ANALOG DEVICES ADuCM3027/ADuCM3029 EEMBC ユーザー・ガイド

ADuCM3027/ADuCM3029の EEMBC スコアを再現する方法

はじめに

このリファレンス・マニュアルは、ADuCM3027/ADuCM3029 マ イクロコントローラの組み込みマイクロプロセッサ・ベンチマー ク・コンソーシアム (EEMBC) ULPBench[™] Core Profile スコアと CoreMark® スコアを再現する方法について説明します。

このリファレンス・マニュアルでは、両方のスコアを測定するためにソフトウェアをインストールして、ハードウェアをセットアップするうえで必要な手順について説明します。

このドキュメントを最後まで読むと、ADuCM3029 EZ-KIT[®] ボードの EEMBC ULPBench Core Profile スコアと CoreMark スコアを 再現できるようになります。

このリファレンス・マニュアルでは、ベンチマークで使用する各 電力モードで ADuCM3027/ADuCM3029 マイクロコントローラが 消費する電力について説明し、データシートに記載された値を確 認します。

ADUCM3027/ADUCM3029 について

ADuCM3027/ADuCM3029 プロセッサは、処理、制御、接続に使 用される超低消費電力の内蔵型ミックスド・シグナル・マイクロ コントローラ・システムです。マイクロコントローラ・ユニット (MCU) サブシステムは、ARM[®] Cortex™-M3 プロセッサをベー スにし、一連のデジタル周辺機器、キャッシュ組み込み SRAM、 フラッシュ・メモリ、アナログ・サブシステム、A/D コンバータ (ADC) だけではなく、クロック、リセット、パワー・マネージ メントの機能も備えています。

日本語参考資料

最新版英語ユーザー・ガイドはこちら

ADuCM3027/ADuCM3029 プロセッサは、ダイナミックなソフト ウェア制御によるクロック・ゲートや電力ゲートなどの電力モー ドと機能を備え、超低ダイナミック・パワー・マネージメントと 休止パワー・マネージメントをサポートします。

ADuCM3027/ADuCM3029のすべての仕様は、製品データシートに 記載されています。EZ-KITを使用する場合は、このリファレンス・ マニュアルとデータシートを組み合わせて参照してください。



図 1. EEMBC ULPBench スコア

アナログ・デバイセズ社は、提供する情報が正確で信頼できるものであることを期していますが、その情報の利用に関して、あるいは利用によって 生じる第三者の特許やその他の権利の侵害に関して一切の責任を負いません。また、アナログ・デバイセズ社の特許または特許の権利の使用を明示 的または暗示的に許諾するものでもありません。仕様は、予告なく変更される場合があります。本紙記載の商標および登録商標は、それぞれの所有 者の財産です。※日本語版資料は REVISION が古い場合があります。最新の内容については、英語版をご参照ください。

Rev. 0

©2016 Analog Devices, Inc. All rights reserved.

本 社/〒105-6891 東京都港区海岸 1-16-1 ニューピア竹芝サウスタワービル 電話 03(5402)8200 大阪営業所/〒532-0003 大阪府大阪市淀川区宮原 3-5-36 新大阪トラストタワー 電話 06(6350)6868

アナログ・デバイセズ株式会社

目次

はじめに1
ADuCM3027/ADuCM3029 について1
改訂履歴2
EEMBC について
CoreMark
ULPBench
IAR セットアップ4
IAR ツールのインストール
IAR プロジェクトの構成4
CoreMark

改訂履歴

3/2017-Revision 0: Initial Version

EEMBC について

EEMBC[®]は、新しいベンチマークを実現するために、組み込み 業界の大手サプライヤと共同で取り組む必要性に気付いた非営利 団体です。

EEMBCは、世界をリードする半導体、知的所有権、コンパイラ、 RTOS、システム企業 40 社以上で構成されています。さらに、 EEMBC Benchmark は世界中の 80 社を超える企業と100 を超える 大学から認可を受けています。メンバーによる努力が実り、EEMBC ベンチマークは、客観的かつ明白に定義され、アプリケーションを ベースにした基準に従い、組み込みプロセッサとシステムの機能を 評価する業界標準になりました。

EEMBC には、クラウドやビッグ・データ、モバイル・デバイス (電話やタブレット)のネットワーキング、超低消費電力マイク ロプロセッサ、モノのインターネット(IoT)、デジタル・メディ ア、自動車、その他のアプリケーション分野をターゲットとする 一連のベンチマークがあります。また、EEMBC には、CoreMark、 MultiBench™(マルチコア)、FPMark™(浮動小数点)など、性 能分析向けの汎用ベンチマークもあります

このリファレンス・マニュアルでは、電力処理やMCUの電力効率の測定を目的とした CoreMark と超低消費電力マイクロコントローラのベンチマークに重点を置いて説明します。これらの項目は、ADuCM3027/ADuCM3029 プロセッサの主な機能であるためです。

COREMARK

特定のアプリケーションに合わせて MCU を選択するには、MCU の処理能力が要件に適合するか確認する必要があります。複数の ベンチマーク・オプションを使用できます。Dhrystone が最も一 般的に使用されるベンチマーク・オプションです。ただし、一部 のライブラリ・コールに時間制限があることや、動作を最適化す る際にコンパイラの機能による影響を受けやすいなど、いくつか の問題があります。これらの問題に対処してオープン・ソースの 簡素なベンチマークを提供するため、EEMBC は CoreMark を作 成しました。 CoreMark は、組み込みシステムで使用する CPU(中央処理装置) の性能を測定することを目的としたベンチマークです。2009 年 に EEMBC で開発されたこのベンチマークは、時代遅れになった Dhrystone ベンチマークに代わる業界標準になることを目指して います。C 言語で記述されたコードには、次のアルゴリズムが実 装されています。

- リスト処理(検索と並べ替え)
- マトリックス操作(共通のマトリックス操作)
- ステート・マシン(入力ストリームに有効な数値が含まれているかどうかを判断する)
- CRC

ULPBENCH

ターゲットが IoT のエッジ・ノードであるか、バッテリで駆動する アプリケーションであるかどうかで、超低消費電力(ULP)の意味 が変わります。電源が極端に制限されている場合(エナジー・ハー ベストなど)、有効電流を最小限に抑える必要があります。ほとん どの時間、システムがスタンバイ・モードまたはスリープ・モード の状態にあります。まれに(定期的にまたは非同期で)ウェーク アップしてタスクを処理する場合は、スリープ電流を最小限に抑 える必要があります。また、ULP は電力効率も高いので、限ら れた時間内にほとんどの作業を実行できます。全体的に、アプリ ケーションでは、前に説明したすべての条件のトレードオフの組 み合わせが必要になります。月、年、10年単位で ULP 操作を行 うため、アプリケーション開発者は多数の課題に直面します。 ULP機能を搭載したマイクロコントローラは多数存在しますが、 開発者はデータシートのパラメータのみに頼ることはできません。 EEMBC ULPBench は、データシートのパラメータを標準とし、 信頼性が高く、公正な MCU 電力効率の測定方法を提供します。

ULPBench の基礎は、以下のとおりです。

- 互換性 デバイスの比較が簡単になります。
- 透過性-すべての測定とセットアップのプロセスがわかりや すくなります。
- 再現性-誰でもベンチマーク・スコアを再現できるようになります。

ユーザー・ガイド

IAR セットアップ IAR ツールのインストール

IAR Embedded Workbench と付属の IAR C/C++ コンパイラを使用 すれば、ARM ベースのアプリケーション向けに高速で実行でき る、業界で最もコンパクトなコードを生成できます。そのため、 アナログ・デバイセズは、IAR Workbench 向けに ADuCM3027/ ADuCM3029 用のボード・サポート・パッケージ(BSP)を提供 しています。

ADuCM3027/ADuCM3029のサポートは、BSP内で提供されます。

KickStart エディションは、無料の IAR スターター・キットの評価版です。このエディションには、コード・サイズ (32 KB)の他に、提供されるサービスとサポートに制限があります。

IAR Embedded Workbench は、IAR の Web サイトからダウンロードできます。

IAR プロジェクトの構成

ここでは、適切に動作させるための IAR 構成について説明しま す。デフォルト値から変更する必要がある部分のみ説明します。

 プロジェクト名を右クリックし、 [Options…] を選択しま す(図2参照)。



図 2. プロジェクト・オプション

[General Options] で、ご使用のマイクロコントローラに合わせて[AnalogDevices ADUCM3027]または[AnalogDevices ADUCM3029]がターゲットとして選択されていることを確認します。

ategory:						
eneral Options Ratic Analysis	Â					
C/C++ Compiler Assembler		Processor va	Library Configuration	Library Options	MISRA-C:2004 M	ISF * *
Custom Build Build Actions		Core	Cortex-M3	*		
Linker Debugger	I	Device	AnalogDevices AD	UCM3029	D	
Angel CMSIS DAP	11	Endian mode		ΨÜ		
GDB Server		@ Little		None	*	
I-jet/JTAGjet J-Link/J-Trace TI Stellaris		BIG BE32 BE8				
Macraigor PE micro RDI						
ST-LINK Third-Party Driver				_		

図 3. [General Options] - [Target] の設定

3. [C/C++ Compiler]で、高速処理のために最適化が選択され ていることを確認します(図4参照)。[pltInitialize]など の一部の機能は、最適化されないように保護されています。 次のコードは関数を保護し、コンパイラが関数コードを変 更しないようにします。

#pragma optimize=none

図 5 に、ULPBench Core Profile プロジェクトを適切にコンパ イルするために必要な組み込みディレクトリ・パスと[define] 設定を示します。

Category:		Not fin Compilation			Factory Setti	ngs
Seneral Options Ratic Analysis	^	Discard Unused Publics				
C/C++ Compler		Language 1 Language 2 Code	Optimizations	Output List	Preproce	
Austmoor Output Converter Custom Build Build Actions Linker Debugger Simulator Angel Chills DAP GOE Server LAR ROM-sonitor L-jet/JTAGjet Juki/J-Trace TI Stellaris Macraigor PE micro RDI ST-LINK		Cove None Low Medium Uigh Speed Ng size constraints	✓ Common su ✓ Loop unrolli ✓ Function inli ✓ Code motio ✓ Type-basei ✓ Stefic cluste ✓ Instruction s ✓ Vectorizatio	nnaouns. bexpression el ning n d alias enolysis ring cheduling n	imination	

図 4. [C/C++ Compiler] - [Optimizations] の設定

UG-944

4302-005



-ザー・ガイド

図 5. [C/C++ Compiler] - [Preprocessor] ULPBenchの設定

図 6 に、CoreMark プロジェクトの適切なコンパイルに必要 なディレクトリ・パスと [define] の設定を示します。

Category: General Options Ratic Analysis	Multi-file Compilatio	n ed Publics			Factory Set	ttings		
Runtime Checking	Language 2 Code Detinizations Dutred List Preprocessor							
Assembler Output Converter Output Converter Output Converter Debugger Simulator Angel CMSIS DAP GDB Server IDAR ROM-monitor I-set/JTAGjet J-link/J-Trace TJ Stefaris Macraigor PE micro RDI ST-LINK Third-Party Driver	Ignore standard Additional include d SPR0J_DIRS.Vin S	include directories: (one p c c-CloreMark clusse clussed inclAnalogDevic ne per line)	es	proces Preserv Genera	sor output to file re comments te #line directive			

図 6. [C/C++ Compiler] - [Preprocessor] CoreMark の設定

不要な警告を避けるため、 [Suppress following diagnostics] オプションに次の診断を追加します: Pa050 と Pa082 [Signature] フィールドに32ビットの巡回冗長検査(CRC) チェックサムが保存されているので、ユーザー・コードに よるユーザー・スペースの整合性チェックの要求が可能に なります。そのため、図7と図8に示すようにチェックサ ムを構成する必要があります(どちらの設定も[Linker]に 含まれます)。

Category:							Factory Settings
Seneral Options	*						
Static Analysis Runtime Checking		Advanced	Output List	#define	Diagnostics	Checksum	Extra Options
C/C++ Compiler Assembler		Fill unu	sed code me	mory			
Custom Build Build Actions		Start	address:	0x0	End	address:	0x7fb
Linker.		🔽 Ge	nerate check	sum			
Debugger		c	hecksum size	4 bytes	▼ Alig	nment	4
Angel		A	Igorithm:	CRC32	•	0×11021	
CMSIS DAP		225	Result i	n full size		Initial value	
GDB Server IAR ROM-monitor		c	omplement	As is	•	0×######	
I-jet/JTAGjet		в	it order:	MSB first	-	Use as	input
3-Link/3-Trace TI Stellaris Macraigor PE micro		C	Reverse by hecksum unit	te order with size:	in word 32-bit	•	
RDI							

図 7. [Linker] - [Checksum]の設定

Category		Facto	ry Settings
General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custon Build Build Actions	•	Output List #define Diagnostics Checksum Extra Options Use command line options Command line options: (one par line) -keep _checksum	
Unite Debugger Simulator Angel ChSIS DAP GDE Server LAR ROM-monitor I JAR ROM-monitor I Stellaris Macraigor PE misrico ROI			
ST-LINK Third-Party Driver TI XD5		OK	Cancel

14302-106

-ザー・ガイド

UG-944

使用されるデバッガは [J-Link/J-Trace] です(図9参照)。 5. [Debugger] > [Download] メニューで [Verify download] と [Use flash loader] の両方をチェックする必要があります (図10参照)。

ategory:			Factory Settings
General Options Static Analysis Runtime Checking C/C++ Compiler	*	Setup Download Images Extra Options Multicore Plugins	
Assembler		Driver Run to	
Output Converter		J-Link/J-Trace main	
Build Actions		Setup macros	
Linker		Lise merco file(s)	
Debugger		Cose motio me(s)	
Simulator			
Angel	Ξ		
CMSIS DAP GDB Server IAR ROM-monitor 1-jet/JTAGjet		Device description file	
3-Link/3-Trace TI Stellaris Macraigor		TOOLKIT_DIR\$\CONFIG\debugger\AnalogDevices\joADUC	M302
PE MICTO BOJ			
ST-LINK			
Third-Party Driver TI XDS	-	ОК	Cancel

図 9. [Debugger] - [Setup] の設定



図 10. [Debugger] - [Download] の設定

図 11 および図 12 に、[J-Link/J-Trace]の設定を示します。 6. [Halt after bootloader] ターゲット・リセットの方策を使用 してください。そうしないと、カーネルが破損し、デバイス がロックダウンします。デバイスのロックダウンから回復す るには、カーネルを再度フラッシュする必要があります。

Category:		Factory Settings
General Options Static Analysis Runtime Checking C/C++ Compiler Assembler	Setup Connection Breakpoints Reset	
Output Converter Custom Build Build Actions Linker Debugger Simulator Angel E CMSIS DAP GDB Server	JTAQ/SWD speed Clock Auto Initial 1000 kHz CPU Fixed 1000 kHz SWD Adoptive	setup clock: MHz 0 clock: Auto 2000 kHz
LAR ROM-monitor 1-jet/JTAGjet SLEHJ-Sfrace TI Stellaris Macraigor PE micro RDI ST-LINK Takid Path Datase	ETM/ETB	

図 11. [J-Link/J-Trace] - [Setup]の設定

Category:					Factory Settings
General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Artions	•	Setup Connection Communication © USB: © TCP/IP:	Breakpoints Device 0 IP address	Serial no:	
Linker Debugger Sinulator Angel CMSIS DAP GDB Server JAR ROM-monitor	CHIL:	IP address: Interface © JTAG © SWD	asa bbb.ccc.ddd JTAG scan chain I JTAG scan chain TAP number Scan chain Preceed	Serial no.	ices
1-yet 31 Maget 3-1/14/1-17/sce TI Stellaris Macraigor PE micro ROI ST-1 Itak		Log communic SPR0J_DIR\$	ation cspycomm.log		
Third-Party Driver	-			OK	Cancel

図 12. [J-Link/J-Trace] - [Connection] の設定

COREMARK COREMARK プロジェクトのロード

CoreMark ソースファイル、core_portme.c ファイル、core_portme.h ファイルを含むプロジェクトはアナログ・デバイセズのプラット フォームに合わせてチューニングされます。次の手順では、IAR で CoreMark プロジェクトを追加する方法を説明します。

- 1. IAR ワークベンチを開きます。
- 2. IAR でプロジェクトを開きます。
- 3. 図 13 に示すように、[Project] で、[Add Existing Project…] を クリックします。

File Edit View	Project Tools Window Help	
🗅 🗃 🖬 🕼	Add Files	
Workspace	Add Group	
	Import File List	
Files	Add Project Connection	
	Edit Configurations	
	Remove	
-	Create New Project	
	Add Existing Project	

図 13.既存のプロジェクトの追加

 取得したプロジェクトを参照し、.ewp 拡張子のファイルを 開きます。ワークスペースで使用できるファイルを図 14 に 示します。

<u>F</u> ile	<u>E</u> di	t <u>V</u> iew	<u>Proj</u>	ject	Sim
	ê 🕻	0 é	5 %		B
Works	pace				
Dek	oug				
File	es				
	AD	LCM302	x_Co	reMa	ırk
HG	9 🗀 B	SP sour	ces		
	— 🕀 🖸	adi_pv	/r.c		
	-Ð 🖸] adi_ua	irt.c		
	- ± C] startup	.C		
1	-] system	n.c		
15		oreMark	sourc	es	
	- E C] core_li	st_join	.C	
	- E C] core_n	nain.c		
] core_n	natrix.c		
		core_s	tate.c		
] core_u	til.C	2	
I Fi		lattorm s	source	5	
		p	orme.	С	
		utput			
义	14.7	゚゚ロジェ	クト・	ファ	イル

[BSP sources]フォルダには、デバイスを適切に構成するためのBSP ファイルがあります。[CoreMark sources]フォルダには、EEMBC で指定されたソース・ファイルがあります。 [Platform sources] フォルダには、EEMBC で指定された core_portme.c ファイルがあ りますが、テスト済みデバイス(ADuCM3027/ADuCM3029 プロセッ サ)を適切に構成できるようにチューニングされています。

EEMBC では、アナログ・デバイセズのプラットフォームに合わ せて core_portme.c ファイルと core_portme.h ファイルの変更に対 する制限はありません。core_portme.c ファイルと EEMBC によっ て指定されたファイルの相違を以下に示します。

- UART 出力のコード
- オシレータや水晶発振器を使用して実行する刻みを計算す るためのコード
- マイクロコントローラを適切に構成するためのコード
- これらのファイルをサポートするヘッダー・ファイル
- デバイスの構成消費電力を削減するため、降圧レギュレー タがイネーブルになっています。CoreMarkの実行中に電力を 測定する際に便利です(電力測定セクションを参照)。

Core_portme.h ファイルには、2 つの定義があります。

- UART_PRINT 定義は、UART を介して結果を出力するため に使用されます。これをコメント処理すると、結果は端子 の入出力でのみ出力されます。
- XTAL 定義は、外部の水晶発振器と内部 RC オシレータのどちらを使用して、刻みを測定するかどうかを決定するために使用されます。これをコメント処理すると、内部オシレータが使用され、そうでない場合は水晶発振器が使用されます。

アプリケーションの実行

アプリケーションのビルド

アプリケーションのコードをビルドするには、次の手順のいずれ かに従います。

- [Project] をクリックしてから、 [Rebuild All] をクリック します(図 15 参照)。
- プロジェクト名を右クリックし、[Rebuild All]をクリックします(図16参照)。.eww 拡張子を使用してワークスペースを保存するように求められます。プロジェクトは、エラーなしでビルドする必要があります。



図 15. プロジェクトのビルド

-ザー・ガイド

UG-944



図 16. プロジェクトのビルド

コードのダウンロード

EZ-KIT ボードにコードをロードするには、以下に示すいずれかの手順を実行します。

- [Project] で [Download] をクリックし、 [Download active application] をクリックします (図 17 参照)。
 - **[Download and Debug**]をクリックします(図 18 参照)。



図 17. コードのダウンロード

プロジェクトの実行

ULPBench_Phase1 - IAR Embedded Workbench IDE

コードを実行するには、[Go]をクリックします(図19参照)。

Eile Edit	¥iew	Project	Debug	Disassembly	Simulator	Iools	Windo	W H	lelp						
0 📽 🖬	3 8	XRA	\$ 10 C	-		+ 4	+ 11	3 🖬		Þ	-8.0+	03 9	20 12	۹	1
5-14	122	88	33 ×	1											
Workspace				webstein a latest	up.c. Platform	ic									

図 19. プロジェクトの実行

COREMARKの結果

CoreMark コードは10秒以上実行する必要があります。コードの 設定は10,000回の反復を経て、2分程度で完了します。 結果は端子の入出力から出力されます。端子の入出力を表示する には、[View]をクリックし、[Terminal I/O]をクリックしま す。デバッグ・モードでは、このオプションをアクティブにする 必要があります。

<u>File</u> <u>E</u> dit	⊻iew	<u>P</u> roject	<u>D</u> ebug	Disase
🗅 🗳 🖬	N	lessages		•
5-	v	∕orkspace	9	
Workspace	S	ource Bro	wser	•
Debug	С	-STAT		•
Files	В	reakpoin	ts	
	D	isassemb	olv.	
He C sta	N	lemory	~	
H-⊞ C sys	S	vmbolic	Memory	
- <u>-</u>	R	eaister	,	
	v	/atch		
	L	ocals		
	S	tatics		
	A	uto		
	Li	ive Watch	ı	
	ç	uick Wat	ch	
	M	Iacros		
	С	all Stack		
	S	tack		•
	Т	erminal I,	/0	

図 20. [Terminal I/O] の表示

ube:	Light Of
T performance run persentes for community. Sententis Ser 1991 Unit Link - 1991 District - 1995 District - 1995	beemadiacrit

図 21. 端子の結果

デフォルトでは、core_portme.h ファイルの UART_PRINT 定義が コメント処理されています。UART を介して結果を出力するには、 UART PRINT 定義のコメントを削除します。

以下の手順では、UART を介して結果を出力する方法について説明します。

- 1. UART_PRINT 定義のコメントを削除します。
- USB ケーブルを使用して、EZ-KIT の UART ポートを PC に 接続します。



図 22. USB - UART 変換による接続

4302-

ユーザー・ガイド

- 3. [Control Panel] パネルから、 [Device Manager] をクリッ クします。
- 4. UART が接続されている COM ポート番号を確認します。
- UART ポートに接続できる端子を開きます(ここでは PuTTYを使用)。
- 6. [Connection]タイプを[Serial]に設定し、対応する COM ポー ト番号を入力します。
- 7. その他の設定を図23に示します。



図 23. UART の設定

プロジェクトの実行 セクションの指示に従い、プロジェクトをリ ビルドします。結果はUARTを介して出力されます(図24参照)。

CoreMarkの数値は処理速度を示し、CoreMark/MHzの数値はコアの効率を示します。CoreMark/MHzの値を計算するには、ベンチマークの実行中にCoreMarkの値をクロック速度で除算する必要があります。



このプロジェクトでは、ADuCM3027/ADuCM3029 プロセッサを 26 MHz で実行します。

CoreMark/ MHz = $\frac{85.337589}{26 \text{ MHz}}$

CoreMark/MHz スコアは 3.2822 です。

このスコアは、ARM Cortex-M3 プロセッサの CoreMark/MHz スコア (3.32) とほぼ同じです。

CoreMark では、次の形式でスコアを報告するよう推奨しています。

CoreMark/MHz 1.0:3.2822 / IAR EWARM

7.50.2.10505 --no_size_constraints

--cpu=Cortex-M3 -D __ADUCM3029__

--no_code_motion -Ohs -e --fpu=None

--endian=little/FLASH

電力測定

以下の手順では、CoreMark コードを実行した場合の ADuCM3027/ADuCM3029 プロセッサの消費電流を監視する方法 について説明します。

- 1. UART ピンをフロートさせないように、core_portme.hファイ ルのUART_PRINT 定義がコメント処理されていることを確 認してください。
- 2. マイクロコントローラにコードをロードします。
- 3. ソースの正端子を JP6 コネクタに接続します。
- 4. ボードの GND をソースの GND に接続します。
- 5. すべてのジャンパを取りはずします。
- 6. [Reset] ボタンを押します。
- メーターの消費電流を監視します。プロセッサが26 MHz で コードを実行する場合、消費電流を約1165 µA にする必要が あります。
- 8. ダイナミック消費電流を測定する場合は、異なる周波数でこの手順を繰り返します。周波数を4で除算して、26 MHz/4= 6.5 MHz の値になるように、CLKDIV 定義の変数を4 に変更 します。
- 9. メーターの消費電流を監視します。消費電流は、約425 μAに なる必要があります。

ダイナミック消費電流の値を取得するには、2点(周波数と電流) で構成される直線の傾きを計算します。

傾きは、以下の方法で計算されます。

$$Sope = \frac{1165 - 425}{26 - 6.5} = 38 \,\mu\text{A/MHz}$$

ダイナミック電流消費は38 µA/MHzです。





ULPBENCH CORE PROFILE ENERGYMONITOR

EEMBC ULPBench EnergyMonitor™ ソフトウェアは、正確な電力 測定ツールです。ULPBench スコアを測定するには、EEMBC EnergyMonitor ハードウェア(図 25 に表示)が必要です。このハー ドウェアは、EEMBC の Web サイトから購入できます。

図 25 に、EEMBC EnergyMonitor ハードウェア、および ADuCM3029 EZ-KIT ボードへの電力供給に使用する VCC と GND ピンを示し ます。



図 25. EEMBC EnergyMonitor ハードウェア

EnergyMonitor ソフトウェア・ドライバのインストール

EnergyMonitor ハードウェアを初めて PC に接続すると、USB デ バイスを認識できない旨を知らせるメッセージが表示されます。 この USB ドライバ・メッセージが表示されたら、 [Next] をク リックして、 [Manually locate USB drivers] をクリックします。

ドライバ・メッセージが表示されない場合は、[Device Manager] に移動し、[EEMBC Application UART1]と[EEMBC Energy Tool V1]という名前のデバイスを探し、両方のドライバをインストー ルします。

USB ドライバをインストールします。それぞれ /bin/USB_CDC/ monitor_driver.inf と /bin/USB_CDC/monitor_driver.cat にありま す。発行者を検証できないことを示すセキュリティ警告が表示さ れます。[Install this driver software anyway] をクリックします。

デフォルトでは、64 ビットの Windows® Vista 以降の Windows に おいて、カーネルがドライバ署名を検証できる場合のみドライバ がロードされます。上記の Windows バージョンのいずれかを使 用してドライバをインストールできない場合、適切な設定を使用 して、有効なドライバ署名の強制を一時的に無効にします(適切 な設定は Windows のバージョンによって異なります)。

アプリケーション・コードのビルド アプリケーションのビルド

アプリケーション・コードをビルドまたはコンパイルするには、 [Project] をクリックして [Rebuild All] をクリックします。

Eile Edit View Pr	oject <u>I</u> ools <u>W</u> indow <u>H</u> elp	
D 🍃 🖬 🕼 🏼	Add Files	
Release	Make	F7
Files	Compile	CTRL+F7
	Rebuild All	
- Derichman	Clean	
He lcd.c	Batch build	F8

図 26. プロジェクトのビルド

コードをダウンロードするためのボードのセットアップ

USB 経由でコードをダウンロードするには、両方の JP6 ジャン パを取り付ける必要があります。

EEMBC ULPBench_Phasel プロジェクトがすでにダウンロードされている場合、デバイスはほとんどの期間にわたり休止モードになります。ディープ・スリープ・モード(休止モードとシャッド ダウン・モードの両方)では、シリアル・ワイヤがディスエーブ ルになり、コードをダウンロードできません。

EEMBC ULPBench_Phasel プロジェクトでは、以下のコードを追加してアプリケーションをフラッシュすると、コードをダウンロードできます。

while((pADI GPIO0->IN & (1 << 4)) == (0 << 4));

起動時は、P0.4 が入力として構成されます。コードは while ルー プの内部で待機するため、コードをダウンロードするには、P0.4 ピンをクリアする必要があります。P0.4 ピンをクリアするには、 図 図 27 に示すように、P0.4 (SCL と基板上に表記されている Arduino の J6 ヘッダの 10 番ピン) とグラウンド (GND と基板上 に印刷されている Arduino の J7 ヘッダの7番ピン)を接続します。



図 27. コードをダウンロードするためのボードの設定

コードのダウンロード

コードをダウンロードするには、 [Project] > [Download] > [Download active application] をクリックします。

File Edit View P	Project Tools Window Help		
D 📽 🖬 🕼 🖉	Add Files	I II P P P A A A B II II II A A A	
Norkspace	elease		m.c
Release			1-1
Files	Download	•	Download active application
CorePn CoreP	SFR Setup		Download file
	Open Device Description File		Linkermennery
	Save List of Registers		

図 28. アプリケーションのダウンロード

デバイスの電源を入れなおした後に、ADuCM3027/ADuCM3029デバイスでコードを実行します。

ベンチマークの実行

ULPBench Core Profile の実行

ここでは、ULPBench Core Profile スコアを測定するために ADuCM3029 EZ-KIT ボードを設定する方法について、順を追っ て説明します。

- P0.4 と GND の間に配置したケーブル(前のセクションで説 明)を取りはずします。
- 前に取りはずした P0.4 と VBAT の間のケーブルを配置します(3.3 V と基板上に表記されている Arduino の J7 ヘッダの4番ピン)。この構成では、コードが while 命令で停止しません(コードをダウンロードするためのボードのセットアップセクションを参照)。
- 3. USB ケーブルを取りはずします。
- 4. JTAG を取りはずします。

- 5. ピン1とピン3の間にある JP6 ジャンパを取りはずします。 これは電力測定ジャンパです。
 - ピン1を介してボードの外部部品(LED、スイッチ、 センサー、加速度センサー)に電力を供給します。
 - ピン3を介してマイクロコントローラに電力を供給します。



- EnergyMonitor ハードウェアの VBAT ケーブルを JP6 のピン3 に接続します。
- EnergyMonitor ハードウェアのGND ケーブルをJ6のピン7に 接続します。

図30に示すような接続が確立されます。

EnergyMonitor ソフトウェアを起動し、 [Run ULPBench] をク リックして、スコアの測定に移行します。EnergyMonitor ハード ウェアは、ADuCM3029 EZ-KIT ボードに電力を供給し、コア・プ ロファイルの消費電力を測定します。実行の最後に、ソフトウェ アは EEMBC ULPBench Core Profile スコアを計算し、画面に結果を 表示します。履歴ウィンドウには、前のサイクルの平均消費電力 も表示されます。



図 30. スコアを測定するためのボード設定

通常のデバイスで取得されるスコアは、約250 EEMarks[™]-CPです。この値は、プロセスや温度条件によって変動します。図31に、 通常デバイスのスコア例を示します。



図 31. ULPBench Core Profile スコア

正しい動作の検証

ワークロードが正しく実行されていることを確認するため、ステー タス・ピンが定義されています。P0.2(12と基板上に表記されて いる J6 ヘッダの5番ピンでアクセス可能)は、公認コードでス テータス・ピンとして構成されます(図30で示す紫のケーブル)。

ベンチマークによると、ワークロードは2回実行され、最初にス テータス・ピンが 20回切り替わります。2番目のワークロード が完了したら、エラーが通知されない限り、ステータス・ピンは クリアされます。

図 32 に、前に説明したワークロード実行中の P0.2 の出力を示します。



図 32. 正しい動作の検証

ULP Crystalless Profile の実行

Core Profile は、ULPBench グループで公認されている唯一のスコ アです。ただし、水晶発振器スコアの取り決めは、現在 ULPBench ワーキング・グループ内で協議中です。

水晶発振器が提供するような高い精度が必要でない場合は、低周 波数発振器(LFOSC)をリアル・タイム・クロックのソース・ クロックとして使用し、消費電力を節約できます。LFOSC と LFXTALは、どちらも周波数が32 kHzです。 配信されるコードには、ULPBench Crystalless Profile をテストで きるように、define ディレクティブ(Platform.c ファイルの51行 目)が含まれます。LFOSC を RTC クロックとして使用するには、 以下に示す define USE_LFOSC 行をコメント処理します。

#define USE_LFOSC

通常のデバイスで取得されるスコアは、約265 EEMarks[™]-CPで す。この値は、プロセスや温度条件によって変動します。図33に、 通常デバイスのスコア例を示します。



図 33. ULP Crystalless Profile スコア

結果の分析

ULPMark-CP は電力値の逆数になります(ULPBench 10 サイクル にわたる1秒あたりの平均電力の5倍の中央値)。

$$Energy(\mu J) = \frac{1000}{EEMarkCP}$$

消費電力は、デバイスが (アクティブ・モードで) ワークロードを 実行している間とデバイスが休止している間の合計として計算さ れます。

電力量=有効電力量+スリープ電力量

ADuCM3027/ADuCM3029 データシートに従い、LFXTAL と RTC をイネーブルにした場合、有効電流の代表値は38 μA/MHz に なり、休止電流の代表値は830 nA になります。図 32 に、アクティ ブ期間が420 μs であることを示します。

電力量 = 電圧 × 電流 × 時間

有効電力量 = 3 V × 1188 µA × 0.42 ms = 1.49 µJ

スリープ電力量 = 3 V × 830 nA × 999.58 ms = 2.49 µJ

データシートに記載された数値と実行時間によると、アクティブ 電流の電力量は1.49 µJ、休止時間中に消費された電力は2.49 µJ に なります。これらの値を基にしたスコアは、EEMBC EnergyMonitor ソフトウェアで測定されたものです。

Energy(
$$\mu$$
J) = 1.49 + 2.49 = 3.98 μ J $\cong \frac{1000}{252.1}$ = 3.96 μ J

ESD に関する注意

ESD(静電放電)の影響を受けやすいデバイスです。電荷を帯びたデバイスや回路ボードは、検知されないまま放電することがあります。本製品は当社独自の特許技術である ESD保護回路を内蔵してはいますが、デバイスが高エネルギーの静電放電を被った場合、損傷を生じる可能性があります。したがって、性能劣化や機能低下を防止するため、ESDに対する適切な予防措置を講じることをお勧めします。

法的条項

アナログ・デバイセズの標準販売条項が適用される評価用ボードの購入の場合を除き、ここで説明する評価用ボード(すべてのツール、部品ドキュメント、サポート資料、また評価 用ボードも含む)を使用することにより、以下に定める条項(本契約)にお客様は同意するものとします。本契約に同意した方のみ、評価用ボードを使用することができます。お客様 が評価用ボードを使用した場合は、本契約に同意したと見なします。本契約は、"お客様"と One Technology Way, Norwood, MA 02062, USA に本社を置く Analog Devices, Inc. (以降 ADI と記載)との間で締結されるものです。本契約集項に従い、ADI は、無償、限定的、一身専属、一時的、非独占的、サブライセンス不能、譲渡不能な評価用ボードを、評価目的での み使用するライセンスをお客様に許諾します。お客様は、評価用ボードが上記目的に限定して提供されたこと、さらに他の目的に評価用ボードを使用しないことを理解し、同意す るものです。さらに、許諾されるライセンスには次の追加制限事項が適用されるものとします。(i) 評価用ボードを賃借、賃貸、展示、販売、移転、譲渡、サプライセンス、 または 頒布しないものとします。(ii) 評価用ボードへのアクセスを第三者に許可しないものとします。ここで言う "第三者"には、ADI、お客様、その従業員、関連会社、および社内コン がいかっていていた。1911年19月1日には、1911年19月1日には1915年にあったのではありません。評価用ボードの所有権などの、本契約にて明示的に許諾されていないす べての権利は、ADIに帰属します。本契約と評価用ボードはすべて、ADIの機密および専有情報と見なされるものとします。お客様は、この評価用ボードの如何なる部分も、如何 なる理由でも他者に開示または譲渡しないものとします。評価用ボード使用の中止または本契約の終了の際、お客様は評価用ボードを速やかに ADI へ返却することに同意するもの です。<

く

道加制度事項>お客様は、評価用ボード上のチップの逆アセンブル、逆コンパイル、またはリバース・エンジェアリングを行わないものとします。

お客様は、アメートの構成材料に影響を与えるその他の行為に限らず、評価用ボードに発生したすべての損傷や修正または改変を ADI へ通知するものとします。

詳価用ボードに対する修正は、RoHS 規制に限らずすべての該当する法律に従うものとします。

く

契約の終了>ADI は、お客様に書面通知を行うことで、何時でも本契約を終了することができるもの とします。お客様は、評価用ボードを速やかに ADI に返却することに同意するものです。<<u>責任の制限</u>>ここに提供する評価用ボードは現状有姿のまま提供されるものであり、ADI はそれに関する如何なる種類の保証または表明も行いません。特にADIは、明示か黙示か整問わず、評価用ボードにおけるあらゆる表明、推奨または保証(商品性、権原、特定目的適合性または知的財産権非侵害の黙示の保証を含みますがこれらに限定されません)を行いません。如何なる場合でも、ADIおよびそのライセンサーは、利益の喪失、遅延コス ト、労賃、またはのれん価値の喪失など(これらには限定されません)、評価用ボードのお客様による所有または使用から発生する、偶発的損害、特別損害、間接損害、または派生 的損害については、責任を負うものではありません。すべての原因から発生する ADI の損害賠償責任の負担額は、総額で 100 米国ドル (\$100.00)に限定されるものとします。<輸出 >お客様は、この評価用ボードを他国に直接的または間接的に輸出しないことに同意し、輸出に関する該当するすべての米国連邦法と規制に従うことに同意するものとします。準 拠法。本契約は、マサチューセッツ州の実体法に従い解釈されるものとします(法律の抵触に関する規則は排除します)。本契約に関するすべての訴訟は、マサチューセッツ州サフォー ク都を管轄とする州法廷または連邦法廷で審理するものとし、お客様は当該法廷の人的管轄権と裁判地に従うものとします。本契約には、国際物品売買契約に関する国連条約は適 用しないものとし、同条約はここに明確に排除されるものです。