BCRX210 マイコン開発セット マニュアル

第1版2013.3.17

第1版

【 製品概要 】

本マニュアルはBCRX210 CPUボードのソフトウエア開発を行うために必要なソフトウエアイ ンストゥール手順、添付CDのサンプルプログラムの動作について解説されています。本CPUボード開 発にはルネサスエレクトロニクス社製E1が必要です。



1. 開発環境、事前準備

1-1. 開発環境

- a:開発セット 同梱物
- b: BCRX210 CPUボードの特徴
- c:デバッカE1
- d:無償のHEW、RX用Cコンパイラのダウンロード
- e:CDコピー、デバイスドライバD2XXのインストゥール
- 1-2 動作、デバック
 - a:HEW起動、コンパイル、書き込み、動作
 - b:新しいプログラムを作る

2. サンプルプログラム

2-1. sample1 出力ポートのON, OFF
2-2. sample2 SIO (USB) でパソコンとやりとり
2-3. sample3 A/D変換をUSB出力、D/A出力
2-4. sample4 割り込み
2-5. sample5 新規ワークスペース製作見本
2-6. sample6 三角、対数、平方根関数を使う

1-1. 開発環境

a:開発セット同梱物

RX210 CPUボード CD(サンプルプログラム、デバイスドライバ、ドキュメント) マニュアル(本誌)

電源ケーブル



※開発に必要なルネサスエレクトロニクス社製デバッカE1は同封されておりません。別途必要です。

b:BCRX210 CPUボードの特徴

●RXアーキテクチャコア(32ビットシングルチップCISC 最大78DMIPS 50MHz動作、
 32ビット乗算器内蔵) RX2108 CPUを搭載。

RXはルネサスの従来のマイコンに比べ、処理性能2倍、コード効率30%向上、消費電力1/3、4倍の高集積を目標に設計されています。例えば1.62Vで20MHz動作、2.7V以上で50MHzという高速動作を実現。動作時わずか0.2mA/MHz。デバイスでエコを実現しています。※1 データ処理でH8Sの3.6倍、コードサイズで38%の向上を達成。

※1 3.3V、5Vどちらでもフルスペックで動きますが、消費電力、発熱を考えると3.3V動作が お勧めです。

●コンパクト55×72mm、USBインターフェイス。

●動作電圧3.3V~5.5V、30mA TYPE(3.3V、50MHz動作時)、USB接続時、500mA迄であれば外部電源不要。但し、パソコンにより、USBポートから500mA供給できない場合もあります。

●大容量メモリ内蔵 Flash 512KB、RAM 64KB R5F52108CDFM

●豊富な周辺機能

A/D 変換器:12 ビット、16ch、D/A 変換器:10 ビット 2ch、リアルタイムクロック内蔵、シリアル(SCI): 7 ch、強力なタイマ:MTU2(16bit×6ch)、ウオッチドグタイマ、コンペアマッチタイマ(16bit×2ch)、温 度センサ、コンパレータ内蔵

●USB ドライバ IC FTDI 社の FT232RL 搭載。

●デバッカE1によるデバック用コネクタ搭載。

●オプションで外部クリスタル、シリアルフラッシュ ROM、シリアル EEPROM 搭載パッドあります。お問い 合わせ下さい。 基板大きさ



c:E1デバッカ



概要

E1 エミュレータは、ルネサス主要マイコンに対応したオンチップデバッギングエミュレータです。基本的 なデバッグ機能を有した低価格の購入しやすい開発ツールで、フラッシュプログラマとしても使用可能で す。

C 言語ソースデバックが可能で、1 行実行、ブレークポイント設定、変数、レジスタ、メモリ参照等々、従 来であれば高価な ICE しか出来なかった機能が、安価に実現されています。また、使い方もHEW(統合 開発環境)のE8a と同じで、経験があれば半日で、無くても1日で必要な操作を会得することが出来ると 思います。 マイコンとの通信として、シリアル接続方式とJTAG 接続方式の2種類に対応しています。使用可能なデバッグインタフェースは、ご使用になるマイコンにより異なります。

また、基本デバッグ機能に加え、ホットプラグイン機能(動作中のユーザシステムに後から E1 エミュレー タを接続して、プログラムの動作確認を行うことが可能)を搭載しているため、プログラムのデバッグ・性能評価 に大きく貢献できます。

対応MPU

- V850 ファミリ
- RX ファミリ
- RL78 ファミリ
- R8C ファミリ
- 78K ファミリ



E1を購入するとCDが添付されていて、ドライバーのインストールとセルフチェックを行った後に、ネットから開発環境HEWとCコンパイラのダウンロードを行います。

省略

1-2 動作、デバック

a:HEW起動、コンパイル、書き込み、動作



HEWを起動します。ここでは例としてRX210¥sample1を動作させます。初めてのときは「別のプロジェクトワークスペースを参照する」を選択し

ようこそ!	? 🔀
🌮 🔿 新規プロジェクトワークスペースの作成(C)	ОК
<i>F</i>	キャンセル
⑥ 最近使用したプロジェクトワークスペースを閉く(O):	
C:¥WorkSpace¥RX210_sample¥sample1¥sample1.hws	アドミニストレーション(A)
○ □ □ □ □ □ ○ □ ○ □ □ ○ □ □ □ □ □ □ □ □	

拡張子hwsファイルをダブルクリック。以降、同じsample1でしたら「最近使用した、、」でOK です。

🖗 ワークスペースを開く		
ファイルの場所(I): 📜 sample1		•
名前	~	
👢 sample1		
🖕 sample1.hws		

起動設定が表示されます。

起動設定
デバイス 起動と通信
MCUヴループ: RX210 Group
デバイス名(V): R5F52108 🔽
_ 動作モード
● デバッグモード(D)
□ ホットプラグイン(日) 以下を確認してOKを押してください。
 ユーザシステムとエミュレータが接続されていない。 エミュレータシリアルNo が表示されている。
○ フラッシュライタモード(F)
□ デバッガ終了時にユーザプログラムを実行(区)
┌電源供給設定─────
□ エミュレータから電源供給(P)(最大 200mA)
C <u>3</u> .3V C <u>5</u> .0V
通信設定
シリアルNo.: 682862D351&0&4
 OK キャンセル

外部電源から電源を供給する場合はその電源をONにします。USBケーブルを挿入すると自動的にUS Bケーブルから電源3.3Vが供給されます。OKをクリック。

接続中		
	\sim	
Target		×
ユーザシステムの電源を	入れてからOKを押し	てください。
	ОК	キャンセル

例ではE1から電源を供給してみます。本ボードは電源電圧3.3V、5Vいずれでも動作しますが、例では5Vを選択します。(多くの用途に消費電力、発熱にメリットのある3.3Vを推奨します)

電源	
▼ Tミュレータから電源の	共給(P)(最大 200mA)
□	
O 3.3 V	@ 5.0 V
<u> </u>	キャンセル

コンフィグレーションプロパティが表示されます。「内蔵フラッシュメモリ、、」にチェックが入っている必 要があります。確認後、OK。

コンフィグレーションプロ	ロパティ	×
MCU システム 内蔵フラ	ッシュメモリの上書き 外部	フラッシュメモリ
└MCUの動作────		
動作モード(M):	シングルチップモード	_
エンディアン(E):	リトルエンディアン	T
□ EXTAL周波数:		MHz
外部領域(T)		
領域エン	ディアン バス幅	
「エミュレータからの内蔵フ	ラッシュメモリ書き換え――	
☑ 内蔵フラッシュメモ	リ書き換え時のクロック操作	を許可(P)
ワークRAM開始ア (0x400 バイト 使	パレス 1000 用): 1000	
	ОК	キャンセル
□ 次回	回起動時にこのダイアログボ	ックスを表示しない。

sample1のエディタ画面等が表示されます。



プログラムのコンパイルは、以下の「ビルド」ボタンで行います。ボタンはマウスを乗せると意味が表示 されます。

	/
Workshop - [sample1.c]	
クト(P) ビルド(B) デバッグ(D)	基本設定(U) ツール(T) テス
- M % % ∭ ∰ ⊗	🗄 🛗 👗 🛛 Debug 💽 🔽
et et et et e 6 6 1 e *	RI 💭 🕶 💯 🛄 🖉

sample1は出荷時にすでにコンパイラされていますので、0 Errors、0 Warningsと表示され ます。新たにプログラムを製作してコンパイルしたときにErrorsがOで無い場合、プログラムに文法上の 問題がありますので、エディタでソースファイルを修正し、Errors Oにする必要があります。Errorsが Oで無い場合、書き込み用ファイル xxx. motが新たに作成されません。前のファイルは残りますので、 勘違いして昔のファイルを書き込んでしまわないように注意が必要です。



CPUボードのLEDが点滅しているのが見えると思います。停止はSTOPをクリックします。

プロ	コジェクト(P) ビルド(B) デバッグ(D)	
₽	- M A A	
(<i>1</i> 4]] ET E↓ EI EI EI EI (} (} () (0 I/2 × ×	-

以上がプログラム開発に必要な「コンパイル」「書き込み」「実行」です。 例として s a m p l e 1を書き換えて、コンパイル、書き込み、動作の変化の確認、を行います。

1 1 1		
行… ソースア… オ S. ソ	-7	
73 FFFF8595 74 FFFF8598 75 FFFF8598 76 FFFF859E 77	PORT4.PODR.BYTE = 0xaa; PORTB.PODR.BYTE = 0xaa; PORTC.PODR.BYTE = 0xaa; PORT5.PODR.BYTE = 0xaa;	//ボートON ADでも使用 //ボートON //ボートON //ボートON
78 FFFF85A1 79 FFFF85A4 80 FFFF85A7 81 FFFF85AA 82 FFFF85AD 83	PORTH.PODR.BYTE = 0×aa; PORTI.PODR.BYTE = 0×aa; PORT2.PODR.BYTE = 0×aa; PORT3.PODR.BYTE = 0×aa; PORTJ.PODR.BYTE = 0×aa;	//ボート 0N //ボート 0N //ボート 0N //ボート 0N //ボート 0N
84 FFFF8585 85 86	lwait(25000);	
87 FFFF85BD 90 FFFF85D0 91 FFFF85C3 92 FFFF85C6 93 FFFF85C6 93 FFFF85C2 94 FFFF85C7 94 FFFF85C7 95	PORTA.PODR.BYTE = 0x55; PORTE.PODR.BYTE = 0x55; PORT0.PODR.BYTE = 0x55; PORT4.PODR.BYTE = 0x55; PORT4.PODR.BYTE = 0x55; PORT5.PODR.BYTE = 0x55; PORT5.PODR.BYTE = 0x55; PORT5.PODR.BYTE = 0x55;	//ボートOFF //ボートOFF ADでも使用 //ボートOFF ADでも使用 //ボートOFF ADでも使用 //ボートOFF //ボートOFF //ボートOFF //ボートOFF
96 FFFF85D2 97 FFFF85D5 98 FFFF85D8 99 FFFF85D8 100 FFFF85DB 101 102 102 FFFF85B6	PORTH.PODR.BYTE + 0x55; PORT1.PODR.BYTE + 0x55; PORT2.PODR.BYTE = 0x55; PORT3.PODR.BYTE = 0x55; PORTJ.PODR.BYTE = 0x55; Vx55; Vx55; Vx55;	//ボート0FF //ボート0FF //ボート0FF //ボート0FF //ボート0FF

/Cソースファイルが選択されていること

sample1. cが選択されていること。

Iwait(2500)の中の数値を1桁増やして25000にしてみます(2箇所)。プログラムをセーブします。

ファイルの保存はここをクリックします。エディタで書き込み、保存すると色が変わります(未保存の確認が出来ます)。

Q	sam	nple:	1 -	Hig	h-pe	erfo	rma	nce
1	ファ	7-1)	L(F)	編集	(E)	表	示(V
] D	1 🖻		9	8	8	Ē	ß	{• }
0	U	16	<u>10</u>	<u>8</u>	2	Ņ.	*	# 1
								d vi l

次ぎに、「コンパイル」し、エラーが無い場合、以下が表示されます。はい(Y)をクリックでプログラム がダウンロードされます。

確認要	求 ? 🔀
٠	はいボタンを押すとC:¥WorkSpace¥RX210_sample¥sample1¥sample1¥Debug¥sample1.absをダウンロードします。
	□ 今後表示しない(D)
	はい(Y) いいえ(N) すべてはい(E) すべていいえ(O) キャンセル

「リセット後、実行」をクリックします。LEDの点滅周期が遅くなったのがおわかりいただけると思います。

以上のように、プログラム開発は「エディタ (プログラム作成)」→「セーブ」→「コンパイル」→「エラ ーが無いことを確認」→「書き込み」→結果によって頭の「エディタ」に戻る繰り返しになります。

エディタは使い慣れたものでも使用可能で、その場合、HEWのエディタは使えなくなります。

省略

2. サンプルプログラム

```
2-1 sample1 出力ポートのON, OFF
/*
                                                      */
/* FILE
            ∶sample1.c
                                                      */
/* DATE
            :Wed, Nov 28, 2012
                                                      */
/* DESCRIPTION :Main Program
                                                      */
/* CPU TYPE
            :RX210
                                                      */
/*
                                                      */
/* This file is generated by Renesas Project Generator (Ver. 4.53).
                                                      */
/*
  NOTE: THIS IS A TYPICAL EXAMPLE.
                                                      */
/*
                                                      */
/*
       I/O ON/OFF
       クロックは内部にある低速オンチップオシレータ 125KHzで立ち上がる
*/
①#include "iodefine.h"
//#include "typedefine.h"
#ifdef __cplusplus
//#include <ios>
                             // Remove the comment when you use ios
//_SINT ios_base::Init::init_cnt;
                             // Remove the comment when you use ios
#endif
void main(void);
#ifdef __cplusplus
extern "C" {
void abort(void);
}
#endif
②void lwait(long wait_time)
{
       while(wait_time != 0)
       {
              wait_time--;
       }
}
```

③void main(void)

{

4	PORTA.PDR.BYTE = 0xff;	//全部出力 ポート検査用
	PORTB.PDR.BYTE = 0xff;	//全部出力 ポート検査用
	PORTC.PDR.BYTE = 0xff;	//全部出力 ポート検査用
	PORTD.PDR.BYTE = 0xff;	//全部出力 ポート検査用
	PORTE.PDR.BYTE = 0xff;	//全部出力 ポート検査用
	DODTH DOD RVTE - Avff.	//全虹山力 - 北 6 埃本田
	PORT DR RYTE - Oxff;	
	FUNTU. FUN. DITE - UXIT,	//王라山刀 小一下使重用
	PORTO.PDR.BYTE = 0xff;	//全部出力 ポート検査用
	PORT1.PDR.BYTE = 0xff;	//全部出力 ポート検査用
	PORT2. PDR. BYTE = 0xff;	//全部出力 ポート検査用
	PORT3.PDR.BYTE = 0xff;	//全部出力 ポート検査用
	PORT4. PDR. BYTE = 0xff;	//全部出力 ポート検査用
	PORT5. PDR. BYTE = 0xff;	//全部出力 ポート検査用
5	while(1)	
	{	
6	PORTA. PODR. BYTE = 0xaa;	//ポート出力
	PORTE. PODR. BYTE = 0xaa;	//ポート出力
	PORTO. PODR. BYTE = 0xaa;	//ポート出力 AD でも使用
	PORTD. PODR. BYTE = 0xaa;	//ポート出力
	PORT4. PODR. BYTE = 0xaa;	//ポート出力 AD でも使用
	PORTB. PODR. BYTE = 0xaa;	//ポート出力
	PORTC. PODR. BYTE = 0xaa;	//ポート出力
	PORT5. PODR. BYTE = 0xaa;	//ポート出力
	PORTH. PODR. BYTE = 0xaa;	//ポート出力
	PORT1. PODR. BYTE = 0xaa;	//ポート出力
	PORT2. PODR. BYTE = 0xaa;	//ポート出力
	PORT3. PODR. BYTE = 0xaa;	//ポート出力
	PORTJ. PODR. BYTE = 0xaa;	//ポート出力
7	lwait(2500);	
(8)	PORTA, PODR. BYTE = $0x55$:	//ポート出力
0	PORTE PODR BYTE = $0x55$:	//ポート出力
	PORTO. PODR. BYTE = $0x55$;	//ポート出力 AD でも使用

```
PORTD. PODR. BYTE = 0x55;
                                              //ポート出力
             PORT4. PODR. BYTE = 0x55;
                                              //ポート出力 AD でも使用
             PORTB. PODR. BYTE = 0x55;
                                              //ポート出力
             PORTC. PODR. BYTE = 0x55;
                                              //ポート出力
             PORT5. PODR. BYTE = 0x55;
                                              //ポート出力
             PORTH. PODR. BYTE = 0x55;
                                              //ポート出力
             PORT1. PODR. BYTE = 0x55;
                                              //ポート出力
             PORT2. PODR. BYTE = 0x55;
                                              //ポート出力
                                              //ポート出力
             PORT3. PODR. BYTE = 0x55;
             PORTJ. PODR. BYTE = 0x55;
                                              //ポート出力
             lwait(2500);
      }
}
void abort(void)
{
}
#endif
Option-Setting Memory
(9)#pragma address OFS1_REG = 0xFFFFF88 /* OFS1 register */
const unsigned long OFS1_REG = 0xFFFFFFF;
#pragma address OFSO_REG = 0xFFFFF8C /* OFSO register */
const unsigned long OFSO_REG = 0xFFFFFFF;
【解説】
本プログラムは、出荷検査のポート検査用に作られたもので、全てのポートをON, OFFさせています。
RX210は内部にクロック発振子を内蔵していて、電源立ち上がり時には低速オンチップオシレータ
125KHzで動作しています。動作速度と発振精度をそれほど要求しない動作には消費電力も少なく、
良いモードです。
①#include "iodefine.h"
各レジスタの定義が入っているヘッダです。
②void lwait(long wait_time)
ł
      while(wait_time != 0)
       {
```

wait_time--;

}

メイン関数で使用するLEDのON、OFF時間を設定するウエイトです。

③void main(void)

}

{ メインルーチンです。

 ④
 PORTA. PDR. BYTE = 0xff;
 //全部出力 ポート検査用

 ビットに1を立てて、全て出力にしています。0を設定すると入力になります。

⑤ while(1)
{
ここから無限ループです。

6 PORTA. PODR. BYTE = 0xaa;0 x a a を出力しています。

//ポート出力

⑦ Iwait(2500);ウエイトです。2500が0になるまで戻ってきません。

(8) PORTA. PODR. BYTE = 0x55;

//ポート出力

0 x 5 5 を出しています。2進数で書くと0 b 0 1 0 1 0 1 0 1 です。前が0 x a a でしたが2進数表記 では0 b 1 0 1 0 1 0 1 0 となります。お互いを反転させた値をポートに書き込んでいます。

 $0 \times a a = ~ 0 \times 55;$

以下省略

2-6 三角、対数、平方根関数を使う

```
/*
                                                           */
/* FILE
             ∶sample6.c
                                                           */
/* DATE
             :Tue, Apr 02, 2013
                                                           */
/* DESCRIPTION :Main Program
                                                           */
/* CPU TYPE
             :RX210
                                                           */
/*
                                                           */
/* This file is generated by Renesas Project Generator (Ver.4.53).
                                                           */
/* NOTE: THIS IS A TYPICAL EXAMPLE.
                                                           */
/*
                                                           */
①#include <math.h>
②double d1, d2, d3;
short s1, s2, s3;
3#define PI 3.14159265
//#include "typedefine.h"
#ifdef __cplusplus
//#include <ios>
                                // Remove the comment when you use ios
//_SINT ios_base::Init::init_cnt;
                                // Remove the comment when you use ios
#endif
void main(void);
#ifdef __cplusplus
extern "C" {
void abort(void);
}
#endif
void main(void)
{
(4)
       d1 = \log 10 (10000);
       d2 = sin((PI/180)*45);
       d3 = sqrt(2);
(5)
       s1 = d1;
       s2 = d2;
       s3 = d3;
       while(1)
```

}

#ifdef __cplusplus
void abort(void)
{

;

} #endif

【 解説 】

(1)#include <math.h>

三角関数や、対数、平方根を使うためにはmath. hをインクルードする必要があります。加えて、ビルド(B) \rightarrow RX Standard Roolchain、、をクリック。



以下省略

②double d1, d2, d3; short s1, s2, s3;

結果を入れるdouble(32ビット)3つ、キャストするshort(16ビット)3つをここで定義 しています。

③#define PI 3.14159265三角関数計算で角度を入力して数値を出すために使います。

④ d1 = log10(10000); d2 = sin((PI/180)*45); d3 = sqrt(2);

頭から

d1=log10(10000)、答えは4になるはずです。

d 2 = s i n ((P I / 1 8 0 * 4 5) → s i n (4 5°)という意味です。答えは0.707106、 になるはずです。

d3=sqrt(2) →平方根の2です。答えは1.41421356になるはずです。

d1、d2、d3の結果を見るために、メモリをクリック

	_
e6.c]	
デバッグ(D) 基本設定(U) ツール(T) テスト(\$
3 % .₩ @ ₩ ₩ Δ_ Debug 💽 Def	a
() @ I _{rc} > _c E F ₩ F E Ø	A
XEU	

変数名d1を探します。ここをクリック。

表示開始アドレス		? ×
表示開始アドレス:	d	
スクロール開始アドレス:	00000000	▼ 🗾
スクロール終了アドレス:	FFFFFFF	▼ 🗖
ОК	キャンセル	

ラベル名一覧が表示されます。

ラベル選択		? <mark>×</mark>
フィルタ(F):		ОК
Drtl COM_CONVF32u COM_MUL64 COM_MUL6 COM_SUBF COM_SUBF CTBL DTBL Prmo INITSCT MDEreg Drk CTS COM_SUBF	<u>アドレス</u> FFFF894 FFFF805 FFFF816 FFFF8100 FFFF8180 00001420 FFFF800E FFFFF800 00001420 00001420 00001420 00001420 00001014 00001014 00001014 00001014 00001010 FFFF9045 FFFF9045 FFFF9045 FFFF9045 FFFF9045 FFFF9045 FFFF9045	**>\triple

d 1, d 2, d 3 共連続した 4 バイトとびにあるのが確認出来ます。d 1 をクリック。 同様の方法でs 1、s 2、s 3 も見れるように二つの窓を開いておいて下さい。

例えば下記2箇所にブレークポイントを設定します。「リセット後、実行」をクリックします。



d1、d2、d3はどうなっているでしょうか?

×	1 II m	••• <u>16</u> <u>10</u> <u>+1</u>	0 <u>8</u> 2 db	• 💩 💩 🚋	. 6. A.	16.32		
	Address	+()	+4	+8	+C	FLOAT		
	00001014	40800000	3F3504F3	3FB504F3	00000000	+4.000000e+000	+7.071068e-001	+1.414214e+000
Ш	00001024	00000000	00000000	00000000	00000000	+0.000000e+000	+0.000000e+000	+0.000000e+000
	00001034	00000000	00000000	00000000	00000000	+0.000000e+000	+0.000000e+000	+0.000000e+000
Ш	00001044	00000000	00000000	00000000	00000000	+0.000000e+000	+0.000000e+000	+0.000000e+000
	00001054	00000000	00000000	00000000	00000000	+0.000000e+000	+0.000000e+000	+0.000000e+000
Ш	00001064	00000000	00000000	00000000	00000000	+0.000000e+000	+0.000000e+000	+0.000000e+000
	00001074	00000000	00000000	00000000	00000000	+0.000000e+000	+0.000000e+000	+0.000000e+000
+	4.5.14		001.01				ムモブ ちまー・	

左からd1、d2、d3の4バイトHEXデータです。右が浮動小数点付きデータ表示です。答えは合っていますね。

以下省略

WindowsXP®、WindowsVist®、Windows7®はマイクロソフト社の登録商標です。 フォース®ライタは弊社の登録商標です。

- 1. 本文章に記載された内容は弊社有限会社ビーリバーエレクトロニクスの調査結果です。
- 2. 本文章に記載された情報の内容、使用結果に対して弊社はいかなる責任も負いません。
- 3. 本文章に記載された情報に誤記等問題がありましたらご一報いただけますと幸いです。

4. 本文章は許可なく転載、複製することを堅くお断りいたします。

お問い合わせ先:

〒350-1213 埼玉県日高市高萩1141-1
TEL 042(985)6982
FAX 042(985)6720
Homepage: http://beriver.co.jp
e-mail: info@beriver.co.jp
有限会社ビーリバーエレクトロニクス (C)Beyond the river Inc. 20130325