

## AVR4029: ATMEL Y7 h 1 ア 枠組み - 開始に際して

## 要点

ASF例へのアクセス プロシ<sup>\*</sup>ェクトへのASF単位部追加 ATMEL® Studio 6 AR<sup>TM</sup> GNU make file/GCC

## 1.序説

ATMEL ソフ や」ア枠組み (ASF ATMEL AVR Software Framework www.atmel.com/asf)は ATME LOmegaAVR®、AVR XMEGA®、AVR UC3 SAM デバ1 7用の応用構築に ソフ や」ア ドライル とライ フ う に設計 されました。これは オ い レーティング システム (OS)内に統合、または独立製品として動作す ることが できます。

この応用記述に於いて、開発者はATMEL Studio 6 ARとGNU GCCのmakefileと共に開始する方法について読むことができます。

- ・ASFの インストール方法
- ・ASF参考応用の開始方法
- ・ASF資料の取得方法

この開始に際しての資料を読むのに、ASF基本構造の高度な知識に関してASF参照基準手引書を読むことが推奨されますが、必要とはされません。

## 2.ASFとATMEL Studio 6での開始に際して

## 2.1. インストール

- 1.http://www.atmel.com/atmeltudioでATMEL Studio 6を ダウンロードしてください。ATMEL ASFは Studio 6に含まれ、独立した ダウンロードは不要です。
- 2.122トルしてください。
- 3.ATMEL Studioを開始してください。

ATMEL Studioで開始するには組み込まれたATMEL Studio使用者の手引きを参照して、 ようこそ (we kome画面で開始してください。

#### 22.映像

ATMEL StudioとASFの映像での開始はhttp://www.youtube.com/user/AtmeCorporationでのATMEL映像に従ってください。



ATMEL 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.8431B-03/12,8431BJ0-05/12





## 23.ASF例の開始

#### 1.Fileメニューで、New Example Project from ASFを選択してください。

File	Edit View VAssistX Project Debug	Tools	Window He	elp		
	New	• 67	Project		Ctrl+Shift+N	•
	Open	• 🗅	File		Ctrl+N	Hex 🕞 🗝 🚽 🕮 🗂 🖕
	Close	8	Example Pr	oject from ASF	Ctrl+Shift+E	
đ,	Close Solution					ATAGE
	Import	•				AIMEL
	Save Selected Items Ctrl+S					STUDIO
	Save Selected Items As					
1	Save All Ctrl+Shift+	5				
	Export Template					
	Page Setup	Ge	t Started	Tools Help	Latest News	
-	Print Ctrl+P					
	Exit Alt+F4	W	alcome Tu	utorials Links a	and Resources	
	Open Project				Welcome to Ata	nal Studio
					Get to know Atme	Studio.
-			H	The West of Contract	Atmal Studio Llear	Guida
R	ecent Projects	- 1	- Without a		Getting Started	ouide

キット例えば、ATMEL EVK110Q ATMEL AVR Xplained ~ )または技術的なキーワートのえば、USB、D、割じ込み、~ )または種類 (ドライバ、部品、サービス 応用)によってASF例を並び替えることが可能です。例の資料はNew Example Projectりィント かって パネル内 の ヘ ル プ ア イ コンの クリックによって見ることができます。 "USART Example - XMEGA - A1 Xplained" プロジェク を探して、それを作成するた めにOKを クリックしてください。

#### 図 2-2. キッ 開の新しい 例の 選択

Rew Example	Project			×
ASF Version	3.0.0 • Device Family: A	II Category: All	▼ Search for Example Project: ₽	USART Frample
All Projects Kit Category Technology	TC Example TC Example TC Example Trimeout er Unit test f Unit test	e 1 - XMEGA-A1 Xplained e 2 - XMEGA-A1 Xplained e 3 - XMEGA-A1 Xplained ample 1 - XMEGA-A1 Xplained ample 2 - XMEGA-A1 Xplained r XMEGA DES driver - XMEGA-A1 Xplained or ADC using DAC - XMEGA-A1 Xplained or DAC using ADC - XMEGA-A1 Xplained or DAC using ADC - XMEGA-A1 Xplained or TWI - XMEGA-A1 Xplained DMA module - XMEGA-A1 Xplained ple - XMEGA-A1 Xplained S driver example3 - XMEA S driver example3 - XMEA S driver example3 - XMEA	×	USAKI EXAmple - USAKI EXAmple - USAKI EXAMPLE - WIEGA-AI Xplained USAKI SATE - a received character and echoes the character back to the same USAKI (USARI Example - XIMEGA-AI Xplained - XMEGA-AI XPlained W
Project Name:	USART EXAMPLE1			
Location:	D:\My Documents\AtmelStu	dio		Browse
Solution:	Create New Solution			•
Device:	ATxmega128A1			
				<u>O</u> K <u>C</u> ancel



	、ックするこ 図 2-3.ASF例 Solution Explore ウィントゥ
とが可能です。ATMEL ASFファイルは srcフォルタで新 L Studioプロジェクikpに引き入れられ、インポー にわ、ま	しいATME す。例 ファイ
ld src¥フォルタに置かれます。依存 ファイルは src¥as¥	2 s / 3 / 7 T - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -
かれます。PCにデバッカを接続してASP例を走らせて	ください。 Solution 'USART_EXAMPLE1' (1 project)
	Dependencies
	Cutput Files
	a 🗁 src
	<ul> <li>ast</li> <li>age config</li> </ul>
	in conf_board.h
	conf_clock.h
	conf_usart_example.h
	i usart_example.c
	ASF Explorer
3.今や プロジェクをコンパイル、書き込み設定、そして 図	외 2-4.ASF Expbre視野
ァハックタることか可能です。AIMEL ASF Jァ1ル は srcフォルダで新しいATMEL Studioプロジェク内	SE Explorer 🚽 🛛
	i aproci
に引き入れられ (インポ- にれ ます。例 ファイ ルは src	ở ≜‡ <b>⊞</b>
に引き入れられ (インポー lされ ます。例 ファイ llは src ¥フォルタに置かれます。依存 ファイllは src¥as¥フォル タに置かれます。PCに デバッカを接続して ASF例	
に引き入れられ (インポー lされ ます。例 ファイ llは src ¥フォllタに置かれます。依存 ファイllは src¥as¥フォll タに置かれます。PCに デバッカを接続して ASF列 を走らせてください。	<ul> <li>⋧ \$ :::::::::::::::::::::::::::::::::::</li></ul>
に引き入れられ(インポ- lされます。例ファイ ルは src ¥フォルタに置かれます。依存ファイルは src¥as¥フォル タに置かれます。PCにデバッカを接続してASF例 を走らせてください。 4.ASF ExploreはAPと資料への容易なアクセスを持 つASEコードの論理的 ウィンレ゙がです	<ul> <li>⋧‡ ва</li> <li>▲ USART_EXAMPLE1</li> <li>▶ ■ Generic board support</li> <li>▶ ■ GPIO - General purpose Input/Output</li> <li>▶ ■ IOPORT - Input/Output Port Controller</li> <li>■ USART_Heimerel Sensitives (American President Terresitter)</li> </ul>
<ul> <li>に引き入れられ (インポ- lされ ます。例 ファイ ルは src ¥フォルタに置かれます。依存 ファイルは src¥as¥フォル タに置かれます。PCに デバッカを接続して ASF例 を走らせてください。</li> <li>4.ASF Exploreは AP と資料への容易な アクセ スを持 つ ASF コ- トの論理的 ウィン ドウです。</li> </ul>	<ul> <li>2 2 E</li> <li>USART_EXAMPLE1</li> <li>E Generic board support</li> <li>E GPIO - General purpose Input/Output</li> <li>E IOPORT - Input/Output Port Controller</li> <li>E USART - Universal Synchronous/Asynchronous Receiver/Transmitter</li> <li>API Documentation</li> </ul>
に引き入れられ (インホ <sup>2</sup> - lされ ます。例 ファイ ルは src ¥フォルタに置かれます。依存 ファイルは src¥as¥フォル タに置かれます。PCに デバッカを接続して ASF例 を走らせてください。 4.ASF Exploreは AP ビ資料への容易な アクセ スを持 つ ASF コ - トの論理的 ウィン ドウです。	<ul> <li>2 2 1 E</li> <li>USART_EXAMPLE1</li> <li>Generic board support</li> <li>GPIO - General purpose Input/Output</li> <li>IOPORT - Input/Output Port Controller</li> <li>USART - Universal Synchronous/Asynchronous Receiver/Transmitter</li> <li>API Documentation</li> <li>usart.h</li> </ul>
に引き入れられ (インポー lをれ ます。例 ファイ ルは src ¥フォルタに置かれます。依存 ファイルは src¥as¥フォル タに置かれます。PCに デバッカを接続して ASF例 を走らせて ください。 4.ASF Exploreは AP と資料への容易な アクセ スを持 つ ASF コー トの論理的 ウィン ドウです。	<ul> <li>2 2 1 En</li> <li>USART_EXAMPLE1</li> <li>Generic board support</li> <li>GPIO - General purpose Input/Output</li> <li>IOPORT - Input/Output Port Controller</li> <li>USART - Universal Synchronous/Asynchronous Receiver/Transmitter</li> <li>API Documentation         <ul> <li>usart.h</li> <li>IOPORT - Input/Output Port Controller</li> <li>MOPORT - Input/Output Port Controller</li> <li>MOPORT - Input/Output Port Controller</li> <li>MOPORT - Input/Output Port Controller</li> </ul> </li> </ul>
に引き入れられ (インポー lをれ ます。例 ファイ ルは src ¥フォルタに置かれます。依存 ファイルは src¥as¥フォル タに置かれます。PCに デバッカを接続してASF例 を走らせてください。 4.ASF Exploreは AP と資料への容易な アクセ スを持 つ ASF コー トの論理的 ウィン ドウです。	<ul> <li>2 2 1 En</li> <li>USART_EXAMPLE1</li> <li>Generic board support</li> <li>GPIO - General purpose Input/Output</li> <li>IOPORT - Input/Output Port Controller</li> <li>USART - Universal Synchronous/Asynchronous Receiver/Transmitter</li> <li>API Documentation</li> <li>usart.h</li> <li>IOPORT - Input/Output Port Controller</li> <li>IOPORT - Input/Output Port Controller</li> <li>System Clock Control - XMEGA A1/A3/A3B/A4/D implementation</li> </ul>
に引き入れられ (インポー lをれ ます。例 ファイ ルは src ¥フォルタに置かれます。依存 ファイルは src¥as¥フォル タに置かれます。PCに デバッカを接続してASF例 を走らせてください。 4.ASF Exploreは AP と資料への容易な アクセ スを持 つ ASF コー トの論理的 ウィン ドウです。	<ul> <li>2 2 1 E</li> <li>USART_EXAMPLE1</li> <li>Generic board support</li> <li>GPIO - General purpose Input/Output</li> <li>IOPORT - Input/Output Port Controller</li> <li>USART - Universal Synchronous/Asynchronous Receiver/Transmitter</li> <li>API Documentation</li> <li>usart.h</li> <li>IOPORT - Input/Output Port Controller</li> <li>IOPORT - Input/Output Port Controller</li> <li>System Clock Control - XMEGA A1/A3/A3B/A4/D implementation</li> <li>XMEGA compiler driver</li> </ul>
に引き入れられ (インポー lされ ます。例 ファイ ルは src ¥フォルタに置かれます。依存 ファイルは src¥as¥フォル タに置かれます。PCに デバッカを接続して ASF列 を走らせて ください。 4.ASF Exploreは AP と資料への容易な アクセ スを持 つ ASF コー トの論理的 ウィン ドウです。	<ul> <li>State Depicts</li> <li>State De</li></ul>
に引き入れられ (インポ- l>れます。 例 ファイ ルは src ¥フォルタに置かれます。 依存 ファイルは src¥as¥フォル タに置かれます。 PCに デバッカを接続して ASF例 を走らせて ください。 4.ASF Exploreは AP と資料への容易な アクセ スを持 つ ASF コ - トの論理的 ウィン ドウです。	<ul> <li>2 2 2 E</li> <li>USART_EXAMPLE1</li> <li>Generic board support</li> <li>GPIO - General purpose Input/Output</li> <li>IOPORT - Input/Output Port Controller</li> <li>USART - Universal Synchronous/Asynchronous Receiver/Transmitter</li> <li>API Documentation</li> <li>usart.h</li> <li>IOPORT - Input/Output Port Controller</li> <li>IOPORT - Input/Output Port Controller</li> <li>System Clock Control - XMEGA A1/A3/A3B/A4/D implementation</li> <li>XMEGA compiler driver</li> </ul>
に引き入れられ (インポー lをれ ます。例 ファイ ルは src ¥フォルタに置かれます。依存 ファイルは src¥as¥フォル タに置かれます。PCに デバッカを接続して ASF例 を走らせて ください。 4.ASF Exploreは AP と資料への容易な アクセ スを持 つ ASF コー トの論理的 ウィン ドウです。	<ul> <li>2 2 2 En</li> <li>USART_EXAMPLE1</li> <li>Generic board support</li> <li>GPIO - General purpose Input/Output</li> <li>IOPORT - Input/Output Port Controller</li> <li>USART - Universal Synchronous/Asynchronous Receiver/Transmitter</li> <li>API Documentation</li> <li>usart.h</li> <li>IOPORT - Input/Output Port Controller</li> <li>IOPORT - Input/Output Port Controller</li> <li>System Clock Control - XMEGA A1/A3/A3B/A4/D implementation</li> <li>XMEGA compiler driver</li> </ul>
に引き入れられ (インポー lをれ ます。例 ファイ ルは src ¥フォルタに置かれます。依存 ファイルは src¥as¥フォル タlc置かれます。PCに デバッカを接続してASF例 を走らせてください。 4.ASF Exploreは AP と資料への容易な アクセ スを持 つ ASFコードの論理的 ウィンドウです。	<ul> <li>State Control</li> <li>State Co</li></ul>
に引き入れられ (インポ- Юれます。例 ファイ Wは src ¥フォルタに置かれます。依存 ファイWは src¥as¥フォル タに置かれます。PCに デバッカを接続して ASF例 を走らせて ください。 4.ASF Exploreは AP と資料への容易な アクセ スを持 つ ASF コ- トの論理的 ウィン ドウです。	<ul> <li>Structure</li> <li>Structure</li> <li>Structure</li> <li>USART_EXAMPLE1</li> <li>Generic board support</li> <li>GPIO - General purpose Input/Output</li> <li>IOPORT - Input/Output Port Controller</li> <li>USART - Universal Synchronous/Asynchronous Receiver/Transmitter</li> <li>API Documentation</li> <li>usart.h</li> <li>IOPORT - Input/Output Port Controller</li> <li>IOPORT - Input/Output Port Controller</li> <li>System Clock Control - XMEGA A1/A3/A3B/A4/D implementation</li> <li>XMEGA compiler driver</li> </ul>
に引き入れられ (インポ- Юれます。例 ファイ Wは src ¥フォルタに置かれます。依存 ファイWは src¥as¥フォル タに置かれます。PCに デバッカを接続して ASF例 を走らせて ください。 4.ASF Exploreは AP と資料への容易な アクセ スを持 つ ASF コ- トの論理的 ウィン ドウです。	<ul> <li>Struct</li> <li>Struct</li> <li>Struct</li> <li>USART_EXAMPLE1</li> <li>Generic board support</li> <li>GPIO - General purpose Input/Output</li> <li>IOPORT - Input/Output Port Controller</li> <li>USART - Universal Synchronous/Asynchronous Receiver/Transmitter</li> <li>API Documentation</li> <li>usart.h</li> <li>IOPORT - Input/Output Port Controller</li> <li>IOPORT - Programmable Multi-level Interrupt Controller</li> <li>System Clock Control - XMEGA A1/A3/A3B/A4/D implementation</li> <li>XMEGA compiler driver</li> </ul>





24.ASF例の資料取得

- ・プロジェク 名上で右 クロックしてください。
- ・"View ASF Project Example help"を選択し てください。
- 図 2-5.ASF例資料



## 25.既存 プロジェクトヘの ASF単位部追加

AVR4029

どのプロジェクトヘもASF単位部の追加 削除が可能です。

1.プロジェク はたは Projectメニュー上で右 クリックし、そして "ASFW izar d"上を クリックして ください。

#### 図 2-6.プロジェク への ATMEL ASF単位部追加

Solution Explorer		
🔁   🗗		
Solution 'USART	_EXA	MPLE1' (1 project)
USART_EX^	MADI.	Build
i Output		Rebuild
a 🗁 src		Clea <u>n</u>
⊳ 💼 asf ⊿ 💯 con		Add
ы		Set as St <u>a</u> rtUp Project
<u>h</u>	8.	ASF Wizard
asf.	0	View ASF Example Project Help
💁 usai	Ж	Cu <u>t</u>
	×	Remove
		Rena <u>m</u> e
		Unload Project
	•	Properties
ASF Explorer 💐 So	lutio	n Explorer



2.1つまたは多数のASF単位部を選択し、そして "Add to selection"上で クリックしてください。例えは タイマ / カウンタ ドライハ です。

図 2-7.7 ロジェクトヘの ATMEL ASF単位部追加

Available Modules     Selected Modules       Show:     All     Search for modules       Name     Type     Licence       Name     Type     Licence       GFX Monochrome - Monochrome Graphic Library service     ASE     IDPORT - Input/Output (servic)       GFX Monochrome - Spinner/Spin control widget     service     ASE       GFX Monochrome - System Font     service     ASE       Huge Memory Interface     service     ASE       Interrupt management (Common API)     driver     ASE       Memory Control Access Interface     service     ASE       NVM - Non Volatile Memory     driver     ASE       NVM - Non Volatile Memory     driver     ASE       VMM service using timer/counter     service     ASE       Sensors - ATAVRSBIN1 Sensor Board     component     ASE	e)
Show:       All       Search for modules         Name       Type       Licence         GFX Monochrome - Monochrome Graphic Library service       ASF         GFX Monochrome - Spinner/Spin control widget service       ASF         GFX Monochrome - Spinner/Spin control widget service       ASF         GFX Monochrome - System Font       service         LCD - ST7565R Controller       component         LCD - ST7565R Controller       component         MVM - Non Volatile Memory       driver         PMIC - Programmable Multi-level Interrupt contructriver       ASF         PMIC - Programmable Multi-level Interrupt contructriver       ASF         Sensors - ATAVRSBIN1 Sensor Board       component         Sensors - ATAVRSBIN1 Sensor Board       component         ASF       sensore	e)
Name     Type     Licence       GFX Monochrome - Monochrome Graphic Library service     ASF       GFX Monochrome - Spinner/Spin control widget     service       GFX Monochrome - System Font     service       GFX Monochrome - System Font     service       Muge Memory Interface     service       Memory Control Access Interface     service       Memory Control Access Interface     service       NVM - Non Volatile Memory     driver       PMIC - Programmable Multi-level Interrupt Controlrivirer     ASF       PVM - service     ASF       PVM - service     ASF       PVM - Service     ASF       Sensors - ATAVRSBIN1 Sensor Board     component       ASF     senvice       Sensors - ATAVRSBIN1 Sensor Board     component       ASF     sensore	
GFX Monochrome - Monochrome Graphic Library service       ASE         GFX Monochrome - Spinner/Spin control widget service       ASE         GFX Monochrome - System Font       service       ASE         GFX Monochrome - System Font       service       ASE         Muge Memory Interface       service       ASE         Interrupt management (Common API)       driver       ASE         Memory Control Access Interface       service       ASE         NVM - Non Volatile Memory       driver       ASE         PWM service using timer/counter       driver       ASE         PWM service using timer/counter       driver       ASE         Sensors - ATAVRSBIN1 Sensor Board       component       ASE         Sensors - ATAVRSBIN1 Sensor Board       component       ASE         Sensors - ATAVRSBINA1 Sensor Board       component       ASE         Sensors - ATAVRSBINA1 Sensor Board       component       ASE	r)
SFX Monochrome - Spinner/Spin control widget service       ASE         SFX Monochrome - System Font       service       ASE         Huge Memory Interface       service       ASE         Interrupt management (Common API)       driver       ASE         VMO - Non Volatile Memory       driver       ASE         VMM - Non Volatile Memory       driver       ASE         VMC - Programmable Multi-level Interrupt Controdriver       ASE         VMT - Non Volatile Memory       driver       ASE         VMT - Sensor Board       component       ASE         iensors - ATAVRSBIN1 Sensor Board       component       ASE         iensors - ATAVRSBIN1 Sensor Board       component       ASE         iensors - ATAVRSBINA1 Sensor Board       component       ASE	us Receiver/Transmitter (dr
SFX Monochrome - System Font service ASF Huge Memory Interface service ASF nterrupt management (Common API) driver ASF (C) - ST7565K Controller component ASF Wemory Control Access Interface service ASF WTM - Non Volatile Memory driver ASF WTM - Non Volatile Memory driver ASF WTM service using timer/counter service ASF IC - Real Time Counter driver ASF iensors - ATAVRSBIN1 Sensor Board component ASF	is needed and inter (an
Huge Memory Interface     service     ASE       Interrupt management (Common API)     driver     ASE       C.D - ST7565R Controller     component     ASE       Wemory Control Access Interface     service     ASE       WW - Non Volatile Memory     driver     ASE       VMC - Programmable Multi-level Interrupt Controlriver     ASE       VMC - Real Time Counter     service     ASE       Versors - ATAVRSBIN1 Sensor Board     component     ASE       Sensors - ATAVRSBIN1 Sensor Board     component     ASE       Sensors - ATAVRSBIN1 Sensor Board     component     ASE	
Interrupt management (Common API) driver ASF C.C - ST7565R Controller component ASF Wemory Control Access Interface service ASF VVM - Non Volatile Memory driver ASF PMIC - Programmable Multi-level Interrupt Controdriver ASF PWM service using timer/counter service ASF EXEC - Real Time Counter driver ASF Sensors - ATAVRSBIN1 Sensor Board component ASF Sensors - ATAVRSBIN2 Sensor Board component ASF Sensors - ATAVRSBIN2 Sensor Board component ASF	
LCD - ST7565R Controller     component     ASF       Wemory Control Access Interface     service     ASF       VVM - Non Volatile Memory     driver     ASF       PMIC - Programmable Multi-level Interrupt Controdriver     ASF       VVM service using timer/counter     service       StG- Real Time Sounder     driver       ASTG- Real Time Sounder     driver       Sensors - ATAVRSBIN1 Sensor Board     component       ASF     ASF	
Memory Control Access Interface     service     ASE       VIVM - Non Volatile Memory     driver     ASE       MIC - Programmable Multi-level Interrupt Controlriver     ASE       WM service using timer/conter     service     ASE       VIC - Real Time Counter     driver     ASE       einsors - ATAVRSBIN1 Sensor Board     component     ASE       einsors - ATAVRSBIN2 Sensor Board     component     ASE	
IVM - Non Volatile Memory     driver     ASE       MIC - Programmable Multi-level Interrupt Contrictiver     ASE       WM service using timer/counter     service     ASE       Cor- Real Time Counter     driver     ASE       einsors - ATAVRSBIN1 Sensor Board     component     ASE       einsors - ATAVRSBIN2 Sensor Board     component     ASE       einsors - ATAVRSBIN2 Sensor Board     component     ASE	
MIC - Programmable Multi-level Interrupt Controdriver ASF WM service using timer/counter service ASF ensors - ATAVRSBIN1 Sensor Board component ASF ensors - ATAVRSBIN2 Sensor Board component ASF ensors - ATAVRSBIN2 Sensor Board component ASF	
WM service using timer/counter     service     ASE       ITC - Real Time Counter     driver     ASE       iensors - ATAVRSBIN1 Sensor Board     component     ASE       iensors - ATAVRSBIN2 Sensor Board     component     ASE       iensors - ATAVRSBIN2 Sensor Board     component     ASE	
RTC - Real Time Counter         driver         ASE           Sensors - ATAVRSBIN1S Sensor Board         component         ASE           Sensors - ATAVRSBIN2 Sensor Board         component         ASE           Sensors - ATAVRSBIN2 Sensor Board         component         ASE	
iensors - ATAVRSBINI Sensor Board component ASE iensors - ATAVRSBINI Sensor Board component ASE iensors - ATAVRSBINI Sensor Board component ASE	
iensors - ATAVRSBIN2 Sensor Board component <u>ASF</u>	
ensors - ATAVRSBINA1 Sensor Board component ASE	
componente <u>mar</u>	
Sensors - ATAVRSBLP1 Sensor Board component ASE	
Sensors - ATAVRSBPR1 Sensor Board component ASF +	
Add to selection >>	tion

3. "Next"を クリックしてください。これは現在の プロジェク ルに対する ファイ ルの追加 削除を要約します。

4.最後に、"Finish"を クリックして ください。例えば、実時間計数器 (RTC) ドライバの rtc.cと rtc.hの ファイルが src¥as¥xmega¥drivers¥rtcフォル タに追加されます。

<ul> <li>i i i i i i i i i i i i i i i i i i i</li></ul>	Solution Exp	plorer
<ul> <li>asf</li> <li>common</li> <li>common<!--</th--><th></th><th></th></li></ul>		
<ul> <li>common</li> <li>common</li> <li>commons</li> <li>boards</li> <li>codrivers</li> <li>adc</li> <li>adc</li> <li>acpu</li> <li>acpu</li> <li>acpu</li> <li>acput</li> <li>acput</li></ul>	4	asf
<ul> <li>image</li> &lt;</ul>		> ommon
<ul> <li>boards</li> <li>crowdrivers</li> <li>adc</li> <li>cpu</li> <li>cpu</li> <li>ioport</li> <li>o nvm</li> <li>o pmic</li> <li>rtc.</li> <li>rtc.</li> <li>rtc.h</li> <li>sleep</li> <li>co usart</li> <li>usart.h</li> <li>o utils</li> </ul>		a 🙆 xmega
<ul> <li>drivers</li> <li>adc</li> <li>cpu</li> <li>ioport</li> <li>nvm</li> <li>pmic</li> <li>prtc</li> <li>rtc.c</li> <li>rtc.h</li> <li>sleep</li> <li>io usart</li> <li>usart.c</li> <li>usart.h</li> <li>usart.h</li> </ul>		boards
<ul> <li>adc</li> <li>cpu</li> <li>ioport</li> <li>nvm</li> <li>pmic</li> <li>prtc.</li> <li>rtc.c</li> <li>rtc.h</li> <li>sleep</li> <li>io usart</li> <li>usart.c</li> <li>usart.h</li> <li>usart.h</li> </ul>		a 🙆 drivers
<ul> <li>cpu</li> <li>ioport</li> <li>nvm</li> <li>pmic</li> <li>rtc</li> <li>rtc.c</li> <li>rtc.h</li> <li>sleep</li> <li>sleep</li> <li>io usart.c</li> <li>usart.h</li> <li>usart.h</li> </ul>		adc
<ul> <li>ioport</li> <li>nvm</li> <li>pmic</li> <li>rtc</li> <li>rtc.c</li> <li>rtc.h</li> <li>sleep</li> <li>io sleep</li> <li>io usart</li> <li>usart.h</li> <li>usart.h</li> </ul>		p i cpu
<ul> <li>invm</li> <li>in pmic</li> <li>in rtc.</li> <li>in rtc.h</li> <li>in sleep</li> <li>in usart.</li> <li>in usart.h</li> <li>in usart.h</li> </ul>		b ioport
<ul> <li>pmic</li> <li>rtc</li> <li>rtc.c</li> <li>rtc.h</li> <li>sleep</li> <li>isleep</li> <li>isusart.c</li> <li>usart.h</li> <li>usart.h</li> </ul>		D I I I I I I I I I I I I I I I I I I I
<ul> <li>Interview of the second second</li></ul>		pmic
<ul> <li>c] rtc.c</li> <li>b] rtc.h</li> <li>▷ □ sleep</li> <li>⊿ @ usart</li> <li>c] usart.c</li> <li>b] usart.h</li> <li>▷ □ utils</li> </ul>		a 🙆 rtc
<ul> <li>▶ intc.h</li> <li>▶ intc.h</li> <li>▶ intc.h</li> <li>▲ intcine</li> <li>▶ usart.c</li> <li>▶ usart.h</li> <li>▶ intcine</li> </ul>		c) rtc.c
<ul> <li>isleep</li> <li>isleep</li> <li>isleep</li> <li>usart</li> <li>usart.c</li> <li>usart.h</li> <li>usart.h</li> </ul>		h rtc.h
<ul> <li>✓ i usart</li> <li>☑ usart.c</li> <li>☑ usart.h</li> <li>☑ utils</li> </ul>		b is sleep
i usart.c i usart.h ⊳ i utils		a 🙆 usart
bù usart.h ⊳ i⊡ utils		💁 usart.c
b in utils		🔥 usart.h
		b a utils
ASE Explorer Solution Explorer	ASF Explor	er Solution Explorer







26.雛形での新しいプロジェク開始:使用者応用雛形

使用者応用雛形は以下とでプロジェクを使用する準備が整っています。

- ・main.cファイルとmain(関数
- ・選択したATMEL MCU用の src¥as¥フォルタに置かれたASFドライバ汎用入出力、割り込み の基本的な組
- ・ASF基板定義 ファイル。例えば、ATMEL XMEGA-A1 Xplainedキッルこついては、src¥asf¥xmega¥boards¥xmega\_a1\_xpalined¥xmega\_a1\_xpalined±xmega\_xpalined±xpalined±xpalined±xpalined±xpalined±xpalined±xpalined±xpalined±xpalined±xpalined±xpalined±xpalined±xpalined±xpalined±xpalined±xpali

使用者応用雛形にアクセスするのは次の通りです。

- 1.File New Projecを加かしてください。
- 2.C/C++ A the I boards メニューで、望む ATMEL基板用の使用者応用雛形を選択してください。

図 2-9.ATMEL基板用の新し	\使用者雛形	プロジェクト
-------------------	--------	--------

Recent Templates		Sort by:	Default 🔹 📗		S	earch Installed Templates	1
Installed Template	s	XMEGA	User application template - STK600 - ATxmega32A4U	C/C++		Type: C/C++	ontains
Atmel Boards User Boards Assembler Atmel Studio Solutio	s	XMEGA	User application template - STK600 - ATxmega384C3	C/C++		the bare minimum of files needed a project which works with the AS application builder wizard. [User	d to star SF
	lution	XMEGA	User application template - STK600 - ATxmega64A3	C/C++		application template - XMEGA-A Xplained - ATxmega128A1]	1
		UCSA	User application template - UC3-A3 Xplained	C/C++			
		(MCAC	User application template - UC3C-EK	C/C++			
		UCAL	User application template - UC3-L0 Xplained	C/C++			
		UCAL	User application template - UC3L-EK	C/C++			8
		XMEGA	User application template - XMEGA-A1 Xplained	C/C++			
		XMEGA	User application template - XMEGA-A3BU Xplained	C/C++	ш	Alter Lagrangem 7 (1993)	
		XMEGA	User application template - XMEGA-B1 Xplained	C/C++			
		XMEGA	User application template - Xplain	C/C++	*		
<u>N</u> ame:	USER_APPLICATI	ON1					
Location:	D:\\My Documer	nts\Atmel	Studio		Browse		
Solution:	Create new soluti	on		•			
Solution name:	USER APPLICATI	ON1			V	Create directory for solution	

### 3.OKをクリックしてください。



main.c X	<ul> <li>Solution Explorer</li> </ul>	<b>→</b> ‡ ×
main.c	↓ • <mark>(°Go</mark> 🖾 🗳	
<pre>/**  * file  *  * file  *  * file  *  * brief Empty user application template  *  */ */ * Include header files for all drivers that have been imported from  * Atmel Software Framework (ASF).  */ #include (asf.h) int main (void)  {     board_init();     // Insert application code here, after the board has been initialized }</pre>	Dependencies     Output Files     Output Files     Output Files     or src     or gr sef     or gr mega     or gr mega     or gr mega al_xplained     or gr mega al_xplained     or gr vires     on utils     or gr vires     on utils     or gr vires     or gr vires	- #×

AVR402

AVR4029

## 3.ASFと AR Embedded Workbenchでの開始に際して

## 3.1. インストール

- 1.http://www.iar.comで AR Embedded Workbenct®をダウンロードしてください。
- 2. ARを インストールして ください。
- 3.http://www.atmel.com/asからASF独立書庫 (zipファイルをダウンロートしてください。
- 4.ハードドライブでASF独立書庫を展開してください Window®での潜在的な長い名前長の問題を避けるために、ルー Iに近い場所でが 望まれます )

## 32.ヘッダファイル更新

8ビットAVR使用者に関して、ツールチェーン ヘッタ・ファイルの更新が推奨され、それを行う方法についての記述は xmega¥utils¥header\_files¥ フォルタ下の readme.txtファイルに置かれます。

AVR UC3使用者に関して、ARヘッタ<sup>\*</sup>ファイルの更新が推奨されます。これを行うには、(avr32¥utils¥header\_files¥フォルタ<sup>\*</sup>に置かれた avr32-headers.zipファイルを "ARインス トルフォルタ<sup>\*</sup>下の AR EWAVR32インス トルフォルタ<sup>\*</sup>(¥Embedded Workbench x.y¥avr32¥inc¥)に展開 してください。

## 33.ASF独立書庫内の移動

先頭フォルタ構成は次の通りです。

- ・avr32¥フォルタはAVRUC3デル゙イZ専用のソフトウェア単位部(ソース コートとプロジェクトを含みます。
- ・mega¥フォルタはATMEL megaAVRデバイZ専用のソフトウェア単位部 (ソース コートとプロジェクトを含みます。
- ・xmega¥フォルタはATMEL AVR XMEGAデバイZ専用のソフトウェア単位部 (ソース コートとプロジェクトを含みます。
- ・common¥フォルタは全てのATMELAVRデル゙イスによって共用されるソフトウェア単位部(ソース コートとプロジェクトを含みます。
- ・sam¥フォルタはATMEL SAMデバイフ専用のソフトウェア単位部(ソース コートとプロジェクトを含みます。
- ・thirdparty¥フォルタは全てのAVRデバイ及する第三者の提供者からのソフやェア単位部(ソース コートとプロジェクトを含みます。 thirdparty¥フォルタは第三者のソフやェア単位部即ち、第三者のソフやェア単位部当たり1つのフォルダ)によって構成されます。 avr32¥ xmega¥ sam¥ common¥フォルタは次のように構成されます。
- ・drivers¥フォルタはATMEL MCUチップ上の資源用の低位 ソフ トウェア ドライバ例えば、CPU、USART、ADC ドライバを含みます。
- ・boards¥フォルタは基板特有 ソース コードファイルを含みます。
- ・utilstフォルタは他の全ての単位部によって使用されるファイルを含みます。それは様々なリンカ スクリプト ファイル、いくつかの ASF単位部の ARとGCCの予め コンパイルされた ライブラリ そして全般用法定義、マクロ 関数を持つ C /C++ューティリティ ファイルを保持します。
- services¥フォルタは基板や チッフを特定しない応用志向の ソフ トウェアの一部 例えば、FAT、TCP/ Pスタック OS JPEC復号 を含みます。
   common¥先頭 フォルタに関しては、services¥フォルタが チッフ特有の コー トも含みます。
- ・components¥フォルタは支援される各 ハートウェア部品に関し、その部品 (SDRAM、SDカートのような メモリ または表示器 と相互作用する ソフトウェア インターフェー スを含みます。
- ・applications¥フォルダはサービスとドライバを使用する応用の大量の例を含みます。

### 34.ASF例の開始

ATMEL AVR UC 3汎用入出力 (GPD) ドライル単位部の使い方の例を使用すると ARプロジェクは以下に置かれます。

- ・ATMEL EVK1100基板については、avr32¥drivers¥gpio¥peripheral\_bus\_examp b¥at32uc3a0512\_evk1100¥ ar¥
- ・ATMEL EVK1105基板については、avr32¥drivers¥gpio¥peripheral\_bus\_examp b¥at32uc3a0512\_evk1105¥ ar¥
- ・ATMEL EVK110 基板については、avr32¥drivers¥gp io¥peripheral\_bus\_examp le¥at32uc3a0256\_evk1101¥ iar¥
- ・ATMEL EVK110/基板については、avr32¥drivers¥gpib¥peripheral\_bus\_examp b¥at32uc3a3256\_evk1104¥ iar¥
- ・ATMEL STK® 600-RCUC3Lの構成設定については、avr32¥drivers¥gpio¥peripheraLbus\_examp le¥at32uc3 1064\_stk600-rcuc3 10¥ iar¥
- ・ATMEL AT32UC3C-EK基板については、avr32¥drivers¥gpib¥peripheral\_bus\_examp le¥at32uc3c0512c\_uc3c\_ek¥iar¥
- ・ATMEL AT32UC3L-EK基板については、avr32¥drivers¥gpib¥peripheral\_bus\_examp b¥at32uc3 l064\_uc3 l\_ek¥ ar¥
- 各 ar¥フォルタは以下と共に完全な ARプロジェクを含みます。
- ・ AR EW AVR32作業空間 (Workspace)ファイル (avr32\_drivers\_gpio\_peripheral\_bus\_example\_uc3l\_ek.eww :完全なプロジェクを開くには このファイル上をダブル クリックしてください。)
- ・ AR EWAVR327 ロシェクトファイル (avr32\_drivers\_gp io\_peripheral\_bus\_examp le\_uc31\_ek.ewp)
- ・ARWWAVR32デバック形態設定 ファイル (avr32\_drivers\_gpio\_peripheral\_bus\_example\_uc31\_ek.ewd)





図 3-1.ASFでの ARプロジェクl位置					
asf- 3.0.0	^	Name	Date modified	Туре	Size
<ul> <li>avr32</li> <li>avr32</li> <li>avr32</li> <li>boards</li> <li>boards</li> <li>components</li> </ul>		asf.h avr32_drivers_adc_example_evk1100.ewd avr32_drivers_adc_example_evk1100.ewp avr32_drivers_adc_example_evk1100.ewp avr32_drivers_adc_example_evk1100.ewp	26/05/2011 15:35 26/05/2011 15:35 26/05/2011 15:35	H File EWD File EWP File	3 KB 16 KB 41 KB
<ul> <li>docsrc</li> <li>drivers</li> <li>abdac</li> <li>acifa</li> <li>acifb</li> <li>acifb<td>Ш</td><td>avisz_unvers_aut_example_evittuu.eww</td><td>20/03/2011 13:33</td><td>AN UP WORSPace</td><td>1 KB</td></li></ul>	Ш	avisz_unvers_aut_example_evittuu.eww	20/03/2011 13:33	AN UP WORSPace	1 KB
iar     iar     iar     iar     iat32uc3a3256_evk1104     iat32uc3a3256_uc3-a3_xplained     iat32uc3a0256_evk1101					

## 35.ASFプロジェク資料の取得

### 35.1.オンライン資料

全てのASF単位部と参照応用のDoxyger資料はhttp://asf.atmel.comで得られます。

#### 352.手動資料生成 (オフライン)

全ての単位部は doxyger付箋を用いて完全に資料化されています。ATMEL ASF内の各 プロジェク lは htm資料を生成するための (E しい資料生成のために doxygerを形態設定するのに使用される )doxy file.doxygerを含み、doxygerが インストルされて (http://www.dox ygen.org/down bad.htmをご覧 ください) doxy file.doxygerは doxygerに対する入力形態設定 ファイルとして使用されなければなりません。

例として ATMEL AVR UC3 GPD ドライル単位部の使い方の例を使用すると、ATMEL AT 32UC3C - EK基板に対して、連携する doxyfil e.doxygen ファイルは av r32¥dr ivers¥gp io¥per ipheral\_bus\_examp le¥at 32uc3c0512c\_uc3c\_ek¥doxygen¥フォルタ下です。

doxygerを走らせて doxygerに対する形態設定 ファイルとしてこの doxyfile.doxygerを使用してください。

コマンド行を使用し、これは以下の命令で行われます。

資料の入口ファイルは以下です。

avr32/drivers/gpio/peripheral\_bus\_example/at32uc3c0512c\_uc3c\_ek/doxygen/html/index.html

## 4.ASFとGNU makefileでの開始に際して

#### 4.1. インストール

1.http://www.atmel.com/asからATMELASF独立書庫 (zipファイルをダウンロートしてください。

- 2. ハードトライブ上にASF独立書庫を展開してください。
- 3.コンパイルとプログラミング ツールのためにATMELAVR Studi® 5(http://www.atmel.com/avrstudio5をインストルしてください。
- 4. ソフ や IP か 組み構築 システムは構築 マシン上で インストールされたいくつかの基本的な ソフ や IP ッールを持つ ど反定 します。それらの ッールは 最近の殆どオ ペレ - テ ィング システムで インストールが容易です。Microsofe Windowsシステムでそれらの ソフ や IP が 用に関する WinAVR一括 (http://winavr.sourceforge.net/)でと8ビッと32ビッ め使用者に関して ATMEL AVR GNU ッールチェーン (http://www.atm el.com/dyn/products/ tools\_card.asp?tool\_id=4118)で インストルされ、一方 Linu® システムでは通常それらが配布一括 システムを用いて 利用可能です。有用であるべき ソフ や IP が 組みのために インストルされなければならない ソフ や IP は次の通りです。
  - ・make, sed, grep, sort, tac, bc, etc 本質的要素構築として度々参照。
  - ・<mark>sh</mark> 命令解釈部 (シェル )

#### 42.ヘッダファイル更新

8ビットAVR使用者に関しては ツールチェーン ヘッダファイルの更新が推奨され、それを行う方法についての記述は xmega¥utib¥header\_files¥ フォルタ下の readme.txtファイルに置かれます。

ATMEL AVR UC3使用者に関しては、ARヘッタ<sup>\*</sup>ファイルの更新が推奨され、それを行う方法についての記述は avr32¥utib¥header\_files ¥フォルタ下の readme.txtファイルに置かれます。

# AVR4029



#### 43.ASF独立書庫内の移動

先頭フォルタ構成は次の通りです。

- ・avr32¥フォルタはAVRUC3デバイZ専用のソフトウェア単位部(ソース コートとプロジェクトを含みます。
- ・mega¥フォルタはATMEL megaAVRデバイス専用のソフトウェア単位部 (ソース コートとプロジェクトを含みます。
- ・xmega¥フォルタはATMEL AVR XMEGAデバイス専用のソフトウェア単位部 (ソース コードとプロジェクトを含みます。
- ・common¥フォルタは全てのATMELAVRデル゙イスで共用されるソフトウェア単位部(ソース コートとプロジェクトを含みます。
- ・thirdparty¥フォルタは全てのAVRデバイスに対して第3の供給者社からのソフトウェア単位部(ソース コートとプロジェクトを含みます。

thirdparty¥フォルタは第3者社)ソフトウェア単位部即ち、第3者社)ソフトウェア単位部毎に1つのフォルダ)によって構成されます。

avr32¥ xmega¥ common¥の フォルタは次のように構成されます。

・driver¥フォルタはAVRチップ上資源用の低位ソフトウェア ドライバ例えば、CPU、USART、ADCドライバを含みます。

- ・boards¥フォルタは基板特有 ソース コードファイルを含みます。
- ・utilsギフォルタは他の全ての単位部によって使用されるファイルを含みます。それは様々なリンカスクリフトファイル、いくつかのASF単位部のARとGCCの予めコンパイルされたライフ・ラリそして全般用法定義、マクロ関数を持つC/C++ユーティリティファイルを保持します。
- services¥フォルタは基板やチップを特定しない応用志向のソフトウェアの一部 例えば、FAT、TCP/Pスタック OS、JPEG復号 を含みます。 common¥先頭 フォルタに関しては、services¥フォルタがチップ特有のコードも含みます。
- ・components¥フォルタは支援される各 ハードウェア部品に関し、その部品 (SDRAM、SDカートのような メモリ または表示器 と相互作用する ソフ ŀウェア インターフェー スを含みます。

・app lications¥フォルタはサービスとドライバを使用する応用の大量の例を含みます。

#### 44.ASF例の開始

ATMEL AVR UC3汎用入出力 (GPD) ドライハ単位部の使い方の例を使用すると GCC プロジェクは以下に置かれます。

- ・ATMEL EVK1100基板については、avr32¥drivers¥gpio¥peripheral\_bus\_examp le¥at32uc3a0512\_evk1100¥gcc¥
- ・ATMEL EVK1105基板については、avr32¥drivers¥gpio¥peripheral\_bus\_examp le¥at32uc3a0512\_evk1105¥gcc¥
- ・ATMEL EVK110 堪板については、avr32¥drivers¥gp io¥peripheral\_bus\_examp le¥at32uc3a0256\_evk1101¥gcc¥
- ・ATMEL EVK1104基板については、avr32¥drivers¥gpio¥peripheral\_bus\_examp le¥at32uc3a3256\_evk1104¥gcc¥
- ・ATMEL STK 600-RCUC 3LQ構成設定については、avr 32¥drivers¥gp ib¥peripheral\_bus\_examp le¥at 32uc 3 1064\_stk 600-rcuc 3 10¥gcc¥
- ・ATMEL AT32UC3C-EK基板については、avr32¥drivers¥gpib¥peripheralbus\_examp b¥at32uc3c0512c\_uc3c\_ek¥gcc¥
- ・ATMEL AT32UC3L-EK基板については、avr32¥drivers¥gpio¥peripheral\_bus\_examp le¥at32uc3 l064\_uc3 l\_ek¥gcc¥

各gcc¥フォルタは以下と共にGCCプロジェクを含みます。

#### • mekefile

・プロジェク 形態設定 ファイルの configmk







## 45.プロジェク構築

応用を構築するには、単に適切なプロジェクトフォルダ(¥gccフォルダ)こ移行し、makeと入力してください。 例えば、ATMEL EVT1102基板上で走行するATMEL AT32UC3A0512 GPD周辺 バス例については、avr32¥drivers¥gpio¥peripheral \_bus\_examp le¥at32uc3a0512\_evk1100/gccフォルタでのmake入力に帰着します。

\$ make										
MKDIR	avr32/dr	ivers/intc/								
30	avr32/drivers/intc/intc.o									
MKDIR	avr32/drivers/gpio/									
30	avr32/drivers/gpio/gpio.o									
MKDIR	avr32/dr	ivers/gpio/p	eriphe	ral_bus	s_example/					
30	avr32/dr	ivers/gpio/p	eriphe	ral_bus	s_example/gpio_periphe					
MKDIR	avr32/ut	ils/startup/								
AS	avr32/ut	ils/startup/	ˈstartu	p_uc3.c	C					
AS	avr32/ut	ils/startup/	'trampo	line_u	c3.o					
AS	avr32/dr	ivers/intc/e	excepti	on.o						
LN	avr32_dr	ivers_gpio_p	oeriphe	ral_bus	s_example_evk1100.elf					
SIÆ	avr32_dr	ivers_gpio_p	eriphe	ral_bus	s_example_evk1100.elf					
avr32_d	rivers_gpi	io_periphera	I_bus_	example	e_evk1100.elf :					
section		size		addr						
.reset		0x2004	0x80	000000						
.text		0x134	0x80	002004						
.except	ion	0x200	0x800	002200						
.rodata		0xa0	0x80	002400						
.dalign		0x4		0x4						
.bss		0xf0		0x8						
.heap		0xef08		0xf8						
. commen	t	0x30		0x0						
.debug_a	aranges	0x158		0x0						
.debug_p	bubnames	0x3ad		0x0						
.debug_	info	0x11ca		0x0						
.debug_a	abbrev	0x554		0x0						
.debug_	line	0x1b6f		0x0						
.debug_	frame	0x294		0x0						
. debug_s	str	0x8f0		0x0						
.debug_		0x408		0x0						
.debug_r	racinto	0x2b7053		0x0						
.stack		0x1000		0001x0						
.debug_	ranges	0x170		0x0						
lotal		0x2cdce5								
tevt	data	hee	dec	hev	filename					

0x23d8 0x0 0xfffc 74708 123d4 avr32\_drivers\_gpio\_peripheral\_bus\_example\_evk1100.elf OBJDUMP avr32\_drivers\_gpio\_peripheral\_bus\_example\_evk1100.lss NM avr32\_drivers\_gpio\_peripheral\_bus\_example\_evk1100.sym OBJCOPY avr32\_drivers\_gpio\_peripheral\_bus\_example\_evk1100.hex OBJCOPY avr32\_drivers\_gpio\_peripheral\_bus\_example\_evk1100.bin

構築の結果はmakefil的で置と同じフォルタに置かれます。

- ・ELFファイル (.elf)
- ・リストファイル (.**bs**)
- ・シンボル ファイル (.sym)
- ・HEXファイル (.<mark>hex</mark>)
- ・2進ファイル (.bin)

AVR4029

46.7°D2 >>2

### 46.1.Windows使用者

46.1.1.GUに基づく解決策

- ・ATMEL Studioを開始してください。
- ・"File Open Open Object File for Debugging"を選択して generated .elf fileを選択してください。"Next"を選択してください。
- ・"Device Selection"メニューで、対応するMCUを選んでください。"Finish"をかりかしてください。
- ・プロジェクは今やプログラミンクどデバックに利用可能です。ようこそ画面からATMEL Studiの開始に際してと使用者の手引きを参照し てください。
- 46.12.コマンド行に基づくAVR使用者用解決策
- ・プログラミングューティリテ はATMEL Studioと共に インストルされます。
- ・書き込み器として 例えば C: #Program Files (x86)#A the #A the I Studio 6.0#avrdbordの ) at program exeを使用してください。
- ・例えば JTAG ICE3デバッカどでAT32UC3B0512をプログラミングするには次の通りです。

```
atprogram -t jtagice3 -i jtag -d at32uc3b0512 program -f e:¥file.elf
C: \Program Files (x86) \Atmel \Atmel Studio 6.0 \avrdbg>atprogram.exe
AVR Studio Command Line Interface
Copyright (C) 2011 Atmel Corporation.
使用法: atprogram [任意選択] 命令> []数] [命令> []数] ...]
任意選択:
  -t --tool 引数>
                             ツーI名: avrdragon, avrispmk2, avrone, jtagice3,
                             jtagicemkii, qt600, stk500, stk600 tcl samice
  -s --serialnumber 引数>
                            書き込み器 /デバッカの通番。複数のデバッカ接続時に指定されなければなりません。
  -c --comport 引数>
                            書き込み器 /デバッカに使用されるCOMポート 例えば、-c COMまたは -c 1
                            物理 インターフェース: aWire,debug/VIRE,HVPP,HVSP, ISP, JTAG, PD または TPI
  -i --interface <引数>
  -d --device 引数>
                             デル゙イス名。例えば、atxmega128a1や at32uc3a0256
```

- v	verbuse	
-h	host <引数>	avrdbg処理で走行する目的対象 ホスト
-р	port <引数>	avrdbo処理に使用されるポーを指定。ホス 任意選択が与えられない場合は無視
- f	force	ファームウェアが最新でなくても命令を強制
-cl	clock	デバイスとの通信に使用する周波数 (hz, khz, mhz, 既定は hz), 例えば、-cl 10mhz
-mb	max-baudrate 引数>	awireでの通信に対する最大 ボーレート(kbps)
-xr	external reset	作業開始時に外部 ルッを印加
-dc	daisychain ⊰ 数,>	JTAGデージーチェーン構成設定。
		引数は <devices-before devices-after="" instr-after="" instr-before=""></devices-before>
-tv	target-voltage 引数>	STK600またはSTK500の目標電圧 浮動値を設定
-a0	aref0 ⊰ 数 >	STK600の Aer 化または STK 500の are 生成器電圧 浮動値を設定
-a1	aref1 ⊰ 数>	STK6000 Aerf生成器電圧 浮動値 を設定
-cg	clock-generator <引数>	STK 600または STK 5000 加ッ焼生器周波数を設定 (hz, khz, mhz, 既定は hz)
	timeout 引数>	命令に対する秒での監視時間値設定。既定は180秒。(設定は時間監視なし。
-q	quiet	活動指示器非表示
	-v -h -p -f -cl -mb -xr -dc -xr -dc -a0 -a1 -cg -q	-vvendose -hhost 引数> -pport 引数> -fforce -clclock 引数> -mbmax-baudrate 引数> -xrexternal reset -dcdaisychain 引数,> -tvtarget-voltage 引数> -a0aref0 引数> -a1aref1 引数> -cgclock-generator 引数> - timeout 引数> -qquiet

- --quiet -q
- -? --help

```
命令:
```

•	
chiperase	チッフの完全消去
erase	指定メモ吻消去
help	指定命令に対するヘルフ表示
info	デバイスについての情報を表示
list	接続されたATMELツールについての情報を検出して表示
program	<ファイル からのデータでデル゙イ 建き込み
read	デバイスのメモリ内容読み込み
secure	UC3とARMデバイスで保護とッを設定
verify	ファイルに」基づいてメモリ内容を確認
version	版番号表示
write	命令行で入力した値で、圧虐き込み

ヘルフ情報表示





46.13.コマンド行に基づSAM使用者用解決策

ATMEL SAM -BA<sup>®</sup>実装書き込み器を参照してください。 http://www.atmel.com/tools/ATMELSAM -BAN-SYSTEMPROGRAMMER.aspx

#### 462.Linux使用者

- ・8ビッ使用者についてはMCUプログラミンクにAVRDUDE(http://www.nongnu.org/avrdude/を使用してください。
- ・32ビッIAVR使用者についてはGUに基づく解決策に関する「AVR32Studi使用者」項を参照してください。
- ・ARM使用者についてはSAM BA実装書き込み器 (http://www.atmel.com/tools/ATMELSAM BAN-SYSTEMPROGRAMMER.asp xを参照してください。

#### 47.AVR32 Studid 使用者

32L<sup>\*</sup>ットAVR使用者は32L<sup>\*</sup>ットATMEL AVR StudioでATMEL ASF GNU makefilを使用することができます。それは32L<sup>\*</sup>ットAVR Studio 内から展開されたASF一括で動くことが可能で、これは「AVR32769AVR Studio V2での独立AVR32V7 ウェア枠組みコンパ イル法」応用 記述 (http://www.atmel.com/dyn/resources/prod\_documents/doc32115.pdf)で記述されます。

48.プロジェク資料への アクセス

#### 48.1.オンライン資料

全てのASF単位部と参照応用のDoxyger資料はhttp://asf.atmel.comで得られます。

#### 482. オフライン資料

全ての単位部は doxyger付箋を用いて完全に資料化されています。ASF内の各 プロジェクは htm資料を生成するための 正しい資料 生成のために doxygerを形態設定するのに使用される )doxyfile.doxygerを含み、doxygerが インストルされて (http://www.doxygen.org/ down bad.htmをご覧ください) doxyfile.doxygerは doxygerに対する入力形態設定 ファイルとして使用されなければなりません。

例としてATMEL AVR UC3 GPDドライル単位部の使い方の例を使用すると ATMEL AT32UC3C -EK基板に対して、連携するdoxyfil e.doxygenファイルは avr32¥drivers¥gp b¥per pheral\_bus\_examp b¥at32uc3c0512c\_uc3c\_ek¥doxygen¥フォルタ下です。

doxygerを走らせて doxygerに対する形態設定 ファイルとしてこの doxy file.doxygerを使用してください。

コマンド行を使用し、これは以下の命令で行われます。

doxygen doxyfile.doxygen

AVR4029

資料の入口 ファイルは以下です。

avr32/drivers/gpio/peripheral\_bus\_example/at32uc3c0512c\_uc3c\_ek/doxygen/html/index.html



# <mark>5</mark>.目次

要点 ••••••••••••••••••••••••••••••••••••	1
1. 序説 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	1
2. ASFとATMEL Studio 6での開始に際して ······	1
21. 1×1-1/2 ·····	1
22. 映像 •••••••••••••••••••••••••••••••••••	1
23. ASF例の開始 ······	2
2.4. ASF例の資料取得 ····································	4
25. 既存 プロジェクトヘの ASF単位部追加 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	4
26. 雛形での新しいプロジェク開始 :使用者応用雛形 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	6
3. ASFと AR Embedded Workbenchでの開始に際して ·····	7
3.1. インストール・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	7
32. ヘッダファイル更新 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	7
33. ASP独立書庫内の移動 ····································	7
34. ASF例の開始 ····································	7
35. ASFプロジェク資料の取得 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	8
35.1. オンライン資料・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	8
	8
4. ASFEGNU makefileでの開始に際して ······	8
	8
42. ヘッタ ノア1ル史新 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	8
43. ASF 型 当 単 内 の 移 動 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	9
	9
45. ノロク17 開架 46. プログラミング・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	1
40. ノログノシグ 461 Windowe <sup>使田</sup> 考	1
462 Linu使用者 ····································	12
47. AVR32 Studi使用者 ····································	2
48. プロジェク 資料への アクセス・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	2
48.1. オンライ資料 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	12
<b>482. オフライン資料 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・</b>	2
5. 目次 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	3





#### Atmel Corporation

2325 O rchard Parkway San Jose, CA 95131 USA TEL (+1)(408) 441-0311 FAX (+1)(408) 487-2600 www.atmel.com

## Atmel Asia Limited

 Unit 01-5 & 16, 19F
 Business 0

 BEA Tower, Millennium City 5
 Parking 4

 418 Kwun Tong Road
 D-85748 0

 Kwun Tong, Kow bon
 GERMAN

 HONG KONG
 TEL (+49)

 TEL (+852) 2245-6100
 FAX (+49)

#### Atmel Munich GmbH

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

#### Atmel Japan

141-0032東京都品川区 大崎1-6-4 新大崎勧業ビル16F アトメルシャルン株式会社 TEL (+81)(3)-6417-0300 FAX (+81)(3)-6417-0370

#### © 2012 Atmel Corporation. 全権利予約済

ATMEL®、ATMELロゴとそれらの組み合わせ、AVR® AVR Studio® megaAVR® ,SAM -BA® ,STK® ,XMEGA® ,それとその他は ATMEL Corporationの登録商標または商標またはその付属物です。Window®とその他は米国とその他の国に於いてMicrosoft Corporationの登録商標または商標です。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のりェブサイに位 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は重載応用に 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* 2012.

本応用記述はATMELのAVR4029応用記述(doc8431.pdfRev.8431B-03/12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、原本より須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。

青字の部分はリッとなっています。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。