

By (株)秋月電子通商



- 「RX220 CPU ボード」+「RX220 Base ボード」 -40mm x 29mm 95mm x 47mm ■ 開発ソフトのダウンロード手順:

添付 CD の中のフォルダー名「RX220 開発ソフト」を開き、 Windows7 以上の PC(パソコン)に、開発ソフトをダウンロードします。



1、開発ボードの準備:

RX220 マイコンボードは、「RX220 CPU ボード」と「RX220 Base ボード」で、構成されています。 「RX220 CPU ボード」は、40mm x 30mm の超小型 CPU ボードです。「RX220 Base ボード」は、 95mm x 47mm のベース(土台)となるボードで、AC アダプタ端子(5V)、E1 エミュレータ端子、 シリアル書き込み端子が搭載されています。

第1章は、D-Sub端子(9pin)を使用した、シリアル書き込みによる操作方法を説明しております。
 PC(パソコン)と「RX220 Base ボード」の D-Sub端子(9pin)を接続したプログラム書き込み方法
 について、USB-シリアル変換ケーブル(通販コード: M-02747)を使用して実際に即した内容で、
 説明致します。
 電源は、「RX220 Base ボード」のAC アダプタ端子(5V)から供給します(通販コード: M-06590等)。

第2章は、E1エミュレータを使用した書き込み操作方法について、説明しております。 PCのUSB端子から電源をもらうので、「RX220 Baseボード」のACアダプタ端子(5V)からの 電源供給は不要です。

第3章は、E1エミュレータを使用したデバッグ機能操作方法について、説明しております。 PCのUSB端子から電源をもらうので、ボード電源のACアダプタは不要です。

第4章は、「クロック発生回路」について、説明しております。

■ それでは、「RX220 CPU ボード」に、プログラムを書込み、ボードを動作させる手順まで操作を おこなってみましょう。

(注意):かならず、本、手順書を最初から順番に読んでいってください。 (ページを飛ばさないでください)

1、開発ソフトのインストール:

まず初めに、開発ソフトをパソコンにダウンロードします。

PC は、Windows7 以上の搭載機を使用。(Mac OS 機は、使用できません)

ダウンロードするプログラムは、付属CD内のフォルダー「RX220開発ソフト」の

「RX220 用 C コンパイラ」、「RX220 用 E1 エミュレータソフト」、

「RX220 用フラッシュライタソフト」の順に、お手持ちのパソコンに、ダウンロードします。

(3本のプログラムは、無償評価版です。注意事項をお読みの上、ご使用してください)

「RX220 用 C コンパイラ」は、C 言語で書いたソースプログラムを

RX220マイコンが判る言葉に変換するソフトです。

「RX220 用 E1 エミュレータソフト」は、「RX220 CPU ボード」以外に、E1 エミュレータと

「RX220 Base ボード」を使い、オンチップデバッカ機能を実現する為のソフトです。

「RX220 用フラッシュライタソフト」は、パソコンから RX マイコンのフラッシュ ROM に、

プログラムを書込むためのソフトです。PCと「RX220 Base ボード」の D-Sub に接続して、

直接書き込みする方法と、E1 Emulator 端子に E1 エミュレータを接続して書き込みする方法が可能です。 E1 エミュレータ本体は、弊社からも販売中です(通販コード M-06155)。

それでは PC に、開発ソフトをダウンロードしてみましょう。

1、まず、ダウンロード対象の PC が、外部機器(インターネット、外付けDVD、外付けHDD、USB メモリ他)と接続されている場合、全て、外部機器接続を取り外してください。

PCの内蔵DVD装置に、今回付属のCD「RX220開発ソフト」をセットし、 スタート→コンピューター→DVDドライブ(E:)と、マウス左をクリックします。





フォルダ「RX220 開発ソフト」を、マウス左クリックします。

まず始めに、

1、「RX220用Cコンパイラ」をダウンロードする。



「**RX220**用 C コンパイラ」を、マ ウス左でクリックする。



















「その他の地域(日本、アジア他)」を、 マウス左クリックし、チェックマークを付ける。











次に、

2、「RX220 用 E1 エミュレータソフト」をダウンロードする。











「インストール」が開始される。













く戻る(日) インストール(中

キャンセル

23:33
 2014/07/

しばらくすると、「RX E1/E20 Emulator Debugger」が立ち上がる。

and discuss answere discussion of the

88 7 E 🔉 💊 🖸

ູ ・インストール..









次に、

3、「RX220 用フラッシュライタソフト」のダウンロードを行う。





圧縮ソフトなので、 一般的なフリーの解凍ソフト を使い、「解凍」する。













2) *RV220マイコンのプログラムを行って、 飲かしてみよう (Ef: 1c Monet Wee	- 0	*
フィールビ 編集(日 先行(の 挿入(日 書前)の) ツール(日 軒線)	6 90/P900 NR718		×
	H I III III III IIII IIII IIIIIIIIIIII	• ○ ▲ ×・ ■ ■ ■ ■ Ⅲ Ⅲ Ⅲ Ⅲ № ♥ ▲・. • ○ ○ ● ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲	「次へ (N) >」を、 マウス左クリックする。
 ● ● 〒 9 4 ● ● ● 9 4 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●			













「 High-performance Embedded Workshop」をマウス左クリックする。



「 High-performance Embedded Works」の赤の炎マークを、マウス 左をクリックしたまま、Windows 画面上に導き出し、画面上に貼り付 ける。

次に、



Windows 画面上の「High-performance Embedded Works」の赤の炎マークを マウス左クリックすると、下記、新規画面となる。

> 「新規プロジェクトワークスペー スの作成(C)」にチェックマーク を付け、「OK」をマウス左クリッ クする。







R High-performance Embedded Workshop	NAME OF TAXABLE AND ADDRESS OF TAXABLE ADDRESS OF T	- <u>a</u> ' ×
DERESSEN JOSTON	(A) 10 10 10 10 10 10 10 10 10 10 10 10 10	
	FHOT TO 'SP-3/10-17/300 91-1/6.17/50/2488.0.49. FHOT TO 'SP-3/10-17/300 91-1/6.17. FHOT TO 'SP-3/10-17/300 91-1/6.17. FHOT TO 'SP-3/10-17/300 91-1/6.17. FHOT TO 'SP-3/10-17.17. 91-1/6.17.	「次へ (N) >」を マウス左クリックす
2 12/1044 (Cobus), Red mRs 1), FreimRs 2), 10	News (), Test (), Version Control (2 A投资金 2 mm + A 投资金 2 mm + A 型 T2 40 - 31:44











High-performance Embedded Workshop	the second se			
ファイル(F) 編集(E) 表示(VI プロシュクト(P) ヒッド(D)	テバラグ(ロ) 基本設定(ロ) ワール(T) ナスナ(5)	345450(W) 1662(M)		
DEBO GIARBE 9	「原見思」の自己を言	크미	1 2	
S B C B P S 64 07 44 47 \$1 \$21 \$21 \$21 \$21 \$21 \$21 \$21 \$21 \$21	rs / Tet /, wear Cett /	- 19945. State Research 20 Street core 20040-20 Enter Enter Enter Enter 20040-20 RT #co		「次へ (N) >」を マウス左クリックする。





	・たぶん 出合の西西 A Resource ● DelasSenson ● ス	
TESTI		
C cource file C dosoto S intore to		
D reiefprec D stricc El TESTIC		
🔊 vectrbic E 🄄 Dependencies		
B stackaoth B stackaoth	High-performance Embedded Workshop	
i vecth	▲ セッション "DefaultSession"(プロジェクト"TEST1) が変更されまし	
	A た. 保存しますが7	
	(株)(2) (4)(20) (本センセル)	
		_
OL VI VI RI RI		-
		1
The Build & Debug & Find in Files 1 A Find in Files 2	12 入 Marra 入 Test 入 Verson Campa /	-

「Debbag」→「Release」に 変更する。 (Debbag 作業をしないので、 Release を選択した)

変更すると、下記画面が表示され、 「はい(Y)」を マウス左クリックする。



「TEST1.c」を

マウス左ダブルクリックすると、 C 言語開発環境のエディター 画面が表示される。 ユーザのソースプログラム (main)以外は、既に自動的にプロ グラムが作成されている。

	日 1941 - 1945	
Dependincial disclast disclast	/Tinclude "(yeadefine.h" # forolumbia # forolumb	
	37 3 31 (fet	5
OL OT AL AT XI XI	H 9	
*	я 1.), Тебя РЛе 2.), Мосо), Теят), Теяхол Солез /	
ady	🖪 🖸 🖬 🖸 Defaulti desktop - Read-write - 1/35	A 10 0 0
🦉 (C 🖵 📢		2014/07/24

G TEST! - High-per

ここの main 以下のプログラム に、ユーザの作ったソースプログ ラムを、書き込んでいく。

Void main(Void)

{

← ここに、書き込んでいく。

} #ifdef _cplusplus

■LED1(赤)と LED2(緑)を、 交互に約1秒点滅させる C 言語 プログラムを作ってみよう。





ツールバーからビルド→

"RX Standard Toolchains"を選択。 起動した"RX Standard Toolchains ダ イアログ"で最適化リンカタブを選 択。カテゴリに"セクション"、設定項 目に"セクション"を選択。 編集ボタンを押下。"セクション設定" ダイアログが起動。SU,L をそれぞれ 選択して削除ボタンを押下。 最後に OK ボタンを押下して "セクション設定"ダイアログを閉じ る。"RX Standard Toolchains" ダイアログを"OK"ボタンを押下し 閉じる。「ビルド」を再度、実行する。

1551 - High-performance Protector Yoshino - 1551 よ つ ファイル() 編集() 第5() プロシェクト() という() アイルグ(0) 編集部支(0) グール(1) アスト(5) ウインドウ(0) へんプ(0) D かとし ペール クレート		
Image: State		「ビルド」を再実行すると、 2 Warnings が 0 Warnings と なった(OK)。
Image: Solution of the soluti	4 .	
Redy Image: State of the state	dexistop Read-write (15/48) 1 (2015) 전 8 8월 알 알 알 알 가 다 22:59 고 24 8월 알 알 알 다 다 구 27 1 (22:59) 고 24 8월 알 알 알 다 다 구 27 1 (22:59)	
LD 日本時の180 Christian H. D. Richard Description GF 第4(0): Christian <		「ファイル」をマウス左クリ ックし、「すべて保存」をマウ ス左クリックする。
ttv5a20JJレv5a(k) (11): ************************************	24 24	
EC 20 2 - ドレムモジュール(出) T プリケーションの時7(()) U (trops, 0 Hernings T 1 \ Madel / Debug / Fred miles 1 / Fed miles 2 / Made / Text / Verson Control / E 区 区 Defue C 2 O	ulti desktop Read-write 17/46 (79 (365) ▲ A 前 20 (20) 100 (10	
第 TETT - Haghperforments Entended Watching - (TETT) つ ファイル(F) 構成() 影形(Y) プロシェクト(F) 上のト(B) デバッグ(D) 影響設定(D) ジール(T) デスト(S) ウィンドウ(W) へいプ(R) D G 目 G G I S R(H) (A) コール 名 名 田 G C 田 本 Fareser コ [PriceScience] (TETT) (TE	21A	「X」を
Image: Section 1 Image: Section 1 Image: Section 2 Image: Section 2 Imag		マウス左クリックする。
	dti desktop Read-write 17/48 (79 UNS) ぞれ 税役 20 (20 10 100 年 - 第 20 1 2014/07/27	





スタート→すべてのプログラム と、マウス左クリックする。



「Renesas Electronics utilities」 →書き込みツール→Renesas Flash Programmer をマウス左クリック する。

稲妻マークに**R**と記載された 「**Renesas Flash Programmer** V2.04」アイコンを、マウス左クリッ クしたまま Windows 画面に引き出 し、画面上に貼り付ける。



稲妻マークに**R**と記載された 「**Renesas Flash Programmer V2.04**」アイコンを Windows 画面 上に引き出し、貼り付けたところ。



まとめ:

 C 言語のソースプログラムを 作成する場合には、
 「High-Performance
 Embedded Workshop」アイコンを開 き、プログラムの作成を行う。

②、作成したプログラムを マイコンに書き込む場合には、
「 Renesas Flash Programmer V2.04」アイコンを開き、プログラム の書き込みを行う。

(但し、E1エミュレータでの書き込み時は、本ソフトは使用しません。 B、第二章を参照してください) 今後の開発時には、①と②の2個の アイコンのみで、プログラムの開発 が出来きます(但しA、第一章のシ リアル接続での開発方法のみに適応 します)。

A、第1章:

■ 次に作成した「TEST1」プログラムを、「RX220 CPUボード」に書き込んでみましょう。

ここでは、「RX220 CPU ボード」と、「RX220 Base ボード」(どちらも、弊社で販売中)を 使い、「TEST1」プログラムを書き込んでみます。

最近の PC は、シリアル端子(9 PinD-Sub)が無くなり、USB に置き換わっています。 そこで、USB-シリアル変換ケーブル(通販コード M-02747)を使うと、「RX220 CPU ボード」を、 USB端子のみの PC と、接続することが可能です。 **USB**-シリアル変換ケーブルとして、グレー色、延長ケーブル付(通販コード**M**-02747)を 使用します。



■「RX220 Base ボード」の CN4 (D-Sub) を介して、PC (Windows7 以降)のUSBと接続します。



 ■ さて、上記の接続が済み、USB-シリアル変換ケーブルが PC のポートを認識していることを確認出来たなら、 「RX220 CPU ボード」の、「DIP」 スイッチを書き込みモード (SW No.1~2、すべて ON 側にします。 次に、「RX220 Base ボード」の電源を ON にします。LED1 (赤)、LED3 (赤) が点灯し、 プログラムの書き込み待ち状態となります (「ブートモード」)。
 (注):「DIP」スイッチの切り替え時は、必ず電源を OFF にして、

操作を行ってください。



書き込み準備が出来た、 「RX220 CPU ボード」+「RX220 Base ボード」の状態

簡易入出力チエック部:

「RX220 Base ボード」上には、 簡易的にプログラムの入出力の実行 状態の目視チエックが出来る、入力ス イッチ (SW1)と、出力LED1 (赤), LED2 (緑)が搭載されて います。

入力スイッチ(SW1)を、ON側 にした時、ポートHの2ビット目、 PH2(CN2-17)に、"0"が入力され、 OFF側にした時、"1"が入力され ます。

出力 LED1(赤)は、ポートHの0ビ ット目、PH0(CN2-15)から"1"が 出力されると点灯します。出力 LED2(緑)は、ポートHの1ビット目、 PH1(CN2-16)から"1"が出力され ると点灯します。

("0"が出力されると、消灯します) [LED が点灯すると、LED 両端電圧は 約2V、低くなります)]



「Renesas Flash Programme...」 を、マウス左クリックする。

USER .	Microsoft Word	79986-78		
	Righ-performs Embedded Wo	ender1985		
	Renesas Fissi Regramme		-X-2	
) au			
			 新回送用したワークスペースを開く(1) CHUterSHUESRAppOunt/OcalifPerenas Flash ProgrammerW2284.01NNN/rws 作成成系のグークスペースを開く(0) 	
	18 000 - 300- 14 2003			
é				● A 般 20 0 core of - 日 1 4 23:13 cone - 日 1 4 2014/07/28



R **1** 2 ファイル(F) ツール(T) マイクロコントローラ(M) ~いプ(H マイクロコントローラ 信方式 R5232C 10 50 5-7-0 使用ツール(T) COM4 1865 5 100 ominisii Anomisii 7 De ung 100-11-104 < 戻る(B) 次へ(N) > 并约也多 8 ×E 出力パネルのクリア(C) 8 ູ - 2/3 ● A般警察 ② 100% 2 · 會 包 4 23:39 2014/07/28 ê 🤉 🖹 🗼

使用ツール(T)は、シリアル接続 なので、COM 番号を選択する。 今回の COM4 の4は、PC の PORT 4番が選択されているということ です。自動的に PORT 番号が、割り 付けされ表示されます。 (COM 番号が表示されない場合は

USB ケーブルが正しく接続されて いないので、チェック要)

> 「次へ (N) >」を マウス左クリックする。

USER	Microsoft Word	2000-03			
Microsoft Outlock	Algh-performa Embedded Wo	ema CIMG1935			
N 1781-9-	Renesas Flash Programme		マイクロコントローラ. 通信方式 22		
() 7977-0			確認(Q1010002)		
50 C	Contendand Decempt		ターゲットボードの接続、電源、モードエントリを確認してください。		
2240-11./M					ー 「OK」を マウス左クリックする。
noogle clarame	CIMCHORD		出力//未4@20760		
Norton 360 Prenter Edition	1140-20-k 2995				
1	é 🤉 🛙			● A 报 S S Ø O cors 2 + 音 D ● 2014/07/29	





USER	Microsoft Word	95920-92		/	
Microsoft Outlock	Algh-performs Embedded Wo	CIMG1935	Genesss Rish Programmer (Unsupported Freeware Version) □ □ × Zアイル(F) ジール(T) マイクロコントローラ(M) ヘルブ(H) COUPT 0 0 0		
1 021-9-	Renesas Flash Regionine		マイクロムファローフ. クロック在版 2052の供給		
101-0 101-0	. #		PdB/外部クロナ20 Esternal Reconstor or Clock 入力/2019/0F) Refeat A/からロックに見つわいから時代		
and a second sec	Controlated Dreingte		20-59-ЖСС) () СКМ () СКМ ()		「次へ (N) >」を
2240-11.477 12	्राम्स्ट्रायहा ब्रह्म				マウス左クリックする。
Cougle disease	анусназа				
Norton 340 Prenter School	10-50-5 29-5				
(9 📁 🛙	🗐 🏒		4 A 般 Si Ø Ø (2015) Ø + ● ● ● ● ● 0:17 1000 + ● ● ● 1000 + ● ● ● ● ● 0:17 2014/07/29	





USER	Microsoft Word	95050-73		1	
Meresoft Outlock	Algh-performa Embedded Wo	CIMG1935	※Reness Fash Programmer (Unsupported Freeware Version) ファイル(ド) ジール(T) マイクロコントローラ(M) ヘルプ(H)	1	
4 1782-4-	Renesas Fash Programme		マイクロコントローラ: RV:200 Series (Little Endian) ジューザノテータリッア: 学校:(8) コーサブートリッア: 学校:32 クタス (70)		── 「参照(B)」を マウス左クリックする。
1997-0 1997-0	. 		スタート		
en e	Commonwell Decempt		正常終了 1 掲載デバイン: YO2015 Series (LimaEridan), 使用アール: DOMK, 掲載方式: 2 wire UART デバイスのアードンークンスを結婚す。		RX220 と PC(パソコン)とが、
2011年 1日-1月1月 1月1日-1月1月 1月1日-1月1日 1月1日-1月1日 1月1日-1月1日 1月1日-1月1日 1月1日-1月1日 1月1日-1月1日 1月1日-1月1日 1月1日-1月1日 1月1日-1月1日 1月1日-1月1日 1月1日-1月1日 1月1日-1月1日 1月1日-1月1日 1月1日-1月1日 1月1日-1月1日 1月1日-1月1日 1月1日-1月1日 1月1日			クリンケード線化・フルを包括スタ カックチード線化・フック線化えまり。ニトド ホーレートを344(HobaCX更見えま) ポーレートを344(HobaCX更見えま) 種数が成立すりよ		正常に接続された。
n coogle clinente	CIMCORE		出力//ネルの2070		
Norten 140 Pronter Silbion	1040-50-4 2015				
😤 I 🕧	ê 🍙 🛛	🖹 🏂		● A 般 S @ @ Const 2 + 會 口 (2014/07/29	





ローカルディスクの内容が、 表示される。

「WorkSpace」をマウス左クリッ クし、「開く (O)」をマウス 左クリックする。

USER Microsoft Word 29170-92	
Microsoft High-performa CIMG1935 Outlook Embedded Wo	▲77-112間< ●●●・■ « ローカルテイスク (c:) , WorkSpace , ・ 1-5 WorkSpace , ・ 1-5
ander-o-	
Command Biompt	
Image: Conjections Image: Conjections Image: Conjections <td></td>	
1 6 1 B	● A 線 響 ● ① *** 『 → ● 1 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●

USER	Microsoft Word	25405-9X		
Microsoft Outlook	High-performa Embedded Wo	CIMC1985		- 「Release」を
2012-9- 2012-9- 2012-9 2012	An eas Flath Programme.		E団・新しいフォルダー E目・ ドキュメント 名前 ビウチャ Debug ビラオ・ 2014(07)22 948 フナル フォルー シュージック ペホームクループ モーカル ディス	マウス左クリックする。
esti et esti discome			□ □-DU F>7.2 ④ DVD F>7.7 □ JU-707/b ⁻ +	
Nexton 360 December Edition		R	▲ A般留● D ent F + 資 ■ 15:22 23400/79	




USER N	Icrosoft Word	59389-93				
rosoft H	gh-performa	GIMG1985	Renesas Flash Programmer (Unsupported Freeware Version ファイル(F) ツール(T) マイクロコントローラ(M) ヘルプ() 		
-9- 1	tenesas Flash		マイガロコントローラ: RV200 Series (LittleEndian) マ ユーザ/データエリア: TEST1 mot ロ ユーザデートエリア	参照_(B) 参照_(R)		
9-9			スタート スタート			
S Sam	Commone Recently		正常終了 bFFFF001-bfFFF10FF サバル b0000100 bFFFFF01-bfFFFFFF サバル b0000100 #SNAT-20 X BNJAHB H	•		
9 26-12-12-12 2	CIMO PARA		Costs Fush - デージスロードマスは生め、rol キー 者 き込み が 死 アント ビザバ よす ビザバ よす			
n an the second			*	・ 出カパネルのクリア(C)		
Norton 360 - 61 center Séléce						
7 C	3	R			● A 般 😂 🥥 D 🗯 🖞 🖌 🔮 🛶 🖣	15:40

プログラムの書き込みモード → 実行モードの切り替え



[ボード操作手順]

「RX220 Base ボード」の POWER SW を OFF にする。



DIP SW の SW No.1 のみを、 「OFF 側」にする。

プログラムの実行 (動作)

プログラムがマイコンのフラッシュ ROM に書き込まれました。次は、書き込まれたプログラムを マイコンボード上で、実行(動作)してみましょう。

■RX220マイコンは、RESET時は、

低速オンチップオシレータ(LOCO)の周波数125KHzで動作します。

本プログラムも、周波数125KHzで動作しています。

「RX220 CPU ボード」は、多くのクロック発生回路を内蔵し、きめ細かい周波数、電圧の 低減化制御により、低消費電力化を実現することができます。 下記に、「RX220 CPU ボード」の内蔵するクロック発生回路の種類について、纏めてみました。

■「RX220 CPU ボード」クロック発生回路の種類:

- 1、メインクロック発振器・・・・20MHz(水晶振動子)
- 2、サブクロック発振器・・・・32.768KHz(水晶発振子)
- 3、高速オンチップオシレータ (HOCO)・・・32MHz/36.864MHz/40MHz/50MHz
- 4、低速オンチップオシレータ(LOCO)・・・125KHz(RESET時は、本、オシレータが起動する)
- 5、IWDT 専用オンチップオシレータ・・・・125KHz



プログラムの作成→プログラムの書き込み→マイコンボードの動作(実行)までの、 操作手順は、以上で完了です。

[コラム]

■「第1章」の補足説明:

「RX220 CPU ボード」に、作成したプログラムを書き込む場合、2通りの方法があります。 第1章のパソコンと USB-シリアルケーブル書き込み(ブートモード書き込み)と、 第2章のパソコンと E1 エミュレータ(別売、M-06155)接続による書き込みです。 今回の「第1章」の操作手順について、要点のみを整理しておきたいと思います。

ます、パソコンと USB-シリアルケーブル(USB-シリアル変換ケーブルとも言う)を

「RX220 Base ボード」の D-Sub (CN4) に接続して(但し、「RX220 Base ボード」の POWER SW(SW2) は、必ず OFF にしてください)、JP1 の 1Pin-2Pin,3pin-4pin,5pin-6pin も、それぞれジャンパーピンで接続します。「RX220 CPU ボード」の DIP SW、1 pin,2pin を ON(右側にスライドすることにより、ブートモード設定(シリアル書き込み)します)にしてから、「RX220 CPU ボード」を、「RX220 Base ボード」上にセットします(逆差しに注意!)。AC アダプタ(5V、2A)を PSJ 5V 端子に接続して、「RX220 Base ボード」の LED1(赤)、「RX220 Base ボード」の LED3(赤)が点灯します)。

パソコンの Windows 画面にある「Renesas Flash Programme...」を、起動させ、一連の書き込み操作を 行っていきます。

書き込み処理が正常に終わったなら、「RX220 Base ボード」の POWER SW (SW2) を、OFF にします。 「RX220 CPU ボード」の DIP SW の 1Pin を OFF 側にし、「シングルチップモード」(実行) に設定します (DIP SW の 2pin は、ON/OFF どちら側でも可)。

「RX220 Base ボード」の POWER SW(SW2)を ON にすると、作成したプログラムが実行され、 「RX220 CPU ボード」が動作し、「RX220 Base ボード」の LED1(赤),LED2(緑)が交互に点滅を 繰り返します。

■「動作モード」の補足説明: 「RX220 CPU ボード」には、動作モード用のスイッチがあります。 これが、DIP スイッチです。プログラムの書き込みを行う時は、 DIP スイッチの 1pin,2pin を ON 側にします。 このモードを、「ブートモード」と言います。

プログラムを実行させる場合には、DIP スイッチの 1Pin を OFF 側します (2Pin は、ON/OFF どちらでも可)。 このモードを、「シングルチップモード」と言います。

注、モード切替えをする場合には、「RX220 Base ボード」の POWER SW を 必ず OFF にしてから、「動作モード」の SW 切り替えをおこなってください。

「ユーザブートモード」のモードもありますが、特殊な使用方法のため、 説明は省略致します。 ご使用に当たっては、ルネサスエレクトロニクスが発行する、

「RX220 グループユーザーズマニュアル (ハードウェア編)」を、ご参照ください。

■「JP1」の補足説明: 「RX220 Base ボード」には、JP1 (1-2,3-4,5-6)のピンヘッダがあります。 JP1 の 1-2 をジャンパーピンで接続すると、LED1(赤)と接続します。 ジャンパーピンを外すと、ポート H の 0 bit 目 (CN2-15)が開放状態となります。 JP1 の 3-4 をジャンパーピンで接続すると、LED2(緑)と接続します。 ジャンパーピンを外すと、ポート H の 1 bit 目 (CN2-16)が開放状態となります。 JP1 の 5-6 をジャンパーピンで接続すると、入力 SW(SW1)と接続します。 ジャンパーピンを外すと、ポート H の 2 bit 目 (CN2-17)が開放状態となります。 LED1(赤)、LED2(緑)は、出力機能のチェック端子用として、入力 SW (SW1)は、 入力機能のチェック端子用として、活用することが出来ます。 ジャンパーピンを外すと、汎用 I/O ポートとして、使用することが出来ます。



■「低速オンチップオシレータ(LOCO)」の補足説明:

「RX220 CPU ボード」を「シングルチップモード」で動作させると、 パワーオンリセット回路が自動的に働き、「RESET」(初期化)されます。 その時のクロック発生回路は、「低速オンチップオシレータ(LOCO)」が 優先的に選択され、周波数125KHzで、「RX220 CPU ボード」が動作 します。(この「RX220 CPU ボード」のプログラムも、LOCO125KHzで動作しています) メインクロック20MHz 及び サブクロック32.768KHzで、 動作させる場合は、LOCO起動動作(125KHz)から、メインクロック動作 及び サブクロック動作に切り替わるように、RX220マイコンの関連コントロールレジスタを プログラム上で操作して、クロック発生回路を切り替えなければなりません。 ご使用に当たっては、「ルネサスエレクトロニクス」が発行する、 「RX220 グループユーザーズマニュアル(ハードウェア編)」をご参照ください。

●尚、「D、第4章」に、メインクロック20MHz、 サブクロック32.768KHzへの遷移プログラムについて、記載しております。



B、第2章:

E1 エミュレータ使用による「TEST1」プログラムの書き込み操作方法について、 説明致します。(E1 エミュレータは、弊社通販コード M-06155にて購入出来ます)

注)、電源(5V)は E1 エミュレータから印加されますので、 必ず、「RX220 Base ボード」の POWER SW は、OFF としておきます(厳守!)。 尚、USB ポートは、十分な電流供給能力のあるものをお使いください。

「RX220 Base ボード」の「E1 Emulator (CN3)」に、E1 エミュレータの コード(14本)を接続します。E1 エミュレータの1 Pin は赤色コードですので、 Pin 番号を間違わないように接続します。(コネクタの逆差しに、注意してください)

E1 エミュレータの ACT 側にはミニ USB、PC 側には、USB の形状を接続します。 「RX220 CPU ボード」の、「DIP」 スイッチは、1 Pin は OFF 側、 2 Pin は ON,OFF どちらでも可です(「シングルチップモード」にする)。



USER M	ilcrosoft Word	CIMG1943			
Microsoft Hi Outlook Er	gh-performa mbedded Wo	CIMC1982			Ranasas Flash Programma
₩ ⊐>¥⊐-9- ;	Renesso Plasio Regisamente	通知 2010 - シロート 2019 ト	43028		を、マウス左クリックする。
() ネットワーク)) All	199025-92	 新しいワークスペースの作成気(W) Basicモード(E) Fullモード(F) 新回応用したワークスペースを閉((L) 		
al sea			CVUserVVSERWApDDeVMLcoalWaressar Flash ProgrammerW28A1WWNnvs ● 作品が満みのワーウスペースを描述(00)		
	Command Disampt				「トミンス」」両去が
Soogle clinome					「よりこそ!」 画面が 表示され、「次へ (N)」 を
Norton 360 Premier Scillout	CINACUSES				マウス左クリックする。
🥎 🤅	2	×R		● A 报 🐸 🥥 🕐 🕬 🖉 🔺 🛱 📆 🕴 11:29 2014/07/30	





このメッセージが表示されたら、 「OK」をマウス左クリックする。

USER	Microsoft Word	CIMG1943			
Microsoft Outlook	High-performa Embedded Wo	CIMG1982	≰ Renesss Flash Programmer (Unsupported Freeware Version) = □ × ファイル(F) ジール(T) マイクロコントローラ(M) ヘルプ(H)		「EI」を選択する。
N 1781-9-	Renesas Flash Programme	18-10 - ショート ガント	マイクロコントローラ: 通信方式		
1 1987 - 0		99025-92	16987-6471 E		「次へ (N) >」を
2000	21 (Method				マワス左クリックする。
2240-1275 12	Communit Prompti		(京(8) 法40年 495世年		
n couge discone	enerenten a				
Norton Set Prenter Edition	aterown				
(2)	é 🤉	F R		● A 段 😂 🥔 ② 🖓 ன + 🔐 🌠 4 12:09 2014/07/30	





「ピンの出力」は、io2,io3、 「ピンのレベル(High)」は、 io2にチェックマークを付ける。



マウス左クリックする。







RX220 ボード (RX220 CPU ボード+RX220 Base ボード)

「VCC」が点灯

「Power SW」が OFF であるこ とを確認する(厳守)。 Power SW が OFF でも、E1 エミ ュレータからの電源印加(5 V) により、LED1(赤),LED3(赤) が点灯する(ボードの通電状態を 表示している)。

USER	Microsoft Word	CIMG1943		
Microsoft	ange-performa	210 CIMO1932	愛 Reness Fash Programmer (Unsupported Freeware Version) ー □ × ファイル(F) ジール(T) マイクロコントロー与(M) ヘルフ(h)	
	Renesas Flash	HUHO- SH-F	マイクロコントローラ: 連歩方式 23	
1 7415-0	Rrogramme	9995 1995 1992-12		
8			デバイス違訳 RX200 Series (LinteEnden) -	
2010-21M	pictuliti		建物可能分子/17/2種均衡 - 2 デ/インコード 製造コード 7/ARPC003 Series (LiftleEndum)7/BR/R0200 Series (BigEndum)	
	CENCLESS		2004+4000/A0	
Norten 360 Promiler Edition	elimeniaisi	R	●【 人類 登録 ① 500 ピー 音 12 (1434	マワス左クリックする。





USER	Microsoft Wore	eimerasa		
Microsoft Outlook	Algh performs. Emicoded Wo.	CIMO1943		
1721-9- 1935-9-9 1935-9-9	Renosas Flash Programmer V2.61 	0 CHARLOSP 10-10 >>-+ 20-1-+ 20-1-+ 20-1-+ 20-1-+ 20-1-+	##2.5.1 ##2.6.1 ##2.6.1 ##2.6.1 ##2.0 ##2.7 ##2.0 ##2.0 ##2.0 ##2.0	「完了」を
Dolar Chone	Command Drampt CEL	1 (1946-1955)	(反20) Generic 7/17人の何為べ合わ世界支 7しました 出力パイスの何為べ合わ世界支 7しました	マウス左クリックする。
	e awaraan	R		행 I III 100 100 - III 15:05 2014/07/20





「ユーザ/ データエリア」にチェ ックマークを付け、「参照... (B)」 を、マウス左クリックする。

- 接続が成功しました。

USER	Microsoft Word	CIMG1953			
Microsoft Outlook	Aligh-performs Embedded Wo	CEMG 1943	 スアイルを整く () ● () ● () ● () ● ンピューター + ローカル ディスク (C:) + 	◆ 4g □ ーカル·ディスク(C)的建築 P	「ローカルディスク(C)」を
2021-9- 2021-9- 20117-9 2011	Vaces Fash Programmer Vaces	200 20079322 20040-203-14 2004 2004	歴史 ● 新しんりフォルダー) ドキュメント 2冊 ● ビラオ ● Figitu ● ビラオ ● Figitu ● ビラオ ● Hetropy ● スピューター ● ローフル・ダイズ ● ローフル・ダイズ ● コンピューター ● ローフル・ダイズ ● ローフル・ダイズ ● コンピューター ● ローフル・ダイズ	日・ 2 0 天和日本 地域 2014/02/28 8-49 2014/02/28 8-49 2014/02/28 8-49 2014/02/28 8-49 2014/02/28 9-58 2014/02/29 1-59 2014/02/29 1-59 2014/02/29 1-59 2014/02/29 1-59 2014/02/29 1-59 2014/02/28 1-59 2014/02/28 2014/02/28 2014/02/28 2014/02/28 2014/02/28 2014/02/28	マウス左クリックする。
Source Store	Campanad Securge CINC 1956	040325	€ 000 F347 (・ 「プログラム・ファイル(*hex; ・) 副く(0) ・ 年アンセル	マウス左クリックする。





「Release」を マウス左クリックする。

(Debug 作業を行わないので、 「Release」ファイル選択で、OK)

USER	Microsoft Word	CIMG1953		
Microsoft Outlook	Aligh-performs Embedded Wo	CIMG1948	▲ファイルを務く	= 2
-9-24-9-	Residences Fibels	CEMG1982	建理 ・ 新しいフォルター	<u>語・日 日</u> 18月
101-0 101-0	V2.04 	1 110-20-17 29-17	■ ビラオ ■ TEST1 2014(07)27 9:47 ■ ビデオ ■ ビニーシック ■ ホームグループ	FDT4 Data File
S. Com	EL CHACHDAI	93403-40	● コンピューター ▲ ローカル ディス ローカル ディス	
2040-11.167 15	Command Stempt	CINALE SECURATES	DVD ドライブ () UL-/(ブル デ・・(ファイル(*.hex; +
coogle diseante	CIIMCHARE CIIMCHARE		B<(0)	*r>th
Norton 340 Prenter School	enerere			
1 6	é 🧊	K		● A 投 🐸 🥔 ⑦ rank 音 预 🕴 15:57

__ 「TEST1」を マウス左クリックする。

(今回の LED1 (赤)、
 LED2 (緑)の約1秒間隔で交互に点滅を繰り返すプログラムのファイル名が、「TEST1」です)



USER	Microsoft Word	CIMG1953		2
Microsoft	Algh-performe.		Renesss Flash Programmer (Unsupported Freeware Version) フライル(ノF) ツール(T) マイクロコントローラ(M) ヘルプ(H)	
outlook	Embedded Wo.		マイカロコントローラ、RX200 Series (LittleEndan) ビ ユーザ/データエリア: TEST1 mot 年間(1)	
1 797-0	Programme	N.	ロコヤランド10% コマンド: 書き込み フタート	
S.		2091+ ()))) ()))))))	正常終了 DrffFE000 - DrffFENFF サ/ズ 0x1000700	
	Sammajai	C16401936	ロンデオデオション 書を込みデータンス 書き込み参照 14 [Data Faila] こ 小江口 - 1515世んでした 者を読み時か (Jaila) 収録します	
e Soge direme	Checkenses		2860より之 出力パネルの2017(G)	
A Norfor S60				
rentertikku P		F R		● A 酸容 ● の Carls 2 - 會 切 4 18:58 2014/07/30

「正常終了」・「書き込みが完 了しました」の表示で、E1エ ミュレータによる、「TEST1」 ソースプログラムの書き込み が、正常にマイコンのフラッ シュ ROM に、書き込まれま した。 「RX220 Base ボード」の Power SW を、OFF 状態にしておきます。 E1 エミュレータと「RX220 Base ボード」が接続されているコード(14本)を、取り外します。 「RX220 Base ボード」の PSJ 5V 端子に AC アダプタ(5V±10%、2A)を接続します。 「RX220 Base ボード」の「DIP」 SW を、1 Pin が OFF 側(左側)に(2 Pin は、どちら側でも可) になっていることを確認します。(SWをスライドの際には、ピンセットか、精密マイナス(-)ドライ バの先で、SWのスライド操作を行うとスムーズに移動ができます)

「**RX220 Base** ボード」の POWER SW を ON にします。 ハードウェア的に、パワーオンリセット回路が働き、自動的にリセットされ、 LED1(赤)、LED2(緑)が、約1秒間隔で、交互に点滅を繰り返します。



以上で、「E1エミュレータ」によるプログラムの書き込み操作説明を、終わります。



C、第3章:

E1 エミュレータを使用した「デバッグ機能」操作方法について、説明します。 「デバッグ機能」を利用すると、リアルタイムで、プログラムの書き込みはもちろん、

- ●レジスタ値の参照、設定
- メモリ値の参照、設定
- C 言語変数の参照、設定
- プログラムの指定アドレスでのブレーク(ソフトウェアブレーク、オンチップブレーク)
- 指定した条件でのプログラムの停止(オンチップブレーク)
- プログラム実行履歴の参照(トレース)

と、多くのプログラムの「デバッグ機能」が可能となります。

「RX220 CPUボード」の「DIP」SWは、「1」PinをOFF側、

「2」Pinは、どちら側でも可です(「シングルチップモード」にします」。

RX220ボードの電源(5V)は、E1エミュレータから印加しますので、

「RX220 Baseボード」のPOWERSW (SW2)は、OFFとします(厳守)。

PC(パソコン)のUSB端子-E1エミュレータ-「RX220ボード (CPUボード+Baseボード)」を、それぞれ接続し、セットします。

USBポートの電流供給能力には、十分に余裕のあるものをお使いください。



R X 2 2 0 ボード (СРU ボード+Ваse ボード)

それでは、「デバッグ機能」を動かしてみよう。



Window 画面の 「High-performa... EmbeddedWo...」を、 マウス左クリックする。

「デバッグ機能」とは? ソフトウェアに於ける プログラムのバグ(誤りや 欠陥)を探して、正しく動作 するように修正する機能。



● TEST1 - High-performance Embedded Workshop フォイルド) NEE(E) 名を(V) プロシニクイドド ビルド(B) テバ	こう(1) 御神教部(1) シーン(1) つうていう(1) くういう(1)	— a X
	四朝後後 西南部四本 二 二 夜	20MH z を確認
Construct the Construct the Construct the Construct the Construct the Construct	コンフィグレーシュングロバティ MOU システム MOU アンフム MOU アン MOU アン MOU アン MOU アン MOU アン MOU RO MOU RO MOU RO M	「内蔵フラッシュメモリ書 き換え」 にチェックマークを 付ける
1 (yoodray) 1 (yoodray) 3 (yoodray) 5	アンゴム (1980) 「「「「「」」「シング・クシン」」 パン (1980) 「」「シング・クシン (1)最高校 (2)年の (2) (2) (2) (2) (2) (2) (2) (2) (2) (2)	「OK」をマウス左クリックする
CiWorkSpace Test J. Head / Ceture β Λ Test in Files 2 Λ Hears Λ CiWorkSpace Test J. Head / Λ Λ Test in Files 2 Λ Hears Λ	「「注意記録時にこのがイアログからたあまでしたい。」 F 法記録時代にのがイアログからたあまでしたい。 mt人 Version Control /	
7 2 2 Q 1 E	● 4.80 公司 (1) com #	- ¥e 173 4 2024(07720



 9. TEST! - High-performance Embedded Warkshop - ITEST ・ファイム(F) 単晶(E) 表示(Y) プロシェクト(F) ビルト 	80] 6) (7) (7) (7) (7) (7) (7) (7) (7) (7) (7	(1) フィンドフ(W) へいどブ(n)	- 0 X	 「デバッグ (D)」を マウス左クリックする
	戦闘ラバック(之)	Debug RK_E1_E20_S + SemonRK_E1_E20_S/C+ 21		
0 15 15 10 8 2 19 1 + 14 115 11 11 11	1 デバックセッション(1)…			
	デバッグの設定(2)		1	
TESTI III 4	NT CPUSUE + (U)			
B C source file	111 N 第行(G)			+
A state A	14. リセット後期行(E) Shift+F5 TTO: 12 フリー第行		-	•
E TESTIC 6 / CPU TVP	Street and	1/ 1/		「実行 (G)」を
B Download mod. 8 / NOTE :TH	S ARRAY (TROP)	10r (Ver.4.53) - 1/		
S C Dependencies 11 /	Ref Elazo C Ref (E) PC位置を表示(Y) Ctrl+Shift+Y			マウス左クリックする
Buble h 13 Bind tude "10 Bindstach 15 Bindstach 15 Bindstach Topedring Topedring Topedring Topedring Single	ef (1) ステップイン(1) F11 い) (1) (2) ステップオーパ(1) F10 ド10 ド10 ステップオーパ(2) Shift+11 ステップ(5) ステップモード(1) ,	a scenario and the tor-		
•] •] •] •] •] •] •] •] •] •] •] •] •] •	RTOS デバック ・		-1 -	
S . S C STIC	B.2005L. Hall			
5 III OT AL AT 21 21 0 0 0 0 ?	初期(比(<u>Z</u>)			
Connected				
	メモリの保存(点) メモリのペリファイ(E)			
	The second second			
Normal	 入 ダウンロード(W) アンロード(L) 			
		Defaulti desktop Read-write 1/48	1 INS 23:05	
		■ 1級 ● ● □	2014/07/30	





E1エミュレータの「VCC」電源の ON、OFF 操作について:



TEST1 - High-performance Embedded	Workshop - [TEST1.c]			- 0	
シ ファイル(F) 編集(E) 表示(V) プロ3	ジェクト(P) ビルド(B)	「デバッグ(D)」基本設定(U) ツール(T) 5	ネト(5) ウィンドウ(W) ヘルプ(H)	_ 8 %	
	-	同期テバッグ(Z)	Debug_R0_E1_E20_5 • SemionR0_E1_E20_S/1 • 71		
08 6 8 8 2 5 4 8	和 田 魏 田 科 田	デバッグセッション(目)			
1×1	5 2-2	デバッグの設定(D)		1	
TESTI	2	INT CPUDUEN H(M)	······	7	
G cource file	A FILE	¥1英行(G) F5	1	_	
1) intorec 5 1) reseture 6	A DESCRIPTION :) リセット機実行(E) Shift+F5	1		
a) abek c 7	This file is	国語アリー家の	(Ver.4.53). 1/		
a) vectble 10	A NOTE:THIS IS	目的カーションに加速であるという	1/		E1 エミュレータの VCC
Deventead mod.	/***************	会体を確定して 単行(用)…			
idefineh 14	#include lidefine	S. PORTERALLY COLORD-Y		_	の電源を、接続(ON)す
m stackacth 16	Hifdef _cplusplu	何 スティブ いの (11	temport that you use los		7 r
- il vecth 18	//_SINT ios_base:	$\overline{W}(\lambda,\tau) = \mathcal{F}(\tau \to t \underline{\mathcal{L}}) $ (10)	tossent when you use los		るには、「アハック(D)」
20 21	void main(void);	We way that the state of the			(k c (N)))
22 23	#ifdefceluselu extern{{	ステップモード(M)			\neg 「按於 $(IN)」 と、$
24	wold abort (vold);	RTOS デバック		+	マウス左クリックすろ
1 1		C YOYSLIMICH		<u> </u>	
	STID	- Time ()			
A OL OT AL AT 121 27 2 00 0	2	弹 接触(N)			
Connected		We deralling			
Flash memory writing OK		(本臣()の「(空)(立)」。			
Disconnected		SENDALIZATE -			
11 1 B. B. dr) Dahus (Ends Eles 1 1	Red to Blog 7 3 Marco 3	Tart) Varian Control /			
THE PART OF A CARANTY CONTRACT V	And a straight of the state of	The V. Samerana 1	Default1 desktop Read-write 1/48 1 INS		
🙊 🤌 🦷 🧥			4 A 10 10 10 10 10 10 10 10 10 10 10 10 10	0:45	
				2014/07/31	

CPU のリセット

ツールバーの「CPU リセット」ボタンを押すことで、CPU がリセットされ、リセット開始番地のプログラムをウィンドウに表示します。現在の PC が示す行は黄色のラインで表示されます。





■レジスタの内容表示:

現在動作中のレジスタの内容も、表示することができます。 リアルタイムでプログラムのレジスタ動作状況把握ができ、プログラムの評価に威力を発揮します。



プログラムの実行と停止

プログラムの実行には以下の種類があります。CPU をリセットしてから実行するには、「リセット後実行」 ボタンを押します。

ボードでの動作を確認できたら、「停止」ボタンを押してプログラムを停止させます。



そのほかのデバッグ機能

ここで紹介した機能のほかに、以下のようなデバッグ機能があります。これら機能については、E1のマニュアルおよびアプリケーションノートを参照してください。

- レジスタ値の参照、設定
- メモリ値の参照、設定
- C 言語変数の参照、設定
- プログラムの指定アドレスでのブレーク(ソフトウェアブレーク、オンチップブレーク)
- 指定した条件でのプログラムの停止(オンチップブレーク)
- プログラム実行履歴の参照(トレース)

■オンチップデバッギングエミュレータ E1 導入ガイド http://documentation.renesas.com/jpn/products/tool/apn/r20an0022jj0101_e1_intr.pdf

■オンチップデバッギングエミュレータ E1 トラブルシューティングガイド http://documentation.renesas.com/jpn/products/tool/apn/r20an0045jj0100_e1e20_trbl.pdf





(参考)

■プログラム集:

- 「RX220 Base ボード」の LED1(赤)、LED2(緑) に於いて、 LED1が1秒間点滅し終わると、LED2が1秒間点滅する、 この繰り返し動作を永遠に繰り返す。
 - まず、 #include"iodefine.h" を、下記プログラムの上に、追加記述してください。

追加記述する→ #include"iodefine.h" //#include "typedefine.h" #ifdef __cplusplus

下記プログラムを、void main(void) { この部分に記載する } #ifdef __cplusplus

long int t; (変数 t を定義する) PORTH.PODR.BYTE=0x00; (ポートHの4bit分(16進数)を"0"にクリアする) PORTH.PDR.BYTE=0xff; (ポートHの4bit分(16進数)を出力端子に設定) (While で{} 囲まれたプログラムを、無限に繰り返し実行させる) while(1){ PORTH.PODR.BIT.B0=1; (ポートHのObit目から"1"を出力する。"1"=5V) for(t=0;t<20000;t++); (0から19999までカウントUPさせると、約1秒となる。時間稼ぎ) PORTH.PODR.BIT.B0=0; (ポートHの0bit目から"0"を出力する。"0"=0V=GND) for(t=0;t<20000;t++); (0から19999までカウントUPさせると、約1秒となる。時間稼ぎ) PORTH.PODR.BIT.B1=1; (ポートHの1bit目から"1"を出力する。"1"=5V) for(t=0;t<20000;t++); (0から19999までカウントUPさせると、約1秒となる。時間稼ぎ) PORTH.PODR.BIT.B1=0; (ポートHの1bit目から"0"を出力する。"0"=0V=GND) for(t=0;t<20000;t++); (0から19999までカウントUPさせると、約1秒となる。時間稼ぎ) } =Bit 命令応用によるプログラムの作成 (今回、追加するメインプログラム)=

上記のメインプログラムは、Bit 命令使用により記述したプログラムです。

下記のメインプログラムは、Byte 命令使用により記述したプログラムです。

#include"iodefine.h" ← ①、プログラムの先頭に追加記載します。

②、下記メインプログラムを、{ }の中に記載していきます。

long int t; PORTH.PODR.BYTE=0x00; PORTH.PDR.BYTE=0xff; while(1) { PORTH.PODR.BYTE=0x01; for(t=0;t<20000;t++); PORTH.PODR.BYTE=0x02; for(t=0;t<20000;t++); }

=Byte 命令応用によるプログラムの作成 =

【コラム】

PORTH.PODR.BIT.B1=1 (BYTE 命令) PORTH.PODR.BYTE=0x01(Bit 命令) 2、「RX220 Base ボード」の LED1(赤)、LED2(緑)に於いて、
SW1をON 側にすると、LED1が1秒間隔で点滅を繰り返し(LED2は消灯)、
SW1をOFF 側にすると、LED2が1秒間隔で点滅を繰り返す(LED1は消灯)。
SW1のON,OFFを判断して、LED1なのかLED2なのかを選択し、点滅させるプログラム。
回路的には、SW1がON 側の時、"0"。SW1がOFF 側の時、"1"となります。

#include"iodefine.h"

```
long int t;
PORTH.PODR.BYTE=0x00;
PORTH.PDR.BYTE=0xfb;
while(1)
ł
if(PORTH.PIDR.BIT.B2==1)
{
PORTH.PODR.BIT.B1=1;
for(t=0;t<20000;t++);
PORTH.PODR.BIT.B1=0;
for(t=0;t<20000;t++);
}
else
ł
PORTH.PODR.BIT.B0=1;
for(t=0;t<20000;t++);
PORTH.PODR.BIT.B0=0;
for(t=0;t<20000;t++);
}
}
```

=Bit 命令応用によるプログラムの作成 =

【2、の回答:完成プログラム】

```
#include"iodefine.h" ← (今回、追加記載します)
```

//#include "typedefine.h"
#ifdef __cplusplus
//#include <ios>
//_SINT ios_base::Init::init_cnt;
#endif

// Remove the comment when you use ios
// Remove the comment when you use ios00-

void main(void);

```
#ifdef __cplusplus
extern "C" {
void abort(void);
}
#endif
void main(void)
{
        (変数 t を、定義する)
long int t;
PORTH.PODR.BYTE=0x00; (ポートHの4bit分(0bit~3bit)を、"0"にクリアする)
PORTH.PDR.BYTE=0xfb;
                    (ポートHの2bit 目は入力端子、0,1,3bit 目は、出力端子に設定)
        (While で{ } 囲まれたプログラムを、無限に繰り返し実行させる)
while(1)
{
if(PORTH.PIDR.BIT.B2==1)
                    (ポートHの2bit目の入力端子のデータが、"0"⇔"1"を判断させている)
                    (ポートHの2bit目(入力)が"1"の時、処理は、本 { } 内を実行する)
Ł
PORTH.PODR.BIT.B1=1;
                    (ポートHの1bit目から"1"を出力する。"1"=5V)
for(t=0;t<20000;t++);
                    (0から 19999 までカウント UP させると、約1秒となる。時間稼ぎ)
                    (ポートHの1bit目から"0"を出力する。"0"=0V=GND)
PORTH.PODR.BIT.B1=0;
for(t=0;t<20000;t++);
                    (0から19999までカウント UP させると、約1秒となる。時間稼ぎ)
}
                    (ポートHの2bit 目(入力)が"0"の時、処理は、本 { } 内を実行する)
else
{
PORTH.PODR.BIT.B0=1; (ポートHのObit目から"1"を出力する。"1"=5V)
for(t=0;t<20000;t++);
                   (0から19999 までカウント UP させると、約1秒となる。時間稼ぎ)
                   (ポートHのObit目から"0"を出力する。"0"=0V=GND)
PORTH.PODR.BIT.B0=0;
                   (0から19999までカウントUPさせると、約1秒となる。時間稼ぎ)
for(t=0;t<20000;t++);
}
}
}
#ifdef __cplusplus
void abort(void)
{
}
#endif
```

■朱記の部分が、今回追加したメインプログラムの部分です。

【コラム】

I/O の入出力を指定するには、**PDR(ポート方向レジスタ)**を使う。 I/O の出力データを格納するには、**PODR(ポート出力データレジスタ)**を使う。 I/O の端子の状態を反映するには、**PIDR(ポート入力データレジスタ)**を使う。

■「RX220 開発ソフト」(無償評価版):

「RX220 開発ソフト」無償評価版(有効期間90日間)は、今回添付の開発ソフト以外にも、「ルネサスエレクトロニクス」のホームページ上から、ダウンロードすることができます。

●IDE、Compier(「RX220 用 C コンパイラ」に対応)に関しては、 【無償評価版】RX ファミリ用 C/C++コンパイラパッケージ V.1.02 Release 01 を ダウンロードしてインストールしてください。 http://japan.renesas.com/products/tools/evaluation software/downloads.jsp

●デバッグ(「E1 エミュレータソフト」に対応)に関しては、 E1 エミュレータを使用する場合 以下の URL から RX E1/E20 エミュレータデバッガ V.1.03.00 を ダウンロードしてインストールしてください。 <u>http://japan.renesas.com/request?SCREEN_ID=ViewGRSDownloadSearch&EXECUTE_ACTION=searc</u> <u>h&LAYER_ID=2964,167804</u>

●RX220 への書き込み(「RX220 用フラッシュライタソフト」に対応)は、 フラッシュ開発ツールキットをダウンロードしてインストールしてください。 <u>http://japan.renesas.com/request?SCREEN ID=ViewGRSDownloadSearch&EXECUTE ACTION=searc</u> <u>h&LAYER ID=1050,167804&CATEGORY ID=1</u>

D、第4章:

「RX220 CPUボード」に搭載の RX220 シリーズのマイコンは、きめ細かな低消費電力化を実現するために、 ストップ、スタンバイモードの他に、幾つかのクロック発生回路を内蔵しています。

■RX220シリーズ「クロック発生回路」の種類:

1、メインクロック発振器・・・・20MHz(水晶振動子)

- 2、サブクロック発振器・・・・32.768KHz(水晶発振子)
- 3、高速オンチップオシレータ (HOCO)・・・32MHz/36.864MHz/40MHz/50MHz
- 4、低速オンチップオシレータ(LOCO)・・・125KHz(RESET時は、本、オシレータが起動する)
- 5、IWDT 専用オンチップオシレータ・・・・125KHz

●「RX220 CPU ボード」の RESET 後は、「低速オンチップオシレータ(LOCO)」周波数125KHz 優先で動作します。

そこで、メインクロック20MHz、サブクロック32.768KHz、高速オンチップオシレータ(HOCO) 、IWDT専用オンチップオシレータも選択切り替え出来るようにしなければなりません。

切り替え操作は、プログラムで行わなくてはなりません。





●今後、メインクロック、サブクロック他を起動させる場合には、 上記の紫文字のプログラム全てを、追加する必要があります。

それでは、LOCO125KHzから、メインクロック20MHzに切り替えるプログラムを、 記載することにします。

■前回の「TEST1.c」のプログラムを、下記内容に変更してください。

下記プログラムのシーケンスは、(品名:TEST1. c)

「RX220 Base ボード」の LED 1 (赤)、 LED 2(緑)を、メインクロック 2 0 MH z を使い、 1 6 bit タイマ機能で、正確に 1 秒の交互点滅を繰り返すプログラムです。

下記の黒文字は、「High-performance Embedded」が自動的に作成された部分です。 緑文字は、1秒タイマと入出力端子の制御を行っています。 ピンク文字は、LOCO125KHz からメインクロック 20MHz に、 切り替えるためのプログラムです。緑とピンク色は、今回追加します。

```
/*
                                                         */
/*
  FILE
             :TEST1.c
                                                         */
/*
  DATE
             :Wed, Aug 11, 2014
                                                         */
/*
  DESCRIPTION :Main Program
                                                         */
  CPU TYPE
              :RX220
/*
                                                         */
/*
                                                        */
/* This file is generated by H.Terashita
                                                        */
/*
 NOTE: LED1 秒点滅(メインクロック 20MHz).
                                                         */
                                                        */
/*
           この緑色は、ヘッダーファイルです。
                           今回追加します。
#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
                                             この緑色は、16bit コンペアマッチタイマ(CMT)
void main(void);
                                             使用にて正確な1秒タイマを作ります。
#ifdef __cplusplus
                                             今回追加します。
extern "C" {
void abort(void);
}
#endif
void timer(int msec){//10MCLK
      int cnt = 0;
      MSTP(CMT0)=0;
                                        //Wakeup CMT0, CMT1
      CMT0.CMCOR = 1249;
                                        //8msec
      CMT0.CMCR.WORD = 0x0000;
      CMT.CMSTRO.BIT.STR0 = 1;
      while(1){
             while(CMT0.CMCNT != 0x00);
             cnt++;
             if(cnt == msec)
                    cnt = 0;
                    CMT.CMSTRO.BIT.STRO = 0;
                    break;
             }
             while(CMT0.CMCNT == 0x00);
      }
}
                                      このピンク記述部分が、メインクロック
                                      20MHz に切り替えるプログラムです。
void main(void)
                                      今回追加します。
```

66

{	
SVSTEM PRCR WORD - 0xo50b	リクロックソース選択の保護の解除
$\frac{1}{10000000000000000000000000000000000$	11/11/2/2/2/2017度の府防
$\frac{1}{3151} = \frac{1}{200000000000000000000000000000000000$	川朝什て一下の設定。低陸/十座
1/5151EW.SCKCK.LONG - 0x00001412,	川マーユノル参照。
UCNOMEN COLOR DIM DOLD 0.00	リーーマックロ
//SYSTEM.SUKUR.BIT.PUKD = 0x00,	バマニュアル参照。 パー・マンクロ
SYSTEM.SCKCR.BIT.PCKB = 0x01;	11マニュアル参照。
//SYSTEM.SCKCR.BIT.BCK = 0x00;	//マニュアル参照。
SYSTEM.SCKCR.BIT.ICK = 0x00;	//マニュアル参照。
//SYSTEM.SCKCR.BIT.FCK = 0x00;	//マニュアル参照。
/*クロックソースの選択*/	
SYSTEM.SCKCR3.WORD = $0x0200;$	//大元のクロックにメインクロックを使用する。
//SYSTEM.SCKCR3.WORD = 0x0300;	大元のクロックにサブクロックを使用する。
/*クロックの元栓の設定*/	
SYSTEM.MOSCCR.BYTE = 0; $// \checkmark \checkmark \checkmark$	クロック発振器動作
SYSTEM.SOSCCR.BYTE = 0; //サブク	ロック発振器動作
//SYSTEM.LOCOCR.BYTE = 1; //LOCO	動作
//SYSTEM_HOCOCR_BYTE = 1; //HOCO	動作
//SYSTEM ILOCOCR BYTE =1: //IWDT	車用オシレータ停止
PORTH PODR BYTE=0x00:	この緑色は、ポート日の牌子状能の設定です
PORTH PDR BVTF-0vff	この緑色は、か「日の畑」状態の設定です。
= 1.12(1)	今回迫加します。
while(1)	
PORTH.PODR.BIT.B0=1;	1000=1 秒
PORTH.PODR.BIT.B1=0;	- (例えば、2000 にすると2秒となる)
timer(1000);	この緑色は、今回追加します。
PORTH.PODR.BIT.B0=0;	
PORTH.PODR.BIT.B1=1;	
timer(1000); //timer(ミリ秒)	
}	
)	
}	
, #ifdof enluenlue	
#indercprusprus	
vola abort(vold)	
ł	
`	
}	
)	

次のシーケンスは、リアルタイムクロック(RTC)を使用したシーケンスとなります。 (品名:TEST2.c)

■シーケンス: サブクロックの32.768KHzを使用し、その64Hzカウンタの 2HzビットをLEDに反映させる事でLEDを点滅させるシーケンスの プログラムです。結果的に、LED1(赤)、LED2(緑)を 0.5秒間隔で、交互に点滅表示させることになります。

/**************************************		
/*	*/	I
/*	FILE :TEST2.c *	/
/*	DATE :Wed, Aug 20, 2014 */	/
/*	DESCRIPTION : Main Program */	/
/*	CPU TYPE :RX220 */	/
/*	*/	1
/*	This file is generated by H.Terashita */	
/*	NOTE:リアルタイムクロック(RTC) NO.1	*/
/*	*/	1
/**************************************		



```
/*クロックとカウントモードの設定*/
//RTC.RCR3.BIT.RTCDV = 0x01; //低 CL ドライブ能力 RTCDV がヘッダファイルに存在しない
```

RTC.RCR3.BIT.RTCEN = 1; //サブクロック発振器動作開始

RTC.RCR2.BIT.START = 0;//RTC カウントストップ while(RTC.RCR2.BIT.START != 0);

//RTC.RCR2.BIT.CNTMD = 1;//バイナリカウントモード CNTMD がヘッダファイルに存在しないため、OR で代入する。 RTC.RCR2.BYTE |= 0x80;//バイナリカウントモード

RTC.RCR2.BIT.RESET = 1;//RTC カウンタのリセット while(RTC.RCR2.BIT.RESET == 1);//リセット完了の待ち

```
/*時刻の初期設定*/
RTC.RCR2.BIT.START = 0;//RTC カウントストップ
while(RTC.RCR2.BIT.START == 1);
RTC.RCR2.BIT.AADJE = 0;//自動補正機能禁止
```

```
RTC.RCR2.BIT.START = 1;//RTC カウントスタート
while(RTC.RCR2.BIT.START == 0);
```

```
RTC.RCR1.BYTE = 0x08;//RTC の全割り込み禁止、RTCOUT は 64Hz を出力
//RTC.RCR2.BIT.RTCOE = 1;//RTCOUT 出力許可
```

```
PORTH.PODR.BYTE=0x00;//LED ポートゼロクリア
PORTH.PDR.BYTE=0xff;//LED ポート初期設定
```

```
while(1)
{
t = RTC.R64CNT.BIT.F2HZ;//tに2Hzビットの値(1bit)を格納する
PORTH.PODR.BIT.B0=t;//tの値をそのままLEDに反映させる
PORTH.PODR.BIT.B1=(~t);//tの値にNOTをかけて、LEDに反映させる
}
}
```

```
#ifdef __cplusplus
void abort(void)
{
}
```

```
#endif
```

次に、リアルタイムクロック(RTC)を使ったカレンダー機能のシーケンスプログラムです。 (品名:TEST3.c) RESET(リセット)後、1秒のLED1(赤)が60回点滅(点灯は30回)すると、 LED2(緑)の1分が点灯します。LED1(赤)が120回点滅(点灯は60回)すると、 LED2(緑)の2分が消灯します。継続して動作していきます。LED2(緑)は、 奇数回で点灯、偶数回で消灯となります。本プログラムは、月、年まで管理できます。

```
/*
                                       */
/*
 FILE
         :TEST3.c
                                       */
         :Wed, Aug 20, 2014
                                       */
/* DATE
/* DESCRIPTION :Main Program
                                       */
/* CPU TYPE
          :RX220
                                       */
/*
                                       */
/* This file is generated by H.Terashita
                                       */
/*
 NOTE:リアルタイムクロック(RTC)
                       NO.2
                                        */
/*
                                       */
```



/*クロックとカウントモードの設定*/

//RTC.RCR3.BIT.RTCDV = 0x01; //低 CL ドライブ能力 RTCDV がヘッダファイルに存在しない RTC.RCR3.BIT.RTCEN = 1; //サブクロック発振器動作開始

RTC.RCR2.BIT.START = 0;//RTC カウントストップ while(RTC.RCR2.BIT.START != 0);

RTC.RCR2.BYTE |= 0x00; //カレンダーカウントモード

RTC.RCR2.BIT.RESET = 1;//RTC カウンタのリセット while(RTC.RCR2.BIT.RESET == 1);//リセット完了の待ち

/*時刻の初期設定*/
RTC.RCR2.BIT.START = 0;//RTC カウントストップ
while(RTC.RCR2.BIT.START == 1);
RTC.RSECCNT.BYTE = 0; //秒レジスタの初期値
RTC.RMINCNT.BYTE = 0; //分レジスタの初期値
RTC.RHRCNT.BYTE = 0; //時レジスタの初期値
RTC.RWKCNT.BYTE = 0; //暗日レジスタの初期値
RTC.RDAYCNT.BYTE = 0; //目レジスタの初期値
RTC.RMONCNT.BYTE = 0; //目レジスタの初期値
RTC.RYRCNT.WORD = 0; //年レジスタの初期値

//RTC.RSECCNT.BYTE = 0;//秒レジスタの初期値 //RTC.RSECCNT.BYTE = 0;//秒レジスタの初期値

RTC.RCR2.BIT.AADJE = 0;//自動補正機能禁止

RTC.RCR2.BIT.START = 1;//RTC カウントスタート while(RTC.RCR2.BIT.START == 0);

RTC.RCR1.BYTE = 0x08;//RTC の全割り込み禁止、RTCOUT は 64Hz を出力 //RTC.RCR2.BIT.RTCOE = 1;//RTCOUT 出力許可

PORTH.PODR.BYTE=0x00;//LED ポートゼロクリア PORTH.PDR.BYTE=0xff;//LED ポート初期設定

while(1)

{ t1 = RTC.RSECCNT.BYTE & 0x01;// 秒の1桁目の最下位ビットを格納する t2 = RTC.RMINCNT.BYTE & 0x01;// 分の1桁目の最下位ビットを格納する

PORTH.PODR.BIT.B0=t1;// 秒の LSB を LED に反映させる PORTH.PODR.BIT.B1=t2;// 分の LSB を LED に反映させる }
#ifdef __cplusplus
void abort(void)
{
}

#endif

以上で、第4章を終わります。



以上で、説明を終わります。

