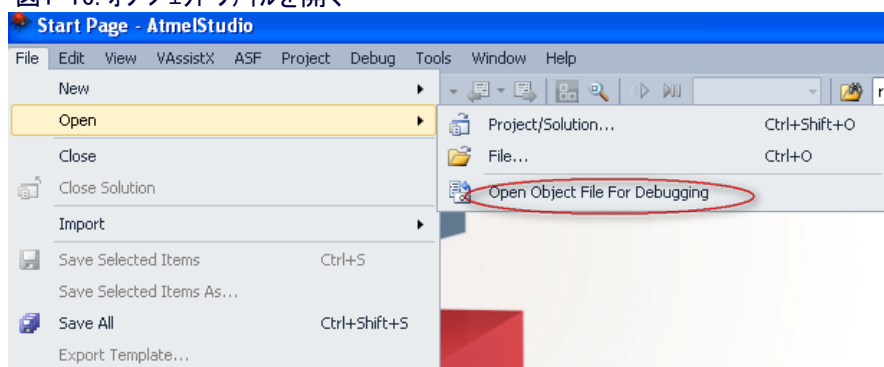
図1-15.で示されるようにProject⇒Download and Debugをクリックしてください。そしてこれは目的対象デバイスに対するブートローダをデバッグします。ダウンロード(書き込み)後、ブートローダのデバッグを開始することができます。

図1-15. ダウンロード(書き込み)とデバッグ



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⇒Open⇒Open Object File For Debugging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。

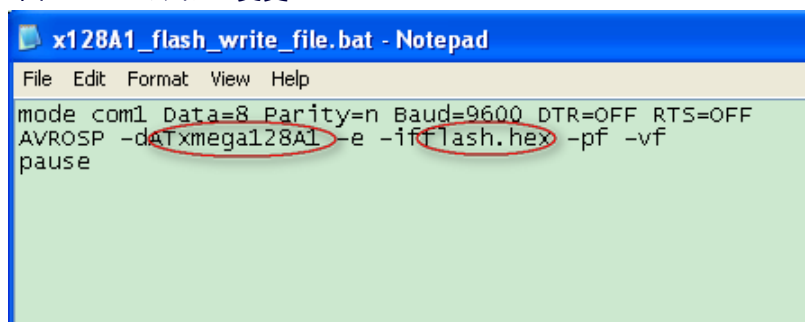
図1-16. オブジェクトファイルを開く



1.8. 段階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用さ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形態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システム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がどう形態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システムデバイスの1つ(換言するとATxmega128A1)でブートローダ応用を使用する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®がどう形態設定され得るかを記述します。

3. 改訂履歴

資料改訂	日付	注釈
42153A	2013年7月	初版



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
新大崎勸業ビル 16F
アトメル ジャパン合同会社
TEL (+81)(3)-6417-0300
FAX (+81)(3)-6417-0370

© 2013 Atmel Corporation. 全権利予約済 / 改訂:42153A-AVR-07/2013

Atmel®, Atmelロゴとそれらの組み合わせ、Enabling Unlimited Possibilities®, AVR®, XMEGA®とその他はAtmel Corporationの登録商標また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。

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、または利益と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。

© HERO 2014.

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。