



Curiosity Nanoを使う低電力動作

事前要件

- ・ハート・ウェア事前要件
 - この文書で記述される小さなハートウェア変更でのAVR128DA48 Curiosity Nano (DM164151)基板
 - Power Debugger (ATPOWERDEBUGGER)
 - 2本のマイクロUSBケーブル: Power Debugger用×1、Curiosity Nano基板用×1
 - ピンに対する両端の穴との2本の線
- ・ソフトウェア事前要件
 - MPLAB® X 5.40
 - MPLAB X 3.95版用MCCプラグイン
 - MCC 2.3.0版用AVRライフ ラリ
 - Atmel Studio最新版

序説

この文書は各種作業を達成するために純粋なソフトウェア手法とCIP(コアから独立した周辺機能)の使用間に於いて消費 電力での違いを示します。

この実地訓練は以下の話題を網羅します。

- ・課題1:割り込みと比較した事象システム
- ・課題2:A/D変換器(ADC)に対してハートウェア累積と比べたソフトウェア累積

各課題に関連する基本的な原理が各課題の初めで紹介され、その後にその機能がAtmel Studioデータ可視器(Data V isualizer)を通して確認されます。

アイコン鍵識別子

各種課題部分を識別して複雑さを減らすためにこの文書内で以下のアイコンが使われます。

┃ 情報:特定の話題についての脈絡上の情報を伝えます。

助言:有用な助言と技術を強調します。

🎦 行うこと: 完了されるべき目標を強調します。

▶ 結果:課題の段階の予測される結果を強調します。

警告 重要な情報を表示します。

•

実行: 必要とさる時に目的対象の中から実行されるべき行動を強調します。

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、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。

目次

事前要件 ••••••••••••••••••••••••••••••••••••	1
序説 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	1
アイコン鍵識別子 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	1
1. 課題1:割り込みと比べた事象システム・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	3
1.1. ハート・ウェア説明 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	3
1.2. 正確な電力測定を得るためのCuriosioty Nano変更 ····································	3
1.3. 使う単位部の説明 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	3
1.4. ソフトウェア実行プログラム作成・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	4
1.5. 事象システム実行プロジェクト作成 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	15
1.6. デバイス書き込みと効率比較・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	18
2. 課題2 : A/D変換器(ADC)に対してハート・ウェア累積と比べたソフトウェア累積 ・・・・・	22
2.1. ソフトウェア実行プロジェクト作成 ·····	22
2.2. ハートウェア実行プロジェクト作成 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	27
2.3. デバイス書き込みと電力効率比較・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	32
3. 改訂履歴	36
Microchipካェフ [*] サイト	37
製品変更通知サービス ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	37
お客様支援・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	37
Microchipデバイス コード保護機能 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	37
法的通知 ••••••••••••••••••••••••••••••••••••	37
商標 ••••••••••••••••••••••••••••••••••••	38
品質管理システム ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	38
世界的な販売とサービス・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	39

1. 課題1:割り込みと比べた事象システム

この課題では2つの異なる手法の消費電力が比較されます。1つ目の手法は温度感知器を読むためにADC変換を開始する計時器 割り込みを通してマイクロコントローラを休止から起こすことを意味します。変換終了時、結果準備可割り込みが起動され、変換の結果が 読まれます。

2つ目の手法では事象システム(EVSYS)が使われます。この単位部は事象を通して違う周辺機能に接続することを使用者に許します。 前の手法は計時器溢れが温度感知器を読むADC変換を開始する事象を起動するように変更されます。変換の終わりで結果を読む ために休止からマイクロコントローラを起こします。

1.1. ハート・ウェア説明

Curiosity Nano基板はAVR128DA48マイクロコントローラ、Nano組み込みデバッガ、LED、釦に基づきます。Nanoデバッガは外部ハートウェアの 必要なしで基板の書き込みとデバッグ走行の機能を使用者に許します。UART-USB橋渡し能力も提供します。

Power Debuggerは構成設定可能な電圧を持つ電力出力も提供する書き込み器で、これに接続された装置の消費電流を測定するのに使うことができます。

1.2. 正確な電力測定を得るためのCuriosioty Nano変更

💦 行うこと: 配線切断と2つのピンの半田付けによってCuriosoty Nano基板を変更してください。

電力がマイクロUSBポートで観測される場合、Power Debuggerは幾許かの電力を消費し、不正確な読み取りにします。



Curiosity Nano基板でPOWERと名付けられた小さな配線があります。鋭利な道具を使うことによって2つの穴がもう接続されなくなるまでこの配線を切断してください。電圧計でこれを調べてください。

この配線が切断された後、この穴に2つのピンを半田付けしてください。今後基板を書くにはこの2つのピンが線(ジャンパ)で接続されなければなりません。



1.3. 使う単位部の説明

この課題で使われる単位部は実時間計数器(RTC)、ADC、基準電圧(VREF)、事象システム(EVSYS)です。

RTCは1秒を計数する計時器として動きます。

ADCはアナログ電圧をデジタル12ビット数値に変換する単位部です。これはEVSYSから変換を開始するようにできます。変換終了時に事象と割り込みも起こします。ADCは内部温度感知器を読むのに使われます。

基準電圧は他の周辺機能に安定した電圧を供給する周辺機能です。ADCに2.048V参照基準を提供するように設定した場合の温度 感知器も許します。

事象システムはCPUの関与なしに相互周辺機能通信を許す配線網です。この通信は1つの周辺機能から別の周辺機能に事象を送って活動を起動することによって行われます。事象は遅れなしで決して失われず、より良い実時間と信頼に足る応用を許します。

1.4. ソフトウェア実行プログラム作成

💦 行うこと: このプロジェクトで使われるハート・ウェア単位部を構成設定し、プロジェクトにコートを追加してください。

- 1. 新しいAVR128DA48用MPLAB Xプロジェクトを作成してください。
 - 1.1. MPLAB Xを開いてください。
 - 1.2. File(ファイル)→New Project(新規プロジェクト)またはNew Pproject(新規プロジェクト)釦を選んでくださ 図1-3. 新規

図1-3	3.	·現フ	シェクト	,
File	Edit	View	Navig	ate
1	P	1	5	

1.3. Next(次へ)をクリックしてください(既定でStandalone Project(Microchip組み込み独立型プロジェクト)が選ばれています)。

図1-4. プロジェクト型式	
😰 New Project	×
Steps	Choose Project
1. Choose Project	Q Filter:
	Categories: Projects: Image: Standalone Project Standalone Project Image: Other Embedded Existing MPLAB IDE v8 Project Image: Standalone Project User Makefile Project Image: User Makefile Project Image: Transmission of the project Image: User Makefile Project Image: Transmission of the project Image: User Makefile Project Image: Transmission of the project Image: User Makefile Project Image: Transmission of the project Image: User Makefile Project Image: Transmission of the project
	Description:
	Creates a new standalone application project. It uses an IDE-generated makefile to build your project.
	< Back Next > Finish Cancel Help

1.4. Device(デバイス)領域でAVR128DA48に対して検索してください。Tool(道具)区分でコンピュータに接続されていればCuriosity N ano基板を選び、さもなければNone(なし)を選んでください。Next(次へ)をクリックしてください。

図1-5. デバイス選択			
🔀 New Project			×
Steps	Select Device		
Choose Project Select Device Select Pleader Select Plugin Board Select Compiler Select Compiler Select Project Name and Folder	Family: Device:	All Families v	
	Tool:	AVR 128DA48 Curiosity Nano-SN: MCH 🗸 🗌 Show All	
MPLAB X IDE			
		< Back Next > Finish Cancel Help	

1.5. XC8 2.20コンハパラを選んでNext(次へ)をクリックしてください。

New Project		×
Steps 1. Choose Project 2. Select Device 3. Select Header 4. Select Plugin Board 5. Select Compiler 6. Select Project Name and Folder	Select Compiler Compiler Toolchains 	
	Rack Nexts Einich Carcel Hale	

1.6. プロジェクト(と保存されるべき場所)に名前を与えてFinish(終了)をクリックしてください。

図1-7. プロジェクト名		
🗴 New Project		×
Steps	Select Project Name a	nd Folder
Choose Project Select Device Select Header Select Plugin Board Select Compiler Select Compiler Select Project Name and Folder	Project Name: Project Location: Project Folder:	avr-da-cnano-low-power-lab-interrupt C: \Wy stuff\Projects stuff\Projects\avr-da-cnano-low-power-lab-interrupt.X
MPLAB X IDE	Overwrite existing p Also delete sources Set as main project Use project location Encoding: ISO-6 Project name and fold Try shortening the pro	project. a sthe project folder 8859-1 v ler path length are nearing the Windows limit. This may cause issues during build or p ject name or path.
		< Back Next > Finish Cancel Help

2. MPLABコート、構成部(MCC)を開いて周辺機能を構成設定してください。

2.1. System Module(システム単位部)でクロック元を24MHzクロックを持つ内部発振器として選んでください。

•	情報: Clo	Source(クロック元)をInternal Oscillator(内部発振器)として、Oscillator Frequency Options(発振器周波数
U	任意	髦択)を24MHz system clock(24MHzシステム クロック)として選び、Prescalor(前置分周器)を禁止してください。

巡1−8. システム単位部構』		
System Module		()
😳 Easy Setup 📄 Registers		
▼ Clock Control		
Main Clock(Hz):	2400000	
Clock Source :	Internal Oscillator	-
Internal Oscillator Frequency:	1-32MHz internal oscillator	-
Oscillator Frequency Options:	24 MHz system clock	-
PLL Enable:		
External Clock(Hz):	1 ≤ 1000000 ≤ 20000000	
Prescaler Enable:		
Prescaler:	6X	-
Olock Out Enable:		
Watchdog Timer		
Brown-out Detector		
Voltage Level Monitor		

2.2. System Module(システム単位部)のRegisters(レシブスタ)タブで休止を許可して動作形態をスタンハイに設定してください。

情報: System Module(システム単位部)のRegisters(レジスタ)タブへ行き、SLPCTRLへ下スクロールしてください。CTRLAレジス タを休止許可(SEN=enabled)に変更して、動作形態(SMODE=Standby Mode)を選んでください。

図1-9. 休止構成設定
System Module
🛞 Easy Setup 📃 Registers
SWRST disabled -
▼ SLPCTRL
▼ Register: SLPCTRLCTRLA 0x3
SEN enabled -
SMODE Standby Mode +
▼ Register: SLPCTRLVREGCTRL 0x0
PMODE AUTO -

2.3. Device resources(デバイス資源)からVREF単位部を追加し、ADCへ2.048V参照基準を提供するように構成設定してください。



情報: VREFはマイクロ コントローラの温度感知器を許可するのに使われます。

図1-10. VREF構成設定	5	
VREF		•
🔅 Easy Setup 📃 Registers		
▼ Software Settings		
API Prefix:	VREF_0	
▼ Hardware Settings		
Enable Force ADC Voltage Reference:		
ADC Voltage Reference:	Internal 2.048V reference	-
Enable Force DAC/AC Voltage Reference:		
DAC/AC Voltage Reference:	Internal 1.024V reference	-
 Enable Force AC Voltage Reference: 		
AC Voltage Reference:	Internal 1.024V reference	*

- **2.4**. Device resources(デハイス資源)からADC単位部を追加し、それを12ビット動作、右揃え結果、累積なし、採取長1刻時、スタンハ イで走行(RUNSTBY)に構成設定してください。
 - **i** *

情報: Left Adjust Result(左揃え結果)任意選択は禁止(チェックなしに)されなければなりません。Sample Length(採取長) とSample Accumulation Number(採取累積数)はHardware Settings(ハート・ウェア設定)枠で選ばれます。12ビット動作 は既定で設定されています。RUNSTBY(スタンハ・イ時走行)ビットはRegisters(レシ、スタ)タブで変更することができ、CT RLAレシ、スタで見つかります。

ž	図1-11. ADC構成設定		
	ADC0	2	
ľ	🕸 Easy Setup 📃 Registers		
	 Software Settings 		
	API Prefix:	ADC0	
	Result Selection :	12-bit mode 👻	
	Oifferential Mode Conversion :	disabled 💌	
	② Left Adjust Result :		
	▼ Hardware Settings		
	② Enable ADC:		
	③ Sampling Frequency(Hz):	272727 ≤ 857142 ≤ 923076	
	ADC Clock(Hz):	12000000	
	Sample Accumulation Number:	No accumulation	
	Sample Length (# of ADC Clock) :	0 \$ 1 \$ 31	

ADC0	•
Easy Setup Easy Setup Registers OmMAND Ox0	
SPCONV disabled STCONV disabled	
▼ Register: CTRLA 0x81	
CONVMODE disabled CONVMODE dis	
FREERUN disabled	
LEFTADJ disabled disabled T disabled T	
RUNSTBY enabled	
▼ Register: CTRLB 0x0	
SAMPNUM No accumulation	

2.5. Device resources(デバイス資源)からRTC単位部を追加し、それを内部32.768kHz発振器使用、1秒の周期、前置分周係数1、 有効にされたスタンバイで走行(RUNSTBY)を持つように構成設定してください。

情報: クロック	7、Prescaling	g Factor(前置分周有	系数)、Perio	d(周期)はHard	lware Settings	s(ハート・ウェア設定)	枠で設定される一
方で	RUNSTBY	(スタンバイ時走行)ビッ	トはCTRLA	<i>レジ</i> スタのレジスタ:	枠で見つかり	ます。	

図1-13. RTC構成設定	•	
RTC		0
🛱 Easy Setup 📃 Registers		
▼ Software Settings		^
API Prefix:	RTC	
 Hardware Settings 		
② Enable RTC:	\checkmark	
RTC Clock(Hz):	32768	
	Internal 32.768 kHz oscillator	*
External Clock(Hz):	1 ≤ 32000 ≤ 32000	
Prescaling Factor:	RTC Clock / 1	-
Ompare:	1 s ≤ 0 s	≤ 2 s
Actual Compare:	0 s	
Period:	1 s ≤ 1 s	≤ 2 s
Actual Period:	1 s	
Periodic Interrupt Timer		
 Interrupt Settings 		
Ocmpare Match Interrupt Enable:		
Overflow Interrupt Enable:		U ~

図1-14. RUNSTBY RTC構成設定
RTC
© Easy Setup
▼ Register: CNT 0x0
▼ Register: CTRLA 0x81
CORREN disabled -
PRESCALER RTC Clock / 1
RTCEN enabled -
RUNSTDBY enabled
▼ Register: DBGCTRL 0x0
DBGRUN disabled

- **2.6.** 全体割り込み(Global Interrupt)を許可してADC用の結果準備可(Result Ready)割り込みとRTC用の(Overflow)溢れ割り込み を許可してください。
 - **情報**: 全体割り込みはInterrupt Manager(割り込み管理部)で許可され、ADC割り込みはADC単位部のInterrupt Setti ngs(割り込み設定)タブで許可され、RTC割り込みは(前の段階に対する図で見られる)RTC単位部のInterrupt Settings(割り込み設定)タブで許可されます。

Interrupt Manager		2
🐯 Easy Setup 📃 Registers		
 Interrupt Setting 		
Global Interrupt Enable:	×	
 Interrupt Priority 		
Round-robin Scheduling Enable:		
Interrupt Level Priority:	0	
Interrupt Vector with High Priority:	0	

2.7. Generate(生成)釦を押してください。

図1-16. 生	成釦			
Projects	Files	Services	Resource Management [MCC] ×	
Tree View	Flat View			
Project	Resource	Generat	te Import Export	
Interr	upt Manag	er		Ê
Pin M	odule			
Syster	m Module			
▼ Peripher	rals			
🛞 🔰	🛾 🔦 ADO	0		
🛞 🔰	🔇 📸 RTC			
🛞 👂	🗸 🦟 VRE	F		Ų

3. 生成したファイルにコードを追加してください。

```
3.1. main.cファイルに以下のコードを追加してください。
```

```
#include <avr/io.h>
#include <avr/sleep.h>
#include "mcc_generated_files/mcc.h"
void TIMER_interrupt(void);
uint16_t result;
int main(void)
    SYSTEM Initialize();
    RTC_SetOVFIsrCallback(TIMER_interrupt);
    while (1) {
        sleep_cpu();
        result = ADC0_GetConversionResult();
}
void TIMER_interrupt()
    ADC0_StartConversion(ADC_MUXPOS_TEMPSENSE_gc);
    情報: SYSTEM_Initialize()はmcc.cで定義され、RTC_SetOVFIsrCallback()はrtc.cで定義され、ADC0_StartConversion
         ()とADC0_GetConversionResult()はadc0.cで定義されます。
```

- ・SYSTEM_Initialize()はCPUと周辺機能用の全ての構成設定レジスタを設定します。この関数はMCCによって生成されます。
- RTC_SetOVFIsrCallback()はmainで定義されるTIMER_interrupt()関数に呼び戻し(割り込みが起動された時に 呼ばれる関数)を設定する関数です。
- ・TIMER_interrupt()関数はADCに対して変換を開始します。
- ・ADC0_StartConversion()関数はADCの温度チャネルでの変換を開始します。
- ・ ADC0_GetConversionResult() 関数は最後の変換結果を返します。

3.2. Source Files(ソース ファイル)⇒MCC Generated files(MCC生成ファイル)⇒src(ソース)で見つかるpin_manager.cファイルに於いてヒ°んの 方向に対するコードを以下で置き換えてください。

PORTA. DIR = 0xFF; PORTB. DIR = 0xFF; PORTC. DIR = 0x3F; PORTD. DIR = 0xFF; PORTE. DIR = 0xFF; PORTF. DIR = 0xFF;

情報:これは電力読み取りを乱す浮きピンを防ぐために行われます。

3.3. ツール ハーでClean and Build(解消して構築)釦を押してプログラムが誤りなしで構築したこ とを確認してください。





1.5. 事象システム実行プロジェクト作成

🚰 行うこと: このプロジェクトで使われるハードウェア単位部を構成設定してプロジェクトにコードを追加してください。

- 1. 新しいAVR128DA48用MPLAB Xプロジェクトを作成してください。
 - 1.1. MPLAB Xを開いてください。
 - 1.2. File(ファイル)→New Project(新規プロジェクト)またはNew Pproject(新規プロジェクト)釦を選んでください。

図1-	·18. ≆	新規フ	゚ロジェク	٢
File	Edit	View	Naviga	te
÷ 🕈	- F	1p		1

1.3. Next(次へ)をクリックしてください(既定でStandalone Project(Microchip組み込み独立型プロジェクト)が選ばれています)。

図1-9. プロジェクト型式		
😰 New Project		×
Steps	Choose Project	
1. Choose Project	Q, Filter:	
	Categories: Microchip Embedded Other Embedded ⊕Èm Samples	Projects: Standalone Project Existing MPLAB IDE v8 Project Prebuilt (Hex, Loadable Image) Project User Makefile Project Library Project Import START MPLAB Project Import Atmel Studio Project
	Description:	
	Creates a new standalone application project	ct. It uses an IDE-generated makefile to build your
	< Back	Next > Finish Cancel Help

1.4. Device(デバイス)領域でAVR128DA48に対して検索してください。Tool(道具)区分でコンピュータに接続されていればCuriosity N ano基板を選び、さもなければNone(なし)を選んでください。Next(次へ)をクリックしてください。

図1-20. デハイス選択 図 New Project		
Steps 1. Choose Project 2. Select Device	Select Device	
 Select Header Select Plugin Board Select Compiler Select Project Name and Folder 	Pamiry: Device:	AVR128DA48
	Tool:	AVR128DA48 Curiosity Nano-SN: MCH v Show All
		< Back Next > Finish Cancel Help

1.5. XC8 2.20コンハパラを選んでNext(次へ)をクリックしてください。

図1-21. コンパイラ選択		
😰 New Project		×
 Steps Choose Project Select Device Select Header Select Plugin Board Select Project Name and Folder 	Select Compiler Compiler Toolchains C XC8 C XC8 C XC8 C XC8 (v2.20) [C:\Program Files\Wicrochip\xc8\v2.20\bin] XC8 (v2.10) [C:\Program Files (x86)\Wicrochip\xc8\v2.10\bin] XC8 (v2.05) [C:\Program Files (x86)\Wicrochip\xc8\v2.05\bin] AVR O avrasm12 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	
	< Back Next > Finish Cancel He	lp

1.6. プロジェクト(と保存する場所)に名前を与えてFinish(終了)をクリックしてください。

図1-22. プロジェクト名			
🔀 New Project			×
Steps	Select Project Name an	nd Folder	
1. Choose Project 2. Select Device 3. Select Header	Project Name:	avr-da-cnano-low-power-lab-evsys	
 Select Plugin Board Select Compiler 	Project Location:	C: Wy stuff\Projects	Browse
6. Select Project Name and Folder	Project Folder:	My stuff/Projects/avr-da-cnano-low-power-lab-evsys.X	
MPLAB X IDE	 Overwrite existing p Also delete sources. Set as main project Use project location 	roject. as the project folder 859-1	
		< Back Next > Finish	Cancel Help

- 2. MPLABコート、構成部(MCC)を開いて周辺機能を構成設定してください。
 - 2.1. System Module(システム単位部)でクロック元を24MHzクロックを持つ内部発振器として選んでください。

情報: Clock Source(クロック元)をInternal Oscillator(内部発振器)として、Oscillator Frequency Options(発振器周波数 任意選択)を24MHz system clock(24MHzシステム クロック)として選び、Prescalor(前置分周器)を禁止してください。

図1-23. システム単位部構	成設定		
System Module			
🔯 Easy Setup 📃 Registers			
Clock Control			
Main Clock(Hz):	24000000		
Olock Source :	Internal Oscillator		
Internal Oscillator Frequency:	1-32MHz internal oscillator		
Oscillator Frequency Options:	24 MHz system clock		
PLL Enable:			
External Clock(Hz):	1 ≤ 1000000 ≤ 20000000		
Prescaler Enable:			
Prescaler:	6X 👻		
Olock Out Enable:			
 Watchdog Timer 			
 Brown-out Detector 			
► Voltage Level Monitor			

1

2.2. System Module(システム単位部)のRegisters(レシ、スタ)タブで休止を許可して動作形態をスタンバイに設定してください。

情報: System Module(システム単位部)のRegisters(レジスタ)タブ へ行き、SLPCTRL ヘ下スクロールしてください。CTRLAレジス タを休止許可(SEN=enabled)に変更して、動作形態(SMODE=Standby Mode)を選んでください。

図1−24. 休止構成設定
System Module
Say Setup Registers
* SLPCTRL
Register: SLPCTRLVREGCTRL Ox0 PMODE AUTO

2.3. Device resources(デバイス資源)からVREF単位部を追加し、ADCへ2.048V参照基準を提供するように構成設定してください。



情報: ADC基準電圧は内部2.048V参照基準に構成設定されなければなりません。AC電圧強制(enable Force AC Vo ltage)任意選択を許可しないでください。

情報: VREFはマイクロ コントローラの温度感知器を許可するのに使われます。

'REF		
🎲 Easy Setup 📃 Registers		
 Software Settings 		
API Prefix:	VREF_0	
 Hardware Settings 		
Enable Force ADC Voltage Reference:		
ADC Voltage Reference:	Internal 2.048V reference	-
Enable Force DAC/AC Voltage Reference:		
DAC/AC Voltage Reference:	Internal 1.024V reference	-
Enable Force AC Voltage Reference:		
AC Voltage Reference:	Internal 1.024V reference	*

- **2.4.** Device resources(デバイス資源)からADC単位部を追加し、それを12ビット動作、右揃え結果、累積なし、採取長1刻時、スタンバイで走行(RUNSTBY)に構成設定してください。事象を受け取る時に変換が開始されるように事象入力開始(STARTEI)ビットも許可してください。
 - 情報: Left Adjust Result(左揃え結果)任意選択は禁止(チェックなしに)されなければなりません。Sample Length(採取長) とSample Accumulation Number(採取累積数)はHardware Settings(ハート・ウェア設定)枠で選ばれます。12ビット動作 は既定で設定されています。RUNSTBY(スタンハ・イ時走行)ビットはRegisters(レジスタ)タフ・で変更することができ、CT RLAレジ、スタで見つかります。STARTEILビットはRegisters(レジ、スタ)タフ・のEVCTRLレジ、スタで見つかります。

図1-26. ADC構成設定

ADC0		@
🔯 Easy Setup 📃 Registers		
Software Settings		
API Prefix:	ADC0	
Result Selection :	12-bit mode	
Oifferential Mode Conversion :	disabled 👻	
② Left Adjust Result :		
▼ Hardware Settings		
② Enable ADC:	\checkmark	
Sampling Frequency(Hz):	272727 ≤ 857142 ≤ 923076	
ADC Clock(Hz):	12000000	
Sample Accumulation Number:	No accumulation	
(2) Sample Length (# of ADC Clock) :	0 ≤ 1 ≤ 31	

図1-27. RUNSTBY ADC構成設定

ADCO
Basy Setup Registers Register: COMMAND 0x0
SPCONV disabled -
STCONV disabled -
▼ Register: CTRLA 0x81
CONVMODE disabled V
ENABLE enabled
FREERUN disabled
ULEFTADJ disabled
RESSEL 12-bit mode
RUNSTBY enabled -
▼ Register: CTRLB 0x0
SAMPNUM No accumulation +

図1-28. 事象で変換開始、ADC構成設定

ADCO
Easy Setup Registers
▼ Register: CTRLD 0x0
INITDLY Delay 0 CLK_ADC cycles
SAMPDLY 0x0
▼ Register CTRLE _{0x0}
WINCM No Window Comparison
DBGRUN disabled
▼ Register: EVCTRL _{0x1}
STARTEI enabled -
▼ Register: INTCTRL 0x0
RESRDY disabled -
WCMP disabled •

2.5. Device resources(デバイス資源)からRTC単位部を追加し、それを内部32.768kHz発振器使用、1秒の周期、前置分周係数1、 有効にされたスタンバイで走行(RUNSTBY)を持つように構成設定してください。

1		ı
	T	
_		

情報: クロック、Prescaling Factor(前置分周係数)、Period(周期)はHardware Settings(ハート・ウェア設定)枠で設定される一 方で、RUNSTBY(スタンハイ時走行)ビットはCTRLAレジスタのレジスタ枠で見つかります。溢れ割り込み(Overflow Inte rrupt)が許可されている場合、それを禁止してください。

図1-29. RTC構成設定	
RTC	0 🖲
🛱 Easy Setup 📃 Registers	
 Software Settings 	^
API Prefix:	RTC
 Hardware Settings 	
@ Enable RTC:	
RTC Clock(Hz):	32768
RTC Clock Source Selection:	Internal 32.768 kHz oscillator
External Clock(Hz):	1 ≤ 32000 ≤ 32000
Prescaling Factor:	RTC Clock / 1
Ompare:	1 s ≤ 0 s ≤ 2 s
Actual Compare:	0 s
Period:	1s ≤ 1s ≤2s
Actual Period:	15
Periodic Interrupt Timer	
 Interrupt Settings 	
Oompare Match Interrupt Enable:	
Overflow Interrupt Enable:	

c	
Easy Setup Registers Register: CMP 0x0	
r Register: CNT 0x0	
Register: CTRLA 0x81	
CORREN disabled	
PRESCALER RTC Clock / 1 👻	
RTCEN enabled 👻	
RUNSTDBY enabled	
Register: DBGCTRL 0x0	
DBGRUN disabled -	

2.6. EVSYS単位部を追加してください。チャネル0事象生成部をRTC_OVFに、起動される事象をADC0STARTに構成設定してください。 い。チャネル1事象生成部をADC0_READYに、起動される事象をEVSYSEVOUTCに構成設定してください。

図1-31. EVSYS構成設定

EVSYS

EVSYS						
Channels						
	ADC0START	CCLLUT0A	CCLLUT0B	CCLLUT1A	CCLLUT1B	CCLLUT2A
CHANNEL0	\checkmark					
CHANNEL1						
CHANNEL2						
CHANNEL3						
CHANNEL4						
CHANNEL5						
CHANNEL6						
CHANNEL7						
CHANNEL8						
CHANNEL9						
	Channels Channels CHANNEL0 CHANNEL1 CHANNEL2 CHANNEL3 CHANNEL4 CHANNEL5 CHANNEL5 CHANNEL5 CHANNEL6 CHANNEL8 CHANNEL8	EVSYS ADC0START Channels CHANNEL0 Image: Colspan="2">Image: Colspan="2">Channel1 CHANNEL1 Image: Colspan="2">Image: Colspan="2" Image: Colspan="2	EVSYS ADC0START CCLLUT0A CHANNEL0 I<	EVSYS Channels ADC0START CCLLUT0A CCLLUT0B CHANNEL0 I I I CHANNEL1 I I I I CHANNEL2 I I I I I CHANNEL3 I <thi< th=""> I <thi< th=""> <thi< th=""></thi<></thi<></thi<>	EVSYS Channels ADC0START CCLLUT0A CCLLUT0B CCLLUT1A ADC0START CCLLUT0A CCLLUT0B CCLLUT1A CHANNEL0 I I I I CHANNEL1 I I I I I CHANNEL2 I I I I I I CHANNEL3 I <thi< th=""> I I <thi< th=""></thi<></thi<>	EVSYS Channels CCLLUT0A CCLLUT0B CCLLUT1A CCLLUT1B ADCOSTART CCLLUT0A CCLLUT0B CCLLUT1A CCLLUT1B CHANNEL0 I I I I I CHANNEL1 I I I I I I CHANNEL2 I

図1-32. EVSYS構成設定

EVSYS								0
🔅 Easy Setup	Registers							
 Software Setti 	ings							
API Prefix:		EVSYS						
▼ Event System	Settings							
CCLLUT5A	CCLLUT5B	EVSYSEVOUTA	EVSYSEVOUTB	EVSYSEVOUTC	EVSYSEVOUTD	EVSYSEVOUTE	EVSYSEVOUTF	EVSYSEVOU
				\checkmark				

2.7. Pin Manager: Grid View(ピン管理部:格子状表示)でEVSYSに対してEVOUTCとしてPC2を選んでください。

Variables	Output	N	otifications [MC	[C]	P	in M	anaç	jer: G	irid \	liew	×																																	
Package:	QFN48	•	Pin No:	44	45	46	47	48	1	2	3	4	5	6	7	8	9	10	11	12	13	16	17	18	3 19	20	21	22	23	24	25	26	27	30	31	32	33	34	35	36	37	38	39	40
							Por	t A 🔻	,					Por	B 🔻	,					Por	t C 🖲	,						Por	t D 🔻	,				Por	t E 🔻	,			P	ort F	•		
Module	Funct	ion	Direction	0	1	2	3	4	5	6	7	0	1	2	3	4	5	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	0	1	2	3	4	5	6
ADC0	AINx		input																							ĵ,	ı în	î	î	î	î	î	î	î	î	ĥ	î.	ĥ	î	î	î	î	î.	
	EVOUTA		output			î.					î																																	
	EVOUTB		output											ì																														
5. (C) (C -	EVOUTC		output																	â					ì																			
EVSYS V	EVOUTD		output																									î.					Ъ											
	EVOUTE		output																																	ì								
	EVOUTF		output																																					ì				
	GPIO		input	Ъ	î.	î.	ĵ.	î.	Ъ	î.	î.	ì	ì	î.	Ъ	î.	î.	Ъ	Ъ	ĵ.	ì	ĥ	î.	1	l î	n n	ı în	î.	î.	î.	ì	î.	î.	î.	î.	Ъ	î.	Ъ	h	î.	ĥ	î.	Ъ	î.
Pin Module V	GPIO		output	ĵ,	î.	î	ĵ.	î.	ĥ	î.	î.	î	î	ì	ĵ.	î	î	ĥ	ĥ	î.	î	î.	ì	ì	î.	1	ı în	î.	î	î.	î	î	ì	ì	î.	î.	î.	ĥ	h	î	ì	î	Ъ	î.
RSTCTRL	RESET		input																						1																			î.

図1-33. ピン管理部:格子状表示

2.8. Pin Module(ビン単位部)に於いて両端で感知するよう(Sence Both Edges)にPC2ビンの変化での割り込み(ISC)を許可してください。

図1-34. ピン単	位部構成設定	定										
Pin Module	Pin Module											
ිදියි Easy Setup	Easy Setup Registers											
Pin Name 🔺	Module	Function	Custom Name	OUTPUT	START HIGH	INVEN	PULLUPEN	ISC				
PC2	EVSYS	EVOUTC		\checkmark				Sense Both Edges	-			

2.9. Interrupt Manager(割り込み管理部)で全体割り込み(Global Interrupt)を許可してください。

I		
Interrupt Manager		
🎲 Easy Setup 📃 Registers		
 Interrupt Setting 		
Global Interrupt Enable:	V	
 Interrupt Priority 		
Round-robin Scheduling Enable:		
Interrupt Level Priority:	0	
Interrupt Vector with High Priority:	0	

2.10. Generate(生成)を押してください。

Projects	Files	Services	Resource Management [MCC] ×	=
Tree View	Flat View			
Project	Resourc	es Genera	te Import Export	
▼ System				
Intern	upt Manag	jer		
Pin M	lodule			
Syster	m Module			
▼ Periphe	rals			
🕜 👂	🛾 🔦 ADO	0		
🕜 👂	🔇 🛃 EVS	YS		
<u> </u>	🛛 🚔 ртс			

3. プロジェクトにコートを追加してください。

3.1. main.cファイルに以下のコードを追加してください。

#include "mcc generated files/mcc.h"
#inglude /own/gloop h
#Include \avr/sleep. n/
void GPIO Interrupt (void):
void drie_interrupt (void) ;
uint16_t volatile result;
int main(vold)
SVSTEM Initialize().
ADCO.MUXPOS = ADC_MUXPOS_TEMPSENSE_gc;
PORTC PC2 SetInterruptHandler(GPI0 Interrupt);
while (1) {
sleep cpu();
}



3.3. ツール ハーでClean and Build(解消して構築)卸を押してプログラムが誤りなしで構築したことを確認してください。

図1-3	7. 解消し	して構築
j.	1	D - 🕹 - 🛣

GitHubでコード例を見てください。 貯蔵庫を閲覧するにはクリックしてください。
--

1.6. デバイス書き込みと効率比較

🕌 行うこと: プロジェクトをマイクロ コントローラに設定して(書いて)電力読み取りを行ってください。

1. ソフトウェア実行プログラムで電力読み取りを得てください。

- 1.1. Curiosity Nano基板をコンピュータに、2つの半田付けしたピンを共に、接続してください。
- 1.2. 主プログラムとして設定したソフトウェア実行プログラグでツール ハーのMake and Program Device(作成(構築)してデハイスに書き込み)釦を押してください。書き込み器として使われるツールを選ぶことを尋ねるポップアップが現れるかもしれません。AVR128 DA48 Curiosity Nano基板を選んでください。

図1-38. 作	F成(構築	き)して	こデバイス	(書き込み
5	• 👸 •	\triangleright	•	~

1.3. Power Debuggerをコンピュータに接続してください。1つ目のピン(Nano組み込みデバッガに最も近いピン)をAチャネル電流計の内向 き矢印に、2つ目のピン(マイクロコンピュータに近いピン)をAチャネル電流計の外側を指す矢印に接続してください。



- 1.4. Atmel StudioでTools(ツール)⇒Data Visualizer(データ可視器)へ行ってください。
- 1.5. 2つの枠が現れます。DGI (Data Gateway Interface) Control Panel(DGI(データ中継器インターフェース)制御盤)と呼ばれる枠をチェッ クしてください。これが現れない場合、左側のConfiguration(構成設定)メニューをアクセスすることができ、Modules(単位部)⇒Exter nal Connection(外部接続)部分でデータ中継器インターフェースを見つけることができます。

义	1-40. データ中継器イン	ターフェース
Data	Visualizer +> × Configuration - Modules ?	DGi Control Panel
onfiguration	External Connection Data Gateway Interface (DGI) Senal Port Visualization	Curiosity Data Gateway Interface
	 Utilities Protocols 	Interfaces Serial Port Control Panel v x

1.6. Connect(接続)を押すと'Power(電力)'と名付けられた枠が現れます。メニューを開くためにそれを選んでその後に歯車を押してください。Bチャネルを禁止してください。

▲警告 Power(電力)領域が現れない場合、基板を	切断して再び接続してください。	
図1-41. DGI接続 Digi Control Panel		× ×
Power Debugger Data Gateway		Connect
Interfaces		ADP Logging Autodetect protocols 🔽 Reset MCU
図1−42. 電力設定		
Power Debugger Data Gateway	Power Configuration X Enable B Channel Trigger calibration	Disconnect Start ADP Logging Autodetect protocols
Interfaces: SPI ☆ USART ☆ TWI ☆ GPIO ☆ Ø Power ☆ Code Profiling ☆	Lock ChA to High Range	

Enable Voltage Output

No

OK Cancel

Voltage Output

Averaging

1600

m

Code Location

C ... 0

B Current C

B Voltage

2 💕 💿

3 🌈 🙆

図1-44. 作成(構築)してデバイス書き込み

Y

P_

1.7. 全てが正しく構成設定された時にStart(開始)を押すと、実時間で消費電力を表示する新しい枠が現れます。

提供します。 図1-43. 監視開始 Power Debugger Data Gateway Disconnect Start ADP Logging Auto tect protocols 🔽 Re nterface 🗆 SPI 🙀 🗆 USART 🏠 🗆 TWI 🏠 🗆 gpio 🔅 ✓ Power XX
Code Profiling XX **6 0** Car Code Location 0 🕬 💿 A Current A Voltage 🗨 2 💕 💿 3 ch A o 3910.5 3911.0 3911.5 3912.0 3917.0 3910.0 3912.5 3913.0 3913.5 3914.0 3914.5 3907.5 3908.0 3908.5 3909.0 3909.5 3915.0 3915.5 3916.0 3916.5 ର୍ ତ୍ om in the time axis. 🗵

助言: 表示を調整するには枠の左側で制御盤を開いてください。Show Zero(0表示)のチェック解除はより鮮明な画像を

1.8. 読み取りが安定した後、Stop(停止)を押してその後に切断してください。電力測定の枠は開いたままです。

- 2. 事象システム実行プログラムからの消費電力読み取りを取得してください。
 - 2.1. Curiosity Nano基板上の2つの半田付けしたピンを共に接続してください。
 - 2.2. 事象システム実行プロジェクトを主プログラムとして設定し、MPLAB Xのツール ハーでMa ke and Program Device(作成(構築)してデバイスに書き込み)釦を押してください。
 - 2.3. Curiosity Nano基板をPower Debuggerに接続してください。詳細と写真については本節の1.3.段階をご覧ください。
 - 2.4. Atmel Studioでデータ可視器(Data Visualizer)へ行き、DGI盤でConnect(接続)を押してください。

図1-45. データ中継器インターフェース	
DGI Control Panel	^ X
Power Debugger Data Gateway ISCOCCCCS66	Connect
Interfaces	ADP Logging Autodetect protocols ✔ Reset MCU

2.5. Power(電力)枠をチェックしてBチャネルを禁止するように設定を編集してください。

凶1-40. 电刀汉上	
DGI Control Panel	**
Power Debugger Data Gateway	Ø Power Configuration X
///////////////////////////////////////	Enable B Channel Start
	Trigger calibration
Interfaces:	Enable Range Source
□ SPI 🏠 □ USART 🏠 □ TWI 🏠 □ GPIO 🏠 🗹 Power 🏠 □ Code Profiling 🏠	Lock ChA to High Range
Code Location	Enable Voltage Output
1 🌑 💿 A Voltage 🚭	Voltage Output
2 🌓 B Current 🌓	Averaging No Y
3 🌑 🖉 B Voltage 🌑	OK Cancel

2.6. Satrt(開始)を押すと、実時間消費電力を表示する新しい枠が現れます。より鮮明な画像のためShow Zero(0表示)を禁止して ください。

図1−47. 監視開始	
DGI Control Panel Power Debugger Data Gateway S00000566	Disconnect Start
Interfaces: □ <t< th=""><th>itect protocols 🗹 Reset MCU</th></t<>	itect protocols 🗹 Reset MCU
Power Analysis	Channel A Channel B Chane
3725 3726 3727 3728 3729 3730 3731 3732 3733 3734 Window Average Instant Lip 	

- 2.7. 読み取りが安定した後、固定した結果を持つようにStop(停止)を押してください。
- 3.2つの測定を比較してください。
 - 3.1. 両方の消費電力図が同じ画面で開いています。 今やプログラムの挙動の様子を観測するためにこれらを分析することができます。

図1-48	. EVSYSとと	比較される	るソフトウェブ	′実行									
Power Analysis													
Ch A Window Aver	age Instant	0 3725.5 3	726.0 3726.5 key to zoom in the time a	3727.0 37	27.5 3728.0	3728.5 372	9.0 3729.5 :	3730.0 3730.5	3731.0 37	31.5 3732.0	3732.5 3	733.0 373	3.5 3734.0
400μA – 340μA – 220μA – 220μA – 160μA – 100μA – -20μA – -140μA – -140μA – -200μA –													
Ch A Window Aver	3907.5 3908.0	3908.5 3909.0	3909.5 39	3910.5	3911.0 391	1.5 3912.0	3912.5 3913.0	3913.5 3914	0 3914.5	3915.0 3	915.5 3916.0	3916.5	3917.0

i ^情

情報:上部は事象システム実行の消費電力を表示、同時に下部は割り込み実行の消費電力を表示します。図から見ることができるように、違いは約1.9µAとかなりの小ささです。これはマイクロコントローラが休止で費やす長い時間のためです。

2. 課題2 : A/D変換器(ADC)に対してハート・ウェア累積と比べたソフトウェア累積

この課題では2つの異なるADC採取手法の消費電力が比較されます。ADCでの結果累積は結果の精度を増して雑音の影響を最小にする役割を持ちます。

1つ目の手法はそれら128個が完了されるまで再三変換を起動します。結果は累積器として働く変数に追加されます。

2つ目の手法はADC用組み込み累積器を利用します。これは128変換を走行して結果をハードウェア累積器に加算するのにADCの累 積変換動作を使います。

2.1. ソフトウェア実行プロジェクト作成

「行うこと: このプロジェクトで使われるハート・ウェア単位部を構成設定し、プロジェクトにコートを追加してください。

1. 新しいAVR128DA48用MPLAB Xプロジェクトを作成してください。

- 1.1. MPLAB Xを開いてください。
- 1.2. File(ファイル)→New Project(新規プロジェクト)またはNew Pproject(新規プロジェクト)釦を選んでくださ 図2-1. 新規プレシ。

図2−1. 新規フ ロシ ェクト								
File Edit View Navigate								
: "II "								

1.3. Next(次へ)をクリックしてください(既定でStandalone Project(Microchip組み込み独立型プロジェクト)が選ばれています)。

図2-2. プロジェクト型式		
🔀 New Project	;	×
Steps	Choose Project	_
1. Choose Project	Q Filter:	
2	Categories: Projects: Image: Constraint of the state of	
	Description:	_
	Creates a new standalone application project. It uses an IDE-generated makefile to build your project.	
	< Back Next > Finish Cancel Help	

1.4. Device(デバイス)領域でAVR128DA48に対して検索してください。Tool(道具)区分でコンピュータに接続されていればCuriosity N ano基板を選び、さもなければNone(なし)を選んでください。Next(次へ)をクリックしてください。

図2-3. デバイス選択			
🔀 New Project			×
Steps	Select Device		
Choose Project Select Device Select Header Select Plugin Board Select Compiler Select Compiler Select Project Name and	Family: Device:	All Families V AVR 128DA48 V	
Folder	Tool:	AVR 128DA48 Curiosity Nano-SN: MCH 🗸 🗌 Show All	
MPLAB X IDE			
		<back next=""> Finish Cancel Help</back>	

1.5. XC8 2.20コンハパラを選んでNext(次へ)をクリックしてください。

図2-4. コンパイラ選択		~
 New Project Steps 1. Choose Project Select Device Select Header Select Project Name and Folder 	Select Compiler Compiler Toolchains CXC8 (V2.20) [C: \Program Files \Microchip\xc8\v2.20\bin] CXC8 (V2.10) [C: \Program Files (x86) \Microchip\xc8\v2.10\bin] CXC8 (V2.10) [C: \Program Files (x86) \Microchip\xc8\v2.05\bin] CXC8 (V2.05) [C: \Program Files	
	< Back Next > Finish Cancel He	lp

1.6. プロジェクト(と保存されるべき場所)に名前を与えてFinish(終了)をクリックしてください。

図2-5. プロジェクト名		
😰 New Project		×
Steps	Select Project Name a	and Folder
1. Choose Project 2. Select Device 3. Select Header	Project Name:	avr-da-cnano-low-power-lab-interrupt
 Select Plugin Board Select Compiler 	Project Location:	C: Wy stuff Projects Browse
6. Select Project Name and Folder	Project Folder:	stuff/Projects/avr-da-cnano-low-power-lab-interrupt.X
MPLAB X IDE	Overwrite existing Also delete source Set as main projec Use project locatio Encoding: ISO Project name and fol Try shortening the pr	project. s. t n as the project folder 8859-1 der path length are nearing the Windows limit. This may cause issues during build or ¢ oject name or path.
		< Back Next > Finish Cancel Help

- 2. MPLABコート、構成部(MCC)を開いて周辺機能を構成設定してください。
 - 2.1. 内部発振器を24MHzで動くようにSystem Module(システム単位部)を構成設定してください。
 - 情報: Clock Source(クロック元)はInternal Oscillator(内部発振器)、Oscillator Frequency Options(発振器周波数任意選択)は24MHz system clock(24MHzシステム クロック)です。Prescalor(前置分周器)任意選択は禁止されなければなりません。

図2-6. システム単位部構成	找設定	
System Module		?
🔯 Easy Setup 📄 Registers		
Clock Control		
Main Clock(Hz):	24000000	
Clock Source :	Internal Oscillator	-
Internal Oscillator Frequency:	1-32MHz internal oscillator	-
Oscillator Frequency Options:	24 MHz system clock	•
PLL Enable:		
External Clock(Hz):	1 ≤ 1000000 ≤ 20000000	
Prescaler Enable:		
Prescaler:	6X	~
Olock Out Enable:		
► Watchdog Timer		
Brown-out Detector		
Voltage Level Monitor		

2.2. Device resources(デバイス資源)からVREF単位部を追加し、ADCへ2.048V参照基準を提供するように構成設定してください。



情報: VREFはマイクロコントローラの温度感知器を許可するのに使われます。

REF		•
🕃 Easy Setup 📃 Registers		
 Software Settings 		
API Prefix:	VREF_0	
 Hardware Settings 		
Enable Force ADC Voltage Reference:		
ADC Voltage Reference:	Internal 2.048V reference	-
Enable Force DAC/AC Voltage Reference:		
OAC/AC Voltage Reference:	Internal 1.024V reference	*
Enable Force AC Voltage Reference:		
AC Voltage Reference:	Internal 1.024V reference	-

2.3. Device resources(デハイス資源)からADC単位部を追加し、それを12ビット動作、右揃え結果、累積なし、採取長31刻時に構成 設定してください。



情報: Left Adjust Result(左揃え結果)任意選択は禁止(チェックなしに)されなければなりません。Hardware Settings(ハートウェア設定)枠でSample Length(採取長)が選ばれます。12ビット動作は既定で設定されています。これは累積選択なしでも同じです。

図2-8. ADC構成設定	
ADC0	
🔯 Easy Setup 📃 Registers	
 Software Settings 	
API Prefix:	ADC0
Result Selection :	12-bit mode
Oifferential Mode Conversion :	disabled 👻
② Left Adjust Result :	
 Hardware Settings 	
Parable ADC:	\checkmark
() Sampling Frequency(Hz):	272727 ≤ 272727 ≤ 923076
ADC Clock(Hz):	12000000
Sample Accumulation Number:	No accumulation
(2) Sample Length (# of ADC Clock) :	0 ≤ 31 ≤ 31
► Interrupt Settings	
Select Channels	
Window Settings	

2.4. Generate(生成)釦を押してください。

図2-9. 生成	釦			
Projects	Files !	Services	Resource Management [MCC] ×	_
Tree View F	lat View			
Project Re	esources	Generate	Import Export	
▼ System				
Interrup	t Manager			
Pin Moo	dule			
System	Module			
 Peripheral 	s			
🔍 🖸	ADC0			
	VREF			

3. プロジェクトにコートを追加してください。

3.1. main.cファイルに以下のコードを追加してください。



情報: SYSTEM_Initialize()はmcc.cで定義され、ADC0_StartConversion(channel)、ADC0_IsConversionDone()、ADC0_GetConversionResult()はadc0.cで定義されます。

・SYSTEM_Initialize()はMCCで何が設定されたかに従ってマイクロコントローラと周辺機能を構成設定する関数です。これはMCCによって生成されプログラムの始めで呼ばれることが必要です。

- ・ADC0_StartConversion(channel)は与えられたチャネルでADC変換を開始します。
- ・ADC0_IsConversionDone()は変換の状態を返します。進行中に対して0、終了済みに対して1です。
- ・ ADC0_GetConversionResult()は最後の変換結果を返します。
- ・for繰り返しは変換を開始してその終了を待ち、結果を128回累積器に加算する手順を繰り返します。
- 3.2. Source Files(ソース ファイル)⇒MCC Generated files(MCC生成ファイル)⇒src(ソース)で見つかるpin_manager.cファイルに於いてピんの 方向に対するコードを以下で置き換えてください。

PORTA. DIR = 0xFF; PORTB. DIR = 0xFF; PORTC. DIR = 0x3F; PORTD. DIR = 0xFF; PORTE. DIR = 0xFF; PORTF. DIR = 0xFF;

情報:これは電力読み取りを乱す浮きピンを防ぐために行われます。

3.3. ツール バーでClean and Build(解消して構築)釦を押してプログラムが誤りなしで構築したことを確認してください。





2.2. ハート・ウェア実行プロジェクト作成

「行うこと: このプロジェクトで使われるハート・ウェア単位部を構成設定してプロジェクトにコートを追加してください。

- 1. 新しいAVR128DA48用MPLAB Xプロジェクトを作成してください。
 - 1.1. MPLAB Xを開いてください。
 - 1.2. File(ファイル)→New Project(新規プロジェクト)またはNew Pproject(新規プロジェクト)釦を選んでくださ 図2-11. 新規プロジェクト い。

図2-11. 羗	訂規 ノロン エク	
File Edit	View Naviga	te
P P		i I

1.3. Next(次へ)をクリックしてください(既定でStandalone Project(Microchip組み込み独立型プロジェクト)が選ばれています)。

図2-12. ブロジェクト型式	
🔀 New Project	×
Steps	Choose Project
1. Choose Project	Q Filter:
2	Categories: Projects: Image: Samples Standalone Project Image: Samples Standalone Project Image: Samples User Makefile Project Image: Samples Image: Samples
	Description:
	Creates a new standalone application project. It uses an IDE-generated makefile to build your project.
	< Back Next > Finish Cancel Help

1.4. Device(デバイス)領域でAVR128DA48に対して検索してください。Tool(道具)区分でコンピュータに接続されていればCuriosity N ano基板を選び、さもなければNone(なし)を選んでください。Next(次へ)をクリックしてください。

😰 New Project	_	×
Steps	Select Device	
Choose Project Select Device Select Header Select Plugin Board Select Project Name and Folder	Family: Device:	All Families v
	Tool:	AVR 128DA48 Curiosity Nano-SN: MCH 🗸 🗌 Show All
MPLAB X IDE		
		< Back Next > Finish Cancel Help

1.5. XC8 2.20コンパイラを選んでNext(次へ)をクリックしてください。

New Project		×
Steps 1. Choose Project 2. Select Device 3. Select Header 4. Select Compiler 5. Select Compiler 6. Select Project Name and Folder	Select Compiler Compiler Toolchains - XC8 - XC8 (v2.20) [c:\Program Files\Microchip\xc8\v2.20\bin] - XC8 (v2.05) [C:\Program Files (x86)\Microchip\xc8\v2.05\bin] - AVR - avrasm2 - TAR for AVR - pic-as	

1.6. プロジェクト(と保存されるべき場所)に名前を与えてFinish(終了)をクリックしてください。

図2-15. プロジェクト名		
😰 New Project		×
Steps	Select Project Name a	nd Folder
Choose Project Select Device Select Header Select Plugin Board	Project Name:	avr-da-cnano-low-power-lab-interrupt
5. Select Compiler 6. Select Project Name and Folder	Project Location: Project Folder:	C: VMy stuff Projects Browse
	Overwrite existing	project. s. t n as the project folder
	Encoding: ISO- Project name and fold Try shortening the pro-	8859-1 v der path length are nearing the Windows limit. This may cause issues during build or ; oject name or path.
		< Back Next > Finish Cancel Help

2. MPLABコート構成部(MCC)を開いて周辺機能を構成設定してください。

2.1. 内部発振器を24MHzで動くようにSystem Module(システム単位部)を構成設定してください。



stem Module		•
🖟 Easy Setup 📃 Registers		
Clock Control		
Main Clock(Hz):	2400000	
Olock Source :	Internal Oscillator	•
Internal Oscillator Frequency:	1-32MHz internal oscillator	•
Oscillator Frequency Options:	24 MHz system clock	*
PLL Enable:		
External Clock(Hz):	1 ≤ 1000000 ≤ 20000000	
Prescaler Enable:		
Prescaler:	6X	-
Olock Out Enable:		
 Watchdog Timer 		
Brown-out Detector		

2.2. System Module(システム単位部)のRegisters(レシブスタ)タブで休止を許可して動作形態をスタンバイに構成設定してください。

報	: System Module(システム単位部)のRegisters(レジ、スタ)タブ、行き、SLPCTRLに達するまで下スクロールしてください。SL PCTRL.CTRLAレジ、スタンドイ動作にされるべき任意選択を変更してください。
	网 17 년 년 世代記中

82 ^{-17.} 你正确成改定		
iystem Module		
Easy Setup Registers		
SWRST disabled		
▼ SLPCTRL		
Register: SLPCTRLCTRLA 0x3 SEN enabled SMODE Standby Mode Register: SLPCTRLVREGCTRL 0x0		
PMODE AUTO -		

2.3. Device resources(デバイス資源)からVREF単位部を追加し、ADCへ2.048V参照基準を提供するように構成設定してください。

情報: ADC基準電圧は内部2.048V参照基準に構成設定されなければなりません。AC電圧強制(enable Force AC Voltage)任意選択を許可しないでください。

情報: VREFはマイクロコントローラの温度感知器を許可するのに使われます。

情

REF		
🔅 Easy Setup 📃 Registers		
 Software Settings 		
API Prefix:	VREF_0	
 Hardware Settings 		
Enable Force ADC Voltage Reference:		
ADC Voltage Reference:	Internal 2.048V reference	*
Enable Force DAC/AC Voltage Reference:		
DAC/AC Voltage Reference:	Internal 1.024V reference	•
Enable Force AC Voltage Reference:		
AC Voltage Reference:	Internal 1.024V reference	*

- **2.4.** Device resources(デバイス資源)からADC単位部を追加し、それを12ビット動作、右揃え結果、128結果累積、採取長31刻時、スタンバイで走行(RUNSTBY)に構成設定してください。
 - **情報:** Left Adjust Result(左揃え結果)任意選択は禁止(チェックなしに)されなければなりません。Sample Length(採取長) とSample Accumulation Number(採取累積数)はHardware Settings(ハートウェア設定)枠で選ばれます。12ビット動作 は既定で設定されています。RUNSTBY(スタンハイ時走行)ビットはRegisters(レシ、スタ)タブで変更することができ、CT RLAレジスタで見つかります。

図2-19. ADC構成設定		
ADC0		
🔅 Easy Setup 📄 Registers		
 Software Settings 		
API Prefix:	ADC0	
Result Selection :	12-bit mode	
Oifferential Mode Conversion :	disabled	
😗 Left Adjust Result :		
 Hardware Settings 		
③ Sampling Frequency(Hz):	272727 ≤ 272727 ≤ 923076	
ADC Clock(Hz):	12000000	
Sample Accumulation Number:	128 results accumulated	
Sample Length (# of ADC Clock) :	0 ≤ 31 ≤ 31	
 Interrupt Settings 		
Result Ready Interrupt Enable:	\checkmark	
WCMP Interrupt Enable:		
Select Channels		
 Window Settings 		

Easy Setup Registers ADC0 Interrupt Enables RESRDV WCMP COMMAND 0x0 SPCONV disabled * SPCONV disabled * STCONV disabled * STCONV disabled * Enabled * Enabled * Enabled * Enabled * RESEL 12-bit mode * Register: CTRLB 0x7	DC0	
 ADCO Interrupt Enables ✓ RESRDY WCMP ✓ Register COMMAND 0x0 Ø SPCONV disabled * Ø STCONV disabled * ✓ Register CTRLA 0x81 Ø CONVMODE disabled * Ø ENABLE enabled * Ø FREERUN disabled * Ø FREERUN disabled * Ø RESSEL 12-bit mode * Ø RUNSTBY enabled * 	🔅 Easy Setup 📃 Registers	
Interrupt Enables	r ADCO	
RESRDY WCMP Register: COMMAND 0x0 SPCONV disabled T STCONV disabled T CONVMODE disabled T ENABLE enabled FREERUN disabled T ENABLE enabled T RESSEL 12-bit mode T Register: CTRLB 0x7	Interrupt Enables	
w KMP w Register: COMMAND 0x0 SPCONV disabled SPCONV disabled Register: CTRLA 0x81 CONVMODE disabled Rester: CTRLA 0x81 CONVMODE disabled Rester: CTRLA 0x81 A Register: CTRLA 0x7 A Regi	V RESRDY	
Register: COMMAND Ox0 SPCONV disabled STCONV disabled Register: CTRLA Ox81 CONVMODE disabled V Register: CTRLB 0x7	WCMP	
Register: COMMAND 0x0 SPCONV disabled STCONV disabled Register: CTRLA 0x81 CONVMODE disabled CONVMODE disabled ENABLE enabled Enabled Enabled Enabled Enabled Enabled Enabled Enabled		
 G SPCONV disabled ▼ G STCONV disabled ▼ Register: CTRLA 0x81 CONVMODE disabled ▼ ENABLE enabled ▼ FREERUN disabled ▼ UEFTADJ disabled ▼ RESSEL 12-bit mode ▼ RESSEL 12-bit mode ▼ RUNSTBY enabled ▼ 	▼ Register: COMMAND 0x0	
STCONV disabled Register: CTRLA 0x81 OxNVMODE disabled ENABLE enabled FREERUN disabled UEFTADJ disabled RESSEL 12-bit mode RESSEL 12-bit mode	SPCONV disabled +	
 STCONV aisabled ▼ Register: CTRLA 0x81 CONVMODE disabled ▼ ENABLE enabled ▼ FREERUN disabled ▼ LEFTADJ disabled ▼ RESSEL 12-bit mode ▼ RESSEL 12-bit mode ▼ RESSEL 12-bit mode ▼ 		
▼ Register: CTRLB 0x81 @ CONVMODE disabled @ ENABLE enabled @ FREERUN disabled @ FREERUN disabled @ LEFTADJ disabled @ RESSEL 12-bit mode @ RUNSTBY enabled	STCONV disabled	
 CONVMODE disabled ▼ ENABLE enabled ▼ FREERUN disabled ▼ LEFTADJ disabled ▼ RESSEL 12-bit mode ▼ RUNSTBY enabled ▼ ▼ Register: CTRLB 0x7 	▼ Register: CTRLA 0x81	
 enabled ▼ FREERUN disabled ▼ LEFTADJ disabled ▼ RESSEL 12-bit mode ▼ RUNSTBY enabled ▼ ★ Register: CTRLE 0x7 	CONVMODE disabled	
 FREERUN disabled ▼ LEFTADJ disabled ▼ RESSEL 12-bit mode ▼ RUNSTBY enabled ▼ ★ Register: CTRLE 0x7 	ENABLE enabled	
ILEFTADJ disabled ILEFTADJ disabled RESSEL 12-bit mode RUNSTBY enabled	ERFERIN disabled	
Q LEFTADJ disabled ▼ Q RESSEL 12-bit mode ▼ Q RUNSTBY enabled ▼ ▼ Register: CTRLB 0x7 0x7		
RESSEL 12-bit mode RUNSTBY enabled Register: CTRLB 0x7	LEFTADJ disabled T	
∂ RUNSTBY enabled ▼ ★ Register: CTRLB 0x7	RESSEL 12-bit mode -	
✓ Register: CTRLB 0x7	RUNSTBY enabled -	
	- Renister (TRIR 0.7	
	V Register on the UX/	

2.5. ADC用の結果準備可(Result Ready)割り込みと全体割り込み(Global Interrupt)を許可してください。

情報: ADC単位部に於いて、Interrupt Settings(割り込み設定)でResult Ready Interrupt Enable(結果準備可割り込み 許可)を有効にしてください。この設定は前の段階用の図で見ることができます。Interrupt Manager(割り込み管 理部)で全体割り込み(Global Interrupt Enable)を許可してください。

☑2−21.割り込み管理部構成設定					
Interrupt Manager	terrupt Manager				
Easy Setup 📃 Registers					
 Interrupt Setting 					
Global Interrupt Enable:	×				
Interrupt Priority					
Round-robin Scheduling Enable:					
Interrupt Level Priority:	0				
Interrupt Vector with High Priority:	0				

2.6. Generate(生成)釦を押してください。

図2-22. 生	図2−22. 生成釦				
Projects	Files	Services	Resource Management [MCC] ×		
Tree View	Flat View				
Project F	Resource	es Generat	e Import Export		
▼ System					
Interru	upt Manag	er			
Pin M	odule				
Syster	n Module				
 Peripher 	▼ Peripherals				
🛞 🔀	💿 🔀 🔩 ADC0				
8 X	🖌 — VRE	F			

3. プロジェクトにコートを追加してください。

```
3.1. main.c7rイルに以下のコードを追加してください。
#include "mcc_generated_files/mcc.h"
#include <avr/sleep.h>
uint16_t result;
int main(void)
{
   SYSTEM_Initialize();
   while (1) {
      ADCO_StartConversion(ADC_MUXPOS_TEMPSENSE_gc);
      sleep_cpu();
      result = ADC0_GetConversionResult();
   }
}
```

- 情報: SYSTEM_Initialize()はmcc.cで定義され、ADC0_StartConversion(channel)とADC0_ GetConversionResult()は adc0.cで定義されます。
 - ・SYSTEM_Initialize()はMCCで何が設定されたかに従ってマイクロコントローラと周辺機能を構成設定する関数です。これはMCCによって生成されプログラムの始めで呼ばれることが必要です。
 - ・ADC0_StartConversion(channel)は与えられたチャネルでADC変換を開始します。
 - ・ ADC0_GetConversionResult()は最後の変換結果を返します。
 - ・プログラムは変換を始めてその後に休止へ移行します。結果準備可割り込みによって起き上がり、結果を読ん でその後に新しい変換を開始します。
- 3.2. Source Files(ソース ファイル)⇒MCC Generated files(MCC生成ファイル)⇒src(ソース)で見つかるpin_manager.cファイルに於いてビんの 方向に対するコードを以下で置き換えてください。

PORTA. DIR = 0xFF; PORTB. DIR = 0xFF; PORTC. DIR = 0x3F; PORTD. DIR = 0xFF; PORTE. DIR = 0xFF; PORTF. DIR = 0xFF;

情報:これは電力読み取りを乱す浮きピンを防ぐために行われます。

3.3. ツール バーでClean and Build(解消して構築)釦を押してプログラムが誤りなしで構築したことを確認してください。





2.3. デバイス書き込みと電力効率比較

🚰 行うこと: プロジェクトをマイクロ コントローラに設定して(書いて)電力読み取りを行ってください。

1. デバイスに書いてソフトウェア実行プログラムで消費電力読み取りを行ってください。

1.1. Curiosity Nano基板をコンピュータに接続してください。2つの半田付けしたピンを共に接続してください。ソフトウェア累積プログラムを主プログラムとして設定し、ツール ハーのMake and Program Device(作成(構築)してデバイスに書き込み)釦を押してください。書き込み器として使われるツールを選ぶことを尋ねるポップアップが現れるかもしれません。AVR128 DA48 Curiosity Nano基板を選んでください。



AVR[®] DA 訓練の手引き

1.2. Atmel Studioを開いてPower Debuggerをコンピュータに接続してください。Curiosity Nano基板をPower DebuggerのAチャネル電流 計に接続してください。Nano組み込みデバッカーに最も近いピンをA電流計の内向き矢印に、マイクロコンピュータに近いピンを電流 計の外側を指す矢印に接続してください。



- 1.3. Atmel StudioでTools(ツール)⇒Data Visualizer(データ可視器)へ行ってください。
- 1.4. 2つの枠が現れます。DGI (Data Gateway Interface) Control Panel(DGI(データ中継器インターフェース)制御盤)と呼ばれる枠をチェッ クしてください。これが現れない場合、左側のConfiguration(構成設定)メニューをアクセスすることができ、Modules(単位部)⇒Exter nal Connection(外部接続)部分でデータ中継器インターフェースを見つけることができます。

図:	2-26. データ中継器インタ	ターフェース
Data	∕isualizer ⇔ ×	
\odot	Configuration	DGi Control Panel
Configurat	Modules 2 External Connection Data Gateway Interface (DGI)	Curiosity Data Gateway Interface
iĝ	Senal Port Visualization	ADP Logging ♥ Autodetect protocols
	 Utilities Protocols 	Interfaces:
		Serial Port Control Panel 👻 🗙

1.5. Connect(接続)を押すと'Power(電力)'と名付けられた枠が現れます。メニューを開くためにそれを選んでその後に歯車を押してください。Bチャネルを禁止してください。

집2-27. DGI接続 GControl Panel Power Debugger Data Gateway		Connect
nterfaces:		ADP Logging Autodetect protocols
図2−28. 電力設定		
Power Debugger Data Gateway Soccorrosse	Power Configuration X	Disconnect
ntefaces: SPI 🔅 USART C TWI C GPIO C A Current C Code Profiling C Code Profiling C Code Profiling C Code Profiling C Code Location C A Voltage C C C C C C C C C C C C C C C C C C C	Trigger calibration Enable Range Source Lock ChA to High Range Enable Voltage Output Voltage Output Averaging No OK Cancel	ADP Logging Autodetect protocols 🧟 Reset MC

告 Power(電力)領域が現れない場合、基板を切断して再び接続してください。

1.6. 全てが正しく構成設定された時にStart(開始)を押すと、実時間で消費電力を表示する新しい枠が現れます。

そこ 提供します。	
図2−29. 監視開始	
DGI Control Panel	^ X
Power Debugger Data Gateway	Disconnect
ADP Logging Autod	etect protocols 🖌 Reset MCU
Interfaces: SPI Code Profiling \$\$ Code Profiling \$\$ Code Location \$\$	
1 🌑 A Voltage 🌑	
Power Analysis	^ × 1
	Control Panel Control Panel Code Profiling
G 6210µ - G 6210µ - 6150µ - 6150µ - 6120µ -	Cursors Auto-scroll Automatically fit Y Show zero
14139 14140 14141 14142 14143 14144 14145 14146 14147 14148	
Ch A Window Average Instant	
Scot the mouse-wheel while presing and holding the left shift key to zoom in the time axis.	

☆ 助言:表示を調整するには枠の左側で制御盤を開いてください。Show Zero(0表示)のチェック解除がより鮮明な画像を

1.7. 読み取りが安定した後、Stop(停止)を押してその後に切断してください。電力測定の枠は開いたままです。

- 2. ハードウェア累積実行プログラムからの消費電力読み取りを取得してください。
 - 2.1. Curiosity Nano基板上の2つの半田付けしたピンを共に接続してください。
 - 2.2. ハート・ウェア累積プロジェクトを主プログラムとして設定し、MPLAB Xのツール ハーでMake and Program Device(作成(構築)してデバイスに書き込み)釦を押してください。



- 2.3. Curiosity Nano基板の2つの半田付けしたピンをPower Debuggerに接続してください。1.2.段階をご覧ください。
- 2.4. Atmel Studioでデータ可視器(Data Visualizer)へ戻り、DGI盤でConnect(接続)を押してください。

図2-31. データ中継器インターフェース	
DGI Control Panel	^ X
Power Debugger Data Gateway s020000566	Connect Start
	△ ADP Logging → Autodetect protocols ✔ Reset MCU
Interfaces:	

2.5. Power(電力)枠をチェックしてBチャネルを禁止するように設定を編集してください。

図2-32. 電力設定	
DGI Control Panel	× ×
Power Debugger Data Gateway	Power Configuration X
	Enable B Channel Start
	Trigger calibration
Interfaces:	Enable Range Source
□ SPI 🏠 □ USART 🏠 □ TWI 🏠 □ GPIO 🏠 🗹 Power 🏠 □ Code Profiling 🏠	Lock ChA to High Range
Code Location	Enable Voltage Output
1 🌓 💿 A Voltage 🌍	Voltage Output
2 💕 B Current 💕	Averaging No 👻
3 🖝 💿 🛛 B Voltage 🖝	OK Cancel

2.6. Satrt(開始)を押すと、実時間消費電力を表示する新しい枠が現れます。より鮮明な画像のためShow Zero(0表示)を禁止して ください。

図2−33. 監視開始		
DGI Control Panel		× ^
Power Debugger Data Gateway		Disconnect Start
		AUPLogging Autoatect protocols W Reset MCU
	GPIO X A Current C A Voltage C CC	Code Location
Power Analysis Power Analysis		۲× ۲۸
3800µА - 3700µА - 3600µА - 3500µА - 3500µА - 3400µА -	l ning an an tao interlitet par a paralama di t	
6 3300µА		
14319 14320 Ch A Instant Window Average Instant Image: State Average Image: State Average Image: State Average	14321 14322 the left shift key to zoom in the time axis.	322 14323 14324 14325 14326 14327 14328

- 2.7. 読み取りが安定した後、固定した結果を持つようにStop(停止)を押してください。
- 3.2つの測定を比較してください。
 - 3.1. 両方の消費電力図が同じ画面で開いています。 今やプログラムの挙動の様子を観測するためにこれらを分析することができます。



情報:上部はソフトウェア累積を表示、同時に下部はハードウェア累積を表示します。図から見ることができるように、2つの間の違いは3.1mAあり、ハードウェア実装を効率としてほぼ2倍にします。

ĭ

<mark>3</mark>. 改訂履歴

資料改訂	日付	注釈
А	2020年8月	初版文書公開

Microchipウェフ゛サイト

Microchipは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www.microchip.com/supportでのウェブサイト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、その製品系統が安全であると考えます。
- ・Microchipデバイスのコート、保護機能を破ろうとする試みに使われる不正でおそらく違法な方法があります。当社はこれらの方法が 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とされると確信しています。これらのコート、保 護機能を破ろうとする試みは、おそらく、Microchipの知的財産権に違反することなく達成することはできません。
- ・Microchipはそれのコードの完全性について心配されている何れのお客様とも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。コート保護は製品が"破ることができない" ことを当社が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

この刊行物に含まれる情報はMicrochip製品を使って設計する唯一の目的のために提供されます。デバイス応用などに関する情報は 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。

この情報はMicrochipによって「現状そのまま」で提供されます。Microchipは非侵害、商品性、特定目的に対する適合性の何れの黙 示的保証やその条件、品質、性能に関する保証を含め、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。

如何なる場合においても、Microchipは情報またはその使用に関連するあらゆる種類の間接的、特別的、懲罰的、偶発的または結果的な損失、損害、費用または経費に対して責任を負わないものとします。法律で認められている最大限の範囲で、情報またはその使用に関連する全ての請求に対するMicrochipの全責任は、もしあれば、情報のためにMicrochipへ直接支払った料金を超えないものとします。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、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ロゴ、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www.microchip.com/qualityを訪ねてください。

日本語© HERO 2020.

本訓練の手引きはMicrochipの訓練の手引き(DS40002243A-2020年8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: www.microchip.com/support ウェブ アトレス: 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 Fax: 905-695-2078

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

イント - ハンガロール 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

亜細亜/太平洋

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: 45-4485-5910 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 – カルヒング 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