

AVR®用IAR Embedded Workbench®での開始に際して

序説

著者: Alexandru Niculae, Microchip Technologies Inc. Felipe Torrezan, IAR Systems[®] AB

この応用記述の目的はMicrochip AVR®マイクロコントローラ用IAR Embedded Workbench®でのからCプロジェクトを作成し、それを構築し、マイクロコントローラに書いてそれをデバッグするのに必要とされる手順を通して新規使用者を導くことです。 これを実演するため、例として卸押下でLEDを交互切り替えする簡単な"Hellow World"応用が使われます。



この応用記述は以下を説明します。

- ・0から新しいプロジェクトを作成する方法
- ・Cソースコートをコンパイルするための説明と任意選択
- ・ATmega4809 Curiosity Nano基板の準備方法
- ・書き込みとデバッグをするためのIAR Embedded Workbenchの使い方

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、Microchip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。

1. 準備 •••••••••••••••••••••••••••••••••••
2 . 開始に際して・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
2.1. IAR Embedded Workbenchで新しい作業空間を作成 · · · · · · · · · · · · · · · · · · ·
2.2. IAR Embedded Workbenchで新しいプロジェクトを作成 ····································
2.3. IAR Embedded Workbenchで構成を設定 ····································
2.4 . ヒューズを構成設定・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
2.5. 最適化水準
2.6. スタック使用量分析と設定・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
2.7. AVR [®] 用IAR Embedded Workbenchでの書き込みとテハック ····································
3. 改訂 履歴
Microchipウェブ サイト ····· 11
製品変更通知サービス・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
お客様支援 ····································
Microchipデベイス コード保護機能・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
法的通知 ····································
商標 ······12
品質管理システム・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
世界的な販売とサービス ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・

1. 準備

この応用記述はAVR用IAR Embedded Workbench 7.20版用に書かれました。この点から、機能的なIDEが既にインストールされていると 推察されます。AVR用IAR Embedded WorkbenchインストーラはIAR Systems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。この手引き はATmega4809 Curiosity Nano基板用に書かれましたが、記述される考え方は他のAVRデバイスにも適用されます。また、基本的なプ ログラミングとマイクロコントローラの理解も当然と思われます。

2. 開始に際して

プログラム作成時、特に応用がいくつかのソースファイルになるため、プロジェクトを正しく構成することが重要になります。

IAR Embedded Workbench IDEはCまたはC++プロジェクト内容の管理を簡単にするように設計されました。IDEプロジェクト管理部の上部 抽象化は作業空間(Workspace)です。各作業空間に対して、1つ以上のプロジェクトを追加することができます。同じ作業空間に追加さ れたプロジェクトは同じ製作物に関連付けするか、または共通で他の何かを持つことができます。各プロジェクトはその設定とソースファイル の一覧を持ちます。新しい作業空間を作成し、続いて新しいプロジェックトを作成するか、または作業空間に既存のプロジェクト例を追加 してください。この応用記述は応用に対する初期ソースファイルを含む対となるZIPファイルを持ちます。ZIPファイル内容はディクトップで"AVR" と名付けられたフォルダのような既知の場所に抽出(解凍)されて保存されます。

2.1. IAR Embedded Workbenchで新しい作業空間を作成

IAR Embedded Workbenchを開いて最初のプロジェクトを作るために下の手順に従ってください。

1. File(ファイル)⇒New Workspace(新規作業空間)を選んでください。このように見える空状態が現れるでしょう。

図2-1. IAR作業空間		
IAR Embedded Workbench IDE		×
Eile Edit View Project Iools Window Help		
: b b 🗈 🖻 🕼 X 🗈 D b c		
Workspace • • • ×		
v		
Files • •		
Ready		

2.2. IAR Embedded Workbenchで新しいプロジェクトを作成

新しい作業空間が作成された後は新しいプロジェクトを作成する時間です。

1. Project(プロジェクト)⇒Create New Project(新規プロジェクト作成)を選んでください。



 Create New Project(新規プロジェクト作成)'ダイアログでEmpty Proje ct(空プロジェクト)を選んでOKを押してください。

図2-3.新しい空プロジェクトを作成

Create New Pro	ject		×
<u>I</u> ool chain:	AVB	~	
Project template	s:		
Empty asm asm C++ C C D CB DLIB Extern	project ally built executable		
Description:			
Creates an emp	ty project.		
		ОК	Cancel

- 3. 次のダイアログ枠が現れてプロジェクト ファイル(.ewp)が何処に保存されるべきかを問います。デスクトップに'AVR'と名付けた新しいフォル ダを作成してその下にHelloWorld.ewpとしてプロジェクトを保存してください。
- 4. 供給されたZIPファイルからHelloWorld.cソースファイルを'AVR'フォルダに抽出(解凍)してください。
- 5. 作業空間でプロジェクト名(HelloWorld)上を右クリックしてAdd(追加)⇒Add Files...(ファイル追加)を選んでください。
- 6. HelloWorld.cソースファイルをHelloWorldプロジェクトに追加してください。
- 7. 作業空間ウィンドウでプロジェクト名の直後に*(アスタリスク)があることに注意してください。これはプロジェクトが変更されていてその変更が保存されていないことを意味します。File(ファイル)⇒Save All(全て保存)を選んでください。今や*(アスタリスク)は消滅します。
 - 注:初めて作業空間が保存される時に.ewwファイルの名前を追加してください。

図2-4. HelloWorld.cを持つプロジェクト、HelloWorldプロジェクトに追加
HelloWorld - IAR Embedded Workbench IDE - AVR 7.20.1 - X
<u>E</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> roject <u>I</u> ools <u>W</u> indow <u>H</u> elp
1 🖞 🖓 🖳 🖉 🔚 🖾 🖞 🖯 🗤 🗸 🔍 🗸 🦕 🖽 < 🧛 🖉 🖓 🖓 👘 🔛 🖉 👘 👘
Workspace • A X
Debug ~
Files 🔅 •
Helloworld - Debug 🗸
HelloWorld
Ready

- 8. コート^{*} エディタでそれを開くためにHelloWorld.cソース ファイルをダブル クリックしてください。この応用記述用のソース コート はこのファイル内で す。このコート の目的はIAR環境の興味ある機能のいくつかを示すことです。
 - 注: IARコンハ[°]イラ手引書はIARコンハ[°]イラの多数の機能を知るために調べることができる非常に有用な資料です。このコンハ[°]イラ手引書 はIAR Embdded Workbenchインストール フォルタ[°]のavr¥doc子フォルタ[°]下です。

IAR環境のいくつかの機能が以降の項で記述されます。これらの機能はHelloWorld.c内のコードでも使われます。

2.2.1. EEPROM変数の使い方

EEPROM内の変数の格納と使用はそれの宣言で__eepromキーワードを追加することによって行われます。その後、この変数は他のどの 変数とも同じ様に使うことができます。

任意で、その変数を保存する場所のアドレスを指定してください。

uint8_t __eeprom level @ 0x1400;

注: 既定により、EEPROMはデバイス書き込み時に消去されます。開発の間、デバイスが書かれる毎にEEPROMを消去することは有用 でないかもしれません。これについてはProject(プロジェクト)⇒Options(任意選択)⇒Atmel-ICE(Atmel-ICE)⇒Atmel-ICE 2(Atmel ICE 2)⇒Preserve EEPROM contents even if the device is reprogrammed(例えデバイスが再書き込みされても、EEPROM内容を 保護)を行ってください。

ptions for node "HelloW	'orld"				
Category: General Options Static Analysis C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger Atmel-ICE AVR ONE! JTAGICE3 JTAGICE3 JTAGICE3 JTAGICE3 JTAGICE mkII Dragon Power Debugger Simulator Third-Party Driver	Atmel-ICE 1 Atmel-ICE 2 Run timers in stopped Preserve EEPROM co Restore fuses when er Enable software break System breakpoints on exit putchar getchar	Communication mode ntents even if dev nding debug session points	Extra Options ice is reprogram on Preserve <u>EL</u> A <u>None</u> <u>Boot Are</u> <u>Applicati</u>	Factory Settings medi	

2.2.2. フラッシュ メモリで定数を保存

__flashキーワート゛の使用はフラッシュ メモリに格納される定数を作ります。これは多くのプログラム用メモリが利用可能だけれどもデータ用メモリが少 ない時に有用です。

<pre>const charflash led_state[2][13] = { "LED is off\fryn", "LED is</pre>	s on¥r¥n″	};
注: constキーワードそれ自体は値がプログラム用メモリに保存されることをまったく意味しま せん。これは変数が定数であることだけを意味し、値が保存されるメモリ領域はコン	標準関数	プログラム空間値用等価関数
パイラ実装固有です。	printf	printf_P
フラッシュメモリから値を使う時に使われなければならない標準のものと等価な一式の関	memcmp	memcmp_P
数があります。これらの関数はpgmspace.h内です。右表はフラッシュ メモリの値と共に使	strcpy	strcpy_P
われるべき等価関数の少しの例を示します。	strstr	strstr P

strstr

2.2.3. 割り込み

1つまたは多数の割り込みべクタを指定するには#pragma vector疑似命令を使ってください。_interruptキーワートは割り込み関数を指定 します。



2.2.4. printf

いくつかの基板に対して既定によってprintfはデベッガに接続されたUSARTへ書き、これは同様にIAR Workbench IDEの端末I/Oに データを転送するのにCDCを使います。これは非常に有用な直ぐに使える便利な機能です。

size t write(int handle, const unsigned char *buf, size t bufSize)関数を実装することにより、printf出力を手動で扱うことができます。 コンハイラ手引書で使用者はI/O割り当てされたLCDポートにprintf出力を書く例を見つけることができます。提供されたコート例で_write 関数は未だUSARTへ書きますが、提供される実装は既定のものより高速です。

strstr_P

2.3. IAR Embedded Workbenchで構成を設定

作業空間は今やソースファイルが置かれた'HelloWorld'プロジェクトを含みます。

次の作業はIARのコンハ[°]イラとリンカを構成設定することです。既定により、AVR用IAR Wmbedded WorkbenchプロジェクトはDebug(デバッグ) とRelease(公開)の2つの既定構成で作成されます。

Debug構成はデバッグに必要な全ての情報を含む実行可能オブジェクトファイルの結果となり、一方でRelease構成はデバッグ情報が取り除かれたバイナリファイルの結果になります。

各構成用の設定は作業空間ウィンドウでProject(プロジェクト)⇒Options(任意選択)を選ぶか、または単にAlt+F7を押すことによってアクセス することができます。構成が互いに独立しているため、或る構成で実行された全ての手順は他の構成に対しても実行されることが必 要です。

図2-6. 有効な構成の選択			
9 IAR Embedded Workbench IDE - AVR 7.20.1	122		×
<u>Eile Edit View Project Iools Window H</u> elp			
1 1 1 🖬 📾 🔚 🕹 🖞 🖆 🗋 1 2 C 1 🔄 🚽 🗸 Q > 5 🕸 🖽 < 📮 > 🖸 🗗 📾 🖷 💽 🔸 📮			
Workspace • # X			
Debug 🗸			
Debug Release			
Helloworld - Debug * V			
HelloWorld			
Ready In 1, Col 1	UTF-8	CAP NUM	OVR

1. 作業空間ウィントウの上部の引き落としメニューで望む構成が選ばれていることを確実にしてください。

- 2. プロジェクト名を右クリックしてOptions(任意選択)を選んでください。
- 3. 'General Options(全般任意選択)'区分のTarget(目的対象)タブ下でプロジェクトで使われるデベイス用のProcessor configuration(プ ロセッサ構成設定)を選んでください。ATmega4809を選んでください。

注: Processor configuration選択時、いくつかの一般的なデバイスだけでなく、系列に基づくいくつかの区分もあります。

注:新デバイスが未支援の場合、最新IDE版に更新してみてください。

- 4. 'Memory Model(メモリ様式)'でsmall(小)を選んでください。
- 5. Debugger(デバッカ)区分を選んでください。Setup(装置準備)タブ下でAtmel-ICEドライバを選んでください。
 - 注: Curiosity Nano基板はAtmel-ICE互換の基板上デバッガ(nEDBG)を持ち、故にこの場合、実際のAtmel-ICEハートウェアデバッガ は必要ありません。
- 6. OKをクリックすることによって'HelloWorld'プロジェクト節点に対する任意選択への変更を受け入れてください。以下を除いて同じ手順がRelease(公開)構成に対して繰り返されます。

・ 'Debugger(デバッガ)' 区分のSetup(装置準備)タブ下でトライバ設定をそれらの既定のままにしてください。Release構成はデバック ジンボルを含まず、従ってデバック「作業に使うことはできません。

ー旦終わると、File(ファイル)⇒Save All(全て保存)を選んでください。その後、デバッグ構成に戻ってプロジェクトを構築するためにProject (プロジェクト)⇒Make(作成)を選ぶか、または同じことをするために単純にF7を押してください。全ての構築でBuild Log(構築記録)ウィント ウが飛び出して警告と異常を示します。この構築に対してBuild Log(構築記録)ウィントウで異常なしと警告なしを確実にしてください。

図2-7. Build Log(構築記録	ウィンドウは応用での異常と警告を指摘することができます。	
HelloWorld - IAR Embedded Workbench ID	E - AVR 7.20.1	– 🗆 🗙
Eile Edit View Project Iools Window	<u>H</u> elp	
Workspace • # X	HelloWorld.c x port_dir	fo
Files	9 T /* 10 Preprocessor Macros for USER LED. 11 These macros should be modified 12 in accordance with the board in use.	^ > v
	Build	▼ ↓ ×
	Messages File HelloWorld.c Linking	Line
HelloWorld	Total number of errors: 0 Total number of warnings: 0	v
Ready	Errors 0, Warnings 0 Ln 26, Col 16	UTF-8 CAP NUM OVR

2.4. ヒュースを構成設定

ヒュース・ビットは(ウォッチト・ック・タイマや低電圧検出器のような)様々な周辺機能、(クロック設定のような)システム設定、メモリ領域を構成設定するのに使うことができます。

IAR Workbench IDEでヒュース。設定をアクセスするには上部ハーでAtmel-ICEメニュー項 目へ行き、その後にFuse Handler(ヒュース、処理部)を選んでください。

ヒューズを読み書きすることができるウィンドウが開きます。

注: ヒューズを見るには基板が接続されていなければなりません。

Atmel-ICE	Tools	Window	Help	_
Power	Debug	ging Setting	IS	-
Fuse H	landler.		N	
Chip F	rase		5	

use Handler	
Device ID 0x1E9651	<u>R</u> ead Fuses Program Fuses <u>C</u> lose
LOCKBIT WDTCFG BODCFG OSCCFG SYSCFG0 SYSCFG1 APPEND BOOTEND Old Value: 0xC5 New Value: 0xC5 Lock Bits : No locks	~
Log <u>M</u> essages , USB ID: ATML3094051800001616, FW version: 1.2.262 UPDI clock: 100 KHz, Target voltage: 3.303 V, Device name: ATmega4809 (0x1E9651) Starting to read fuses. Succeded to read fuses	

2.5. 最適化水準

最適化はコンパイラが大きさと速度ようなコートの様々な 面を改善することを許す程度を制御します。使用事例 に応じて、他を損なってまでそれらの1つが改善され ます。

Project(プロジェクト)⇒Options(任意選択)⇒C/C++ Co mpiler(C/C++コンパイラ)⇒Optimizations(最適化)へ行 くことによって最適化設定をアクセスしてください。

図2-10. 'HelloWorld	節点用任意通	選択			
Options for node "HelloWor	ld"				×
Category: General Options Static Analysis C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger Atmel-ICE AVR ONE! JTAGICE3 JTAGICE3 JTAGICE3 JTAGICE mkII Dragon Power Debugger Simulator Third-Party Driver	Id"	ation used Publics 4 M Language 2 Ena Ena Ena Ena Ena Ena Ena Ena Ena Ena	ISRA-C:1998 Code abled optimizat Common sub- Function inlin Code motion Cross call Clustering of Type-based a Julimited	Factor Diagn Extra Optimizations expression elimina ing variables alias analysis	x ory Settings ostics Options Output tion
			[ОК	Cancel

2.6. スタック使用量分析と設定

1つの高度なデバッグ技法はスタック使用量を調べること です。これはスタック溢れと化けの問題の検出に役立つ かもしれません。使用者はProject(プロジェクト)⇒Option s(任意選択)⇒Linker(リンカ)⇒Stack Usage(スタック使用 量)からこの分析を許可することができます。

スタック領域の大きさはProject(プロジェクト)⇒Options(任 意選択)⇒General Options(全般任意選択)⇒System (システム)から調節することができます。

Category: General Options Static Analysis					
C/C++ Compiler Assembler	Target	Output	Library (Configuration	Library Options
Custom Build	Heap Configur	ration S	ystem	MISRA-C:2004	MISRA-C:1998
Build Actions Linker Debugger Atmel-ICE	<u> </u>	CSTACK)		Return address	s stack (RSTACK)
	Size (bytes):	0x20		Depth (levels)): 16
	Place in	external mer	nory	Place in e	xternal memory
JTAGICE3 JTAGICE mkII Dragon	External Mem	nory Configur external mem	ation ory bus external me	emory accesses	
Power Debugger		RAM		ROM	Non-Volatile
Simulator	Base addre	ss OxO		0x0	0×0
Third-Party Driver	Memory size	e 0x0		0x0	0×0
	✓ Initialize unu ☐ Enable bit d	used interrup lefinitions in l	t <u>v</u> ectors wit /O-Include t	h RETI instruction files	ns

2.7. AVR[®]用IAR Embedded Workbenchでの書き込みとデバッグ

どんな異常もなしに実際の応用を書くのは難しいかもしれません。より大きな応用はより高い予測されるハゲの割合になります。成功 裏のコンパイルは応用がC仕様に従うことを意味します。言語遵守が常に実行の正当性を保証するとは限りません。従って、デバッグが 必要です。これは応用実行中にハゲが見つかる開発段階を意味します。開発者は応用の必要条件に従ってそれらを修正する方法 を見つけます。より多くのデバッグ技法を知ることはコード全体を通して欠陥を追跡することをより易しくします。

AVR用IAR Embedded Workbenchでデバッグ作業を開始するには以下を実行してください。

- 1. PCから基板のDEBUGGER USBポートにUSBケーブルを接続してください。
- 2. POWER LEDがONにされたことを確認してください。
- 3. Workspace(作業区間)ウィントウで選んだデバック構成があることを確実にしてください。
- 4. マイクロ コントローラに書いてデ・バッグ作業を開始するにはProject(プロジェクト)⇒Download and Debug (CTRL+D)(ダウンロート'とデ・バック' (Ctrl+D))を選んでください。毎回、応用が再構築され、マイクロ コントローラは再書き込みされなければならず、従ってデ・バッグ作業は再始動されます。



注: ディ、ッグ作業開始時、IARディ、ッガは自動的にmain()関数の始めにコード中断点(ブレークホイント)を設定します。

5. Tera Termのような端末プログラムを開き、Curiosity Nano基板COMポートを選んでボーレートを9600に設定してください。 6. コート、エディタでprintf関数呼び出しが見つかるまでHelloWorld.cソース ファイルを下スクロールしてください。 7. この行に新しいコード中断点を設定するため、下で示されるように、コードェディタの左列をクリックしてください。

図2−13. コード中	断点	を含む行		
workspace - IAR Embedded W	orkbench IDE	AVR 7.20.1	-	D X
Eile Edit View Project Deb	ug <u>Atmel-IC</u>	Iools Window Help		
🗅 🗅 🖬 🕼 🔚 🗶 🛍	D D C	: < Q > \$ + E < ♥ > 2 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1	
Workspace	* # X	HelloWorld.c 🗙 Intrinsics.h	Disassembly	* a ×
Debug	~	main() f()	Go to CODE	~ [
Files ■ HelloWorld - Debug ■ BelloWorld c ■ Output	•		Disassembly 0001AA 9508 PORTB DIRCLE = PIN2_ba; initButton: 0001AC E004 0001AE 9300 0422 PORTB PIN2CTRL : PORT_J 0001BE 9300 0432 0001BE 9300 0432) 0001BC 9508 init5 membrdie()	RET LDI STS 20LLUPEN_ba LDS ORI STS RET
		<pre>printf_P[led_state[level]]; /* #ait until all data were shifted out of the UGART shift register before goind to sleep. Manually clear the flag afterwards. // while((!(USARTI.STATUS & USART_TXCIF_bm)))</pre>	<pre>main: 0001EE DFDD initUsart(): 0001C0 DFE0 initLed(): 0001C2 DFF0 initLet():</pre>	RCALL RCALL RCALL
Helloworld		۲ ۲	<	>
		G		CAP OVR

- 注: プログラム実行はコード中断点を含む行に達する時に必ず自動的に止まります。
- 8. 実行を続けるにはDebug(デベッグ)⇒Go (F5)(進行(F5))を選び、その後に基板の使用者釦を押してください。コート・中断点が設定さ れている行で実行が止まることに注意してください。

図2-14. 緑の矢	印は	プログラム実行が止まった場所を示します。					
workspace - IAR Embedded W	orkbench IDE	AVR 7.20.1			90. 20 1		×
Eile Edit View Project Deb	ug Atmel-IC	Jools Window Help					
the para a ke	DC	+ < Q > \$ HE < Q > R B B B H O C 0 1 (A F H H H + (
Workspace	* # X	HelloWorld.c x Intrinsics.h	-	Disassembly		+	ąх
Debug	~	main()	fo	Go to	~ CODE		~ 0
Files ■ HelloWorld - Debug HelloWorld c B Gutput	•••	<pre>sleep(); while(1) (delay_ms(BOUNCE_TIME); if(0 == level) { level = i; PORTB.OUTER = FINS_bm; } else if (1 == level) { level = 0; PORTB.OUTER = FINS_bm; } pissif (1 == level) { level = 0; PORTB.OUTER = FINS_bm; } } Pissif (1 == level) { level = 0; PORTB.OUTER = FINS_bm; } Pissif (1 == level) { level = 0; restrict [level = 1; level = 0; restrict [level = 1; level = 0; level = 0</pre>		Disassembly 000205 E0 000208 E0 000204 E0 000202 4 000210 01 000212 EA 000214 E0 000216 0F 000218 IF 000218 IF 000218 E2 000220 E0 000222 81 000224 FF 000222 FF 000021 FF 00021 FF 00022 FF 0002 FF 00	10 4D 50 50 05 05 78 98 00 10 02 13 13 05 0840 00 56 76 00 06 75 75 70 00 08 24	LDI LDI LDI CALL NOVV LDI LDI ADD ADC CALL ADD ADC CALL LDI LDI LDI LDI LDI SBRS RJMP XCIF_bm STS	*
relowond		C	> v	<			>
dy						CAP C	WR

9. 再びF5を打つことがプログラムの継続を許します。while部のSLEEP命令のため、マイクロコントローラは休止へ行き、休止から起き上がるまで再び中断点に当たりません。マイクロコントローラを起こすのに基板上の使用者釦を押してください。その後に再び中断点に当たります。この手順は望まれる限り何度でも実行することができます。実行を続ける時に端末で現れるメッセージに対して実際の基板のLEDの状態を比べてください。

図2-15. 端末へ送られたprintf出力		
🔟 COM16 - Tera Term VT	_	×
<u>F</u> ile <u>E</u> dit <u>S</u> etup C <u>o</u> ntrol <u>W</u> indow <u>H</u> elp		
LED is on LED is off		^
LED is on LED is off		
		Ť

10. デバッグはそれらの応用で何が起きているかを詳細に直視する力を開発者に与えます。コード中断点のような機能は走行中にコード で何が起きているかの助言を使用者に提供することができます。この場合、実際のLED状況と端末出力間に不一致があります。 これを修正し、その後に前の手順に従ってプロジェクトを再構築し、ここまでに学んだことを使い、もう一度デバッグをしてみてください。

3. 改訂履歴

資料改訂	日付	注釈
А	2020年3月	初版

Microchipウェフ゛サイト

Microchipはhttp://www.microchip.com/で当社のウェブサイト経由でのオンライン支援を提供します。このウェブサイトはお客様がファイルや情報を容易に利用可能にするのに使われます。利用可能な情報のいくつかは以下を含みます。

- ・製品支援 データシートと障害情報、応用記述と試供プログラム、設計資源、使用者の手引きとハートウェア支援資料、最新ソフトウェア配布と 保管されたソフトウェア
- ・全般的な技術支援 良くある質問(FAQ)、技術支援要求、オンライン検討グループ、Microchip設計協力課程会員一覧
- ・Microshipの事業 製品選択器と注文の手引き、最新Microchip報道発表、セミナーとイベントの一覧、Microchip営業所の一覧、代理 店と代表する工場

製品変更通知サービス

Microchipの製品変更通知サービスはMicrochip製品を最新に保つのに役立ちます。加入者は指定した製品系統や興味のある開発ツールに関連する変更、更新、改訂、障害情報がある場合に必ず電子メール通知を受け取ります。 登録するにはhttp://www.microchip.com/pcnへ行って登録指示に従ってください。

お客様支援

Microchip製品の使用者は以下のいくつかのチャネルを通して支援を受け取ることができます。

- ・代理店または販売会社
- ・最寄りの営業所
- ・組み込み解決技術者(ESE:Embedded Solutions Engineer)
- ・技術支援

お客様は支援に関してこれらの代理店、販売会社、またはESEに連絡を取るべきです。最寄りの営業所もお客様の手助けに利用できます。営業所と位置の一覧はこの資料の後ろに含まれます。

技術支援はhttp://www.microchip.com/supportでのウェブ サイを通して利用できます。

Microchipデバイスコート、保護機能

Microchipデバイスでの以下のコード保護機能の詳細に注意してください。

- ・Microchip製品はそれら特定のMicrochipデータシートに含まれる仕様に合致します。
- ・Microchipは意図した方法と通常条件下で使われる時に、その製品系統が今日の市場でその種類の最も安全な系統の1つである と考えます。
- コート、保護機能を破るのに使われる不正でおそらく違法な方法があります。当社の知る限りこれらの方法の全てはMicrochipのデータシートに含まれた動作仕様外の方法でMicrochip製品を使うことが必要です。おそらく、それを行う人は知的財産の窃盗に関与しています。
- ・Microchipはそれらのコードの完全性について心配されているお客様と共に働きたいと思います。
- ・Microchipや他のどの半導体製造業者もそれらのコートの安全を保証することはできません。コート、保護は当社が製品を"破ることができない"として保証すると言うことを意味しません。

コート、保護は常に進化しています。Microchipは当社製品のコート、保護機能を継続的に改善することを約束します。Microchipのコート、保護機能を破る試みはデジタルシニアム著作権法に違反するかもしれません。そのような行為があなたのソフトウェアや他の著作物に不正なアクセスを許す場合、その法律下の救済のために訴権を持つかもしれません。

法的通知

デバイス応用などに関してこの刊行物に含まれる情報は皆さまの便宜のためにだけ提供され、更新によって取り換えられるかもしれま せん。皆さまの応用が皆さまの仕様に合致するのを保証するのは皆さまの責任です。Microchipはその条件、品質、性能、商品性、 目的適合性を含め、明示的にも黙示的にもその情報に関連して書面または表記された書面または黙示の如何なる表明や保証もし ません。Microchipはこの情報とそれの使用から生じる全責任を否認します。生命維持や安全応用でのMicrochipデバイスの使用は完 全に購入者の危険性で、購入者はそのような使用に起因する全ての損害、請求、訴訟、費用からMicrochipを擁護し、補償し、免責 にすることに同意します。他に言及されない限り、Microchipのどの知的財産権下でも暗黙的または違う方法で許認可は譲渡されま せん。

商標

Microchipの名前とロゴ、Mcicrochipロゴ、Adaptec、AnyRate、AVR、AVRロゴ、AVR Freaks、BesTime、BitCloud、chipKIT、chipKITロ ゴ、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、 maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemiロゴ、MOST、MOSTロゴ、MPLAB、OptoLyzer、PackeTime、PI C、picoPower、PICSTART、PIC32ロゴ、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SSTロゴ、Super Flash、Symmetricom、SyncServer、Tachyon、TempTracker、TimeSource、tinyAVR、UNI/O、Vectron、XMEGAは米国と他の国に於 けるMicrochip Technology Incor poratedの登録商標です。

APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、HyperLight Load、 IntelliMOS、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plusロゴ、Quiet-Wire、 SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、Vite、WinPath、ZLは米国に於けるMicrochip Technology Incorporatedの登録商標です。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BlueSky、BodyCom、CodeGuard、 CryptoAuthentication、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、EC AN、EtherGREEN、In-Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet¤ ゴ、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified¤ゴ、MPLAB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PowerSmart、PureSilicon、QMatrix、REALICE、Ripple Blocker、SAM-ICE、Se rial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、View Sens e、WiperLock、Wireless DNA、ZENAは米国と他の国に於けるMicrochip Technology Incorporatedの商標です。

SQTPは米国に於けるMicrochip Technology Incorporatedの役務標章です。

Adaptec^{ロコ、}、Frequency on Demand、Silicon Storage Technology、Symmcomは他の国に於けるMicrochip Technology Inc.の登録商 標です。

GestICは他の国に於けるMicrochip Technology Inc.の子会社であるMicrochip Technology Germany II GmbH & Co. KGの登録商 標です。

ここで言及した以外の全ての商標はそれら各々の会社の所有物です。

© 2020年、Microchip Technology Incorporated、米国印刷、不許複製

品質管理システム

Microchipの品質管理システムに関する情報についてはhttp://www.microchip.com/qualityを訪ねてください。

日本語© HERO 2020.

本応用記述はMicrochipのAN3419応用記述(DS00003419A-2020年3月)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。



米国

世界的な販売とサービス

本社

2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 技術支援: http://www.microchip.com/ support ウェブ アトレス: http://www.microchip.com アトランタ Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 オースチン TX Tel: 512-257-3370 ボストン Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 シカゴ Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 ダラス Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 デトロイト Novi, MI Tel: 248-848-4000 **ヒューストン** TX Tel: 281-894-5983 インデアナポリス Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 ロサンセ・ルス Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 D-J- NC Tel: 919-844-7510 ニュ**ーヨーク** NY Tel: 631-435-6000 サンホセ CA Tel: 408-735-9110 Tel: 408-436-4270 カナダ - トロント Tel: 905-695-1980

オーストラリア - シト・ニー Tel: 61-2-9868-6733 中国 - 北京 Tel: 86-10-8569-7000 中国 - 成都 Tel: 86-28-8665-5511 中国 - 重慶 Tel: 86-23-8980-9588 中国 - 東莞 Tel: 86-769-8702-9880 中国 – 広州 Tel: 86-20-8755-8029 中国 – 杭州 Tel: 86-571-8792-8115 中国 - 香港特別行政区 Tel: 852–2943–5100 中国 - 南京 Tel: 86-25-8473-2460 中国 - 青島 Tel: 86-532-8502-7355 中国 - 上海 Tel: 86-21-3326-8000 中国 - 瀋陽 Tel: 86-24-2334-2829 中国 - 深圳 Tel: 86-755-8864-2200 中国 – 蘇州 Tel: 86-186-6233-1526 中国 - 武漢 Tel: 86-27-5980-5300 中国 - 西安 Tel: 86-29-8833-7252 中国 - 廈門 Tel: 86-592-2388138 中国 - 珠海 Tel: 86-756-3210040

亜細亜/太平洋

亜細亜/太平洋 イント・- ハンカ・ロール

Tel: 91-80-3090-4444 イント - ニューデリー Tel: 91-11-4160-8631 イント・フネー Tel: 91-20-4121-0141 日本 - 大阪 Tel: 81-6-6152-7160 日本 - 東京 Tel: 81-3-6880-3770 韓国 - 大邱 Tel: 82-53-744-4301 韓国 - ソウル Tel: 82-2-554-7200 マレーシア – クアラルンプール Tel: 60-3-7651-7906 マレーシア ー ヘ・ナン Tel: 60-4-227-8870 フィリピン ー マニラ Tel: 63-2-634-9065 シンガポール Tel: 65-6334-8870 台湾 - 新竹 Tel: 886-3-577-8366 台湾 - 高雄 Tel: 886-7-213-7830 台湾 - 台北 Tel: 886-2-2508-8600 タイ ー バンコク Tel: 66-2-694-1351 ベトナム ー ホーチミン Tel: 84-28-5448-2100

欧州 オーストリア – ウェルス Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 テンマーク - コヘンハーケン Tel: 45-4450-2828 Fax: 45-4485-2829 フィンラント – エスホー Tel: 358-9-4520-820 フランス – パリ Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 トイツ – ガルヒング Tel: 49-8931-9700 ドイツ – ハーン Tel: 49-2129-3766400 トイツ – ハイルブロン Tel: 49-7131-72400 ドイツ – カールスルーエ Tel: 49-721-625370 ドイツ - ミュンヘン Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 ドイツ - ローセンハイム Tel: 49-8031-354-560 イスラエル - ラーナナ Tel: 972-9-744-7705 イタリア ー ミラノ Tel: 39-0331-742611 Fax: 39-0331-466781 イタリア ー パドバ Tel: 39-049-7625286 オランダーデルーネン Tel: 31-416-690399 Fax: 31-416-690340 ノルウェー - トロンハイム Tel: 47-72884388 ポーラント゛ー ワルシャワ Tel: 48-22-3325737 ルーマニア – ブカレスト Tel: 40-21-407-87-50 スペイン - マドリート Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 スウェーデン – イェーテホリ Tel: 46-31-704-60-40 スウェーデン – ストックホルム Tel: 46-8-5090-4654 イキ・リス - ウォーキンガム Tel: 44-118-921-5800 Fax: 44-118-921-5820

Fax: 905-695-2078