Rev 1.00.00

ハードウェアブレーク(PBC)機能なしMCU用のRAMでデバッグする場合の説明

- 1. 対象 MCU
 - ・H8/300H、H8Sシリーズ、H8SXシリーズ、SH-2シリーズが対象になります。
- 2. 機能
 - ・BSC (バスステートコントローラ) による拡張 RAM でのデバッグになります。
 - ・PBC/UBC 無しタイプの CPU 品種でもプログラムメモリが RAM の場合、C ソース/Asn ソー ス上に直接ソフトブレークが張れます。
- 3. デバッグ開始前の準備

3-1) BSC (バスステートコントローラ) 設定のスクリプトファイルを作成する。

例)ファイル名<H83068-BSC.log> // // H8 用(H8/3068F)バスステートコントローラ初期設定 // エリア 1:SRAM 512Kb 8bit 0x200000 // コメントは、コマンド実行ラインに記述しないで下さい。 // //バス幅コントロールレジスタ CS1 エリア:8bit(default) ABWCR 0xff <S ∥ポート1データディレクションレジスタ A7.A6.A5.A4.A3.A2.A2.A0 \leq S P1DDR 0xff #ポート2データディレクションレジスタ A15,A14,A13,A12,A11,A10,A9,A8 \leq S P2DDR 0xff //ポート5データディレクションレジスタ A19,A18,A17,A16 <S P5DDR 0xf //ポート8データディレクションレジスタ CS1 出力端子 \leq S P8DDR0x8 //コメント行

<S {8ビットアクセス} {レジスタ名} {データ} <SS {16ビットアクセス} {レジスタ名} {データ} <SL {32ビットアクセス} {レジスタ名} {データ} 【内部登録されているシンボルタイプ (ビット長)を使用する場合】 <SQ {8~32ビットアクセス} {レジスタ名} {データ} 3-2)作成したスクリプトファイルを実行させ確認をする。

【DEF8Kメニュー】

-<ファイル>―<スクリプト実行>

ブートロード(B) >	← → × ↑
	整理 ▼ 新しいフォルダー 目== ▼ □
	ProjectH83068_RAM ^ 名前 ^个 更新日時
シンボル読込み(Y)	Debug 2007/04/24 13:44
Makeファイルの指定(Z)	Release
	ProjectH83068_ROM
ベリファイ(V)	ProjectH836049 v <
アップロード(1)	ファイル名(N): H83068-BSC.log v Script File (*.log)
7776 1(0)	開く(O) キャンセ
アブ・川コートファイル設定(Δ)	
CPU設定読込み(S)	
CPU設定読込み(S)	
CPU設定読込み(S) CPU設定登録(R)	座 DEF0K インフォメーション
CPU設定読込み(S) CPU設定登録(R)	EF6K インフォメーション EFE 「N:#Usr4p¥448000¥Cpu411Test_R0M_RAM¥Hew4_RCH8v7_0_0¥ProjectH88888_RAM¥Debug¥H88888-BSC.1
CPU設定読込み(S) CPU設定登録(R) R8C-IDコード確認/変更(I)	DEF6K インフォメーション >DEF "N: WJsrAp¥AH8000¥DpuAllTest_ROM_RAM¥Hew4_RCH8v7_0_0¥ProjectH83868_RAM¥Debuş¥H83068-BSC. I \$2 ABNCR 0xff 00FEED202(w) HexFFF Dect255
CPU設定読込み(S) CPU設定登録(R) R8C-IDコード確認/変更(I)	DEF6K (ソフォメーション >DEF "N: WJsrAe¥4H8000¥DpuAllTest_ROM_RAM¥Hew4_RCH8v7_0_0¥ProjectH83868_RAM¥Debuş¥H83068-BSC. I X3 ABWCR 0xff 00FEE2021('0') HexFFF Dect255 WDEFE0211('0') HexFFF Dect255
CPU設定読込み(S) CPU設定登録(R) R8C-IDコード確認/変更(I) ユーザブログラム消去(E)	EEE DEF0K インフォメーション >EXE 「N:4UsrAp4H000014CpuAllTest_ROM_RAM9Hew4_RCH0v7_0_04ProjectH83088_RAM9Debug4H83088-ESC.1 なるABNCR 0xff のUFEED21:(*) HexFFF Dec7255 な3 PIDOR 0xff のUFEED21:(*) HexFFF Dec7255
CPU設定読込み(S) CPU設定登録(R) R8C-IDコード確認/変更(I) ユーザプログラム消去(E)	DEFak (>74×->=>> >EXE "N:WJsrAcM4H80004CpuAllTest_ROM_RAMMHew4_RCH8v7_0_0MProjectH83868_RAMMDebug4H83068-BSC.1 X5 ABRCR 0xff 00FEED21:(v) HexFFF Dec2255 X5 PDIOR 0xff 00e525 00FEED01:(v) HexFFF Dec2255 00FEED01:(v) HexFFF Dec2255 00FEED01:(r) HexFFF Dec2255 00FEED01:(r) HexFFF
CPU設定読込み(S) CPU設定登録(R) R8C-IDコード確認/変更(I) ユーザプログラム消去(E) スクリプト実行(L)	DEFaX (V/27X->>>> >DEC "N:WUsrAp#H8000WCpuAllTest_FOM_RAWHew4_RCH8v7_0_0WProjectH83068_RAWDebug#H83068-ESC.IX X ABRCR Vorf 00FEED201(w) HexFFF Dec=2255 XS FDICR 0.rff 00FEED01(w) HexFFF Dec=255 00FEED01(w) HexFFF Dec=255 00FEED01(w) HexFFF Dec=255 00FEED01(r) HexFFF Dec=255 00FEED01(w) HexFFF Dec=255 00FEED01(w) HexFFF Dec=255 00FEED01(w) HexFFF Dec=255
CPU設定読込み(S) CPU設定登録(R) RBC-IDコード確認/変更(I) ユーザプログラム消去(E) スクリプト実行(L) オフライン作業(O)	Example DEFax (*)7#X->>>> >EXE "N:WUS-ApVAH8000YCpuAllTest_ROM_RAMMHew4_RCH8v7_0_0VProjectH83068_RAMMDebugVH83068-ESC.It X5 ABRCR Ox Orf 00FEE2021(w) HexFFF Decr255 X5 PIDOR 0xff 00FEE201(v) HexFFF Decr255 X5 PIDOR 0xff 00FEE201(v) HexFFF Decr255 X5 PIDOR 0xff 00FEE201(v) HexFFF Decr255 X5 PEDOR 0xff 00FEE201(v) HexFFF Decr255
CPU設定読込み(S) CPU設定登録(R) R8C-IDコード確認/変更(I) ユーザブログラム消去(E) スクリプト実行(L) オフライン作業(O)	Emp DEFak (ソフォメーション SDE "N:3UsrAp44880004CpuAllTest_ROM_RAM44ew4_RCH6v7_0_04ProjectH83068_RAM4Debug4H83068-ESC.I X3 ABRICR Vorf 00FEER021:(n) Hex:FF Dec:225 X5 F10CR Vorf 00FEE001:(n) Hex:FF Dec:225 X5 P20CR Vorf 00FEE001:(n) Hex:FF Dec:225 X5 P20CR Vorf 00FEE001:(n) Hex:FF Dec:255 X5 P30CR Vorf X5 P30CR Vorf X5 P30CR Vorf X5 P30CR Vor
CPU設定読込み(S) CPU設定登録(R) R8C-IDコード確認/変更(I) ユーザブログラム消去(E) スクリプト実行(L) オフライン作業(O) オフライン環境設定(M)	DEFax インフォメーション >> >> >> >> DEFax インフォメーション >> Sec. Th: Nubrab StateB000 PCpuAll ITest_ROM_RAWFHee4_BCH8v7_0_0 PFrojectH83068_RAMFDebug#H83068-BSC. XS ABURC No.ff 00FEE0021(w) Hex:FF Dec=255 XS FDIOR Volf 00FEE001(w) Hex:FF Dec=255 XS FDIOR Volf 00FEE0015(w) Hex:FF Dec=255 XS FDIOR Volf

3-3) 拡張 RAM が正しく読み書き出来るか「メモリフィル」で確認する。

【DEF8Kメニュー】

<データ>---<メモリフィル/検索/サム計算>

🔤 DEF8K メ	モリフィル/検索/サ	けム計算	×
Address	0x200000		-
Size	0x80000	•	
Pattern	0x55	FIND/S	JC NOT C SUM
	Access	C short	C long
FILL	FIND	SUM	Cancel

200000	- ÷)×200000					•	X	Ŧ!	ノダ	`ン:	プ	•	ct	nar				-	16進	-	周期実	行回数 1	
00200	000:		55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	00	UUUUUUU	UUUUUUU			_
00200	010:		55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	100	UUUUUUU	UUUUUUU			
00200	020:		55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	100	UUUUUUU	UUUUUUU			
00200	030:		55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	100	UUUUUUU	UUUUUUU			
00200	040:		55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	100	UUUUUUU	UUUUUUU			
00200	050:		55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	JUU	UUUUUUU	UUUUUUU			
00200	060:		55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	Ш			1		
Go	Break	RstMon	Reset	t	W	in		R	eg		Wa	tc	h	S	ym		T	rac	e	Step	CTrac	CStep	DI	E

3-4)作成したスクリプトファイルを「CPU設定」に登録する。

【DEF8Kメニュー】

<オプション>—<CPU 設定>



$\leftarrow \rightarrow$	1 ProjectH830	> Debu	g v	Ō	。 Debugの検索	
整理▼	新しいフォルダー				8==	- 🔳 🌘
	ProjectH83068_RAM	^	名前	^	更新日時	9
			H83068	-BSC.log	2007/04/	/24 13:44
	Release ProjectH83068_ROM ProjectH836049					
	7アイル名(N): He	33068-BSC	log	~	Script File (*.log)	~

【スクリプトファイル登録による効果】

・ユーザプログラムのダウンロード時の、開始と終了後に登録された内容を実行します。

・【RstMon】と【Reset】を実施後、登録された内容を実行します。

3-5)「環境設定」を設定する。

【DEF8K メニュー】 <オプション>—<環境設定>



- 3-6) プログラム側の条件と準備
- ・スタートアップ関数は、必ず、ROM 側に配置して下さい。
- ・スタートアップ関数処理でスタックポインタ設定後、ソフトタイマ 200ms 以上を実装して下 さい。理由は、DEF8K より MCU リセット指示からモニタ起動(NMI)するまでにプログラム 未実装の RAM エリアに飛びプログラム暴走を防ぐために必要です。
- 1) ルネサス C の場合【resetprg.c】

#pragma section ResetPRG
//************************************
void Wait1ms(void) //1ms ソフトタイマー(25.000MHz)
{
long cnt;
cnt = 961; // 961*26=(25000)
while(cnt!=0) {} // 26 clock
}
// ***********************************
void SoftWaitNms(long ms)
while(ms != 0) {
Wait1ms0;
<pre>}</pre>
entry(vect=0) void PowerON_Reset(void)
Softwaltinms(200), パーーホイント 200ms Walt(ノート DF の場合特(二必要)
// Set_Imask_CC(_UBYIE/1),
_LN115010, // 円下空略
// 这十月吗
main()
}
,

【セクションの設定】





2) KPIT/Gnu-Cの場合【start.asm】



【セクションの設定】





3) IAR-Cの場合【cstartup.asm】

#define CODESEG STARTUP
RSEG CODESEG:CODE:NOROOT(1) PUBLIC ?cstart_init_sp
?cstart_init_sp:
MOVX #SFE(CSTACK), SP // 200ms wait
jsr @SoftWaitNms
// 省略
//wait1ms(void)
Wait1ms:
mov.w #3125,r0 // 25000/8(clock)
wait
dec.w #1,r0 //2 clock
nop // 2 clock
bne wait //4 clock
rts
//SoftWaitNms(long ms)
SoftWaitNms:
mov.l er0,er1
loop:
jsr @Wait1ms
dec.l #1,er1
bne loop
rts
//

【セクションの設定】 【lnk3068flh.xcl】

-P(CONST)INTVEC=0-3FF -P(CONST)FLIST=40-FF -P(CONST)STARTUP=800-BFF -P(CONST)DATA16_C=200000-27FFFF -P(CODE)CODE24=200000-27FFFF -Z(CONST)DATA8_ID,DATA16_ID,DATA32_ID=200000-27FFFF -Z(CONST)DIFUNCT= -P(CONST)DATA32_C=

-Z(CONST)CHECKSUM=200000-27FFFF



- 4. DEF8K でデバッグ
 - 4-1) アブソリュートファイルでのダウンロードを設定する。

【DEF8Kメニュー】 <ファイル>—<アブソリュートファイル設定>



- 1) シンボルコンバータスイッチの説明
 - 【-r】ディレクトリ情報を作成しない。
 - 【- i】 不整合な Inline 情報を削除する。
 - 【-m】重複モジュール情報をCソースにマージする。
 - 【-g】 スタティック変数をグローバル化する。
 - 【-s】 ラインシンボル情報をソートしない。
 - 【-f】使用インクルードファイルを CView に登録する。
 - 【-1】ローカル変数情報を作成しない場合はチェックします。
 - 【-x】ローカルの構造体シンボル情報を作成しない場合はチェックします。
 - 【-z】構造体メンバのシンボル情報を作成しない場合はチェックします。
 - ◎[-1] [-x] [-z]のオプションスイッチの使用目的は、シンボル数が制限数を超えてしまいグローバルシンボルを優先したい場合にチェックします。また、コンパイラ等のバージョンアップに伴い ELF/Dwarf 情報に不具合がありデバッグ作業が継続できなくなった場合の一時的な退避策として使用する。



4-2) プログラムをダウンロードする。

【DEF8Kメニュー】

<ファイル>--<ダウンロード>



1) ダウンロード方法の確認



4-3) ダウンロード後の DEF8K 画面

ĕı⊦	割込 [DI PC 000	00082		Vi	ew CPI	J H8/306	8 (EMC=	1)	Advanced	25.0	000×1	書认同数	00029TOP	0×000008	0080TT0	×002002E7	Target	
00824	P	owerON	Reset		-		Src C	Mix C	Asm	resetpr	K.C		- -	実行ステートシ	きょう しょうしん しょうしん しんしん しんしん しんしん しんしん しんしん	•			
00000824		78	ent r	y(vect=0) void	Powerl	DN_Reset	t (void))										1
		79	{																
0000082/		81	So	ftWaitNm	s(200);	11.	(ポ	イント	200ms	Wait(ブ-	- ト I/F	の場合。	必要)						
		82																	
00000000		83 .	// s	et_imask	_ccr((_	UBYTE)1);												
00000830		85		NITSCI()	;														
		86	// _0	ALL_INIT	();			// F	Remove	the comm	ent wh	en you	use glob	al class	object				
		87	<i>,,</i> ,							1/0 1 - 1			. / /	TH 1 (0		1/0	、 、		
		88 ,	// _l	NII_IULI	8();			// 6	nabie	1/U in t	ne app	licatio	n(both a	sim i/u a	nd hardwa	are I/U)		
		90 .	// er	rno=0;				// F	Remove	the comm	ent wh	en you	use erm	10					
		91 ,	// sr	and((_UI	NT)1);			// F	Remove	the comm	ent wh	en you	use rand	I()					
		92 .	// _s	1ptr=NUL	;			// F	Remove	the comm	ent wh	en you	use strt	.ok()					
		94	// На	rdwareSe	tup();			// F	Remove	the comm	ent wh	en you	use Haro	ware Set	up				
		95	// se	t_imask_	ccr((_U	BYTE));												
0000000		96		-0.															_
00000834		98	me	in();															
		99	// _0	LOSEALL();			// (Close 1	[/O in th	e appl	ication	(both SI	M I/O an	dhardware	e I/O)			
		100																	
		101 .	// _0	ALL_END();			// •	(emove	the comm	ent wh	en you	use glob	al class	object				
00000838	:	103	sl	eep();															
honnool		10.4	ι					_		00000000							- Linn	mlun i r	21 k
BPT 000	100000			×	ULK N				BPZ	100000000			×			I	//~ト'BP腹	歴 ツト	10-9
						• (な	U)	•	char		Ψ.	16進	<u> </u>	周期 実	行回数 1				
																			1
																			Ŧ
, Go B	reak	RstM	n R	eset	Win	Re	g 1	atch	Sym	Tra	ce	Step	CTrac	CStep	DI	E	I Int	ELS PI	JTCH
000008	24 1B	7												1	1				
																			Ŷ
. 10		- 1			~					Lor							F.e	- 0	

【RstMon】をクリックするとスタートアップ関数の先頭に View 表示が切り替わります。

1) 拡張 RAM エリアのプログラムソースに画面を切り替える

亨止 割辺	DI PC 00	0000824 🔲 周期 🗌 View CPU H8/3068(MC=1) Advanced 25.0000×1 書込回数00029TDP 0x0000080BDTT0x002002E7 Targ	et 🖕
200080	#61:	Src C Mi	K C Asm ProjectH83068_RAM.c 🔽 🗆 実行が上数 高速 💌	
	59	Ulong time;	dbsct.c	
	60	DOD DOD DVTF In Auto	intprs.c	
00200080:	60	BSC.BCR.BTTE [= 0x4;	ProjectH83068_RAM.c	
00200084+	62	PADDR = 0vf.	- KAM に割り	
0020000H.	64	THOON - UXT,		
00200092:	85	MSTCR.BIT. ITU = 0:	// ITU モジュール MSTPL4=0 ITUbmのかけ 付けたソースファ	
00200098:	66	ITU0.TCR.BIT.CCLR = 1;	// TGRA(cmp)	
002000A0:	67	ITU0.TCR.BIT.CKEG = 0;	// 立上がりェッジ イルを指定する。	
002000A6:	68	ITU0.TCR.BIT.TPSC = 2;	// \$/4	
002000AE:	69	ITU0.TCNT = 0;	// TCNT=0	
002000B4:	70	ITU0.GRA = 62500;	// (25000000/4)/62500=100(Hz)	
002000BC:	71	ITU.TISRA.BIT.IMIEA0 = 1;	// IMIEA0=1	
002000C0:	72	ITU.TSTR.BIT.STR0 = 1;	// TCNT0カウント動力作	
	73			
002000C4:	74	TmStart(0,10);	// 100msec	
000000044	75	LoopUnt = 0;		
002000CA:	70	time = 0;		
00200000.	78	epshle():		
00200144:	79	while(1) {		
002000D2:	80	if (TmUp[0] == ON) {		
002000DC:	81	TmStart(0,10);	// 100msec	
002000E2:	82	PADR.BYTE = ("time & 0×f);	
BP1 000000	100	V CLR V	C BP2 000000000 マロス ア バート* BP履歴 ソフ	トプレ
	-	▼ (なし)	▼ char ▼ 16進 ▼ ■ 周期 実行回数 1	
Go Brea	ak RstM	on Reset Win Reg Wat	ch Sym Trace Step CTrac CStep DI EI IntFlg	PUTO

4-4) ソフトブレーク設定画面を開く

い 割り	-タ(D) 実行 込 DI PC 000	(G) ブレーク(B) 割り込み(I) オプショ 000824 □ 周期 □ View CPU H8/306	ョン(O) ヘルブ(H) 88(EMC=1) Advanced 25.0000×1 書込回数(00028TOP 0×00000800B0TT0×002002E7 Target ●	•
200080	#61:	▼ Src C	Mix C Asm ProjectH83068_RAM.c 💌 🗆 実行ステート数 高速 🚬	
	59	Ulong time;		±
	60		DEF8K ソフトフレークの設定(RAM専用) X	-
00200080:	61	BSC.BCR.BYTE = 0×4;	// BCR EMC=1 C 除外 アドレス シンボル モード	
	62	D1000 - 0. (-	GLR ▼ CLR ▼ CLR C	
0020008A:	63	PADUK = Uxf;	// LED Port C SB2 000000000 V CLR V	
00200092+	64	WOTOD DIT ITU - 0.	// TTU T05-50 MCTPI C SB3 000000000 ▼ CLB ▼	
00200032:	00	ITIN TOP BIT COLP = 1.	// T(PA/cmp) C SP4 00000000 - 000 -	
002000000	87	ITUD TCR BIT CKEC = 0:	// 立上がりての app 100000000 ・ CLR ・	
002000A6:	68	ITU0.TCR.BIT.TPSC = 2:	// \$\$4 \$ \$B\$ 00000000 V CLR V	
002000AE:	69	ITUO.TCNT = 0;	// TCNT=0 C SB6 000000000 V CLR V	
002000B4:	70	ITU0.GRA = 62500;	// (25000000/4) C SB7 000000000 - CLR -	
002000BC:	71	ITU.TISRA.BIT.IMIEA0 = 1:	// IMIEA0=1 C SB8 000000000 V CLR V	_
002000C0:	72	ITU.TSTR.BIT.STR0 = 1;	// TCNT0/02/100/	
	73			
002000C4:	74	TmStart(0,10);	// 100msec	
	75	LoopCnt = 0;		
002000CA:	76	time = 0;	「ソフトブレーク設定」画面	
	77			
002000D0:	78	enable();		
0020014A:	79	while(1) {		
002000D2:	80	if (TmUp[0] == ON) {		
002000DC:	81	ImStart(0,10);	// 100msec	-
UU2UUUE2:	82	PADR.BYTE = (time & U	Uxf);	Ŧ
BP1 000000	000	V CLR V	C BP2 000000000 VCLR マ VトドBP履歴 ソフトフィレ	<u> </u>
	÷	▼ (なし)	▼ char ▼ 16進 ▼ ■ 周期 実行回数 1	
				±
				=
				Ŧ
Go Bre	ak RstMo	in Reset Win Reg Wa	atch Sym Trace Step CTrac CStep DJ EI IntF g PUT	CH
	0 CLR			•
	0 0211			0
Lort I				0 P
iart				
art				

1) ソフトブレーク設定の説明



2) Cview 画面で直接ブレークポイントの設定

鲈 割込	DI PC 000000824 T /BJKA View CPU H8/3068(EMC=1) Advanced 25.00	000×1 書込回数00029TOP 0×00000800B0TT0×002002E7 Target
200072	main 🔹 💽 Src O Mix O Asm ProjectH83068.	_RAM.c ▼ 「実行ステート数 高速 ▼
	52 50 // Anti-Anti-Anti-Anti-Anti-Anti-Anti-Anti-	DEF8K ソフトブレークの設定(RAM専用) ×
	54 // maio()	
	55 //##################################	C SP1 000200000 #211Project W2202(- 100 - 1
	56 void main(void)	C SP3 00020000 #01.Projectiosoc M0 •
00200072:	57 {	SB2 000200004 #/4:ProjectH8306₹ MU ▼
	58 static Uchar i;	C SB3 0002000DC #81:ProjectH8306₹ M0 ▼
	59 Ulong time;	C SB4 0002000F4 #84:ProjectH83068 MO -
	60	C SB5 000200100 \$\$9:ProjectH8306€ M0 ▼
00200080:	61 BSC.BCR.BYTE = 0x4; // BCR EMC=1	C SB6 00020015C #103:ProjectH830€ ▼ M0 ▼
00000004.		C SB7 000200172 #107:ProjectH8306 - M0 -
0020008A:	63 PADDR = 0X	C SB8 00020014E #119:Project H8306 - M0 -
00200092+	⁹⁴ KTCR BIT Cソースラインで直接	
00200098:	66 ITU0. TCR. B	0 110/20192019
002000A0:	67 ITWO.TCR.B ダブルクリックして設	
002000A6:	68 ITUO. TCR.B	
002000AE:	⁶⁹ ITU0.TONT 定/解除する。	
002000B4:	70 ITU0.GRA =	(2500=100(Hz)
002000BC:	71 ITU.TISRA.BIT.IMIEA0 = 1; // IMIEA0=1	
002000C0:	72 ITU.TSTR.BIT.STR0 = 1; // TCNT0#92N	動作
00000004	73 24 T-01-1/0 103-	
00200064:	74 Instart(0,10), 77 Toomsec	
002000CA:	76 time = 0:	
	77	
BP1 0000000	U C BP2 00000000	✓ ICLR ✓ // // 「 // * SP履歴 ソフトブレ
-	▼ (なし) ▼ char ▼	16進 🔽 同期 実行回数 1
· · · · ·		

3) ソフトブレークポイントまで実行

ファイル(F)	データ(D)	実行	(G) ブ	ノーク(B)	割り	込み(l)	オプション	<u>۱</u> (0) ۸	ルプ(H)											
停止	割込 DI	PC 0002	00080	□ 周期	٦V	ew CPU	H8/3068	(EMC=1)	Adva	inced 25.0	000×1 4		女00029	'OP 0×0	000080	OBOTTO×	:002002E	7 Tars	et 🖕	•
0200072	a ina. i	n		-		@ S	rc 🔿 M	ix C A	sm Pr	ojectH83068	_RAM.c	•	実行な	い数	高速	•				
	5	5 /	/*******	heteroid)	*CHORENEO #	****	****	******	*****	kokokokokokokokokok	_	🚥 D	F8K Y7	トブレー	クの設定	E(RAM	専用)		×	*
. 002000	72: 5	7 {	JTG man	(voru)							-	_ 除夕	・アドレ	ス	シンボ	いレ		-±	-	-
	5	8	stat	ic Ucha	r	i;						SB	0002	0080	#61:6	Project	H83061 -	MO	•	
	5	9		Ulona	٢.	time;					-	C SB:	0002	00C4	\$74:6	Project	H8306{ -	MO	•	
• 002000	30: 6	1	BSC.	BCR.BYT	E =	0x4; 🔥			// BC	R EMC=1		C SB	0002	OODC	#81:F	Project	H8306{ -	MO	•	
	6	2				7	$\overline{}$					C SB	0002	00F4	# 84:F	Project	H8306{ -	MO	•	
• 002000	BA: 6	3	PADD	R = 0×			`					C SB	0002	0100	\$89:F	Project	H8306{ -	MO	•	
. 002000	92: 6	5	MSTC	R.BIT.	步力	\$帯(立置、	でプ	ログ	ラム	PL4=0	C SB	0002	015C	\$103	Projec:	tH830(-	MO	-	
• 002000	98: 6	6	I TUO	.TCR.B	1-	÷ 1	×. =		+	+		C SB	0002	0172	#107:	Projec	tH830(•	MO	-	
• 002000/	AO: 6	7	I TUO	TCR.B	15	計上な	ど衣		いよ	-9 o	ッジ	O SB	0002	01AE	II 119:	Projec:	tH830(-	MO	-	
. 002000/	40. 0 AE: 6	9	ITU0	.TCNT =	0;				11	TCNT=0	-									
. 002000	84: 7	0	I TUO	.GRA = 6	62500	;			11	(25000000	/4)/6250	0=100	Hz)							
• 002000	BC: 7	1	ITU.	TISRA.B	IT.IM	IEA0 =	1;		11	IMIEA0=1										
• 0020000	50: 7		T THE	TOTO DI	T OTO	0 - 1.			- 11	TONTO #614	the De									
	7	3	ITU.	TSTR.BI	T.STR	0 = 1;			11	TCNT0 かりント	動作									
→ 002000	7 04: 7	2 3 4	ITU.	TSTR.BI	T.STR 0);	0 = 1;			//	TCNT0カウント Omsec	動作									
→ 002000	7 24: 7 7	2 3 4 5	ITU. TeSt Loop	TSTR.BI art(0,10 Dnt = 0;	T.STR 0); ;	0 = 1;			//	TCNT0カウント Omsec	動作									
→ 0020000	7 24: 7 7 CA: 7	2 3 4 5 6 7	ITU. TmSt. Loop time	TSTR.BI art(0,10 Cnt = 0; = 0;	T.STR 0); ; ;	0 = 1;			// 10	TCNT0カウント Omsec	動作									
→ 0020000	7 24: 7 7 CA: 7 7 20: 7	2 3 4 5 6 7 8	ITU. TmSt Loop time enab	TSTR.BI art(0,10 Cnt = 0; = 0; le();	T.STR 0); ;	0 = 1;			// 10	TCNT0カウント Omsec	動作									
 0020000 0020000 0020000 0020000 0020001 	7 24: 7 7 CA: 7 7 00: 7 4A: 7	2 3 4 5 6 7 8 9	ITU. TmSt Loop time enab while	TSTR.BI art(0,11 Cnt = 0; = 0; le(); e(1) {	T.STR 0); ;	0 = 1;			// 10	TCNT0カウント Omsec	動作									
 0020000 0020000 0020000 0020001 0020001 0020001 	7 24: 7 7 24: 7 7 20: 7 44: 7 22: 8	2 3 4 5 6 7 8 9 0	ITU. TmSt Loop time enab while	TSTR.BI art(0,1) Cnt = 0; = 0; le(); e(1) { if (TmU;	T.STR 0); ; ;	0 = 1; == ON)	{		// 10	TCNT0カウント Dasec	動作									
→ 0020000 0020000 0020000 0020000 0020000 0020000 0020000 0020000 0020000	7 24: 7 7 CA: 7 00: 7 4A: 7 02: 8	2 3 4 5 6 6 7 8 9 0	ITU. TeSt Loop time enab while	TSTR.BI art(0,1) Cnt = 0; = 0; le(); e(1) { if (TmU; ()	T.STR 0): ; ; p[0] DLR [0 = 1; == 0N)	{	C BF	// // 10 P2 00000	TCNT0カウント Dassec	動作		CLR		<u>_</u>		ħ~*°₿₽,₩	見歴 [ソ	ı⊦フ°レ	 * ・ ・
 → 0020000 → 0020000 → 0020000 → 0020000 → 0020000 → 0020000 → BP1 00 	7 24: 7 7 2A: 7 7 20: 7 4A: 7 02: 8 0000000	2 3 4 5 6 7 8 9 0	ITU. Test Loop time enab while	TSTR.BI art(0.1] Cnt = 0; = 0; le(); e(1) { if (TmU; []	T.STR 0); ; p[0] DLR	0 = 1; == 0N) 	.)	C BF	// 10 // 10	TCNT0 かう)ト Onsec	動作] CLR	✓	▼ 1		バート。BP相	匪刃	11-7*6	· · ·
→ 0020000 0020000 0020000 0020000 0020001 0020001 0020001 0020001 0020001 0020001 0020001 0020001	7 7 7 7 7 7 00: 7 7 7 00: 7 7 4 4 7 00: 7 7 00: 7 8 0000000	2 3 4 5 6 7 8 9 0	ITU. Inst Loop time enab while	TSTR.BI art(0,1) Cnt = 0; = 0; le(); e(1) { if (TmU; (0)	T.STR 0); ; ; P[0] DLR	0 = 1; == 0N) ▼ (<i>t</i> _R)	× ,)	C BF	// 10 // 10	TCNT0分分入 Dasec	動作 1 6進] CLR 「周期	▼]	▼ 回数 1		<u>ħ~ト*BP</u> 用	<u>夏歴</u> [ソ]	11-7*6	·
D02000 D02000 D02000 D02000 D02000 D02000 D02000 BP1 00	7 24: 7 7 2A: 7 7 20: 7 4A: 7 902: 8 00000000	2 3 4 5 6 7 8 9 0	ITU. ImSt Loop time enab while	TSTR.BI art(0,1) Cnt = 0; = 0; le(); e(1) { if (TmU; []	T.STR 0): ; ; p[0] DLR [0 = 1; == 0N) ▼	{ 	C BF	// 10	00000	動作 16進] CLR 「周期	▼ 実行	✓ ■数 1		<u>_ħ~卜*₿₽</u> 履	夏歴 [ソ	1 1 7 7	· · · · · ·
D020000 D020000 D020000 D020001 D020001 D020001 D020001 D020001 D020001	7 24: 7 7 CA: 7 7 00: 7 4A: 7 20: 8 0000000 CA: 8 0000000 CA: 7 7 8 0000000 CA: 8 0000000 CA: 8 0000000 CA: 8 0000000 CA: 9 CA:	2 3 4 5 6 6 7 8 9 0	ITU. Inst Loop time enab while	TSTR.BI' art(0.1] Cnt = 0; = 0; le(); e(1) { if (TmU; (0)	T.STR 0): ; ; ; [[] []	== 0N) → (<i>t</i> _k)	{ 		// 10 P2 00000 har	TONTO #224	動作 16進] [CLR] 周期	▼ 実行	回数 1		ħ~ト°ΒΡΠ I In	貢歴[ソ] ŧ F I ø	PIIT	
0020000 0020000 0020000 0020001 0020001 0020000 BP1 00 Go Go Co	7 24: 7 7 7 7 7 7 7 7 7 7 7 7 7 7	2 3 4 5 6 7 8 9 0 0	ITU. Inst Loop time enab whili	TSTR.BI art(0.1 Cnt = 0 = 0; le(); e(1) { if (TmU; v (0) et 1	T.STR D): ; ; P[0] CLR []	0 = 1; == 0N) • (/z) • (/z)	{ 	C BF	// 10 P2 00000 har Sym	TCNT0#924	動作 1 6 進 Step	CIra	CLR Altr	▼】 実行	▼ □鼓 1 DI		<u>ハードBP</u> 肩 I In	፼歴[ソフ tFlg	PUT	
0020000 0020000 0020001 0020001 0020001 0020001 0020001 0020001 0020001 000 000 000 000 000 000 000 000 000	7 24: 7 7 20: 7 44: 7 90: 7 44: 7 90: 7 44: 7 90: 7 90	2 3 4 5 6 7 8 9 0 0	ITU. ImSI Loop time enab whili	TSTR.BI' art(0,1) Cnt = 0; = 0; = 0; if (1) { if (TmU; 	T.STR 0): ; ; ; P[0] CLR [0 = 1; == 0N) ▼ (<i>t</i> _k t) ▼ (<i>t</i> _k t)	() z #at	C BF	// 10 P2 00000 har Sym [TCNT0#924	動作 16進 Step	CIra] CLR 一 周期 c CSt	▼ 実行 ep	 回鼓 1 DI		л-к*вря I I n	<u>履歴</u> 𝒴	PUT	
→ 0020000 → 0020000 → 0020001 → 0020001 → 0020000 → 002000 → 002000 → 002000 → 002000 → 00200 → 002000 → 002000 → 00200 → 0000 → 00000 → 0000 → 00000 → 00000 → 0000 → 0000 → 00000 → 000	7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	2 3 4 5 6 7 8 9 0	ITU. Instruction	TSTR.BI art(0,1) Cnt = 0; = 0; le(); e(1) { if (TmU; v (0) et 1	T.STR 0): ; ; P[0] CLR]	0 = 1; == 0N) ▼ (ない ▼ (ない	(C BF	// 10 // 10 P2 00000 har Sym [TCNT07/924	動作 16進 Step	CIra] CLR 一月期 c CS1	▼ 実行 ep	 回鼓 1 D I		л-⊦*вря I In	履歴 ソフ tFlg sc	PUT Sto	
0020000 0020000 0020000 0020000 0020000 0020000 0020000 0020000 0020000 0020000 00200 00200	7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	2 3 4 5 6 7 8 9 0 0	ITU. TeSI Loop time enab whil	TSTR.BI art(0,1 Cnt = 0 = 0 le(); e(1) { if (TmU; et l	D); ; ; [0] [] [] [] [] [] [] [] [] [] [] [] [] []	0 = 1; == 0N) (t&L (t&L (t&L)	{ 	C BF	// 10 // 10 P2 00000 har Sym [o] L	TCNT07/97/1	動作 1 G進 Step	CIra	CLR Fille CS1	▼】 実行	▼ □鼓 1 DI		<u>л-⊦*вря</u> I In Es	<u>屁歴</u> [у; tFlg sc	PUT Sto	
D020000 D020000 D020001 D02001 D020001 D02001	7 7 7 7 2A: 7 7 00: 7 4A: 7 7 02: 8 0000000 8 8 reak	2 3 4 5 6 6 7 8 9 9 0 0	ITU. Inst Loop time enab while n Res	TSTR.BI art(0,1 Cnt = 0 = 0 le(); e(1) { if (TmU; et }	T.STR D); ; ; ; DLR [] 	0 = 1; == 0N) ▼ (tau	(C BF	// 100 P2 [00000 har Sym [o] L	TCNT0//)/	動作 1 G)進 Step		CLR I IIII 2 CS	▼】 実行 ep	マレン 「 」 、 、 、 、 、 、 、 、 、 、 、 、 、	 	Λ-ト°ΒΡΠ I In Es	夏歴 yy tFIg sc	PUT Sto	
D020000 D020000 D020001 D02000 D02000 D02000 D02000 D02000 D0200	7 7 7 2A: 7 7 00: 7 4A: 7 7 02: 8 0000000 Break 200080	2 3 4 5 6 7 8 3 0 0	ITU. Inst Loop time enab whili	TSTR.BI art(0.1) Cnt = 0 = 0 le(); e(1) { if (TmU; e(1) { if (TmU; et }	T.STR 0); ; ; ; CLR [0 = 1; == 0N) ▼ ▼ (/&L Res	{ 	C BF	// 10 // 10 P2 [00000 har Sym [0] L	TCNT0#92F	助作 1 G)進 Step	CIra	CLR NUM C CS1	▼】 実行	URA DI DI		л-ŀ*в₽я I In Et	夏歴 Y7 tFlg sc	PUT Sto	
D020000 D020000 D020000 D020001 D020000 D02000 D0200	7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	2 3 5 6 7 8 9 0 0	ITU. InSU Loop time enab while n Res	et	T.STR 0): ; ; ; P[0] CLR [0 = 1; == 0N) ▼ (/at Re:	[⊂ BF	// 10 // 10 P2 [00001 har 。 、 」 し (、 大 、	TCNT0約20 10000 [了Trace] .0g]	動作 1 6 進 Step		CLR 0 CS	▼ 実行 ep			<u>л-</u> ь'вря I In Еа	最歴 サ tFlg sc	PUT Sto	
● 0020000 ● 0020000 ● 0020001 ● 0020000 ● 0020000 ● 0020000 ● 0020000 ● 0020000 ● 0020000 ● 002000 ● 002000 ● 002000 ● 002000 ● 002000 ● 002000 ● 002000 ● 002000 ● 00200 ● 002000 ● 00200 ●	7 7 7 2A: 7 7 00: 7 7 4A: 7 7 22: 8 0000000 Break	2 3 5 6 7 8 9 0 0	ITU. InSU Loop time enab while n Res	stient of the second se	T.STR 0): ; ; ; CLR]	0 = 1; == 0N) ▼	(v) v) v at	C BF C cl	// // 10 P2 [00001 har Sym [C 力ぶ	Trace .og 	動作 1 6週 Step SAM] (CLR ■ ■ ■ ■ □ ■ ■ ■ □ ■ ■ ■ □ ■ ■ ■	マリ 実行 ep			л ⁺ ВР Л I In E t	夏歴 Ŋ tFIg sc	PUT Sto	

以上

- 5. 注意事項
 - ・本文書の著作権は、エーワン(株)が保有します。
 - ・本文書を無断での転載は一切禁止します。
 - ・本文書に記載されている内容についての質問やサポートはお受けすることが出来ません。
 - ・本文章に関して、ルネサス エレクトロニクス社への問い合わせは御遠慮願います。
 - ・本文書の内容に従い、使用した結果、損害が発生しても、弊社では一切の責任は負わないものとします。
 - ・本文書の内容に関して、万全を期して作成しましたが、ご不審な点、誤りなどの点がありましたら弊社までご連絡くだされば幸いです。
 - ・本文書の内容は、予告なしに変更されることがあります。

 $\overline{+}486-0852$

愛知県春日井市下市場町 6-9-20 エーワン株式会社 https://www.aone.co.ip



メンシューワン株式会社