

UCB-ADUCM3029 ユーザーズマニュアル

第1版

金子システム株式会社

1



- 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。
 当社製品のご購入およびご使用にあたりましては、当社ホームページを通じて公開される情報を参照ください。
- 当社から提供する情報の正確性と信頼性には万全を尽くしていますが、誤りがないことを保証する ものではありません。当社はその使用に対する責任を一切負いません。その使用によって第三者の 特許権、著作権その他知的財産が侵害された場合でも、同様に責任を負いません。
- 3. 本資料は、当社の書面による事前の明示同意がない限り、いかなる形式でも複製できません。
- 4. 当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一 切その責任を負いません。

目次

1	はじ	こめに	3
	1.1	パッケージ内容	3
2	ハー	-ドウェア・リファレンス	4
	2.1	製品外観	4
	2.2	ブロック図	4
	2.3	外形寸法	5
	2.4	電気特性	6
	2.5	ブートモードの設定	6
	2.6	CN1/CN2 コネクタ仕様	6
	2.7	SWD	8
	2.8	CrossCore Embedded Studio の設定	9
	2.9	CrossCore Embedded Studio でのデバッグに関する注意	0
	2.10	内蔵 FLASH の書き換え10	0
	2.11	フラッシュイメージ作成方法1	2
3	更親	履歴1	3

KANEKO

1 はじめに

このたびは当社製品をご購入いただき、ありがとうございます。

本製品は、アナログ・デバイセズ社 MCU である ADuCM3029 を使用した CPU ボードです。本 製品 UCB-ADUCM3029 の特徴は以下の通りです。

- 63.5mm(横)×17.8mm(縦)×1.6mm(基板厚)と小型です
- ブレッドボードに挿入でき、実験に最適です
- 2.54mm ピッチコネクタなので、ユニバーサル基板にて実験が可能です

また、ADuCM3029 プロセッサの特徴は以下の通りです。

- 最大 26MHz で動作する ARM Cortex-M3 プロセッサ
- 超低消費電力
- 256KBの ECC 機能つきフラッシュメモリを内蔵
- 64KB のパリティ機能付き SRAM
- SPI、I2C、SPORT、RTC、SAR ADC 等の豊富なペリフェラルを内蔵
- 25 個の DMA チャンネル
- $64 \lor \sim LFCSP$

詳細は、アナログ・デバイセズ社の ADuCM3029 のサイトを参照ください。

http://www.analog.com/jp/products/processors-dsp/microcontrollers/precision-microcontroller s/aducm3029.html

 回路図やサンプルプログラムは、以下のサイトを参照ください。 <u>http://kaneko-sys.co.jp/support/</u>

【注意!】CorssCore Embedded Studio のバージョンは、2.6.0 以降をお使いください。それ以前 のバージョンをご使用になる場合、DPF の Update を行わないと、内蔵フラッシュの書き換えができ なくなります。

1.1 パッケージ内容

UCB-ADUCM3029のパッケージには、以下が含まれます。

表 1 パッケージ内容

内容	数量
UCB-ADUCM3029 ボード	1枚



- 2 ハードウェア・リファレンス
 - 2.1 製品外観



1 pin

図 1 表面写真

2.2 ブロック図

ここでは、UCB-ADuCM3029ボード上のプロセッサの構成を説明します。





UCB-ADUCM3029は、ADuCM3029プロセッサを中心に、動作に最低限必要となる以下の機能 で構成されています。

- SWD コネクタ (2×5 列ハーフピッチ)
- 26MHz 水晶
- 32.7680kHz 水晶(RTC 用)

2.3 外形寸法





2.4 電気特性

	表 2 電	该特性			
項目	条件	記号	min	typ	max
供給電圧	_	VBAT	1.74V	3.0V	3.6V
ADC用リファレンス電源	_	VREF	1.74V	3.0V	3.6V
3.0V 供給時の消費電流	SWD デバッグ中 待機している状態	ICC1	_	440uA	_
(美測・参考値)	while(1)実行状態	ICC2	_	1197uA	_
Flash ECC 有効 LFXTAL 有効	SRAM にリード/ ライトを繰り返して いる状態	ICC3	_	1118uA	_

※ベースボードからの供給電流は余裕をもった設計にしてください。

2.5 ブートモードの設定

P1 で ADuCM3029 のブートモードの設定を行います。デフォルト(オープンの状態)では内蔵 フラッシュからのブートに設定されます。

ブートモードの詳細は、アナログ・デバイセズ社「ADuCM302x Ultra Low Power ARM Cortex-M3 MCU with Integrated Power Management Hardware Reference」を参照ください。

SYS_BMODE 設定	備考
0	P1 = SHORT
UART download mode	GPIO17='L'
1	P1 = OPEN
Flash boot.	(デフォルト設定)
	GPIO17='H'

表 3 ブートモードの対応

2.6 CN1/CN2 コネクタ仕様

- ※ 各表の入出力は、ADuCM3029 からみたもので、信号名の最後に # が付く名前は、負 論理(Low アクティブ)であることを示します。
- ※ 各ピンの詳細は、アナログ・デバイセズ社「ADuCM302x Ultra Low Power ARM Cortex-M3 MCU with Integrated Power Management Hardware Reference」を参照く ださい。



ピン番号	信号名	入出力	説明
1	VREF	_	ADC リファレンス電源
2	GND		電源グラウンド
3	P2_03	入出力	GPIO35
4	P2_04	入出力	GPIO36
5	P2_05	入出力	GPIO37
6	P2_06	入出力	GPIO38
7	P2_07	入出力	GPIO39
8	P2_08	入出力	GPIO40
9	P2_09	入出力	GPIO41
10	P2_10	入出力	GPIO42
11	GND		電源グラウンド
12	P0_05	入出力	GPIO05
			2kΩの抵抗で VCC_3V3 にプルアップされています.
13	P0_04	入出力	GPIO04
			2kΩの抵抗で VCC_3V3 にプルアップされています.
14	P1_09	入出力	GPIO25
15	P1_08	入出力	GPIO24
16	P1_07	入出力	GPIO23
17	P1_06	入出力	GPIO22
18	P2_11	入出力	GPIO43
19	P0_12	入出力	GPIO12
20	P2_00	入出力	GPIO32
21	P1_15	入出力	GPIO31
22	P1_01	入出力	GPIO17
			SYS_BMODE0 ピン兼用 / P1 に接続
23	P0_09	入出力	GPIO07
24	P0_08	入出力	GPIO08
25	GND		電源グラウンド

表 5 CN2 コネクタ仕様

ピン番号	信号名	入出力	説明
26	GND		電源グラウンド
27	SYS_HWRST#	入力	100kΩの抵抗で VBAT にプルアップされています
28	P1_11	入出力	GPIO27
29	P1_12	入出力	GPIO28
30	P1_13	入出力	GPIO29

31	P1_14	入出力	GPIO30
32	P2_02	入出力	GPIO34
33	P0_14	入出力	GPIO14
34	P1_00	入出力	GPIO16
35	P0_15	入出力	GPIO15
36	P0_13	入出力	GPIO13
37	P2_01	入出力	GPIO33
38	P1_05	入出力	GPIO21
39	P1_04	入出力	GPIO20
40	P1_03	入出力	GPIO19
41	P1_02	入出力	GPIO18
42	P0_11	入出力	GPIO11
43	P0_10	入出力	GPIO10
44	P1_10	入出力	GPIO26
45	P0_03	入出力	GPIO03
46	P0_02	入出力	GPIO02
47	P0_01	入出力	GPIO01
48	P0_00	入出力	GPIO00
49	GND		電源グラウンド
50	VBAT		+3.0V 電源

2.7 SWD

UCB-ADUCM3029には、以下の JTAG-ICE を使用することが可能です。

表 6 JTAG-ICE

製品名	備考
Analog Devices	http://www.analog.com/jp/evaluation/eval-adsp-bf70x/eb.html
ADZS-ICE-2000	
Analog Devices	http://www.analog.com/jp/evaluation/eval-adsp-bf70x/eb.html
ADZS-ICE-1000	



2.8 CrossCore Embedded Studioの設定

※CorssCore Embedded Studio のバージョンは、2.6.0 以降をお使いください。

CrossCore Embedded Studio で開発を行う場合、初回のみ CMSIS Pack Manager から DFP (Device Support)をインストールします。次の手順から DFP をインストールしてください。

Open Perspective から CMSIS Pack Manager を開きます。



Analog Deviecs.ADuCM302x_DFP と ARM.CMSIS をインストールします。

🔀 UCB-ADUCM3029 - CMSIS Pack Manag	ger - CrossCore	Embedded Studio			– 🗆 X
<u>F</u> ile <u>E</u> dit <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject	<u>R</u> un <u>W</u> indow	v <u>H</u> elp			
	0 - 🤷 - 🖓	∛ ▼ ∲ ▼ 🖗 ▼ 🖗 ▼ 🔿 ▼	Qu	uick Access 🛛 😰 🗟 C/C++ 🔅 🛙	Debug 🐞 CMSIS Pack Manager
🔳 Devices 🛿 📑 Boards	- 0	🛞 Packs 🛛 📑 Examples	E E	0 2 🖉 🗞 🕍 🗸 🗆	😑 Pack Prop 🛛 🗖 🗖
E E	? 💥 🗸	Search Pack			∃ ⊡ ? ⊽
type filter text		Pack	Action D	Description ^	type filter text
Device Sur	mmary ^	✓ ■ Device Specific	155 Packs A	II Devices selected	✓ ⊕ AnalogDevices.ADuCM30
🗸 🔧 All Devices 398	8 Devices	> ABOV.CM0_DFP	🔅 Install 🛛 A	BOV Semiconductor CM0 Device S	> 💠 Components
> 🔗 ABOV 10 E	Devices	> the ABOV.CM3_DFP	🔅 Install 🛛 A	BOV Semiconductor CM3 Device S	> Devices
> 🔗 Active-Semi 4 De	evices	> Active-Semi.PAC52XX	🔅 Install 🛛 P/	AC52XX Family of Power Application	
> 🔗 Ambiq Micro 6 De	evices	AmbiqMicro.Apollo_DFP	😵 Install 🛛 🗛	mbig Micro Apollo Series Device S	
> 🔗 Analog Devices 21 [Devices	AnalogDevices.ADuCM302x_DFP	💠 Up to date 🛛 🗛	nalog Devices ADuCM302x Device	
> 🔗 ARM 40 E	Devices	AnalogDevices.ADuCM302x_EZ_KIT	😵 Install 🛛 🗛	nalog Devices ADUCM302x EZ-Kit	
> 🔗 Atmel 273	Devices	AnalogDevices.ADuCM320_DFP	🔅 Install 🛛 🗛	nalog Devices ADuCM320 Device !	
> 🗸 Cypress 425	Devices	AnalogDevices.ADuCM36x_DFP	🔅 Install 🛛 A	nalog Devices ADuCM36x Device !	
> 🔗 GigaDevice 70 [Devices	AnalogDevices.ADuCM4x50_DFP	🔅 Install 🛛 A	nalog Devices ADuCM4x50 Device	
> 🖗 Holtek 22 [Devices	AnalogDevices.ADuCM4x50_EZ_KIT	🔅 Install+ 🛛 A	nalog Devices ADuCM4x50 EZ-Kit	
> 💡 Infineon 166	Devices 🗸	AnalogDevices.CM41x_M0_DFP	🔅 Install 🛛 🗛	nalog Devices ARM Cortex-M0 CN 🗸	
<	>	<		>	< >
🗐 Console 🛛 🕞 Progress					
CMSIS Pack Manager					
					A
<					>



🔀 UCB-ADUCM3029 - CMSIS Pack	Manager - Cross	ore	Embedded Stu	tudio			– 🗆 X
<u>F</u> ile <u>E</u> dit <u>N</u> avigate Se <u>a</u> rch <u>P</u> r	oject <u>R</u> un <u>W</u> ir	dov	/ <u>H</u> elp				
🔁 - 🛛 🕤 🗟 😂 🔗	🔤 🔿 - 💁		🔗 🕶 🖢 👻	• 4 • ⇔ ⇔ • ⇒ •		Quick Access 🔡 🔡 🖬 C/C++ 🔅	Debug 🛞 CMSIS Pack Manager
📕 Devices 🛿 🗾 Boards	- 1	3)	ጰ Packs 🖇	🛛 📑 Examples	Ē	🖻 🕐 😂 💝 🗁 🔻 🗖	📒 Pack Prop 🛛 📃 🗖
	9 🖸 🖓 🙀	\bigtriangledown	Search Pack	k			
type filter text			Pack		Action	Description ^	type filter text
Device	Summary	^	> 🖶	TexasInstruments.MSP432P4xx_DFf	🔅 Install	Device Family Pack for Texas Instrun	V 🖶 ARM.CMSIS.5.0.1
✓ ⁴ S All Devices	3988 Devices		> 🖶	Zilog.ZNEO32_DFP	🚸 Install	Zilog ZNEO32! Family Device Suppo	> 🖪 Boards
> 🔗 ABOV	10 Devices		🗸 🔍 Ger	neric	18 Packs	Software Packs with generic conten	> 💠 Components
> 🔗 Active-Semi	4 Devices		7 1	AnalogDevices.ADI-SensorSoftware	😵 instail+	Analog Devices Sensor Drivers and I	> E Devices
> 🔗 Ambiq Micro	6 Devices		> 🖶	ARM.CMSIS	💠 Up to date	CMSIS (Cortex Microcontroller Soft	
> Analog Devices	21 Devices		> #	ARM.CMSIS-Driver_validation	😻 instail	CIVISIS-Driver validation	
> 🖉 ARM	40 Devices		> 🖶	ARM.CMSIS-FreeRTOS	🔅 Install	Bundle of FreeRTOS for Cortex-M ar	
> 🔗 Atmel	273 Devices		> "	ARM.CMSIS-RTOS_Validation	🔅 Install	CMSIS-RTOS Validation	
> 🔗 Cypress	425 Devices		> 🖷	ARM.mbedClient	🔅 Install	ARM mbed Client for Cortex-M dev	
> 🔗 GigaDevice	70 Devices		> "	ARM.mbedTLS	🔅 Install	ARM mbed Cryptographic and SSL/	
> 🔗 Holtek	22 Devices		> "	ARM.minar	🔅 Install	mbed OS Scheduler for Cortex-M de	
> 🔮 Infineon	166 Devices	~	> "	Huawei.LiteOS	🔅 Install	Huawei LiteOS kernel Software Pacl 🗸	
<	>		<			>	
🗐 Console 🕱 🖷 Progress							
CMSIS Pack Manager							·
							\$
<							>

2.9 CrossCore Embedded Studio でのデバッグに関する注意

CrossCore Embedded Studio上にて SWD を使ったデバッグを行う場合、ADuCM3029の内蔵 フラッシュにプログラムが書き込まれている状態ではデバッグができません。内蔵フラッシュの書 き換えを行った場合、「2.10 内蔵 FLASH の書き換え」を参考に、Erase を行ってください。

2.10 内蔵 FLASH の書き換え

内蔵フラッシュメモリを書き換える場合、CrossCore Utilities の CrossCore Serial Flash Programmer をご利用ください。

こちらよりダウンロードできます。

<u>http://www.analog.com/jp/design-center/processors-and-dsp/evaluation-and-development-s</u> oftware/crosscore-utilities.html

Flash 書き換えの際、UART download mode に設定する必要があります。P1 ピンをショート したままリセットまたは電源を投入してください。



🔀 CrossCore Serial Flash Program	mer	_	
Iarget ADuCM302x ▼	Serial <u>P</u> ort COM7 (USB Serial Port)	-	<u>B</u> audrate 115200 ▼
Action Program	<u>K</u> ey		
Second stage kernel [C:\Analog Devices\CrossCore Utilitir	es 1.3.0\etc\ccsfp\program	mers\ADul	CM: Browse
File to <u>d</u> ownload			
C:\tmp\led_blink.hex			Browse
Status			
	9	<u>à</u> tart	Cancel



2.11 フラッシュイメージ作成方法

CrossCore Embedded Stuio で内蔵フラッシュに書き込むためのイメージを作成する場合は、 次の設定を行います。

Project Properties を開き、C/C++ Build – Tool Settings タブ – CrossCore GCC ARM Embedded C Linker – Libraries の Semihosting support を nosys_spaces に変更します。





M Properties for led_billik	_	
type filter text	Settings	<⊳ ▼
 Resource Builders C/C++ Build Build Variables Environment 	Configuration: Debug [Active]	figurati
Logging Settings	🛞 Tool Settings 🔝 Processor Settings 🎤 Build Steps 🕎 Build Artifact 🗟 Binary Parse	rs 😢
Warnings	Pre-build steps	
> C/C++ General Droject References	Command:	
Run/Debug Settings		
	Description:	
	Post-build steps	
	Command:	
	arm-none-eabl-objcopy -O inex sterojivame; sterojivame; nex	
	puon:	
下記追加します。		
arm-none-eghi-ohio	ony - () ihay \${ProiName} \${ProiName} hay	
arm none cabi obje	opy o mex $\varphi(\mathbf{r} \mathbf{r} \mathbf{o} \mathbf{j} \mathbf{v} \mathbf{a} \mathbf{m} \mathbf{c}) $ $\varphi(\mathbf{r} \mathbf{r} \mathbf{o} \mathbf{j} \mathbf{v} \mathbf{a} \mathbf{m} \mathbf{c}) $.	

これで、ビルド後に ihex 形式ファイルが作成されます。

3 更新履歴

版	更新日	更新内容
第1版	2017/07/08	初版発行