



AVR1936: XPLORE開始の手引き

Atmel 8ビット マイクロ コントローラ

要点

- XPLORE評価キットを支援
- Atmel® ATxmega32A4(U)/D4のPDIプログラミング
- Atmel Studioまたは/とIAR Embedded Workbench®を使用する実演例演習
- Atmel JTAGICE mk II または3を利用するチップ。上デバック

序説

この資料はXPLORE評価キットを使用してAtmel ATxmega32A4(U)/D4マイクロ コントローラを開始する方法についての情報を含みます。

この資料の最初の4つの章は設計と実装のソフトウェア(Atmel StudioのインストールとAtmel AVR®用IARTM Embedded Workbenchのインストール)、コンパイル、プログラミング、それとファームウェア設計をデバッグでするのに必要とされる実演演習インターフェース ソフトウェアを始める前に(仮想COMホートドライバのインストール/更新)についての情報を含みます。次の2つの章は実演設計例(開始実演例の説明)に対する説明と、Atmel XMEGA®デバイスのチップ・上デバッグ(コンパイルとチップ・上デバッグ)とPDI実装プログラミング(ISP)(XPLORE上のAtmel XMEGAの実装プログラミング)用のXPLORE評価キット基板構成設定方法(ハートウェア構成設定)についての詳細な手順を含みます。プロジェクトを作成してファームウェア設計コードをコンパイルし、Atmel Studio(Atmel Studio内の組み込みコンパイルとデバッグ)とJTAGICE mk II またはJTAGICE 3(IARとAtmel Studioでのコンパイルとデバッグ)を使用してXPLORE評価基板上の設計のデバッグを開始する方法を実演する3つの章もあります。最終章は参照を含みます。

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。

目次

1.	Atmel Studioのインストール・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	3
	Atmel AVR用IAR Embedded Workbenchのインストール ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	
3.	仮想COMポート ドライバのインストール/更新 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	3
4.	Puttyソフトウェアのインストール ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	4
5.	拡張ツールのインストール(任意選択)・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	4
	開始実演例の説明・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	
	ハードウェア構成設定 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	
8.	コンパイルとチップ。上デバッグ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	7
	8.1. Atmel Studio内の組み込みコンパイラでのコンパイルとディック・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	
	8.2. Atmel StudioとIARでのコンパイルとデバッグ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	
	XPLORE上のAtmel XMEGAの実装プログラミング・・・・・・・・・・・・・・・・・1	
10.	参照 · · · · · · · · · · · · · · · · · · ·	
	10.1. オンライン参照資料・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	
	10.2. 設計ソフトウェアと通信装置ドライバ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	17



1. Atmel Studioのインストール

Atmel Studio(6.0版またはそれ以降版)は組み込みアセンブラ、C/C++コンパイラ(本来のGNU、WINAVR GNU)だけでなく、Atmelマイクロコントローラ設計用のシミュレータと前置デバッカも含みます。この資料で記述されない有用な機能もありますが、それらはAtmel Studioソフトウェアのヘルプ・手引書から容易に得られます。Atmel AVRプログラミング・/デバッグ・ツール(JTAGICE mk II またはJTAGICE 3)はXPLORE評価キット基板でAVR XMEGA系統マイクロコントローラの実装プログラミング(ISP)とインサーキットエミュレーション(ICE)を実効するためにAtmel Studioソフトウェアと共に使用することもできます。

XPLORE評価キット基板とで開始するために、AtmelのウェブサイトからAtmel Studioインストーラ ソフトウェアと公開記述PDFをダウンロート してください。ソフトウェア公開記述を読み、その後にインストーラを走らせ、ソフトウェアのインストールを完了するための指示に従ってください。

プロジェクトを開始し、組み込みC/C++コンハプイラを使用してコートでコンハプイルする方法はこの資料の参照章の[10]から得られます。

2. Atmel AVR用IAR Embedded Workbenchのインストール

Atmel AVR用のIAR EWキックスタート版、期間制限版、または完全版を含むIARTM Embedded Workbench 6.11またはそれ以降版は Atmelのマイクロ コントローラを支援する任意選択の第三者のコンパイラとデバック、ツールです。

IAR Embedded Workbenchインストール ソフトウェアのダウンロード・ウェフ・リンクは、この資料の参照章の[11]で得られます。

ソフトウェアのインストールを開始するには、"auto run"応用を選択し、そして"Install IAR Embedded Workbench"を選択してください。

インストール中、許諾期間が過ぎていなければ、IAR Systemsから得た以前の版のIAR Embedded Workbench用と同じ"許諾番号"と"許諾鍵"を使用することが可能です。

何か技術支援またはIARソフトウェア許諾に関連する疑問があれば、IAR Systemsの技術支援にお問い合わせください。

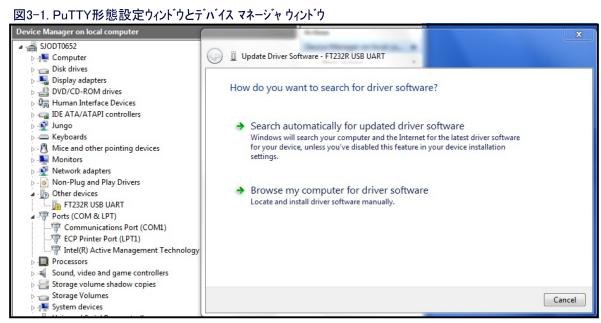
IARのインストール中に既定設定を使用することも推奨されます。IAR EWでプロジェクトを開始してコート・をコンパイルする方法はこの資料の8.2.項で記述されます。

3. 仮想COMポート ドライバのインストール/更新

USBケーブルをXPLORE評価キット基板のUSBコネクタ(J4)からパーソナルコンピュータ(PC)またはラップトップのUSBポートに接続する前に、USB-COM模倣動作を実行するために基板上のFTDIデバイス(FT232RL)に対し、あなたのシステムにFuture Technology Devices Internation al Ltd(FTDI)の仮想COMポート(VCP:Virtual COM Port)ソフトウェアをダウンロート・してインストール/更新することが必要とされます。

Windows®用の"CD x_xx_xx WHQL Certified.zip"のようなVCPソフトウェアトライハ、zipファイル(Microsoft®からの32ビットと64ビットのWindowsオヘ°レーティンク、システム(OS)に対して異なるzipファイルが使用されます。)は、最初にFuture Technology Devices Internation al Ltdのウェブサイトからあなたのシステム(PC/ラップトップ)にダウンロート、することができます。

XPLORE基板からPCまたはラップトップへのUSBケーブル接続後、"CD x_xx_xx WHQL Certified"ディルクトリからソフトウェアトライバを更新するために閲覧することができます。更新/インストールのためにVCPソフトウェアトライバを選ぶように指示するために"新しいハートウェア発見"での "USB接続の自動検出"メッセージが表示されない場合、USBケーブルを外してその後に再びPCのUSBポートに嵌め込み、デスクトップ・ウィントウで"マイコンピュータ"の"プロパティ"設定、または"コンピュータ"から"システムのプロパティ"任意選択の"ハートウェア"タブ・ウィントウから"デバイスマネージャ"を手動で開いてください。次に、"デバイスマネージャ"ウィントウから、図3-1.で示されるように"USB Serial Port"または"FT232R USB UART"設定(いくつかのシステムは"その他のデバイス"設定下で示されます。)の検出を得るために"活動"メニューから"ハートウェア変更を走査"を選択することができます。また、"トライバ・ソフトウェアを更新"任意選択を選ぶためにマウスの"右クリック"を使用し、最後に解凍された "CD x_xx_xx WHQL Certified"ディルクトリからソフトウェアトライバ・ファイルを更新するために、あなたのシステムを通して閲覧してください。ソフトウェアトライバ・更新成功後、"デバイストライバ"ウィントウの"ポート (COMとLPTx)"任意選択下で次の"USB Serial PORT"設定に新しいCOM ポートが表示されます。





異なるWindowsオペレーティング・システムでVCPソフトウェアト・ライハ・を成功裏に更新/インストールする方法についてのより多くの情報に関しては、Future Technology Devices Internation al Ltdのウェブ・サイトからソフトウェアト・ライハ・ダウンロート・指示を再調査してください。

VCPドライバ インストール ソフトウェア ダウンロードのウェブ リンクはこの資料の参照章の[9]から得られます。

4. Puttyソフトウェアのインストール

USB-COM模倣動作データを表示するための端末画面を使用するために、各種オペーレーティング・システムを支援するPuttyソフトウェアのインストールも必要とされます。

Puttyインストール ソフトウェア ダウンロート、のウェブ・リンクはこの資料の参照章の[14]から得られます。

5. 拡張ツールのインストール(任意選択)

Puttyソフトウェアを使用する他にも、使用者は拡張ツールとしてAtmel Studio(6.0版またはそれ以降版)に任意選択の端末ウィントウをインストールすることもできます。以下の手順はインストール手順を記述します。

拡張管理部(Extension Manager)はAtmel Studioソフトウェアへの各種プラク・イン追加を手助けします。

段階1. "Tool"そしてその後に"Extension Manager"任意選択を選ぶことによってAtmel Studioソフトウェアで拡張管理部を開いてください。

段階2. "Extension Manager"ウィント・ウから"Available Downloads(利用可能なダウンロート)"任意選択をクリックしてください。

段階3. ダウンロードするために"Terminal Window Plugin(端末ウィンドウ プラグイン)"任意選択をクリックしてください。

段階4. アカウントまたはサインインを作成してください。

注: 新しいアカウントを作成する場合、アカウントを作成し、Atmel Studio 6.0を再始動して上の手順を繰り返してください。

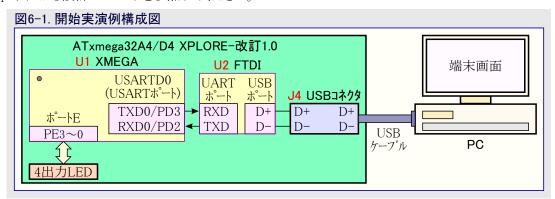
段階5. 一旦拡張がダウンロードされると、下側の状態バーで"You must restart Atmel Studio in order for the change to take effect(変更 が有効になるためにAtmel Studioを再始動しなければなりません。)"が表示されます。

段階6. Atmel Studioを再始動してください。

6. 開始実演例の説明

この開始実演例はAtmelの意義が何かについて使用者に知らせるまたは気付かせるように構成設定され、そしてそれはAtmelが何を表すかでもあります。実際に、それは"PASSION"、"INTEGRITY"、"TEAMWORK"、"ACHIEVMENT"、"ACCOUNTABILITY"の組み合わせです。これの目標はAtmel AVR XMEGAマイクロコントローラのUARTホートからFT232RL(FTDI)デバイスへAtmelの意義メッセージ(ASCII形式文字)を送り、その後にそれらをUSBを通して仮想COMポート(VCP)模倣動作インターフェースへ渡し、最後に送信されたメッセージをパーソナルコンピュータ(PC)またはラップトップの端末画面に表示することによって達成されます。表示されたメッセージはPC/ラップトップの端末画面から"Agree(同意)"を意味するASCII文字'A'を入力するように使用者にも指示し、その後にそれをUSB-UARTインターフェースを通してXMEGAマイクロコントローラへ送り戻し、最後にマイクロコントローラの入出力ポートEを通してXPLORE評価キット基板に置かれたLEDに2進の'A'値(1010: LED3=OFF(1)、LED2=ON(0)、LED1=OFF(1)、LED0=ON(0))を表示します。

図6-1.はこの開始実演設計例に対する主な部品とそれのインターフェース接続を示します。実際の実装についてのより多くの詳細に関してはプロジェクトzipファイルから設計ソース コードを参照してください。



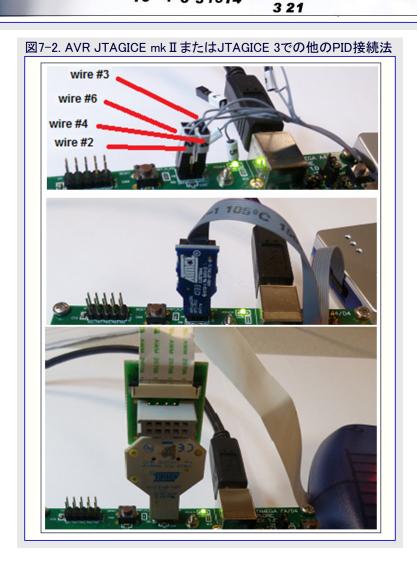
7. ハードウェア構成設定

XPLORE評価キット基板は44ピンVQFP外囲器のAtmel AVR ATxmega32A4、ATxmega32A4U、ATxmega32D4マイクロ コントローラを支援するように設計され、また別の外囲器形式を支援するための容易な変更を許す、44ピンTQFP外囲器の配線パターンと共にやって来ます。現在、既定デバイスとしてXPLORE評価キット基板上にATxmega32A4Uが組み上げられます。

本章は開始実演設計例の演習のためにXPLORE評価キット基板を構成設定する段階的手順を示します。 図7-1.はAVR JTAGICE mk II を使用する、XPLORE評価キット基板上のXMEGAマイクロ コントローラのPDIプログラミングとチップ。上ディブッグの接続を示します。 JTAGICE m k II のPDIコネクタが使用される、またはそれのPDIコネクタか線でJTAGICE 3が使用される場合は図7-2.で示される僅かに異なるPDI ISP /チップ。上ディブッグ接続法を参照してください。



図7-1. それの線接続でAVR JTAGICE mk II を使用する開始実演例ハート・ウェア構成設定 12 16 4 6 5 15 14





ハート・ウェア構成設定手順:

段階1. XPLORE基板のJPL8ヘッタでの左側にジャンパを設定してください。

注: そのような方法でのこのシャンハ。設定はUSBコネクタ(J4)からのVBUSピンを(LDO)低損失直線調整器(U3:LT3022)の入力ピンに接続します。

段階2. JPL9ヘッダの左側にジャンパを設定してください。

注: そのような方法でのこのシャンハ。設定はLDO調整器(U3:LT3022)のLDO_OUTピンをJP2ヘッタ・のVCC網に接続し、そしてこれはJP2ヘッタ・にシャンハ。が設定された時にAtmel XMEGAマイクロコントローラのVCC/AVCCピンに電力を供給するのにも使用されます。

段階3. H7ヘッダの左側にジャンパを設定してください。

注: そのような方法でのこのジャンパ設定はLDO調整器(U3:LT3022)の出力に接続された分圧回路から3.3V出力設定を選びます。

段階4. JP2ヘッダにジャンハ。を設定してください。

注: このシャンパ設定はJPL9ヘッダからのVCC網をJP2ヘッダのU1_VCCに接続します。

段階5. JP0ヘッダにジャンパを設定してください。

注: そのような方法でのこのジャンパ設定はUART通信のためにXMEGAマイクロ コントローラ(U1)のRXD0/PD2ピンをFTDIデバイス (U2:FT232RL)のTXDピンに接続します。

段階6. JP1~ッタ にジャンハ を設定してください。

注: そのような方法でのこのシャンパ設定はUART通信のためにXMEGAマイクロコントローラ(U1)のTXD0/PD3ピンをFTDIデバイス (U2:FT232RL)のRXDピンに接続します。

段階7. JPL0~ッダの左側にジャンパを設定してください。

注: そのような方法でのこのジャンパ設定はXMEGAマイクロ コントローラ(U1)のポートEの0番ピン(PE0)をLED0に接続します。

段階8. IPL1へッダの左側にジャンパを設定してください。

注: そのような方法でのこのジャンパ設定はXMEGAマイクロ コントローラ(U1)のポートEの1番ピン(PE1)をLED1に接続します。

段階9. JPL2ヘッタ・の左側にジャンパを設定してください。

注: そのような方法でのこのジャンパ設定はXMEGAマイクロ コントローラ(U1)のポートEの2番ピン(PE2)をLED2に接続します。

段階10. JPL3へッターの左側にシャンハ。を設定してください。

注: そのような方法でのこのシャンパ設定はXMEGAマイクロコントローラ(U1)のポートEの3番ピン(PE3)をLED3に接続します。

重要な構成設定:

次の段階へ進む前に、FT232RLのVCPソフトウェア ドライバ インストール ファイルをダウンロードし、この資料の前の3.章で案内されたようにあなたのWindowsシステム(OS)にそれをインストールするためにインストールの手引きに従ってください。このソフトウェア インストール段階はWindows PCへのFT232RLデバイスのUSB接続に必要とされます。

段階11. USBバス インターフェースからのVBUS接続を通して基板に通電するために、XPLORE評価キット基板と共に含まれるUSBケーブルをPCのUSBポートからXPLORE評価キット基板のUSBコネクタ(J4)に接続してください。

段階12. 図7-1.で示されるようにJTAGアダプタ基板をAtmel AVR JTAGICE mkⅡコネクタに接続してください。

注: 他のPDI/チップ・上デバッグ接続法が使用される場合、この段階を飛ばすことができますが、**図7-2**.で示されるように適切な接続を構成設定することがきっと必要です。

この開始実演例で作業する前に、Atmel StudioソフトウェアのインストールによってAtmel JTAGICE mk II /3用のUSBソフトウェア (AVR Jungo USB)も正しく構成設定されるべきです。

段階13. USBケーブルをAVR JTAGICE mk II からPCのUSBポートに接続してください。

注: JTAGICE mk II の代わりにJTAGICE 3が使用される場合、USBケーブルをAVR JTAGICE 3からPCのUSBポートに接続してください。

段階14. 図7-1.で示されるように赤線をJTAGICE mk Ⅱ のJTAGアダブタからXPLORE評価キット基板上のH6ヘッダの1番ピン(PDI_DATA) に接続してください。

注: 他のPDI/チップ上デバッグ接続法が使用される場合、この段階を飛ばすことができますが、**図7-2**.で示されるように適切な接続を構成設定することがきっと必要です。

段階15. 図7-1.で示されるように紫線をJTAGICE mk Ⅱ のJTAGアダプタからXPLORE評価キット基板上のH6ヘッダの2番ピン(VTG/U1_VC C IN/VCC)に接続してください。

注: 他のPDI/チップ・上デバッグ接続法が使用される場合、この段階を飛ばすことができますが、**図7-2**.で示されるように適切な接続を構成設定することがきっと必要です。

段階16. 図7-1.で示されるように緑線をJTAGICE mk Ⅱ のJTAGアダプタからXPLORE評価キット基板上のH6ヘッダの5番ピン(PDI_CLK)に接続してください。

注: 他のPDI/チップ上デバッグ接続法が使用される場合、この段階を飛ばすことができますが、図7-2.で示されるように適切な接続を構成設定することがきっと必要です。



- 段階17. 図7-1.で示されるように白線をJTAGICE mk II のJTAGアダブタからXPLORE評価キット基板上のH6ヘッタ・の6番ピン(GND)に接続してください。
 - **注**: 他のPDI/チップ上デバッグ接続法が使用される場合、この段階を飛ばすことができますが、**図7-2**.で示されるように適切な接続を構成設定することがきっと必要です。
 - 注: XPLORE評価キット基板は外部DC電力供給または電池供給によって通電することもできます。この開始実演例は基板通電のために1つの方法だけを示します。基板の他の通電方法の詳細についてはXPLOREハードウェア使用者の手引きを参照してください。

8. コンハ[°]イルとチッフ[°]上テ゛ハ゛ック゛

本章はAtmel StudioとIAR Embedded Workbenchでの両方の組み込みコンパイラで開始実演例プロジェクルを作成、コンパイル、構築し、その後にAtmel Studioを使用してデバイスのチップ・上デバックがを実行する方法を記述します。 Atmel Studioの組み込みコンパイラを利用する開発については8.1.項を読んでください。 IAR Embedded Workbenchを使用する開発については8.2.項を読んでください。

8.1. Atmel Studio内の組み込みコンパイラでのコンパイルとディック

Atmel Studio内の組み込みコンパーイラを使用してAtmel ATxmega32A4Uデバイス用のプロジェクトを開始するには下の記述のように行ってください。

- **段階1**. この開始実演例を演習するためのCソースコート、とヘッタ・ファイルを含む"Xplore_Getting_Started"実演例ディレクトリを構成設定するのに"Cト・ライフ・"のようなローカルト・ライフ・にこのXPLORE開始資料と共に来る"Xplore_Getting_Started.zip"ファイルを解凍してください。
- 段階2. ディクトップ コンピュータから、"Atmel Studio"ソフトウェアを開始するために"Atmel Studio"ディクトップ アイコンをダブルクリックしてください。
- 段階3. "New Project"ウィントウを開くために、"Start Page"ウィントウから"New Project..."任意選択、または"File"⇒"New"を、その後にソフトウェア メニューから"Project..."任意選択を選んでください。
- **段階4**. "Installed Templates"として"C/C++"を選び、"GCC C Executable project"任意選択をクリックし、"Create directry for solutio n"任意選択をチェック解除し、そして"GccApplication1"のような"Project Name"を入力し、最後に従うべきプロジェクト(位置)"Location"を構成設定し、その後に"OK"釦を押してください。
 - 注: このような場合に"GccApplication1"プロジェクト フォルダと共に"C:\{\text{Yxplore_Getting_Started}\{\text{Atmel}\{\text{Getting_Started}\{\text{Getting_Started}\{\text{Getting_Started}\{\text{Getting_Started}\{\text{Getting_Started}\{\text{Finter_Getting_Started}\{\text{Getting_Started}\{\text{Getting_Started}\{\text{Getting_Started}\{\text{Getting_Started}\{\text{Finter_Getting_Started}\{\text{Getting_Sta
- 段階5. "Device Selection"ウィントウの最後へスクロール ダウンし、"Atcmega32A4U"デバイスを選び、そして再び"OK"釦を押してください。
- 段階6. "Solution Explorer"ウィントウから"GccApplication1.c" 雛形ファイルを強調表示にし、そして図8-1.で示されるように"Remove"任意選択を選ぶのにマウスの右クリックを使用してください。

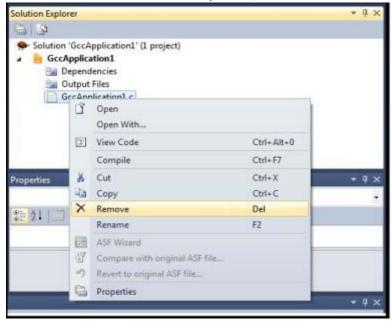


図8-1. Atmel StudioでのSolution Explorerウィントゥ

- 段階7. "GccApplication1.c"雛形ファイルを削除するために"Atmel Studio"ダイアログウィンドウカンら"Delete"釦を選んでください。
- **段階8**. "Solution Explorer"ウィンドウから"GccApplication1"を強調表示にし、そして"Add"を選ぶのにマウスの右クリックを使用し、その後に"Existing item..."任意選択を選んでください。
- **段階9**. "C:\footnote{Started\footnote{Atmel\footnote{Getting_Started\footnote{Getting_Started_Apploace}"デルクトリから"Getting_Started_XPLORE.c"ファイルを選択し、そして"Add"釦を選ぶ前に、"Ctrl"キーボート、キーを押下して"usart_driver.c"、"usart_driver.h"、"avr_compiler.h"ファイルも共に選択してください。



- 段階10. 全ての設定とプロジェクト全体を保存するためにAtmel Studioウィントウから"File"その後に"Save all"任意選択を選んでください。
 - **注**: "example_code"フォルタ からAtmel Studio内の現在の設計ディレクトリへ"Getting_Started_XPLORE.c"、"usart_driver.c"、"usart_driver.h"、"avr_compiler.h"ファイルのような例設計ソースファイルの追加により、それらのファイルの複製が保存され、現在の設計ディレクトリ("C:\text{YXplore_Getting_Started}*Atmel\text{YGetting_Started}*GccApplication1\text{\text{Y}}")に格納されます。
- **段階11**. Atmel Studioソフトウェアでファイルを開くために"Solution Explorer"ウィント・ウから"Getting_Started_XPLORE.c"ファイルをダブルクリックしてください。
- 段階12. コート・をコンパーイルしてプロジェクトを再構築するために"Atmel Studio"ウィント・ウから"Build"を選択して"Rebuild Solution"を選んでください。
 - 注: Atmel Studioの既定コンパイラは組み込まれた本来のGNUコンパイラを意味する"Native"に構成設定されます。使用者はAtmel Studioソフトウェア内での他の任意選択コンパイラ選択としてプロジェクト"GccApplication1 Properties..."の"Advance"設定から"WinAVR"GNUコンパイラである"Toolchain Flavour"を選ぶこともできます。

重要な構成設定:

この開始実演設計のチップ上デバッグのために次の段階へ続行するために、7.章(ハードウェア構成設定)で記述されるようにハードウェア構成 設定が既に行われていることを確実にしてください。また、この資料の4.章(Puttyソフトウェアのインストール)で指示されるようにあなたのシステ ムにPuttyソフトウェア(Windows端末画面ソフトウェア)が既にインストールされているべきです。

- 注:端末ウィントウとしてPuttyソフトウェアを使用する他に、使用者は直接的なインストールや、この資料の5.章で記述される拡張ツールとしてAt mel Studioソフトウェア内の別の端末ウィントウを使用することもできます。
- **段階13**. "PuTTY Configration"ウインドウを開始するために、"Start"、"Program"または"All Programs"、"PuTTY"、そして再び"PuTTY" 任意選択を選択してください。
 - 注: この開始実演例はUSB-COMポート模倣動作データを表示するための端末画面ウィントウとして"Putty"ソフトウェアを利用する方法を記述します。使用者はそれが先にAtmel Studioソフトウェアにインストールされている場合に、プラグイン端末ウィントウを開始して使用するためにAtmel Studioから"View"その後に"Terminal Window"任意選択を選択的に選ぶこともできます。このような場合、使用者はAtmel Studioの拡張管理部からこのようなプラグイン端末ウィントウに対して利用可能な設定を調整するために、Atmel Studioから"Tools"⇒"Options"⇒"Extensions"⇒"Terminal Window"を選択することもできます。
- 段階14. "PuTTY Configration"ウインドウから"Category"下の"Session"を選び、その後に"connection type"として"Serial"任意選択を許可/チェックし、"speed"設定として"9600"を入力し、次に"Serial line"設定として"COMxx"を入力し、そこでそれは"デバイスマネージャ"ウイント・ウ(コンピュータ→フィーコンピュータ→フプロペティ→(ハート・ウェア タフ・)→デバイス マネージャ)から"USB Serial Port"に対して検出された"COMxx"ポート番号設定と一致しなければなりません。
 - 注: この開始実演例はUSB-COMポート模倣動作データを表示するための端末画面ウィントウとして"Putty"ソフトウェアを利用する方法を記述します。使用者はそれが先にAtmel Studioソフトウェアにインストールされている場合に、プラグイン端末ウィントウを開始して使用するためにAtmel Studioから"View"その後に"Terminal Window"任意選択を選択的に選ぶこともできます。このような場合、使用者はAtmel Studioの拡張管理部からこのようなプラグイン端末ウィントウに対して利用可能な設定を調整するために、Atmel Studioから"Tools"⇒"Options"⇒"Extensions"⇒"Terminal Window"を選択することもできます。

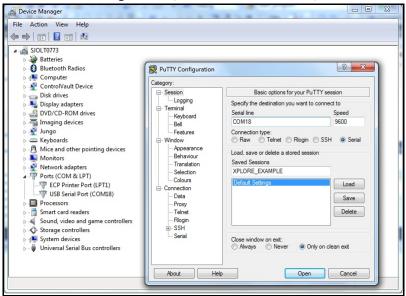


図8-2. PuTTY Configurationウィント・ウとデバイス マネーシャ ウィント・ウ

- 段階15. "Saved Sessions"として"XPLORE_EXAMPLE"を入力して"PuTTY Configuration"ウインドウから"Save"任意選択を選んでください。
- **段階16**. "PuTTY Configuration"ウィントウから"Category"下の"Terminal"任意選択を選び、その後に"Local echo"に対して"Force on" を許可/チェックし、"Local line editing"設定として"Force on"任意選択も許可/チェックしてください。



段階17. "PuTTY Configuration"ウィントウから"Connection"の"SSH"下の"Serial"任意選択を選び、その後に"Serial line to connect to"の右側に正しいCOMポート(COMxx)番号と以下のシリアルポート設定も指定されていることを確実にしてください。

シリアル線形態設定

Speed(Baud): 9600 Data bits : 8 Stop bits : 1 Parity : None

Flow control: XON/XOFF

- 段階18. "COMxx PuTTY"端末ウィンドウを開始するために"PuTTY Configuration"ウィンドウから"Open"を選んでください。
- 段階19. "COMxx PuTTY"端末ウィント・ウと"テ`ハ`イス マネーシ`ャ"ウィント`ウを最小化し、その後にAtmel Studioウィント`ウを使用するように選択 を切り返してください。
- 段階20. Atmel Studioソフトウェアカゥら"Tools"⇒"AVR Tools Firmware Upgrade"を選んでください。
- 段階21. 接続ツールが検出された場合に"AVR Tools Firmware Upgrade"ウィントウから"Upgrade"を選んでください。ソフトウェアによって正 しいツールが検出されない場合、基板のPDI接続と電力供給を調べてください。
- 段階22. "AVR Tools Firmware Upgrade"ウィントウで"Upgrade Done"メッセーシが表示された時は"Close"を選んでください。
- 段階23. "Solution Explorer"ウインドウから"GccApplication1"または"ObjProject1"を強調表示にし、その後に"Properties"任意選択を 選択するためにマウスの右クリックを使用してください。
- 段階24. "Tool"を選び、その後に使用する実際のチップ上デバッカーに依存して"JTAGICE mkII"または"JTAGICE 3"のような"selected debugger/programmer"を選択してください。
- 段階25. 再び"Solution Explorer"ウィントウから"Getting_Started_XPLORE.c"ファイルをダブルクリックしてください。
- 段階26. 設計コート・のチップ・上テ・ハ・ック・を開始するために"Atmel Studio"ウィント・ウの"Debug"メニューカゝら"Start Debugging and Break"を選 ぶか、または"Alt+F5"キーボート"キーを使用してください。
- 段階27. 図8-3.で示されるように設計コードの最後へ 図8-3. コードでの中断点(ブレークポイント)設定 スクロール ダウンしてください。右(図)の命令文 上をクリックし、その後に"Debug"を選んでく ださい。その後に"Breakpoint"または"Tog gle Beakpoint"を選ぶのにマウスの右クリックを 使用してください。その後、デバイスのUART ポートから1バイトが受信された後でソフトウェア 中断点(ブレークポイント)を設定するのに"Inse rt Breakpoint"を使用してください。

```
Fetch received data as it is received
int w = 0;
while (w < NUM_BYTES)</pre>
    if (USART_RXBufferData_Available(&USART_data))
        receiveArray[v] = USART_RXBuffer_GetByte(&USART_data);
        // if received byte is 41 in hex, which is the "A" in ASCII
        // then display A (hex value) to port E [3:0]
              1
// disply the character "A" to LEDs
LEDPORT.OUT = 0x0A; // LED[3:0] = 1010 = Off, ON, Off, ON
```

- 段階28. マウス カーソルを"ReceiveArray[0]"に置き、"& 0x0F"の前のコードからの配列命令文はそこでのクリックなしでコンパイラが選択した SRAMアドレス位置メッセージを表示します。このメッセージはマウスの近くに表示されます。UARTから受信するバイトに対するカーソル はAteml Studioソフトウェアでコンハ°イルしたコート、に対して"receiveArray (uint8_t[1] {data}@0x21ab}"、IARでコンハ°イルされたコート、に 対して"receiveArray|{uint8_t[1] {data}@0x21f8}"のようになります。
- 段階29. Atmel Studioで"Memory1"ウィント うが自動的に開かれて表示されない場合、"Memory1"ウィント うを表示するためにAtmel Studi oのメニューから"Debug" \Rightarrow "Windows" \Rightarrow "Memory1"を選ぶか、または"Alt+6"キーボート"キーを使用してください。
- 段階30. "Memory1"ウィント・ウから、メモリ設定のために"data INTERNAL_SRAM"を選んでください。その後に図8-4.で示されるようにAtm el Studioでコンパイルしたコート`に対する"0x21AB,data"または図8-5.で示されるようにIARでコンパイルしたコート`に対する"0x21F8,

data"のように、受信するバイト"ReceiveArra 図8-4. SRAMデータ Memory1ウィントウ - 本来のGNUコンパイラ使用 y[0]"として使用される検出された"アドレス" 値を入力してください。("Adddress"ダイ アロ グボックスを見るために"Memory1"ウィンドウが 最大に拡大されていることを確実にしてく ださい。)

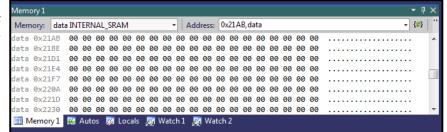
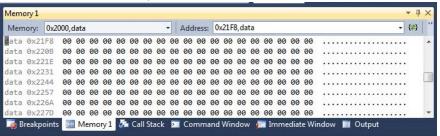




図8-5. SRAMデータ Memory1ウィンドウ - IARコンパイラ使用



- 注: この例プロジェクトに対するSRAMでのデータ アトレス位置0x21ABはコートのコンハプイルにAtmel Studio(GNUコンハプイラ)が利用された場合に受信するパイトの格納に使用され得、データ アトレス位置0x21F8はIAR EWコンパプイラの場合に受信するパイトのために使用され得ます。このSRAMアトレス位置は実際、現実に検出されたSRAM16進アトレス位置を示す前の段階に依存します。
- **段階31**. "Atmel Studio"ソフトウェアから"Debug"そして"Continue"任意選択を選ぶか、または"F5"キーボート キーを使用、その後にAtmel XMEGAデバイスから端末画面ウィントウに送られた以下の初期メッセージを見るために"COMxx PuTTY"端末ウィントウを最大に切り替えてください。

"What does ATMEL stand for?

The correct answer is:

A = PASSION, T = INTEGRITY, M = TEAMWORK, E = ACHIEVEMENT,

L = ACCOUNTABILITY;

Please enter the letter 'A' to agree with above answer."

- **段階32**. それで、"COMxx PuTTY"端末画面ウィントウからカーソル/ブロックを選択してクリックし、何が起こるか見るために指示されたものに従わずに、"Shift"キーボート・キーと"C"キーを共に押し、その後に"Enter"キーボート・キーを押すことによって端末ウィントウから"A"ではなく大文字の"C"を入力することによって行ってみてください。
- 段階33. "COMxx PuTTY"端末ウィンドウを最小化してください。さて、データ メモリ ウィンドウは図8-6.または図8-7.で示されるようにアドレス位置のx21ABまたは0x21F8(データ メモリ(SRAM)のUART受信ハイト位置)がASCII文字の"C"である16進値(43)を含むことを示します。

段階33. "COMxx - PuTTY"端末ウィンドウを最小化し 図8-6. SRAMデータ Memory1ウィンドウ - 本来のGNUコンパイラ使用

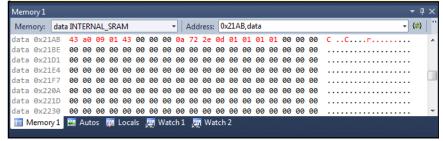
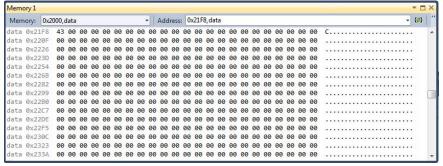


図8-7. SRAMデータ Memory1ウィント・ウ - IARコンハ・イラ使用



- 注: この例プロジェクトに対するSRAMでのデータアトンス位置0x21ABはコートのコンパイルにAtmel Studio(GNUコンパイラ)が利用された場合に受信するバイトの格納に使用され得、データアトンス位置0x21F8はIAR EWコンパイラの場合に受信するバイトのために使用され得ます。このSRAMアトンス位置は実際、現実に検出されたSRAM16進アトンス位置を示す前の段階に依存します。
- 段階34. プログラムの走行を続けるために"Debug"そして"Continue"任意選択を選ぶか、または"F5"キーボードキーを使用してください。
- 段階35. "COMxx PuTTY"端末ウィンドウを選択し、その後にそれを最大化してください。さて、端末画面ウィンドウに以下のような新しいメッセージが表示されるでしょう。

"C - Please try again. You can only allow entering the letter 'A' to agree with above answer."

段階36. そこで、再び"COMxx - PuTTY"端末画面ウィントウを選択してクリックして、何が起こるか見るために、今回は指示されたものに従って"Shift"キーホート・キーと"A"キーを共に押し、その後に"Enter"を押すことによって端末ウィントウから大文字の"A"を入力することによって行ってみましょう。



段階37. "COMxx - PuTTY"端末ウィントウを最小化してください。さて、データメモリウィントウは図8-8.または図8-9.で示されるようにアトレス位置のx21ABまたは0x21F8(データメモリ(SRAM)のUART受信バイト位置)がASCII文字の"A"である16進値(41)を含むことを示します。プログラムの走行を続けるために"Debug"そして"Continue"任意選択を選ぶか、または

"F5"キーボート キーを使用してください。

図8-8. SRAMデータ Memory 1ウィンドウ - 本来のGNUコンパイラ使用

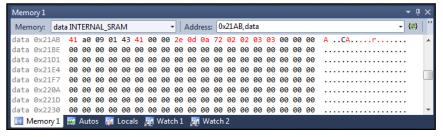
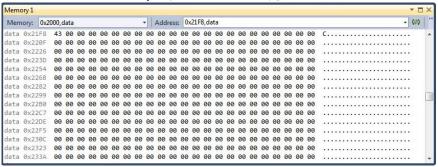


図8-9. SRAMデータ Memory1ウィンドウ - IARコンパイラ使用



注: この例プロジェクトに対するSRAMでのデータ アトンス位置0x21ABはコートのコンパイルにAtmel Studio(GNUコンパイラ)が利用された場合に受信するバイトの格納に使用され得、データ アトンス位置0x21F8はIAR EWコンパイラの場合に受信するバイトのために使用され得ます。このSRAMアトンス位置は実際、現実に検出されたSRAM16進アトンス位置を示す前の段階に依存します。

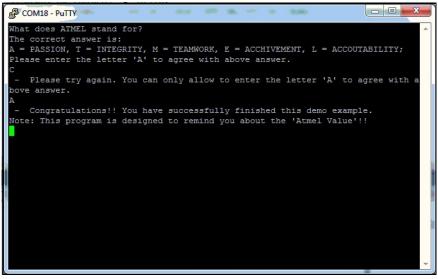
段階38. "COMxx - PuTTY"端末ウインドウを選択し、その後にそれを最大化してください。さて、この実演例の完了のために以下のような別の新しいメッセージが示されます。これは**図8-10**.でも示されます。

"A

- Congratulations!! You have successfully finished this demo example."

注: このプログラムは'Atmelの意義'についてあなたに思い出させるように設計されています!

図8-10. 実演完了成功メッセージを示すCOMxx - PuTTY端末ウィンドウ



注: 'A'ASCII文字は2進値にも変換されてXPLORE評価キット基板上のLED3~0で表示されます。これはXPLORE評価キット ハート・ウェアがAtmelの意義であなたの同意に従うことを示します。

以下はデバッグ処理を停止し、プロジェクトを保存し、そしてAtmel Sudioソフトウェアを抜け出すための最終段階を示します。

段階39. 端末画面を閉じるために"COMxx - PuTTY"端末ウィンドウの右上側隅から"×"印をクリックしてください。

段階40. 設計コート、のデ、バッグを停止するために"Debug"その後に"Stop Debugging"任意選択または"Ctrl+Shift+F5"キーボート、キーを選んでください。

段階41. プロジェクトを保存するために"File"その後に"Save All"または"Ctrl+Shift+S"キーーボード キーを選んでください。

段階42. "Atmel Studio"ソフウェアを抜け出すために"File"その後に"Exit"または"Alt+F4"キーボート、キーを選んでください。



8.2. Atmel StudioとIARでのコンパイルとディック

Atmel StudioとIAR EWでAtmel ATxmega32A4Uデバイス用のプロジェクトを開始するには下の記述のように行ってください。

- **段階1**. この開始実演例を演習するためにCソース コードとヘッタ・ファイルを含む"Xplore_Getting_Started"フォルタ・を構成設定するのに"Cトライフ・"のようなローカルトライフ・にこの開始資料と共に来る"Xplore_Getting_Started.zip"ファイルを解凍してください。
- 段階2. PCのテ`スクトップ゚から、"Atmel Studio"ソフトウェアを開始するために"Atmel Studio"テ`スクトップ゚アイコンをダブルクリックしてください。
- 段階3. "Atmel Studio"ウィント・ウから"Tools"その後に"External Tools..."任意選択を選んでください。
- 段階4. "External Tools"ターイアログ ウィントウから"Add" 釦を選んでください。
- 段階5. "External Tools"ダイアログウィンドウの"Title"区画から"[New Tool1]"のような文を"IAR Compiler"に変更してください。
- 段階6. 以下と同様なIARコンパイラの実行可能ファイル パスとファイル名を入力するために"Browse"釦を選んでください。
 - "C:\Program Files (x86)\IAR Systems\Embedded Workbench 6.0 0\cdot\common\text{\text{bin}}\IarIdePm.exe"
 - 注: 64ビットのWindows 7オペレーティング システム(OS)は通常"Program Files (x86)"ディレクトリを使用します。他のWindows OSについては単に"Program Files"ディレクトリが使用されるでしょう。上のパス設定は実際、先にインストールしたIAR Embedded Workbenchコンパイラ ソフトウェアの現実のパス位置に基づきます。
- **段階7**. "External Tools" ダイアログ ウィンドウから"Apply"釦その後に"OK"釦を選んでください。 今や、外部ツールとしてAtmel Studio 6.0またはそれ以降版のソフトウェア ウィンドウからAVRコンパイラ用のIAR Workbenchを呼び出すことができます。
- 段階8. IAR Workbenchソフトウェアを開始するために"Atmel Studio"ソフトウェアから"Tools"その後に"IAR Compiler"任意選択を選んでください。
- 段階9. "IAR Embedded Workbench"ウインドウから"Project"その後に"Create New Project"任意選択を選んでください。
- 段階10. "Create New Project"ウィントウから、図8-11.で示されるように、最初に"Externally Build Executable"任意選択選び、その後に"OK"釦をクリックしてください。

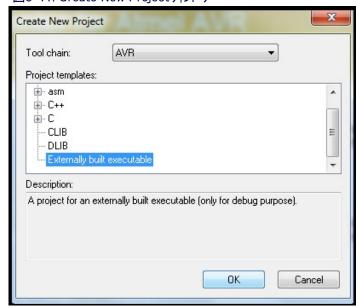
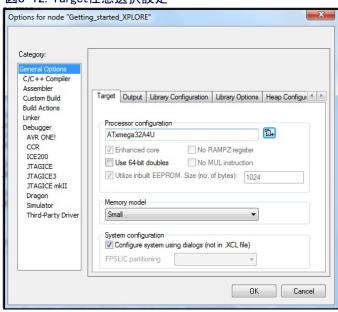


図8-11. Create New Projectウィント・ウ

- **段階11**. "C:\formular C:\formular Started\formular IAR\formular Getting_Started\formular "ディレクトリを閲覧し、その後に新しいプロジェクト名として"Getting_started_X PLORE.ewp"を入力し、最後に新しいプロジェクトを保存するために"Save" 釦を選んでください。
- 段階12. "Workspace"ウィントゥのFiles下で"Getting_started_XPLORE Debug"を選び、そして"Add"その後に"Add Files..."任意選択を選ぶためにマウスの右クリックを使用してください。
- **段階13**. 先に"Getting_started_XPLORE.c"ファイルを選択し、その後に"Ctrl"キーボード キーを押して同時に"usart_driver.c"ファイルを選択し、最後に"Open"釦をクリックしてください。
- 段階14. ファイルを開くためにWorkspaceウィントウから"Getting_started_XPLORE.c"ファイル上でダブルクリックしてください。
- **段階15**. 最初にWorkspaceウィントゥから"Getting_started_XPLORE Debug"を強調表示にし、その後に"IAR Embedded Workbench I DE"ウィントゥウから"Project"そして"Options"を選ぶか、または"Alt+F7"キーボート・キーを使用してください。
- 段階16. "Options for node ..."ウィントウが飛び出た時に"General Options"を選んでください。
- **段階17**. "General Options"分類の"Target"タブから、図8-12.で示されるように"Processor configuration"として"ATxmega"任意選択その後に"ATxmega32A4U"を選んでください。

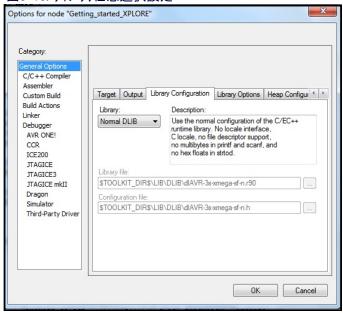


図8-12. Target任意選択設定



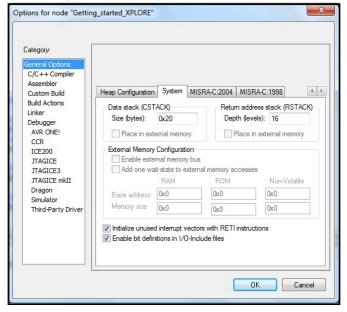
段階18. "General Options"分類から"Library Configuration"タブを選び、その後に図8-13.で示されるようにライブラリ名として"Norm al DLIB"を選んでください。

図8-13. ライブラリ任意選択設定



段階19. 右手側のウィントウに移動するために右矢印釦上でクリックするのにマウスを使用し、"General Options"分類から"System"タブを選び、その後に"enable bit difinitions in I/O include files" 任意選択を許可(チェック)して、その後にOKをクリックしてください。この設定は図8-14.で示されます。

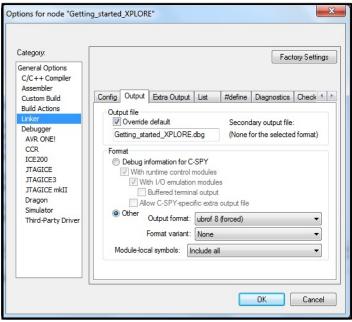
図8-14. System設定任意選択





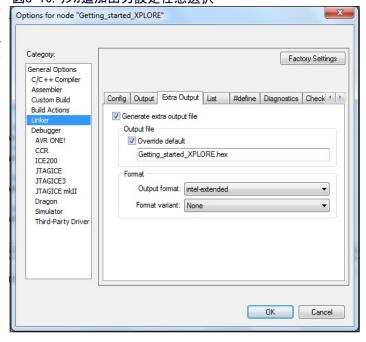
段階20. 分類から"Linker"その後に"Output"タフ'を選び、次に"Output file"に対するデ`ハ`ッグ出力ファイル名として"Getting_started_XP LORE.dbg"ファイルでの"override-default"構成設定を許可(チェック)し、その後に図8-15.で示されるように"Other"を選択して"Output format"として"ubrof-8 (forced)"を選んでください。

図8-15. リンカ出力設定任意選択



段階21. "Category(分類)"から"Linker"その後に"Extra Output"タフ'を選び、次に"General extra output file"任意選択と"Override default"任意選択を許可(チェック)し、ファイル名も"Getting_started_XPLORE.hex"に変更し、その後に"Output format"を"Intel-standard"にすべく選び、最後に"OK"釦を選んでください。これらの設定は図8-16.で示されます。

図8-16. リンカ追加出力設定任意選択



- 段階22. IAR Embedded Workbenchソフトウェアから"File"そして"Save Workspace"を選んでください。
- 段階23. "Save Workspace As"ウインドウが飛び出した時に"Getting_started_XPLORE.eww"を入力し、その後に"Save"釦をクリックしてください。
- 段階24. "Workspace"ウィンドウから"Getting started XPLORE.c"ファイルを選んでください。
- 段階25. プロジェ外をコンパイルして構築するために"IAR Embedded Workbench IDE"メニューから"Project"そして"Rebuild All"任意選択を選んでください。

プロジェ外を保存してIARを抜け出すために以下を実行してください。

- 段階26. "File"その後に"Save All"任意選択を選んでください。
- 段階27. "File"その後に"Exit"任意選択を選んでください。

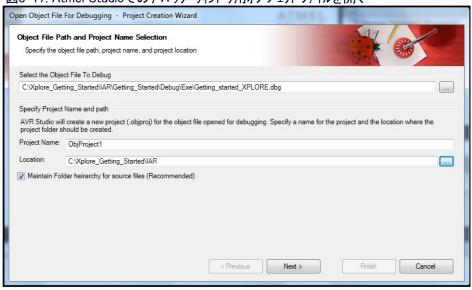
IARでコンパイルしたコートのデバックをAtmel Studioで続けるために以下を実行してください。

- 段階28. Atmel Studioウィントウを最大化してください。
- 段階29. Atmel Studioソフトウェア ウィント・ウから"File"その後に"Open"を選び、そして"Open Object File for Debugging"を選ぶか、または "Ctrl+O"キーボート・キーを使用してください。



段階30. "Open Object File for Debugging"ウィントウから、"C:\footnote{Name" Cetting_Started\footnote{Name" Cetting_Started Started St

図8-17. Atmel Studioでのディック ウィンド ウ用オブシェクト ファイルを開く



- **段階31**. "Next" 釦選択後、"Device Family"として"AVR XMEGA 8-bit"を選ぶために"Device Selection"ウィントウの最後へスクロール ダウ ンし、その後に"Atxmega32A4U"デッ・イスを選び、最後に"Finish" 釦を押してください。
- 段階32. こんなファイルを開くために"Atmel Studio"ソフトウェアの"Solution Explore"ウィンドウから"Getting_started_XPLORE.c"ファイルでダブル クリックしてください。
- 段階33. 全ての設定とプロジェクトを保存するために"File"その後に"Save All"任意選択を選んでください。

重要な構成設定

設計のチップとディックを実行するために次の段階を続行するには、前の章で記述されるようにハードウェア構成設定が行われることを確実にしてください。

段階34. そこで、この開始設計例に対してチップ上ディックを実行するために、この資料の8.1.項の**段階13**.から記述されたのと同じ手続きに従ってください。

9. XPLORE上のAtmel XMEGAの実装プログラミング

本章はXPLORE評価キット基板上のPDIインターフェースを通してAtmel ATxmega32A4Uデバイスの実装プログラミング(ISP)を実行する方法を記述します。これはIAR Embedded WorkbenchまたはAtmel Studioからのコンパイラ使用で設計例のコート・をコンパイルしてデバッグした後で書き込みファイル(*.hex)が既に生成されていると仮定されます。

前章で記述されたのと同じハート・ウェア構成設定で、XPLORE評価キット基板上のXMEGAデバイスのPDI実装プログラミングを実行するために、使用者はAtmel Studioを開始し、その後に下で記述される段階的手順に従ってください。

- 段階1. "Atmel Studio"ソフトウェアを開始するために、PCのデスクトップから、"Atmel Studio"デスクトップ。アイコン上でダブルクリックしてください。
- 段階2. "Open Project" ダイアログ ウィンドウを開始するために、"Start Page" ウィンドウから" Open Project..."任意選択を選ぶか、またはソフト ウェア メニューから"File"、"Open"、そして"Project/Solution"任意選択を選んでください。
- 段階3. 以前に保存されたAtmel Studioプロジェクトを選ぶために、設計プロジェクト ディレクトリを閲覧してください。

Atmel StudioプロジェクトファイルがAtmel Studio(本来のGNUまたはWinAVR GNU)でコンハーイルされて構築されたファイルに基づいて作成されたなら、以下のファイルを開くために選択してその後に"Open"釦を選ぶことができます。

"C:\text{!Xplore_Getting_Started}\text{!Atmel\text{!YGetting_Started}\text{!GccApplication1\text{!YGccApplication1\text{!Grown}}}

Atmel StudioプロジェクトファイルがIARでコンハプイルされて構築されたファイルに基づいて作成されたなら、以下のファイルを開くために選択することができます。

"C:\forall Yplore_Getting_Started\forall IAR\forall ObjProject1\forall Objproject1\forall ObjProject1.objproj"

重要な構成設定

設計のチップ上デバッグを実行するために次の段階を続行するには、前の章で記述されるようにハードウェア構成設定が行われることを確 実にしてください。

現在使用している現実のJTAGICE mk II またはJTAGICE 3ツールで"AVR tools Firmware Upgrade"手続きが実行されていなかったなら、以下の段階を行う前に8.1.項からの<mark>段階20.、段階21.、段階22.</mark>を終えてください。

段階4. 既存Atmel Studioプロジェ外読み込み後、"Tools"その後に"Device programming"任意選択を選んでください。



- **段階5**. "AVR Programming"ウィンドウから、"Tool"設定として(使用する実際のツールに依存して)"JTAGICE mkII"または"JTAGICE 3"を 選んでください。
- 段階6. "Device"設定として"ATxmega32A4U"を選び、"Interface"設定として"PDI"も選び、その後に"Apply"釦を押してください。
- 段階7. 図9-1.で示されるように"AVR Programming"ウインドウ から"Target Voltage"を読むのに"Read"釦を、また "Device ID"を読むのに別の"Read"釦を選んでく ださい。

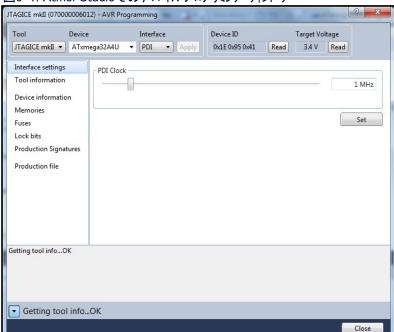


図9-1. Atmel Studioでのデバイス プログラミング ウィンドウ

段階8. "AVR Programming"ウィント・ウの"Interface settings"から"Memories"タブを選び、その後にデバイスのフラッシュメモリプログラミングのためのフラッシュ入力HEXファイル(Flash Input Hex file)とディレクトリ パスを指定するのに閲覧釦を選んでください。

前にプロジェクトをコンパイルして構築するのにAtmel Studio組み込みコンパイラ(本来のGNU/WinAVR GNU)が使用された場合、以下のプログラミング用HEXファイルを選ぶことができます。

"C:\text{YXplore_Getting_Started}\text{YAtmel}\text{YGetting_Started}\text{YGccApplication1}\text{YDebug}\text{YGccApplication1.hex"}

前にプロジェクトをコンパイルして構築するのにIARコンパイラが使用された場合、以下のプログラミング用HEXファイルを選ぶことができます。

プログラミング用HEXファイルを構成設定するために"Open"釦を選ぶことを確実にしてください。

- 段階9. フラッシュ メモリに入力HEXファイルをプログラミング する(書き込む)ために"Program" 釦を選んでください。 今や、Atmel XMEGAデバイスはプログラミングされ(書き込まれ)ます。使用者はプログラミングされた(書き込まれた)コードを検証するために"Verify" 釦を選ぶこともできます。
- 段階10. システムのデスクトップから、"PuTTY Configuration"ウィントウを開始するために、"Start"、"Programs"または"All Programs"、"PuTTY"を、その後に再び"PuTTY"任意選択を選んでください。
- **段階11**. "PuTTY Configuration"ウインドウから"Category"下の"Session"を選び、その後に"Saved sessions"として前に保存された"XPL ORE_EXAMPLE"を選んでください。
 - 注: "XPLORE_EXAMPLE"が未だ"Saved sessions"として構成設定されていなければ、この段階を再び試みる前に、この資料の8.1.項からの段階14.と段階15.を繰り返してください。
- 段階12. この資料の8.1.項からの段階16.、段階17.、段階18.を繰り返してください。
- 段階13. さて、開かれた"COMxx PuTTY"端末ウィンドウとで、一度XPLORE評価キット基板からリセット(SW8)釦を押して放してください。 これは"COMxx - PuTTY"端末画面で以下のメッセージを示すでしょう。

"What does ATMEL stand for?

The correct answer is:

A = PASSION, T = INTEGRITY, M = TEAMWORK, E = ACCHIEVEMENT,

L = ACCOUNTABILITY;

Please enter the letter 'A' to agree with above answer."

段階14. 先の端末画面からの指示された命令に従わないために、ASCII文字"C"("Shift"+"C"キーボードキーを使用する大文字の"C") を入力し、その後に"Enter"キーを押してください。

そこで、"COMxx - PuTTY"端末画面から以下の更新されたメッセージを見るでしょう。

"C

- Please try again. You can only allow entering the letter 'A' to agree with above answer."



段階15. 次に、端末画面からの指示された命令に従う"同意(Agree)"のために、ASCII文字"A"("Shift"+"A"キーボード キーを使用する大文字の"A")を入力し、その後に"Enter"キーを押してください。

そこで、"COMxx - PuTTY"端末画面から以下の更新されたメッセージを見るでしょう。

"A

- Congratulations!! You have successfully finished this demo example."

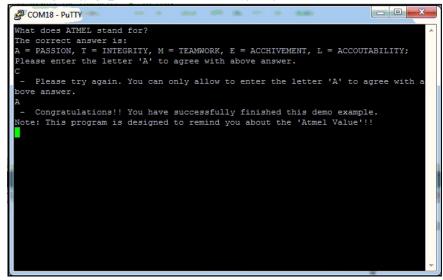
注: このプログラムは'Atmelの意義'についてあなたに思い出させるように設計されています!

"COMxx - PuTTY"端末画面メッセーシ が図 図9-2. 検査結果を示すCOMxx - PuTTY端末ウィンドウ9-2.で示されます。 図9-2. 検査結果を示すCOMxx - PuTTY端末ウィンドウ

XPLORE評価キット基板上に配置されたLE D3~0も"A"文字を2進値で変換されて示され、それは"1010"-LED3=OFF、LED2= ON、LED1=OFF、LED0=ONです。

これはハート・ウェアが最後に同意された応答を確かに追認したことを意味します。

今や、あなたはこのXPLORE評価キットに対する開始の手引きを、成功裏に完了しました。



10. 参照

10.1. オンライン参照資料

- [1] ATxmega32A4Uデバイス データシート: http://www.atmel.com/Images/doc8387.pdf
- [2] ATxmega32A4デバイス データシート: http://www.atmel.com/dyn/resources/prod_documents/doc8069.pdf
- [3] ATxmega32D4デバイス データシート: http://www.atmel.com/Images/doc8135.pdf
- [4] Atmel XMEGA AUデバイス手引書: http://www.atmel.com/Images/doc8331.pdf
- [5] Atmel XMEGA Aデバイス手引書: http://www.atmel.com/dyn/resources/prod_documents/doc8077.pdf
- [6] Atmel XMEGA Dデバイス手引書: http://www.atmel.com/Images/doc8210.pdf
- [7] FTDI(Future Technology International Ltd.) FT232RLデータシート: http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf
- [8] LDO (Linear Technology) LT3022データシート: http://www.linear.com/product/LT3022
- [9] FT232RLデバイスのVCPドライバ用インストールの手引き: http://www.ftdichip.com/Support/Documents/InstallGuides.htm

10.2. 設計ソフトウェアと通信装置ドライバ

- [10] Atmel Studioソフトウェア ダウンロート:http://www.atmel.com/tools/atmelstudio.aspx
- [11] (Atmel AVRを支援しなければならない)IARコンパイラ インストール ソフトウェア ダウンロート: http://www.iar.com/website1/1.0.1.0/675/1/
- [12] (PCへのUSB接続に使用される)Future Technology International IncのFTDI FT2232仮想COMポートトライハ ゙ ダウンロート : http://www.ftdichip.com/Drivers/VCP.htm
- [13] Microsoft Windows 2000,XP,Vista,7 OS使用の場合、FT232RL用のVCP(仮想COMポート)は次のウェブサイトから直接ダウンロート・することができます。: http://www.ftdichip.com/Drivers/CDM/CDM 2.08.24 WHQL Certified.zip
- [14] Windows OS用Puttyソフトウェア(Putty.zip)は次のウェブサイトからダウンロート、することができます。: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html





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 新大崎勧業ピル 16F アトメル ジャパン合同会社 TEL (+81)(3)-6417-0300 FAX (+81)(3)-6417-0370

© 2012 Atmel Corporation. 全権利予約済/改訂:42015A-AVR-07/2012

Atmel®、Atmel¤¬¬とそれらの組み合わせ、AVR®、Enabling Unlimited Possibilities®、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。Windows®とその他は米国とその他の国に於いてMicrosoft Corporationの登録商標です。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AVR1936応用記述(doc42015.pdf Rev.42015A-07/12)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。