Rev 1.00.00

Renesas RZ/T1 用サンプル(e2studio-GCC ベアメタル版)の説明

## (MP-RZT1-01+EV-RZT1-01 対応)

- 1. Sampleの免責について
  - ・Sample に関する Tel/Fax でのご質問に関してはお受けできません。ただし、メールでのご質問に関してはお答えするよう努力はしますが、都合によりお答えできない場合もありますので 予めご了承願います。
  - ・Sample ソフトの不具合が発見された場合の対応義務はありません。また、この関連ソフトの 使用方法に関する質問の回答義務もありませんので承知の上ご利用下さい。
  - ・Sample ソフトは、無保証で提供されているものであり、その適用可能性も含めて、いかなる 保証も行いません。また、本ソフトウェアの利用により直接的または間接的に生じたいかなる 損害に関しても、その責任を負わないものとします。

サンプル名	ワークスペース名	コア	プロジェクト名
	概要		
RZT1_Sample_BARE	RZT1_Sample_BARE	M3	Sample_BARE_M3
	ベアメタル版サンプル		(ソース公開)
		R4F	Sample_BARE_R4F
			(ソース公開)
RZT1_Sample_HWRTOS	RZT1_Sample_HWRTOS	M3	Sample_HWRTOS_M3
			(ソース公開)
	M3 側で HWRTOS を使用	R4F	Sample_HWRTOS_R4F
	したサンプル		(ソース公開)
RZT1_Sample_ECAT	RZT1_Sample_ECAT	M3	Sample_ECAT_M3.mot を使用
			( <u>ソース非公開</u> )
	EtherCAT制御サンプル	R4F	Sample_ECAT_R4F
			(ソース公開)
RZT1_Demo_BARE	RZT1_Demo_BARE	M3	Sample_ECAT_M3.mot を使用
			( <u>ソース非公開</u> )
	ベアメタル版デモソフト	R4F	Demo_BARE_R4F
			(ソース公開)

2. サンプル (ベアメタル版) のプロジェクト名

統合開発環境	コンパイラ (ツールチェイン)
e2 studio(バージョン 5.4.0.015)	GNUARM-NONEv1601-EABI

Cソースに#ifdef等のマクロ定義してい	いる場合に使用します。コア【R4F】専用
USED_DEFnano=x	x = DEFnanoを使用[1]する・[0]しない。

注記

コア【M3】側 EtherCAT®制御用サンプルのソース公開を希望される場合は、JSL Technology 社 とのライセンス契約が必要です。 3. 各サンプル (ベアメタル版) の説明

3-1. 「RZT1\_Sample\_BARE」 サンプルの説明

3-1-1. コア [M3] 側

1) フォルダ構成とファイル名

RZT1 Sample BA	RZT1 Sample BARE¥Sample BARE M3					
HardwareDebug	*.mot/*.x 実行ファイル					
_	*.mapファイル	/				
	*.lstファイル					
.setting	e2studio 管理の	)設定ファイル				
src	common	inc	common ∕	ンクルード用 DIR		
		r_mpc.c	MPC処理			
		r_reste.c RESET レジスタ処理		ジスタ処理		
		vectors_M.c	ベクタテー	ーブル		
	Renesas	CMSIS	Cortex-M3	関係定義集		
		RZT1_RIN_Engine	RIN関係定	官義集		
	Sample	app	inc	app インクルード用		
				DIR		
			main.c	メインプログラム		
			cmt_m3.c	CMT処理		
			led_sw.c	LED/SW 処理		
	Startup_GCC	start_asm	startup処理	理		

🎢 エーワン株式会社

2) インクルードパス設定

"\${TCINSTALL}/arm-none-eabi/optlibinc" "\${workspace\_loc;/\${ProjName}/src/common/inc}" "\${workspace\_loc;/\${ProjName}/src/Renesas/CMSIS/Include}" "\${workspace\_loc;/\${ProjName}/src/Renesas/RZT1\_RIN\_Engine/Include}" "\${workspace\_loc;/\${ProjName}/src/sample/app/inc}"

3) 動作説明

- ・メインループ 200,000 回(ソフトタイマ 1usec)ごとに基板上の LED5 を点滅
- ・CMT-CH2タイマー割り込みにより基板上のLED4を100msec毎に点滅

## 3-1-2. コア【R4F】 側

RZT1_Sample_BA	ARE¥Sample	RE¥Sample_BARE_R4F			
HardwareDebug	*.mot/*.x 実	*.mot/*.x 実行ファイル			
	*.mapファ	イル			
	*.lstファイ	ル			
.setting	e2studio 管	理の設定ファイル			
script_file	Locate.LD	セクションロケート定	義ファイル		
src	common	inc	common Inclue	de DIR	
		g_irq.asm	GCC-IRQ 関数	ζ	
		g_loader_init.asm	GCC-startup 久	心理	
		exit.c	exit処理		
		loader_init2.c	MCU 初期化处	理	
		r_atcm_init.c	ATCM処理		
		r_cpg.c	CPG処理		
		r_ecm.c	ECM 処理		
		r_icu_init.c	ICU処理		
		r_mpc.c	MPC処理		
		r_ram_init.c	RAM-ECC 処理	里	
		r_reset.c	RESET レジス	、夕処理	
	monitor	g_initer_handler.asm	GCC-monitor	割込処理	
		g_vector_robin.asm	GCC-monitor 5	対応ベクタテーブル	
		Umonitor.c	USB-monitor 5	処理	
		Umoniter.h	USB-monitor ヘッダー		
	sample	app	inc	app Include DIR	
			board.c	ボード上デバイス	
				処理	
			cm3start.c	CM3 リセット解	
				除処理	
			led_sw.c	LED/SW 処理	
			rspi0_comm.c	RSPI-CH0 処理	
			sci_comm.c	SCI処理	
			sfram.c	FRAM処理	
		cBARE	inc	cBARE Include	
				DIR	
			main_s.c	メイン処理	
			cmt_s.c	CMT処理	

1)フォルダ構成とファイル名

2) インクルードパス設定

"\${workspace\_loc:/\${ProjName}/src/common/inc}" "\${workspace\_loc:/\${ProjName}/src/monitor}" "\${workspace\_loc:/\${ProjName}/src/sample/app/inc}" "\${workspace\_loc:/\${ProjName}/src/sample/cBARE/inc}" "\${TCINSTALL}/arm-none-eabi/optlibinc"

3) 動作説明

・(Step==0) CMT+CH0 タイマー割り込みにより基板上の LED2 を 100msec 毎に点滅

・(Step=1)メインループ 200,000 回(ソフトタイマ 1usec)ごとに基板上の LED2 を点滅

 ・電源 Down 検出 LED3 を点灯、FRAM に内蔵 RAM[0x2000\_0000]から Size[0x8000]書き込 み終了後、LED3 を消灯する。 3-1-3. 動作構成



1) コア【R4F】 側とコア【M3】 側を L-Link のみ使用してデバッグする場合

2) コア【R4F】 側を DEFnano、コア【M3】 側を ROM 化してデバッグする場合



3-1-4. 動作確認方法 アプリケーションノート「RZT1\_Toolchain\_debug\_e2\_gnu.PDF」を参照 3-2. 「RZT1\_Sample\_HWRTOS」サンプルの説明

3-2-1. コア【M3】 側

1)フォルダ構成とファイル名

RZT1_Sample_HV	ZT1_Sample_HWRTOS¥Sample_HWRTOS_M3				
HardwareDebug	*.mot/*.x 実行ファイル				
	*.mapファイル	/			
	*.lst ファイル				
.setting	e2studio 管理の	)設定ファイル			
src	common	inc	commonイン	νクルード用 DIR	
		r_mpc.c	MPC処理		
		r_reste.c	RESET レジ	スタ処理	
		vectors_M.c	ベクタテーブル		
	Renesas CMSIS		Cortex-M3 関係定義集		
		RZT1_RIN_Engine	RIN 関係定義	<b></b> 長集	
	Sample	app	inc	app インクルー	
				ド用 <b>DIR</b>	
			main.c	メインプログラ	
				ム	
			kernel_cfg.c	RTOS カーネル	
				定義	
			led_sw.c	LED/SW 処理	
	Startup_GCC	start.asm	startup処理		
		System_RIN_Engine.c	RIN依存関係	系	

2) インクルードパス設定

"\${TCINSTALL}/arm-none-eabi/optlibinc" "\${workspace\_loc;/\${ProjName}/src/common/inc}" "\${workspace\_loc;/\${ProjName}/src/Renesas/CMSIS/Include}" "\${workspace\_loc;/\${ProjName}/src/Renesas/RZT1\_RIN\_Engine/Include}" "\${workspace\_loc;/\${ProjName}/src/sample/app/inc}"

3) 動作説明

- ・HW-RTOS の初期化と使用例サンプル
- ・main\_task(メインタスク)は、基板上の LED4 を 100msec 毎に点滅
- ・sub\_task(サブタスク)は、基板上の LED5 を 200msec 毎に点滅

3-2-2. コア【R4F】側

1) フォルダ構成とファイル名

2) 動作説明

- 3-1-2. コア【R4F】 側と同等
- 3-2-3. 動作構成
  - 1) コア【R4F】 側とコア【M3】 側を Ijet のみ使用してデバッグする場合
  - 2) コア【R4F】 側を DEFnano、コア【M3】 側を Ije を使用してデバッグする場合
- 3-1-3.動作構成と同等
- 3-1-4. 動作確認方法
  - ・アプリケーションノート「RZT1\_Toolchain\_debug\_e2\_gnu.PDF」を参照

3-3. 「RZT1\_Sample\_ECAT」 サンプルの説明

## 3-3-1. コア【M3】 側

1)フォルダ構成とファイル名

RZT1_Sample_ECAT¥M3				
Exe	Sample_ECAT_M3.mot 実行 Hex ファイル			
List	MAPファイル			
*eni	MP_RZT1_1.xml AdEXP1572用 eni File			
	スレーブ1台接続用			
	MP_RZT1_2.xml	AdEXP1572 用 eni File		
	スレーブ2台接続用			
*esi	AONE_ESC_RZT1_Samp.xml   TwinCAT 用 esi File			
	ple_ECAT¥M3 Exe List *eni	ple_ECAT¥M3 Exe Sample_ECAT_M3.mot 実行 H List MAPファイル *eni MP_RZT1_1.xml MP_RZT1_2.xml *esi AONE_ESC_RZT1_Samp.xml		

・アプリケーションノート「RZT1\_ECAT\_Testing.PDF」を参照

2) 動作説明

- ・【R4F】からの指示によりHW-RTOSの有効/無効を切り替える。
- ・EtherCAT の「PDO-Output」からの受信データを【R4F】側に通知するため共有メモリ ーにセットする。
- ・【R4F】からの送信データを共有メモリーから取得し、EtherCATの「PDO-Input」に向けて出力する。
- ・EtherCATの「CoE-Download」メッセージデータを【R4F】側に通知するため共有メモリ ーにセットする。
- ・【R4F】からのメッセージデータを共有メモリーから取得し、EtherCAT の「CoE-Upload」にて出力する。
- ・基板上の LED4 を 100msec 毎に点滅

3) 共有メモリマップ

	<i></i>	
Mem アドレス	シンボル	用途
0x0080_0000	APP_HIF_OFS_RESET	【M3】 側の初期化完了フラグ
(BTCM)		[M3] 0x01
		【R4F】 クリアー
0x0080_0002	APP_HIF_OFS_ALSTAT	【EtherCAT】 ステータス
		[0x0001] ::initialization
		[0x0002]::pre-opeeational
		[0x0003]::BootStrap
		[0x0004]::safe opeeational
		[0x0008]::opeeational 接続確立
0x0080_0004	APP_HIF_OFS_TXPDO_SEQ	【PDO送信シーケンス番号】
		0x0000~0xFFFF [PDO-Input]
		更新は、【R4F】 側
0x0080_0006	APP_HIF_OFS_RXPDO_SEQ	【PDO受信シーケンス番号】
		0x0000~0xFFFF [PDO-Output]
		更新は、【M3】 側
0x0080_0008	APP_HIF_OFS_TXMSG_SEQ	【CoEMsg送信シーケンス番号】
		0x0000~0xFFFF [CoE-Upload]
		更新は、【R4F】 側
0x0080_000A	APP_HIF_OFS_RXMSG_SEQ	【CoEMsg受信シーケンス番号】
		0x0000~0xFFFF [CoE-Download]
		更新は、【M3】 側

0x0080_000C	予備(シケンス番号)	
0x0080_000F	APP HIF OFS OSLESS	【M3 HWRTOS 有効・無効】
010000_0010		[0x0000]有効(使用する)
		[0x0001]無効(使用しない)
		セットは、【R4F】側
0x0080_0014	APP_HIF_OFS_CONSOLE	【M3 Console モニタ有効・無効】
		[0x0000]無効(使用しない)
		[0x0001]有効(使用する)
		セットは、【R4F】 側
0x0080_0018	予備 (指示系)	
0x0080_07FF		
0-0000 0000	ADD HIE OES TYDDO 08	【PDO-Input への 8bit 送信アーダ】
0x0080_0800	AFF_HIF_OFS_IAFDO_08	B00 D01
0x0080_0802		B01 B02
0x0080 0803		B03
0x0080 0804	予備(8bit)	
0x0080_081F		
		【PDO-Input への 16bit 送信データ】
0x0080_0820	APP_HIF_OFS_TXPDO_16	W00
0x0080_0822		W01
0x0080_0824	予備(16bit)	
0x0080_083F		
0x0080_0840	APP_HIF_OFS_TXPDO_32	【PDO-Input への 32bit 送信アータ】 D00
0x0080_0844	予備(32bit)	
0x0080_0FFF		
0-0000 1000	ADD LIFE OFC DVDDO 09	【PDO-Outputからの8bit受信アータ】
$0x0080_{1000}$	APP_HIF_OFS_RAPDO_08	B00 D01
$0 \times 0080 \_ 1001$ $0 \times 0080 \_ 1002$		B01 B02
0x0080 1003		B03
0x0080_1004	予備(8bit)	
0x0080_101F		
		【PDO-Output からの 16bit 受信データ】
0x0080_1020	APP_HIF_OFS_RXPDO_16	W00
0x0080_1022		W01
0x0080_1024	于/前(16bit)	
0x0080_105F		【PDO-Output からの 32bit 受信データ】
0x0080_1040	APP_HIF_OFS_RXPDO_32	
0x0080_1044	予備(32bit)	
0x0080_17FF		
0x0080_1800	APP_HIF_OFS_TXMSG	【CoE-Upload 送信メッセージ】
0x0080 180F		MAA 160yte
0x0080 1810	予備(CoE-Unload)	
0x0080_1BFF	1 MILCOLL OPPORT	
0x0080_1C00	APP_HIF_OFS_RXMSG	【CoE-Download 送信メッセージ】
		MAX 16byte
0x0080_1C0F		



0x0080_1C10	予備(CoE-Upload)	
0x0000_1111		
0x0080_2000	空(他用途に使用可能)	
0x0080_7FFF		

# 3-3-2. コア【R4F】 側

1)フォルダ構成とファイル名				
RZT1_Sample_E0	CAT¥Sample_ECAT_R4F			
HardwareDebug	*.mot/*.x 実行ファイル			
	*.map ファ	イル		
	*.lst ファイ	<i>I</i> L		
.setting	e2studio 管	理の設定ファイル		
hif_inc	app_hif_ma	ig.h EtherCAT 共有メモ	ミリマッピング気	議
script_file	Locate.ld 1	マクションロケート定義	遠ファイル	
src	common	inc	common Inclue	de DIR
		g_irq.asm	GCC-IRQ 関数	ζ
		g_loader_init.asm	GCC-startup 久	心理
		exit.c	exit処理	
		loader_init2.c	MCU 初期化处	理
		r_atcm_init.c	ATCM処理	
		r_cpg.c	CPG処理	
		r_ecm.c	ECM処理	
		r_icu_init.c	ICU処理	
		r_mpc.c	MPC処理	
		r_ram_init.c	RAM-ECC 処理	里
		r_reset.c	RESET レジス	、タ処理
		syscall.c	低水準入出力	関数処理
	monitor	g_initer_handler.asm	GCC-monitor	割込処理
		g_vector_robin.asm	GCC-monitor	対応ベクタテーブル
		Umonitor.c	USB-monitor §	心理
		Umoniter.h	USB-monitor -	ヘッダー
	sample	app	inc	app Include DIR
			board.c	ボード上デバイス
			cm3start.c	CM3 リセット解
				除処理
			led_sw.c	LED/SW 処理
			rspi0_comm.c	RSPI-CH0处理
			riic0_comm.c	RI2C-CH0处埋
			sci_comm.c	SCI処埋
			sfram.c	FRAM処理
			stchar.c	文字変換処理
		cBARE	inc	cBARE Include
			main	DIK マイン/加田
			main_s.c	<u> ハイン処理</u> <b>CMT</b> 加理
		oot	ine	OMI 光史
		tal		Ethor CAT
			ecat_demo.c	LtherCATテモ

2) インクルードパス設定

"\${TCINSTALL}/arm-none-eabi/optlibinc" "\${workspace\_loc:/\${ProjName}/hif\_inc}" "\${workspace\_loc:/\${ProjName}/src/common/inc}"

"\${workspace\_loc:/\${ProjName}/src/monitor}"

"\${workspace\_loc;/\${ProjName}/src/sample/app/inc}"

"\${workspace\_loc'/\${ProjName}/src/sample/cBARE/inc}"

"\${workspace\_loc;/\${ProjName}/src/sample/ecat/inc}"

3)動作説明

- ・EtherCATデバイスの初期化
- ・CPU 間割り込みを利用して EtherCAT 用データの共有メモリー間を介してデータの受け 渡し処理【M3】<->【R4F】
- ・(Step==0) CMT+CH0 タイマー割り込みにより基板上の LED2 を 100msec 毎に点滅
- ・(Step=1)メインループ 200,000 回(ソフトタイマ 1usec)ごとに基板上の LED2 を点滅
- ・電源 Down 検出 LED3 を点灯、FRAM に内蔵 RAM[0x2000\_0000]から Size[0x8000]書き込み終了後、LED3 を消灯する。

・3-3-1. コア【M3】 側 3) 共有メモリマップを参照

5) 定義説明

シンボル	説明	宣言 FILE
OSLESS_M3	【M3】 側の HWRTOS を使用レスにするかの定	ecat_demo.h
	義 · · · · · · · · · · · · · · · · · · ·	
	[true] HWRTOS を使用しない。	
	[false] HWRTOS を使用する。	
	注意	
	MP-RZT1-01 で Ethernet を使用する場合は、ハ	
	ードウェアの制限により、必ず、「true」を定義	
	して下さい。 <u>HWRTOS</u> を使用しない。	
ECAT_LOOPBACK	EtherCAT からの受信データを Loopback 送信す	ecat_demo.h
	る。	
	(PDO-utput->PDO-Input)	
	(CoE-Download->CoE-Upload)	
	[1] LOOPback する。	
	[0] しない。	and down h
ECAI_INCDAIA	い部送信用ハッノアーナータを悪余件にインク リマントリーズ Pile	ecat_demo.n
	リメントして Ether CAI 达信 9 る。	
	(FDO Input)	
	[1]データインクリメント送信	
	[0]しない。	
Console_M3	【M3】側の事象ごとの推移をコンソール(SCI-	main_s.c
	CH2)に出力する。	
	[true] コンソール(SCI-CH2)に出力する。	
	[false] しない。	
Console_R4F	【R4F】側の事象ごとの推移をコンソール(SCI-	main_s.c
	CH0)に出力する。	
	[true] コンソール(SCI-CH0)に出力する。	
	[false] しない。	

<sup>4)</sup> 共有メモリマップ

3-3-3. 動作構成

3-3-4. 動作確認方法

・アプリケーションノート「RZT1\_ECAT\_Testing.PDF」を参照

3-4. 「RZT1\_Demo\_BARE」サンプルの説明

3-4-1. コア【M3】側 ・3-3-1. コア【M3】側を参照

3-4-2. コア【R4F】 側

1)フォルダ構成とファイル名

RZT1_Demo_BARE¥Demo_BARE_R4F					
HardwareDebug	*.mot/*.x	*.mot/*.x 実行ファイル			
	*.mapファ	*.mapファイル			
	*.lst ファイル				
.setting	e2studio 管理の設定ファイル				
hif_inc	app_hif_mag.h 共有メモリマッピング定義				
script_file	Locate.LD セクションロケート定義ファイル				
src	common	inc	common Include DIR		
		g_irq.asm	GCC-IRQ 関数		
		g_loader_init.a	GCC-startup 奴	理	
		sm			
		exit.c	exit処理		
		loader_init2.c	MCU初期化処	理	
		r_atcm_init.c	ATCM処理		
		r_cpg.c	CPG処理		
		r_ecm.c	ECM 処理		
		r_icu_init.c	ICU処理		
		r_mpc.c	MPC処理		
		r_ram_init.c	RAM-ECC 処理		
		r_reset.c	RESET レジスタ処理		
		syscall.c	低水準入出力関数処理		
	monitor	g_initer_handl	GCC-monitor 割込処理		
		er.asm			
		g_vector_robin.	GCC-monitor 🛪	対応ベクタテーブル	
		Imonitora	LICD		
		Umonitorh	USB-monitor 处理		
	sampla	ann	inc	opp Include DIR	
	sample	app	boorda	ボードトデバイフ加理	
			Doard.c		
			cm5start.c	UNI3 リビット <b>PH</b> 床処理	
			command.c	コマンド処理	
			e2p.c	E2PROM 処理	
			help.c	HELP表示処理	
			led_sw.c	LED/SW 処理	
			rspi0_comm.c	RSPI-CH0 処理	
			riic0_comm.c	RI2C-CH0 処理	
			rtc.c	RTC処理	
			sci_comm.c	SCI処理	
			sfram.c	FRAM処理	
			spibsc.c	SPIBSC処理	

🎢 エーワン株式会社



	stchar.c	文字変換処理
cBARE	inc	cBARE Include DIR
	main_s.c	メイン処理
	cmt_s.c	CMT処理
ecat	inc	ecat Include DIR
	ecat_demo.c	EtherCATデモ
ev	inc	ev Include DIR
	WiViCom	WiFiモジュール
		WVCWB-R-022
		ファームウェア
		[ReadMe.txt]
		RedpineSignals,Inc
		RS.CN.22.GENR.SP.4.7.1a
		との相違を記載
	c_lcd.c	char LCD 处理
	encoder.c	エンコータ処理
	ev_led_sw.c	EV LED/SW 処理
	pwm_comp.c	相補 PWM 処理
	rscan.c	RSCAN処理
	rspi1_comm.c	RSPI-CH1 処理
	s12adc.c	S12ADC 処理
	WiFiSpi.c	WiFiモジュール
		WVCWB-R-022
	TTRINT	SPI制御処埋
	WiFiUart.c	WiFiモジュール
		WVCWB-R-022
(°1 0]	•	UART 制個処理
П120	inc	TIZD INCLUDE DIR
	optin	漢字コード定義
	diskio.c	FatFs
		DISKIO处理
	ff.c	FatFs ·西淮加亚
mad	inc	标毕处理 mad Include DID
msa		IIISO IIICIUOE DIN
uchh	hagia	μSD 制砷化学生 LICDL 甘卡加田
uson	Dasic	
	hmsc	USBH HMSC処埋
	inc	usbh Include DIR
	r_usb_hmsc	USBH-HMSC 基本処理
	apl.c	
	r_usb.main.c	USBH-HMSC メイン処埋
	usbh.c	USBメモリー処理

2) インクルードパス設定

"\${TCINSTALL}/arm-none-eabi/optlibinc"

- "\${workspace\_loc'/\${ProjName}/src/common/inc}"
- "\${workspace\_loc;/\${ProjName}/src/sample/app/inc}"
- "\${workspace\_loc;/\${ProjName}/src/sample/cBARE/inc}"
- "\${workspace\_loc'/\${ProjName}/src/sample/ecat/inc}"

"\${workspace\_loc:/\${ProjName}/src/sample/ev/inc}"

"\${workspace\_loc;/\${ProjName}/src/sample/ev/WiViCom}"

"\${workspace\_loc:/\${ProjName}/src/sample/ev/WiViCom/Driver/Driver\_TCP/API\_Lib}"

"\${workspace\_loc:/\${ProjName}/src/sample/ev/WiViCom/Driver/Driver\_TCP/Applications/MCU}"

"\${workspace\_loc'/\${ProjName}/src/sample/ff12b/inc}"

"\${workspace\_loc;/\${ProjName}/src/sample/msd/inc}"

"\${workspace\_loc;/\${ProjName}/src/sample/usbh/inc}"

"\${workspace\_loc:/\${ProjName}/hif\_inc}"

"\${workspace\_loc:/\${ProjName}/src/monitor}"

- 3)動作説明
  - ・Tera Term からのコマンド指示により各デバイスを動作させる。
  - ・各コマンド体系は後記にて説明します。

4) 共有メモリマップ

・3-3-1. コア【M3】 側 3) 共有メモリマップを参照

5) 定義説明

シンボル	説明	宣言 FILE
OSLESS_M3	【M3】 側の HWRTOS を使用レスにするかの定	ecat_demo.h
	義	
	[true] HWRTOS を使用しない。	
	[false] HWRTOS を使用する。	
	注意	
	MP-RZT1-01 で Ethernet を使用する場合は、ハ	
	ードウェアの制限により、必ず、「true」を定義	
	して下さい。 <u>HWRTOS</u> を使用しない。	
ECAT_LOOPBACK	EtherCAT からの受信データを Loopback 送信す	ecat_demo.h
	(PDO-utput->PDO-Input)	
	(CoE-Download->CoE-Upload)	
	[1] Loophack t Z	
	$\begin{bmatrix} 1 \end{bmatrix} \mathbf{L} \mathbf{v} \mathbf{v} \mathbf{v}^{*} \mathbf{v}^{*} \mathbf{v}^{*}$	
ECAT INCDATA	内部送信用バッファーデータを無条件にインク	ecat demo.h
	リメントして EtherCAT 送信する。	
	(PDO-Input)	
	-	
	[1]データインクリメント送信	
	[0]しない。	
Console_M3	【M3】側の事象ごとの推移をコンソール(SCI-	main_s.c
	CH2)に出力する。	
	[true] コンソール(SCI-CH2)に出力する。	
Coursels D4E	[talse] しない。	
Console_K4F	【R4F】 側の事家ことの推移をコンソール(SCI- OIIO)に山カナズ	main_s.c
	[true] コンソール(SCI-CHO)に出力する	

6) コマンド実行を指示するため「TeraTerm Pro」をインストールする。



漢字イン(N) ^[\$B +

漢字アウト(0) ^[(B -

言語コード(P): 932

赤丸の設定にする。

漢字-受信(K)

Thit カタカナ

ロケール(C): japanese

SJIS -

漢字-送信(J)

SJIS

# 3-4-3. 動作構成



①PC機と接続する RS232C ケーブルは、市販「クロスケーブル」でも可能です。
 ②USB・シリアル変換ケーブルを使用される場合は、「StarTech.com 社 ICUSB232FTN」推奨
 ③自作する場合は、下記の配線になります。

EV-RX/RZ-xx			PC-Windows	
DSUB 9pin			DSUB 9pin	
オス			オス	
1	NC		1	NC
2	RXD •		2	RXD
3	TXD		3	TXD
4	NC		4	DTR
5	GND		5	GND
6	NC		6	DSR
7	RTS	••	7	RTS
8	CTS	<b>♦</b> (	8	CTS
9	NC		9	NC

17

- 1) 「Demo\_BARE\_R4F」 プロジェクトのプログラムを MCU 基板にダウンロードして実行 させます。
- ・ダウンロード方法はアプリケーションノート「RZT1\_Toolchain\_debug\_ewarm\_ICC.PDF」 を参照



TeraTerm proの画面にオープニングメッセージが表示されます。

🧶 СОМ1:1152	200baud - Tera Term VT
ファイル(F)	編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
? HELP HELP DATE DATE TIME MEM1 MEM2 E2P0 E2P1 E2P2 FRAM MSD USBH SWM LEDM DELAY CM3START WDOG PowerDown	<ul> <li>HELP LCOMMENT</li> <li>HELP LCOMMENT</li> <li>LOOP ON/OFF</li> <li>RTC Set/Get Date</li> <li>RTC Set/Get Time</li> <li>Memory Access (8bit)</li> <li>Memory Access (16bit)</li> <li>Memory Access (16bit)</li> <li>Memory Access (32bit)</li> <li>EEPROM Access used EtherCAT</li> <li>EEPROM Access used Ethernet MAC</li> <li>EEPROM Access used WiFi/etc.</li> <li>FRAM Access</li> <li>µ SD Read Sample</li> <li>WCU Get DIP-SW</li> <li>MCU Get LED</li> <li>Software Delay Test</li> <li>Cortex-M3 is execute &amp; EtherCAT Demo Start.</li> <li>WatchDogTimer Test</li> <li>NMI-&gt;Mem[0x2001_0000]-&gt;FRAM[0x0]Size[0x8000] Write</li> <li>FRAM Writing LED5[0N]</li> </ul>
SWB : LEDB : AN003 : CTEXT : RSCAN : WIFIS : WIFIU : ENC : PWM3 :	: BaseBord Get SW : BsdeBord Set LED : AN003 12bitAD Sampling : Text Write [X:0->19] [Y:0->3] ["Text"] : RSCAN-0 External Loop Back Test : Start WiFi SPI-mode Loopback Test : Start WiFi UART-mode Loopback Test : Input Encoder : MTU3a to Complementary PWM Mode(6 phase)
KEY[?] : KEY[*] : KEY[<] : KEY[<] : KEY[BS] : KEY[ESC] :	: ? [Command]{HELP} : Recall Command : Left Move : Right Move : Back Space

3-4-4. 各コマンドの説明

### LOOP コマンド

各コマンドを繰り返し実行させたい時に使用します。

 LOOP \_1
 //LOOP 指示
 \_

 LOOP \_0
 //LOOP 解除
 、

 ジ記述はリターン表現とします。
 以下省略

LOOP 1 にてコマンド処理を繰り返し実行している時に「ESC」キー入力で中断します。

DATE コマンド

MCU 内蔵の RTC に年月日曜を設定します。

DATE \_年\_月\_日\_曜日 ジ // DATE \_2015 \_4 \_5 \_0 2015/4/5 日曜日 // 曜日 0:日 1:月 2:火 3:水 4:木 5:金 6:土 DATE ジ // 現設定データを表示

TIME コマンド

MCU 内蔵の RTC に時間を設定します。

TIME時分秒や	//TIME_9_0_0	9時0分0秒
TIME	現設定データを表法	示

MEM1 コマンド

メモリーを 8bit アクセスで Read/Write/FILL/インクリメント FILL します。

MEM2 コマンド

メモリーを 16nit アクセスで Read/Write/FILL/インクリメント FILL します。

MEM4 コマンド

メモリーを 32bit アクセスで Read/Write/FILL/インクリメント FILL します。

MEM{1/2/4}\_{R/F/I/W}\_先頭アドレス\_サイズ\_{パターン}

## {READ}

MEM1\_R\_0x2000 0000\_0x100 // 0x2000 0000 から 0x100 要素分 8bit ダンプ MEM2\_R\_0x2000\_0000\_0x100 // 0x2000\_0000 から 0x100 要素分 16bit ダンプ MEM4\_R\_0x2000 0000\_0x100 // 0x2000 0000 から 0x100 要素分 32bit ダンプ {FILL} MEM1\_F\_0x2000 0000\_0x100\_02 // 0x2000 0000 から 0x100 要素分(0)8bitFILL MEM2\_F\_0x2000 0000\_0x100\_02 // 0x2000 0000 から 0x100 要素分(0)16bitFILL MEM4\_F\_0x2000 0000\_0x100\_02 // 0x2000 0000 から 0x100 要素分(0)32bitFILL {Increment FILL} MEM1\_I\_0x2000 0000\_0x100\_0/ // 0x2000 0000 から 0x100 要素分(0++)8bitFILL MEM2\_I\_0x2000\_0000\_0x100\_0/2 // 0x2000\_0000 から 0x100 要素分(0++)16bitFILL MEM4\_I\_0x2000\_0000\_0x100\_02 // 0x2000\_0000 から 0x100 要素分(0++)32bitFILL {WRITE} MEM1\_W\_0x2000\_0000\_0x12 //0x2000 0000 に 0x12 を Write MEM2\_W\_0x2000 0000\_0x1234 #0x2000 0000に 0x1234をWrite MEM4\_W\_0x2000\_0000\_012345678 // 0x2000\_0000 に 0x12345678 を Write

{Read Only Memory アドレス} ・シリアルフラッシュ ROM エリア {0x1000 0000 ~0x10FF FFFF} ・内蔵 RAM エリア'(ATCM) {0x0000 0000 ~0x0007 FFFF} {0x0080\_0000 ~0x0080\_7FFF} ・内蔵 RAM エリア(BTCM) ・内蔵 RAM エリア'(InstRAM) {0x0400\_0000 ~0x0407\_FFFF} ・内蔵 RAM エリア(共有 RAM) {0x2000 0000 ~0x2000 FFFF} {Read/Write Memory アドレス} ・MCU 内蔵 RAM エリア {0x2000 0000~0x2006 FFFF} ・MCU内蔵周辺モジュール {周辺モジュールの仕様による} E2P0 コマンド EtherCAT 用 EEPROM の Read 処理をします。 E2P0\_ $\{R\}$ \_EEPROM アドレス」サイズ {READ} E2P0\_R\_EEPROMアドレス\_サイズ ex) E2P0\_0x0\_0x100 // EEPROM の 0x0 番地から 0x100 サイズ分ダンプ表示  $\{Read Only Memory \mathcal{P} \vdash \mathcal{V} \mathcal{A}\}$ ・EEPROMエリア {0x0000~0x7FFF} E2P1 コマンド Eternet 用 EEPROM の Read/Write 処理をします。 E2P1\_{R/W}\_EEPROMアドレス\_{メモリアドレス}\_サイズ {READ} E2P1\_R\_EEPROMアドレス\_サイズ ex) // EEPROM の 0x0 番地から 0x100 サイズ分ダンプ表示 E2P1\_0x0\_0x100 {WRITE} E2P1\_W\_EEPROMアドレス\_メモリアドレス\_サイズ ex) E2P1\_W\_0x0\_0x2000\_0000\_0x80 // EEPROM の 0x0 番地に 0x2000\_0000 番地の内 #容を 0x80 サイズ分 Write この EEPROM は、MAC アドレス内蔵の EEPROM です。 EEPROMの(0x80~0xFF)は、ライトプロテクトになっていますのでWriteできません。 MACアドレスは、【OxFA~OxFF】の6バイトに格納してあります。 {Read Only Memory アドレス} ・EEPROMエリア  $\{0x80 \sim 0xFF\}$ {Write Memory アドレス}  $\{0x0 \sim 0x7F\}$ ・EEPROMエリア

・内蔵 RAM エリア	{0x2000_0000 ~0x2006_FFFF}
E2P1 コマンドには LAN 用データ設定	コマンドが用意してあります。
①E2P1_MAC {Read Only} MACの表示	
②E2P1_NAME {Read/Write} ネットワークインターフェイス名の家	長示と設定
{Read} E2P1_NAME	
{Write} E2P1_NAME_络前之 // MAX	【7 ASCII文字 E2P_NAME_RZsamp14
③E2P1_PORT {Read/Write} PORT番号の表示と設定	
{Read} E2P1_PORT	
{Write} E2P1_PORT_{xxxxx}신 // E2P _	PORT_50000년
④E2P1_IP {Read/Write} デフォルトIPアドレスの表示と設定	
{Read} E2P1_IP	
{Write} E2P1_IP_{xx.xx.xx}会 // E2P _	IP_192.168.21.12 ¢J
⑤E2P1_SUB {Read/Write} サブネットマスクの表示と設定	
{Read} E2P1_SUB≮┚	
{Write} E2P1_SUB_{xx.xx.xx.xx}	// E2P_SUB_255.255.255.0
⑥E2P1_GATE {Read/Write} ゲートウェイアドレスの表示と設定	
{Read}	

E2P1\_GATE

{Write} E2P1\_GATE\_{xx.xx.xx.xx} // E2P \_GATE 192.168.21.126 <⊅ (7)E2P1, DHCP {Read/Write} DHCPアドレスの表示と設定 {Read} E2P1\_DHCP¢<sup>□</sup> {Write} E2P1\_DHCP\_{xx.xx.xx.xx} // E2P \_GATE 192.168.21.63 <⊅ (8)E2P1\_DNS {Read/Write} DNSアドレスの表示と設定 {Read} E2P1\_DNS {Write} E2P1\_DNS\_{xx.xx.xx.xx} // E2P \_GATE \_ 192.168.21.126 < 注記 使用するルータに設定されている IP アドレスを参照して正しいデータを設定して下さ 12 E2P2 コマンド WiFi/etc 用 EEPROM の Read/Write 処理をします。 ・WiFi-EEPROMエリア  $\{0x0 \sim 0xFF\}$ ・etc-EEPROMエリア  $\{0x100 \sim 0x7FFF\}$ E2P2\_{R/W}\_EEPROMアドレス\_{メモリアドレス}\_サイズ {READ} E2P2\_R\_EEPROMアドレス」サイズ ex) E2P2\_0x0\_0x100 // EEPROM の 0x0 番地から 0x100 サイズ分ダンプ表示 {WRITE} E2P2\_W\_EEPROMアドレス\_メモリアドレス\_サイズ ex) E2P2\_W\_0x100\_0x2000\_0000\_0x80 // EEPROM の 0x100 番地に 0x2000\_0000 番地の // 内容を 0x80 サイズ分 Write {Write Memory アドレス} ・ EEPROM エリア  $\{0x100 \sim 0x7FFF\}$  内蔵 RAM エリア {0x2000 0000~0x2006 FFFF}

E2P2 コマンドには WiFi 用データ設定コマンドが用意してあります。

{Read/Write} (1)E2P2, CH BSS channel の表示と設定 {Read} E2P2\_CH<₽ {Write} E2P2\_CH\_{1>14公 // 1~14チャネル番号公 {Read/Write}  $2E2P2_TYPE$ Security type の表示と設定 {Read} E2P2\_TYPE {Write} E2P2\_TYPE\_{OPEN/WPA1/WPA2/WEP}//Security typeの設定 ③E2P2\_SSID {Read/Write} SSID の表示と設定 {Read} E2P2\_SSID¢<sup>□</sup> {Write} E2P2\_SSID\_{SSID(MAX 32 文字)}。 #SSIDの設定 4 E2P2\_PSK {Read/Write} PSKの表示と設定 {Read} E2P2\_PSK<₽ {Write} E2P2\_PSK\_{PSK(MAX 32 文字)} #PSKの設定 5E2P2\_PORT {Read/Write} PORT番号の表示と設定 {Read} E2P2\_PORT {Write} E2P2\_PORT\_{xxxx} // E2P2\_PORT\_50020 FRAM コマンド FRAMの内容を内蔵メモリーに Read します。また、内蔵 RAMの内容を FRAM に Write {READ} FRAM\_R\_FRAMアドレス\_Storeアドレス\_サイズ ex) FRAM\_R\_0x0\_0x2000\_0000\_0x8000 FRAMアドレス(0x0)からサイズ(0x8000)分 Storeアドレス(0x2000\_0000)に Read します。 {WRITE} FRAM\_W\_FRAMアドレス\_Memoryアドレス\_サイズ ex) FRAM\_W\_0x0\_0x2000\_0000\_0x8000 FRAMアドレス(0x0)に Memory アドレス(0x2000 0000)からサイズ(0x8000)分 Write しま す。 ・FRAMアドレス  $\{0x0 \sim 0x7FFF\}$ · Store アドレス {0x2000 0000~0x2006 FFFF} • Memory アドレス {0x2000 0000 ~0x2006 FFFF} MSD コマンド

µSDの登録ファイルをReadして画面表示します。

MSD\_{Filename.ext}

ex) MSD Sample1.mot

USBH コマンド USBメモリーの登録ファイルを Read して画面表示します。

USBH\_{Filename.ext}

ex)

USBH Sample0.mot

## 注意事項

USBH コマンドを実行する場合は、USB-Function 側のケーブルを抜いてからコマンドを 発行して下さい。また、コマンド実行後、再度。USB-Function を使用する場合は、リセ ットにより再起動して下さい。

SWM コマンド MCU 基板上の DIP-SW3 状態を表示します。

SWM⇔

ex)

MCU DIP-SW3\_1[ON/OFF] SW3\_2[ON/OFF] SW3\_3[ON/OFF] SW3\_4[ON/OFF]

LEDM コマンド MCU 基板上の LED3/4/5 を点灯・消灯します。 LEDM\_{0/1}\_{0/1}\_{0/1}// // LEDM {LED3} {LED4} {LED5} 0.消灯 1:点灯

### DELAY コマンド

ソフトウェアで利用している lusec タイマーの精度を計るため LED2 を点滅させます。

DELAY」、「Time 値 usecc // DELAY 10 2 10 usec の精度

①LED2{time 値} 点灯
 ②LED2{time 値} 消灯
 ③LED2{time 値} 点灯
 ④LED2{10msee} 消灯

#### CM3START コマンド

コア【M3】のリセットを解除します。 CM3START↩

#### WDOG コマンド

WDOG タイマーを起動させ MCU リセットさせます。 MCU リセット後は、電源を再立ち上げして下さい。

#### Power Down(NMI 処理)

停電検出回路が有効になっている場合、電源 OFF 時に内蔵 RAM の内容を 32Kbyte 分 FRAM に Write します。

LED5 点灯

FRAM(0x0)から内蔵 RAM(0x2000\_0000)の内容を 32Kbyte 分 Write する。 LED5 消灯

LED5の点灯時間を計測することにより書き込み時間を得ることができます。

### SWB コマンド

評価基板のSW3~6の状態を表示します。

### SWB<⊅

## BaseBord-SW3[ON/OFF] SW4[ON/OFF] SW5[ON/OFF] SW6[ON/OFF]

**LEDB** コマンド 評価基板の **LED1**~4 を点灯・消灯します。

#### LEDB\_{0/1}\_{0/1}\_{0/1}\_{0/1}// // LEDB {LED1} {LED2} {LED3} {LED4} 0:消灯 1:点灯

AN003 コマンド

AN003に接続されているサーミスタの AD 値(12bit)をくり返し表示します。

AN003

# ex) Start AN003 Sampling EndKey[ESC]

<ADDR> cnt[1234] ad[24xx] // cnt[Read Count] ad[AD 値] // ESC キー入力で終了します。

CTEXT コマンド キャラクタ LCD に英数文字を表示します。

CTEXT\_{0~19}\_{0~3}\_{Text}/2 // CTEXT {X:列{Y行}/英数文字}

注意事項 スペース文字が含む文字列の場合は、""で囲んで下さい。 ex} CTEXT \_0 \_0 \_"Aone \_CO.,LTD" ~

RSCAN コマンド RSCAN-CH0の外部ループバックテスト機能を実行します。

RSCAN¢┚

ex)

<TX>cnt[0] id[1] dlc[8] 00 01 02 03 04 05 06 07 // 00>07 数字を送信 data++ <RX>cnt[0] id[1] dlc[8] 00 01 02 03 04 05 06 07 // 00>07 数字を受信 WIFIS コマンド

WiFi モジュール(WVCWB-R-022)をSPI モードでソケット接続してTCP/IP テストプログラムにてLoopbackを実行します。(E2P2 コマンドで設定したWiFi 設定データ使用)

```
WIFISや

#ESC キー入力で終了します。

ex)

It is connecting SPI-Mode

rsi_bootloader()

rsi_spi_feat_sel()

rsi_query_fwversion()

rsi_query_macaddress()

rsi_band()

rsi_spi_mode_sel()

rsi_init()

rsi_scan()

rsi_join()

rsi_join()

rsi_ipparam_set()

rsi_socket()
```

WiFi MyIP<mark>[192.168.21.22</mark>] Port[<mark>50010</mark>] WiFi MyMAC[00:23:A7:5F:0A:C2]

Start Loopback Test EndKey[ESC]

<TCP\_IP Loopback テスト画面>

TCP_IP LoopBack Test Ver1.10	
IP_Address 192.168.21.22 Port_No 50010	
Size 16 MAX(1024) TCP_IP_LoopBack_Test	Textデータを送信
[TxTime Ave(43)(19 -> 119)us access(1218) 1ms<(0)10ms<(0)cnt] [RxTime Ave(29)(6 -> 19830)us access(1218) 1ms<(1)10ms<(1)cnt]	
接続 ▼ モニタ ■ 1回実行 ▼ 連続実行 ASCII(0x10->0xCF)	Cls 切断
Moniter	
Send>40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F Rece>40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F Send>50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F Rece>50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F Send>50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F	*
Rece>60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F	-

・詳細説明は、「4. TCP\_IP Loopback テストツール」を参照。

WIFIU コマンド

WiFi モジュール(WVCWB-R-022)を UART モードでソケット接続して TCP/IP テストプロ グラムにて Loopback を実行します。(E2P2 コマンドで設定した WiFi 設定データ使用)

WIFIU<sup>2</sup>// ESC キー入力で終了します。

ex)

It is connecting UART-Mode at+rsi\_band=0 at+rsi\_init at+rsi\_numscan=4 at+rsi\_authmode=3 at+rsi\_network=INFRASTRUCTURE at+rsi\_scan=0 at+rsi\_psk=7kimcrdfssk34 at+rsi\_join=WHR-G301N,0,2 at+rsi\_ipconf=1,0,0,0 at+rsi\_ltcp=50020

WiFi MyIP[192.168.21.29] Port[50020] WiFi MyMAC[00:23:A7:5E:00:D6]

Start Loopback Test EndKey[ESC]

<TCP\_IP Loopback テスト画面>

TCP_IP LoopBack Test Ver1.10	
IP_Address 192.168.21.29 Port_No 50020	
Size 16 MAX(1024) TCP_IP_LoopBack_Test	🔲 Textデータを送信
[TxTime Ave(58)(36 -> 120)us access(226) 1ms<(0)10ms<(0) [RxTime Ave(18141)(16 -> 43491)us access(226) 1ms<(225)	)cnt] 10ms<(196)cnt]
接続 🛛 モニタ 📄 1回実行 🔽 連続実行 ASCII(0x10-	Cls 切断
Moniter	
Rece>80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8	^
Rece>90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F Rece>90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F	
Send>A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF Rece>A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AF AF	
	-

・詳細説明は、「4. TCP\_IP Loopback テストツール」を参照。

ENC コマンド

インクリメントエンコーダから A/B/Z 相を入力して、ポジション、スピード、Z 相カウント、ホール素子状態、各相ドライバの A/D 値を表示します。

ENC

//ESCキー入力で終了します。

ex)

<ENC> Pos[+ 47290]pls [ 117]rpm [ 16]zcnt S[ 6]% W[0 2040] V[0 2085] U[1 074]CCW[OFF]

①Pos[ポジション1回転 2048]pls
②[回転スピード]rpm
③[Z相カウンタ]zcnt
④S[PWMデューティ比率]%
⑤W[W相ホール素子状態[0/1] W相ドライバ A/D 値]
⑥V[V相ホール素子状態[0/1] V相ドライバ A/D 値]
⑦U[U相ホール素子状態[0/1] U相ドライバ A/D 値]
⑧CCW[CCW 回転指示(ON/OFF)]

PWM3 コマンド

DC ブラシレスモータ (EBD-223741-MC512 シチズン千葉精密(㈱製) のデモ回転をし ます。(相補 PWM モード 6相 PWM 制御)

①PWM3纪 or PWM3\_0纪

ソフトウェアにより、UF,VF,WFビット設定による出力相の切り替え動作のデモ回転 100usec割り込みによる処理

#### ②PWM3\_1¢┚

ホール素子からの外部入力による出力相の切り替え動作のデモ回転 ハードウェア制御 #ESCキー入力で終了します。

ex)

Manual Start DC Brushless Motore Sample EndKey[ESC] SWB6[ON:Start] SWB5[++duty] SWB4[-duty] SWB3[ON:CW OFF:CCW] <ENC> Pos[+1835810]pls [791]rpm [03]zcnt S[7]% W[0 2102] V[0 2082] U[1 2092] CCW[OFF]

<評価基板の SW 操作> SWB6[ON:デモ回転開始] SWB5[PWM デューティ比率+1%] SWB4[PWM デューティ比率-1%] SWB3[回転方向指示 OFF:正転 ON:逆転]

### KEY操作

簡単な1ラインエディタ機能を入れてあります。

- ・BS バックスペース
- ・← 左にカーソル移動
- → 右にカーソル移動
- ・↑ 1回前に入力した内容のリコール
- ・ESC コマンド処理中の中断

- 4. TCP\_IP Loopback テストツール
  - 1) TCP\_IP Loopback テストツールの起動



・<MCUサンプルプログラム>-<\_PC\_Test>の「TCP\_IP\_TEST.exe」を直接クリック ・インストール操作不要





・Size[]は、1回の送信バイト数の指定

・□Text データを送信は、チェックすると、左枠の文字列を送信します。

以上です。

- 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、ARM 社およびルネサス エレクトロニクス社および SEGGER 社およびエ ンビテック社への問い合わせは御遠慮願います。

21/2 エーワン株式会社

- ・本文書の内容に従い、サンプルソフトを使用した結果、不具合が発生しても、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。

6. 商標

- ・J-Link BASE は、SEGGER 社の登録商標、または商品名称です。
- ARM Cortex, Thumb および ARM Cortex-M3/R4F は ARM Limited の EU およびその他の国に おける商標および登録商標です
- ・RZT1は、ルネサス エレクトロニクス株式会社の登録商標、または商品名です。
- ・その他の会社名、製品名は、各社の登録商標または商標です。

7. 参考文献

- ・「RZT1 グループ ユーザーズマニュアル ハードウェア編」 ルネサス エレクトロニクス株式会社
- ・ルネサス エレクトロニクス株式会社提供のサンプル集
- ・「J-Link User Guide」 SEGGER 社
- ・「J-Flash User Guide」 SEGGER 社
- ・「Flasher User Guide」 SEGGER 社
- ・その他

#### $\mp 486-0852$

愛知県春日井市下市場町6-9-20 エーワン株式会社

http://www.robin-w.com