

Renesas RZ/T1 用サンプル(EWARM-ICC 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ください。
- 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。

2. サンプル (ベアメタル版) のプロジェクト名

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

統合開発環境	コンパイラ
EWARM(バージョン 7.60.2.11350)	icc(バージョン 1.08)

C ソースに #fdef 等のマクロ定義して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_BARE¥M3				
Debug	Exe	*.mot/*.out 実行ファイル		
	List	MAP ファイル		
	Obj	OBJ ファイル		
setting		EWARM 管理の設定ファイル		
src	common	inc	common インクルード用 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_IAR	Cstartup_M.c	startup 処理	

2) インクルードパス設定

\$PROJ_DIR\$/src/common/inc
\$PROJ_DIR\$/src/Renesas/RZT1_RIN_Engine/Include
\$PROJ_DIR\$/src/Renesas/CMSIS/Include
\$PROJ_DIR\$/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

1) フォルダ構成とファイル名

RZT1_Sample_BARE¥R4F					
Debug	Exe	*.mot/*.out 実行ファイル			
	List	MAP ファイル			
	Obj	OBJ ファイル			
setting		EWARM 管理の設定ファイル			
src	common	inc	common Include DIR		
		loader_init.asm	startup 処理		
		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		
	monitor	initer_handler.asm	monitor 割込処理		
		vector_robin.asm	monitor 対応ベクタテーブル		
		Umonitor.c	USB-monitor 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	

2) インクルードパス設定

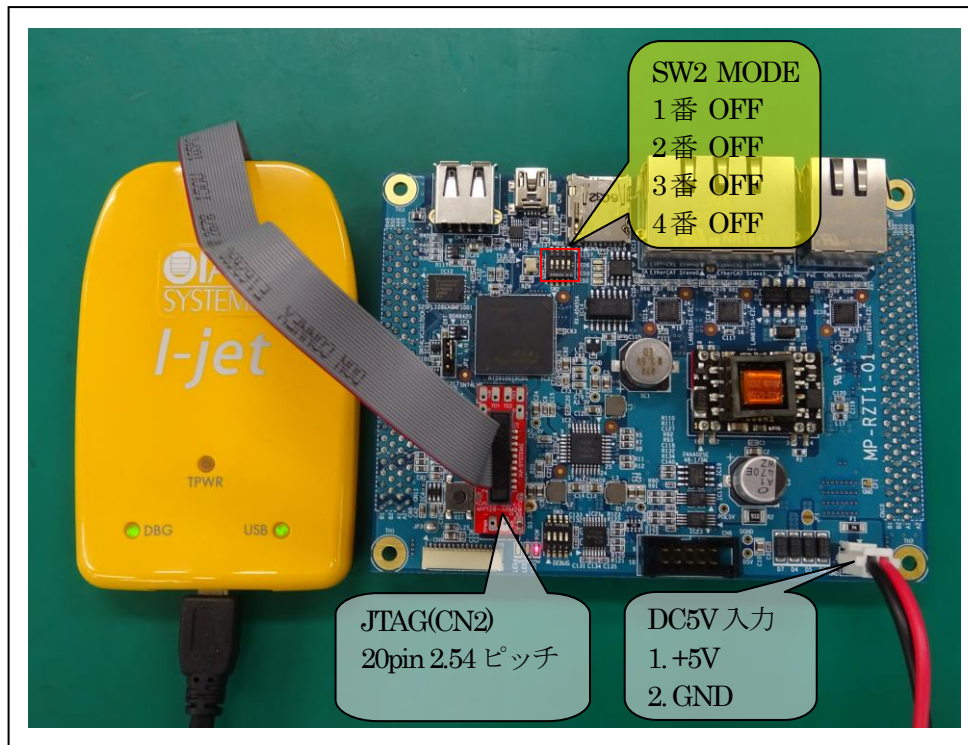
\$PROJ_DIR\$/src/common/inc
\$PROJ_DIR\$/src/sample/app/inc
\$PROJ_DIR\$/src/sample/cBARE/inc
\$PROJ_DIR\$/src/monitor

3) 動作説明

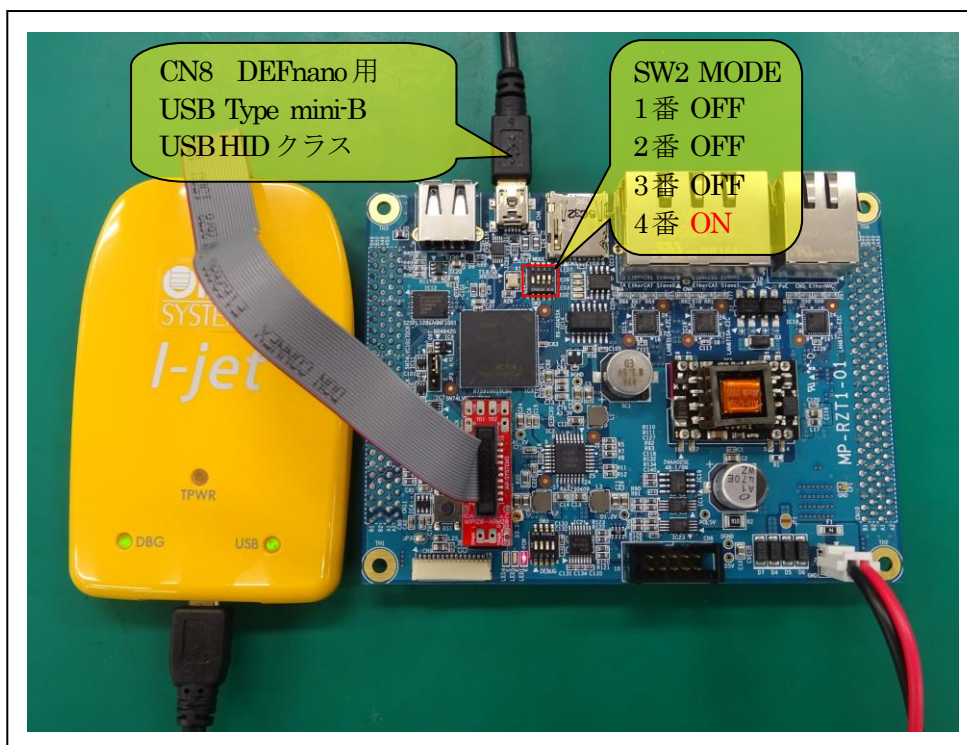
- ・ (Step=0) CMT-CH0 タイマー割り込みにより基板上の LED2 を 100msec 毎に点滅
- ・ (Step=1) メインループ 200,000 回(ソフトタイマ 1usec)ごとに基板上の LED2 を点滅
- ・ 電源 Down 検出 LED3 を点灯、FRAM に内蔵 RAM[0x2000_0000]から Size[0x8000]書き込み終了後、LED3 を消灯する。

3-1-3. 動作構成

1) コア【R4F】側とコア【M3】側を I-jet 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 I-je を使用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_ewarm_ICC.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_HWRTOS_M3				
Debug	Exe	*.mot/*.out 実行ファイル		
	List	MAP ファイル		
	Obj	OBJ ファイル		
setting		EWARM 管理の設定ファイル		
src	common	inc	common インクルード用 DIR	
		r_mpc.c	MPC 処理	
		r_rest.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 関係定義集	
	Sample	app	inc	app インクルード用 DIR
			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_IAR	Cstartup_M.c	startup 処理	

2) インクルードパス設定

```
$PROJ_DIR$/src/Renesas/CMSIS/Include
$PROJ_DIR$/src/Renesas/RZT1_RIN_Engine/Include
$PROJ_DIR$/src/common/inc
$PROJ_DIR$/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_ewarm_ICC.PDF」を参照

3-3. 「RZT1_Sample_ECATTM3」 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_ECATTM3			
Debug	Exe	Sample_ECATTM3.mot 実行 Hex ファイル	
	List	MAP ファイル	
LSLT	*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

・アプリケーションノート「RZT1_ECATTM3_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。
- ・基板上的 LED4 を 100msec 毎に点滅

3) 共有メモリーマップ

Mem アドレス	シンボル	用途
0x0080_0000 (BTCM)	APP_HIF_OFS_RESET	【M3】側の初期化完了フラグ 【M3】 0x01 【R4F】 クリアー
0x0080_0002	APP_HIF_OFS_ALSTAT	【EtherCAT】ステータス [0x0001]::initialization [0x0002]::pre-operational [0x0003]::BootStrap [0x0004]::safe operational [0x0008]::operational 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	【CoE Msg 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	【CoE Msg 受信シーケンス番号】 0x0000～0xFFFF [CoE-Download] 更新は、【M3】側

0x0080_000C 0x0080_000F	予備(シケンス番号)	
0x0080_0010	APP_HIF_OFS_OSLESS	【M3 HWRTOS 有効・無効】 [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 0x0080_07FF	予備 (指示系)	
0x0080_0800 0x0080_0801 0x0080_0802 0x0080_0803	APP_HIF_OFS_TXPDO_08	【PDO-Input への 8bit 送信データ】 B00 B01 B02 B03
0x0080_0804 0x0080_081F	予備(8bit)	
0x0080_0820 0x0080_0822	APP_HIF_OFS_TXPDO_16	【PDO-Input への 16bit 送信データ】 W00 W01
0x0080_0824 0x0080_083F	予備(16bit)	
0x0080_0840	APP_HIF_OFS_TXPDO_32	【PDO-Input への 32bit 送信データ】 D00
0x0080_0844 0x0080_0FFF	予備(32bit)	
0x0080_1000 0x0080_1001 0x0080_1002 0x0080_1003	APP_HIF_OFS_RXPDO_08	【PDO-Output からの 8bit 受信データ】 B00 B01 B02 B03
0x0080_1004 0x0080_101F	予備(8bit)	
0x0080_1020 0x0080_1022	APP_HIF_OFS_RXPDO_16	【PDO-Output からの 16bit 受信データ】 W00 W01
0x0080_1024 0x0080_103F	予備(16bit)	
0x0080_1040	APP_HIF_OFS_RXPDO_32	【PDO-Output からの 32bit 受信データ】 D00
0x0080_1044 0x0080_17FF	予備(32bit)	
0x0080_1800 0x0080_180F	APP_HIF_OFS_TXMSG	【CoE-Upload 送信メッセージ】 MAX 16byte
0x0080_1810 0x0080_1BFF	予備(CoE-Upload)	
0x0080_1C00 0x0080_1C0F	APP_HIF_OFS_RXMSG	【CoE-Download 送信メッセージ】 MAX 16byte

[illegible]

3-3-2. コア【R4F】側

1) フォルダ構成とファイル名

RZT1_Sample_ECAT¥R4F					
Debug	Exe	*.mot/*.out 実行ファイル			
	List	MAP ファイル			
	Obj	OBJ ファイル			
setting		EWARM 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		
src	common	inc	common Include DIR		
		loader_init.asm	startup 処理		
		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	initer_handler.asm	monitor 割込処理	
	vector_robin.asm		monitor 対応ベクタテーブル		
	Umonitor.c		USB-monitor 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	
			riic0_comm.c	RI2C-CH0 処理	
			sci_comm.c	SCI 処理	
			sfram.c	FRAM 処理	
			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		

2) インクルードパス設定

\$PROJ_DIR\$/hif_inc
\$PROJ_DIR\$/src/common/inc
\$PROJ_DIR\$/src/sample/app/inc
\$PROJ_DIR\$/src/sample/cBARE/inc
\$PROJ_DIR\$/src/sample/ecat/inc

\$PROJ_DIR\$/src/monitor

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。

4) 共有メモリアップ

- 3-3-1. コア **【M3】** 側 3) 共有メモリアップを参照

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。HWRTOS を使用しない。	ecat_demo.h
ECAT_LOOPBACK	EtherCAT からの受信データを Loopback 送信する。 (PDO-utput->PDO-Input) (CoE-Download->CoE-Upload) [1] Loopback する。 [0] しない。	ecat_demo.h
ECAT_INCDATA	内部送信用バッファデータを無条件にインクリメントして EtherCAT 送信する。 (PDO-Input) [1] データインクリメント送信 [0] しない。	ecat_demo.h
Console_M3	【M3】 側の事象ごとの推移をコンソール(SCI-CH2)に出力する。 [true] コンソール(SCI-CH2)に出力する。 [false] しない。	main_s.c
Console_R4F	【R4F】 側の事象ごとの推移をコンソール(SCI-CH0)に出力する。 [true] コンソール(SCI-CH0)に出力する。 [false] しない。	main_s.c

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_BAREYR4F				
Debug	Exe	*.mot/*.out 実行ファイル		
	List	MAP ファイル		
	Obj	OBJ ファイル		
setting		EWARM 管理の設定ファイル		
hif_inc		app_hif_mag.h	EtherCAT 共有メモリマッピング定義	
src	common	inc	common Include DIR	
		loader_init.asm	startup 処理	
		exit.c	exit 処理	
		loader_init2.c	MCU 初期化処理	
		r_atem_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	initer_handler.asm	monitor 割込処理	
		vector_robin.asm	monitor 対応ベクタテーブル	
		Umonitor.c	USB-monitor 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
			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
			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
			WiFiCom	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 処理
			encoder.c	エンコーダ処理
			ev_led_sw.c	EV LED/SW 処理
			pwm_comp.c	相補 PWM 処理
			rscan.c	RSCAN 処理
			rspl_comm.c	RSPI-CH1 処理
			s12adc.c	S12ADC 処理
			WiFiSpi.c	WiFi モジュール WVCWB-R-022 SPI 制御処理
			WiFiUart.c	WiFi モジュール WVCWB-R-022 UART 制御処理
		ff12b	inc	ff12b Include DIR
			optin	漢字コード定義
			diskio.c	FatFs DISKIO 処理
			ff.c	FatFs 標準処理
		msd	inc	msd Include DIR
			msd.c	μSD 制御処理
		usbh	basic	USBH 基本処理
			hmsc	USBH HMSC 処理
			inc	usbh Include DIR
			r_usb_hmsc_aplc	USBH-HMSC 基本処理
			r_usb.main.c	USBH-HMSC メイン処理
			usbh.c	USB メモリー処理

2) インクルードパス設定

\$PROJ_DIR\$/src/common/inc
\$PROJ_DIR\$/src/sample/app/inc
\$PROJ_DIR\$/src/sample/cBARE/inc
\$PROJ_DIR\$/src/sample/ecat/inc
\$PROJ_DIR\$/src/sample/ev/inc
\$PROJ_DIR\$/src/sample/ev/WiViCom
\$PROJ_DIR\$/src/sample/ev/WiViCom/Driver/Driver_TCP/API_Lib
\$PROJ_DIR\$/src/sample/ev/WiViCom/Driver/Driver_TCP/Applications/MCU
\$PROJ_DIR\$/src/sample/ff12b/inc
\$PROJ_DIR\$/src/sample/msd/inc

```
$PROJ_DIR$/src/sample/usbh/inc
$PROJ_DIR$/hif_inc
$PROJ_DIR$/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 を使用しない。</u>	ecat_demo.h
ECAT_LOOPBACK	EtherCAT からの受信データを Loopback 送信する。 (PDO-utput->PDO-Input) (CoE-Download->CoE-Upload) [1] Loopback する。 [0] しない。	ecat_demo.h
ECAT_INCDATA	内部送信用バッファデータが無条件にインクリメントして EtherCAT 送信する。 (PDO-Input) [1] データインクリメント送信 [0] しない。	ecat_demo.h
Console_M3	【M3】側の事象ごとの推移をコンソール(SCI-CH2)に出力する。 [true] コンソール(SCI-CH2)に出力する。 [false] しない。	main_s.c
Console_R4F	【R4F】側の事象ごとの推移をコンソール(SCI-CH0)に出力する。 [true] コンソール(SCI-CH0)に出力する。 [false] しない。	main_s.c

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。

- ①「teraterm-4.80.exe」を検索してダウンロードする。
- ②PCにインストールし実行する
- ③シリアルポートの設定

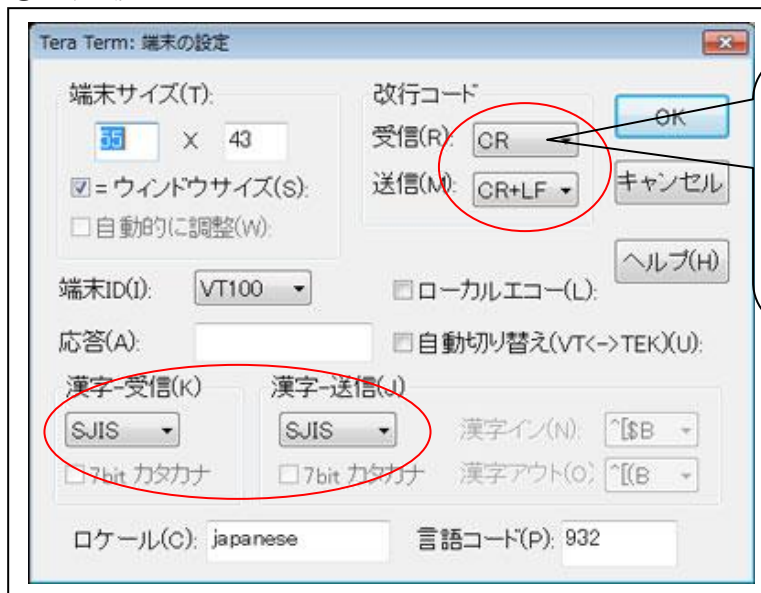


COM 番号は、
PC 側でシリアル
通信可能な番号を
指定する。

115200BPS
8bit
none
1bit
none

の仕様にする。

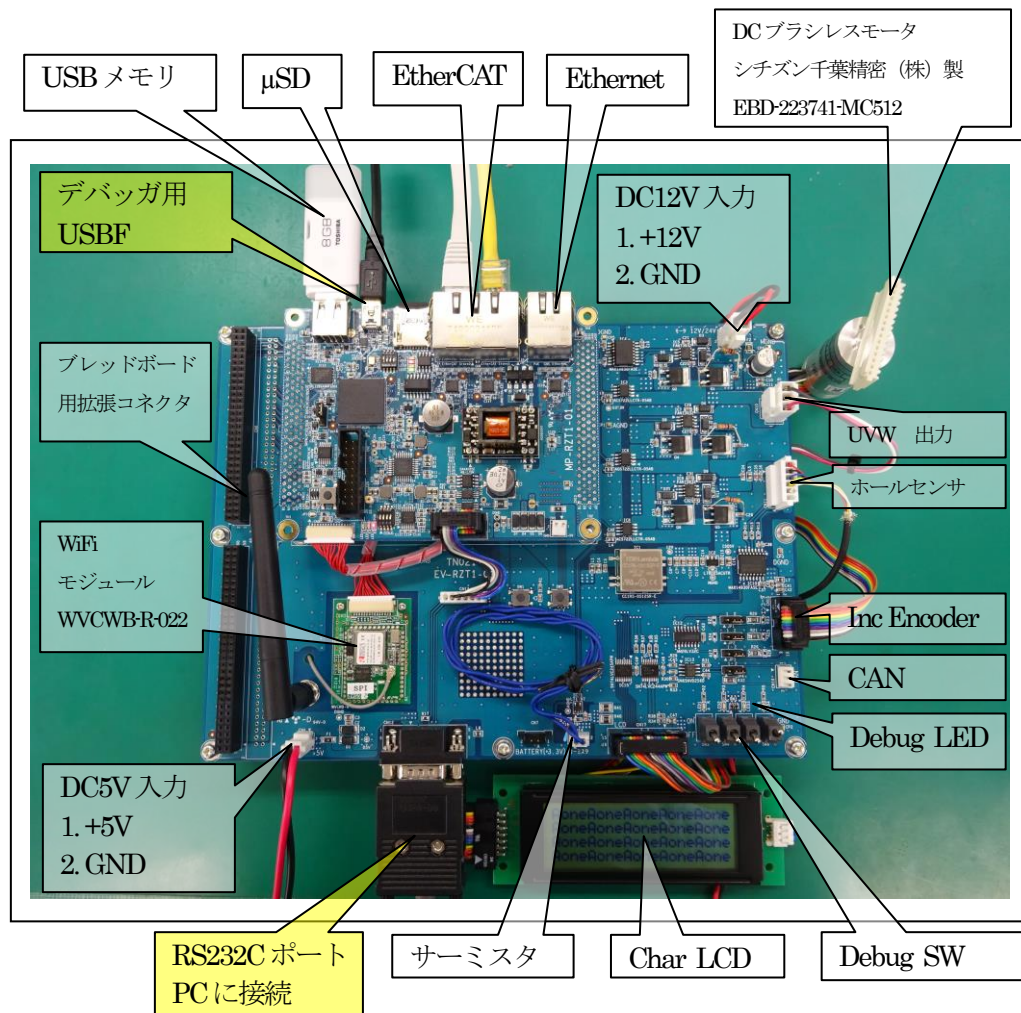
④端末の設定



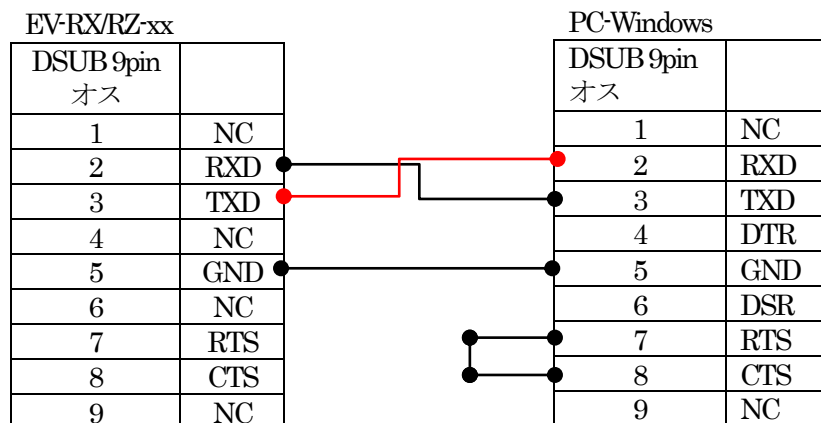
USB シリアルコン
バータ使用時に
CR コードがカッ
トされる設定の場
合は、受信：LF
にして下さい。

赤丸の設定にする。

3-4-3. 動作構成

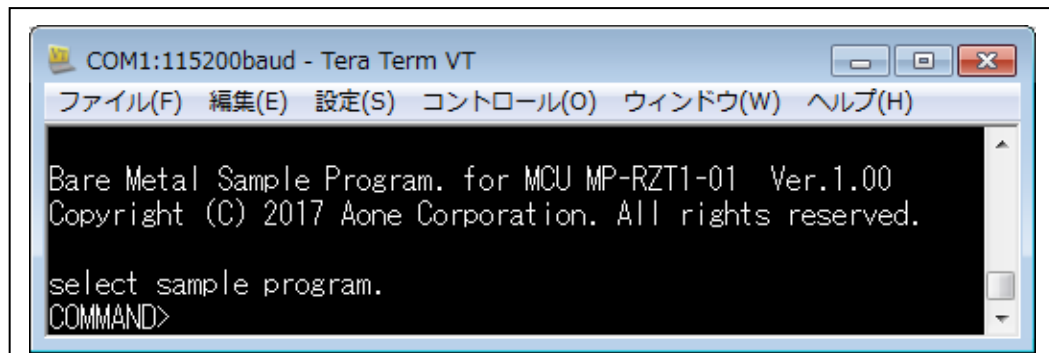


- ①PC機と接続する RS232C ケーブルは、市販「クロスケーブル」でも可能です。
- ②USB-シリアル変換ケーブルを使用される場合は、「StarTech.com 社 ICUSB232FTN」推奨
- ③自作する場合は、下記の配線になります。

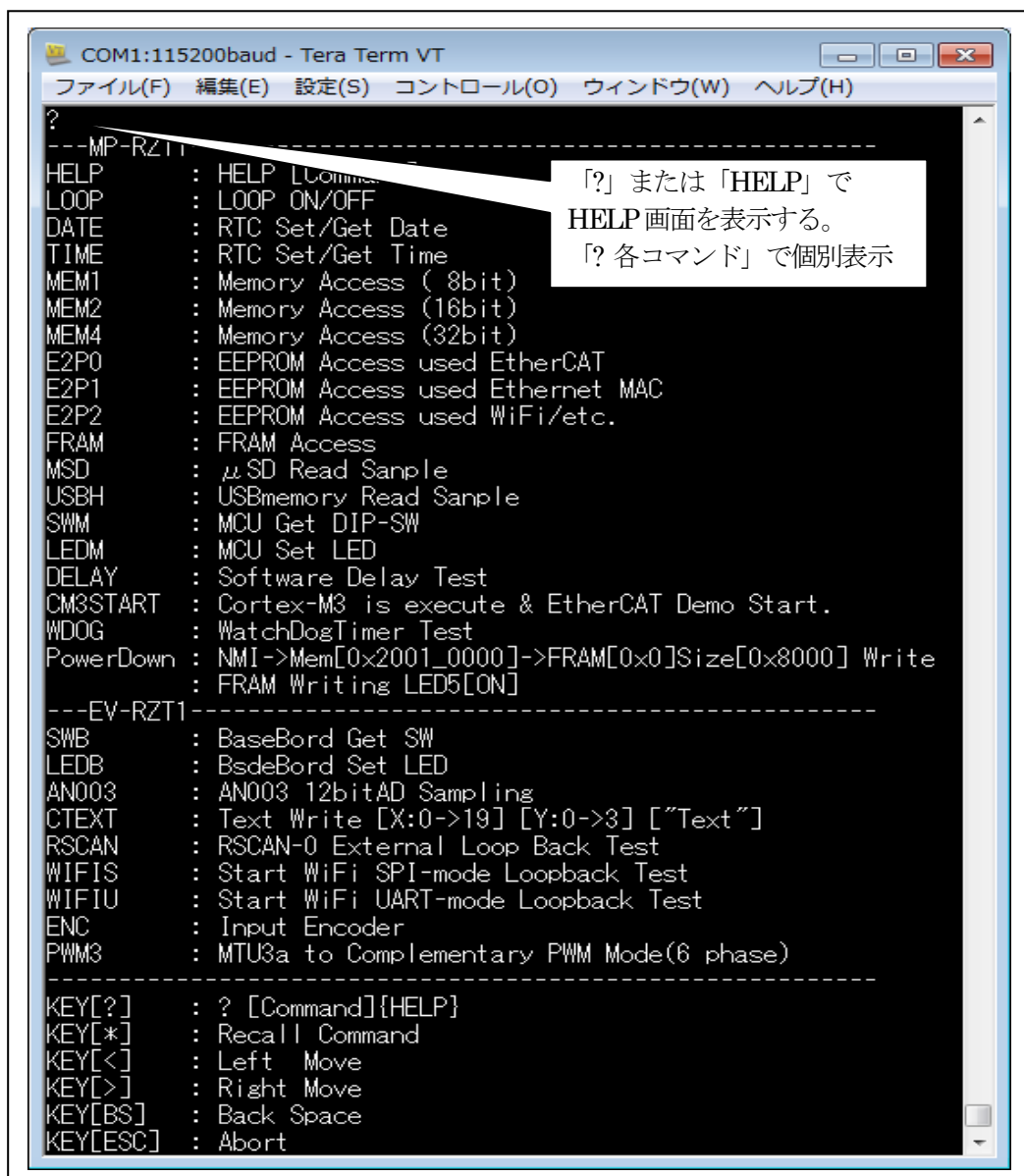


1) 「Demo_BARE_R4F」プロジェクトのプログラムを MCU 基板にダウンロードして実行させます。

- ・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。



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<CR> //LOOP 指示 <CR>記述はスペース表現とします。以下省略

LOOP 0<CR> //LOOP 解除 <CR>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<CR> //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<CR> //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<CR> //TIME 9_0_0 9時0分0秒

TIME<CR> //現設定データを表示

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 16bit 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}<CR>{R/W}<CR>先頭アドレス<CR>サイズ<CR>{パターン}<CR>

{READ}

MEM1<CR>R<CR>0x2000_0000<CR>0x100<CR> //0x2000_0000 から 0x100 要素分 8bit ダンプ

MEM2<CR>R<CR>0x2000_0000<CR>0x100<CR> //0x2000_0000 から 0x100 要素分 16bit ダンプ

MEM4<CR>R<CR>0x2000_0000<CR>0x100<CR> //0x2000_0000 から 0x100 要素分 32bit ダンプ

{FILL}

MEM1<CR>F<CR>0x2000_0000<CR>0x100<CR>0<CR> //0x2000_0000 から 0x100 要素分(0)8bitFILL

MEM2<CR>F<CR>0x2000_0000<CR>0x100<CR>0<CR> //0x2000_0000 から 0x100 要素分(0)16bitFILL

MEM4<CR>F<CR>0x2000_0000<CR>0x100<CR>0<CR> //0x2000_0000 から 0x100 要素分(0)32bitFILL

{Increment FILL}

MEM1<CR>I<CR>0x2000_0000<CR>0x100<CR>0<CR> //0x2000_0000 から 0x100 要素分(0++)8bitFILL

MEM2<CR>I<CR>0x2000_0000<CR>0x100<CR>0<CR> //0x2000_0000 から 0x100 要素分(0++)16bitFILL

MEM4<CR>I<CR>0x2000_0000<CR>0x100<CR>0<CR> //0x2000_0000 から 0x100 要素分(0++)32bitFILL

{WRITE}

MEM1<CR>W<CR>0x2000_0000<CR>0x12<CR> //0x2000_0000 に 0x12 を Write

MEM2<CR>W<CR>0x2000_0000<CR>0x1234<CR> //0x2000_0000 に 0x1234 を Write

MEM4<CR>W<CR>0x2000_0000<CR>012345678<CR> //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}
- ・ 内蔵 RAM エリア(BTCM) {0x0080_0000 ~ 0x0080_7FFF}
- ・ 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 アドレス}

- ・ EEPROM エリア {0x0000 ~ 0x7FFF}

E2P1 コマンド

Ethernet 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)

E2P1_0x0_0x100 // EEPROM の 0x0 番地から 0x100 サイズ分ダンプ表示

{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、【0xFA~0xFF】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 ~ 0xFF}

{Write Memory アドレス}

- ・ EEPROM エリア {0x0 ~ 0x7F}

・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表示と設定

{Read}

E2P1_NAME↵

{Write}

E2P1_NAME_{名前}↵ // MAX 7 ASCII 文字 E2P_NAME_RZsample↵

③E2P1_PORT {Read/Write}

PORT 番号の表示と設定

{Read}

E2P1_PORT↵

{Write}

E2P1_PORT_{xxxxx}↵ // E2P_PORT_50000↵

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.xx}↵ // E2P_IP_192.168.21.12↵

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↵
```

⑦E2P1_DHCP {Read/Write}
DHCP アドレスの表示と設定

```
{Read}
E2P1_DHCP↵
```

```
{Write}
E2P1_DHCP_{xx.xx.xx.xx}↵ // E2P_GATE_192.168.21.63↵
```

⑧E2P1_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い。

E2P2 コマンド

WiFi/etc 用 EEPROM の Read/Write 処理をします。

- WiFi-EEPROM エリア {0x0 ~ 0xFF}
- etc-EEPROM エリア {0x100 ~ 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 ~ 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。

①E2P2_CH {Read/Write}

BSS channel の表示と設定

{Read}

E2P2_CH↵

{Write}

E2P2_CH_{1-14}↵ // 1~14 チャンネル番号↵

②E2P2_TYPE {Read/Write}

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↵

{Write}

E2P2_SSID_{SSID(MAX 32 文字)}↵ // SSID の設定

④E2P2_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_{xxxxxx}↵ // 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 します。

{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 ~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 1usec タイマーの精度を計るため LED2 を点滅させます。

DELAY_ {Time 値}usec < 〽 // DELAY 10 < 〽 10usec の精度

- ①LED2{time 値} 点灯
- ②LED2{time 値} 消灯
- ③LED2{time 値} 点灯
- ④LED2{10msec} 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}↵ // CTEXT {X:列}{Y:行}{英数文字}

注意事項

スペース文字を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<CR>

//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_ipparam_set()

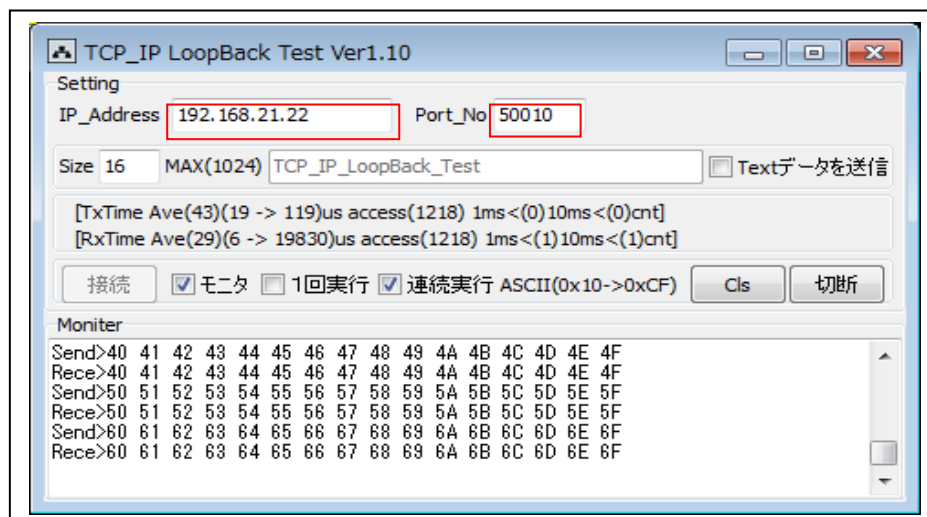
rsi_socket()

WiFi MyIP[192.168.21.22] Port[50010]

WiFi MyMAC[00:23:A7:5F:0A:C2]

Start Loopback Test EndKey[ESC]

<TCP_IP Loopback テスト画面>



・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

//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

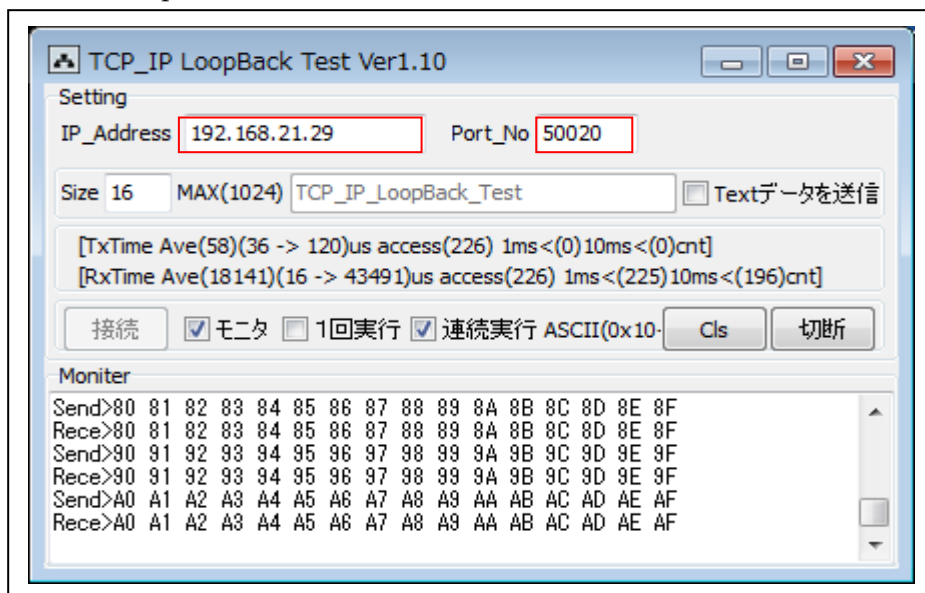
at+rsi_ltcp=50020

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>



- ・ 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(株)製) のデモ回転をし
ます。(相補 PWM モード 6 相 PWM 制御)

①PWM3<⏏> or PWM3_0<⏏>

ソフトウェアにより、UE,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]

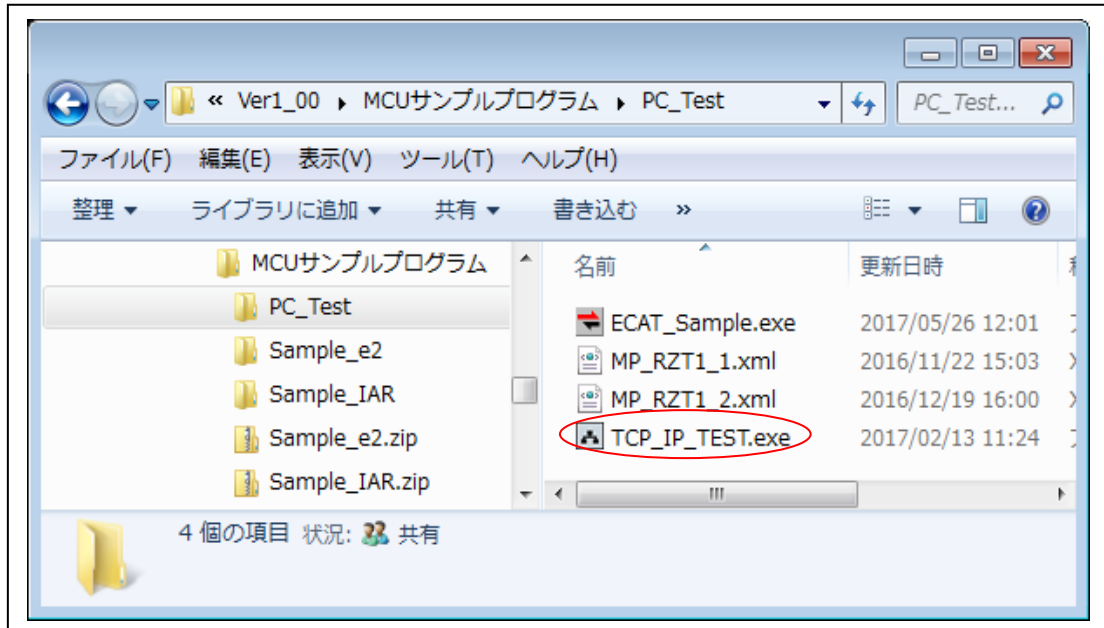
KEY 操作

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

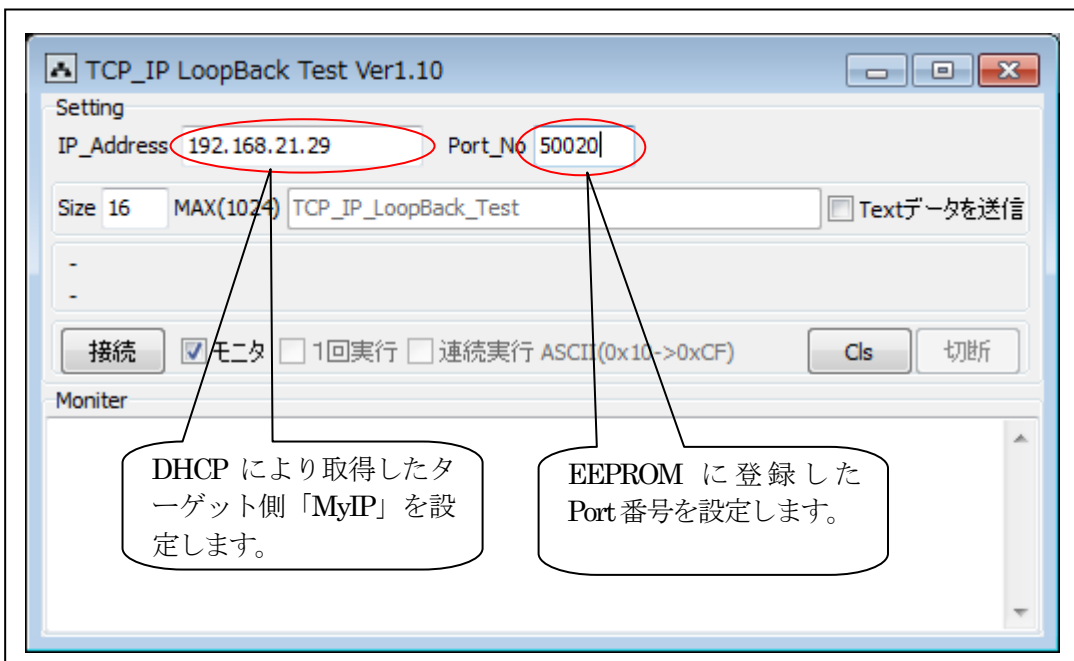
4. TCP_IP Loopback テストツール

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

2)



3)

TCP_IP LoopBack Test Ver1.10

Setting

IP_Address 192.168.21.22 Port_No 50020

Size 16 MAX(1024) TCP_IP_LoopBack_Test ☐ Textデータを送信

[TxTime Ave(31)(24 -> 64)us access(59) 1ms<(0)10ms<(0)cnt]
[RxTime Ave(10)(7 -> 26)us access(59) 1ms<(0)10ms<(0)cnt]

☒ モニタ ☐ 1回実行 ☐ 連続実行 ASCII(0x10->0xCF)

Monitor

接続しました。

Send>10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
Rece>10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
Send>20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
Rece>20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
Send>30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
Rece>30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
Send>40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

③0x10~0xCF まで
指定 Size 分送受信
します。
・ ☐ モニタにする
と表示しません。

①「接続」をクリックします。
コネクションが成功しますと「接続しました」と表示

②連続実行を ☒ すると連続送受信を実行します。
・ 1回実行 ☒ すると 1回の受信を実行します。

④テスト終了は、「切断」をクリック

- ・ 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 IAR 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、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

- ・EWARM は、IAR 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
- ・RZ/T1 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

- ・「RZ/T1 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
- ・「IDE プロジェクト管理およびビルドガイド」 IAR 社
- ・「IAR C/C++開発ガイド コンパイラおよびリンク」 IAR 社
- ・「IAR アセンブリリファレンスガイド」 IAR 社
- ・「IAR デバッグプローブガイド」 IAR 社
- ・その他

〒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>