

Renesas RZ/T1 用サンプル(e2studio-GCC NORTi 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、いかなる保証も行わない。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 (RTOS 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_Demo_NORTi	RZT1_Demo_NORTi RTOS 版デモソフト	M3	Sample_ECAT_M3.mot を使用 (ソース非公開)
		R4F	Demo_NORTi_R4F (ソース公開)

統合開発環境	コンパイラ (ツールチェーン)
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。
RTOS	NORTi 使用時に定義

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. サンプル (RTOS 版) の説明

3-1. 「RZT1_Demo_NORTi」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_Demo_NORTi¥M3			
Debug	Exe	Sample_ECAT_M3.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_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-opeational [0x0003]::BootStrap [0x0004]::safe opeational [0x0008]::opeational 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	APP_HIF_OFS_RXMSG	【CoE-Download 送信メッセージ】 MAX 16byte
0x0080_1C0F		
0x0080_1C10 0x0080_1FFF	予備(CoE-Upload)	
0x0080_2000	空 (他用途に使用可能)	
0x0080_7FFF		

3-1-2. コア【R4F】側

1) フォルダ構成とファイル名

RZT1_Demo_NORTi ¥ Demo_NORTi_R4F				
NORTi	空			
	ReadMe.txt	インストール済み NORTi オリジナルからのインポート手順書		
NORTi_samp	NETSAM P	inc	NETSMP のインクルード	
		noncons.c	コンソール関係	
		nondhcp.c	DHCP 処理	
		nonedns.c	DNS サーバー	
		nonping.c	PING 処理	
		nons shel.c	コマンド shell	
		nonsntp.c	SNTP 処理	
		nonteld.c	TELNET デーモン	
		sntptime.c	SNTP 処理	
		ReadMe.txt	オリジナル NETSMP の変更履歴	
	SMP	inc	SMP のインクルード	
		ethrzt1.c	LAN ドライバの初期化	
		mprzt101.c	NORTi 使用の MCU 依存部	
		n4irzt1.c	RZT1 初期化処理	
		nonethwc	ETHER 割り込み登録	
		nosrzt1.c	SIO 用シリアルドライバ	
		phyrzt1.c	PHY コントロール	
		mpuav7rs79	MMU 処理	
		rzt1.s79	StartUP 処理	
ReadMe.txt	オリジナル SMP の変更履歴			
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			
hif_inc	app_hif_mag.h EtherCAT 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	
		r_icu_init.c	ICU 処理	
		r_mpc.c	MPC 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_handler.s79	GCC-monitor 割込処理	
		g_vector_robin.s79	GCC-monitor 対応ベクタテーブル	
		Nmonitor.c	USB-monitor 処理	
		Nmoniter.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_com.m.c	RSPI-CH0 処理
		riic0_com.m.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
	cRTOS	inc	cBARE Include DIR
		main_r.c	メイン処理
		aonettcp.c	TCP/IP ループバックテスト
		Task_sample.c	TASK サンプル
	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.SP4.7.1a 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_com.p.c	相補 PWM 処理
		rscan.c	RSCAN 処理
		rspi1_com.m.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
	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

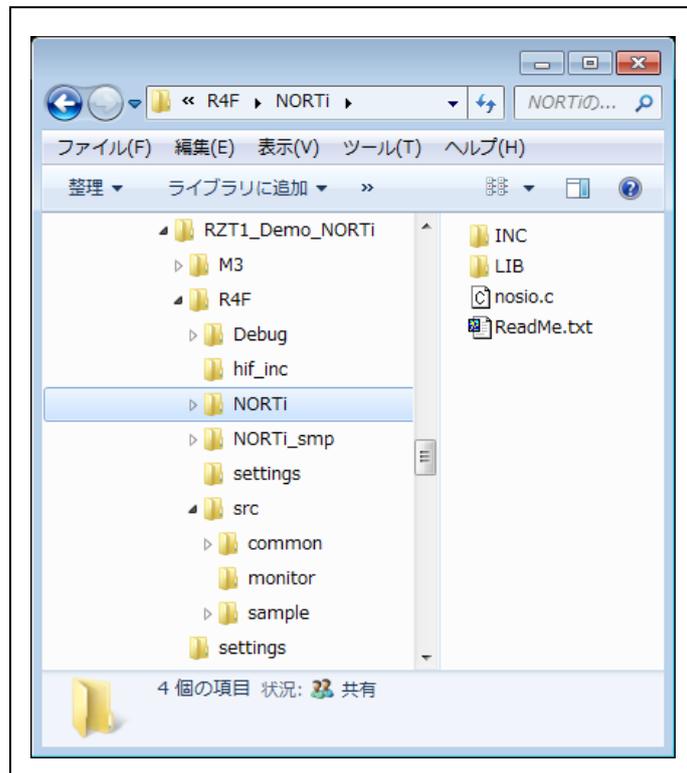
		msd	inc	msd Include DIR
			msdc	μSD 制御処理

2) インストール済み NORTi オリジナル (有償) からサンプルにインポートする手順

- ① 1) 表の黄色部は、空ディレクトリになっていますので、インストール済み NORTi オリジナルから「arm_only」用のライブラリを作成後、下記のように Copy します。

NORTi オリジナル		サンプル(EVRZ_NORTi)
NORTi\INC	→	NORTi\INC
NORTi\LIB	→	NORTi\LIB
NORTi\SRC\nosio.c	→	NORTi\nosio.c

上記のように NORTi オリジナルから、サンプル「R4F\NORTi」の空ディレクトリに Copy して下さい。



- ② ReadMe.txt に Copy 後の変更内容が記述してありますのでソースを変更して下さい。

3) 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}/${ProjName}/src/sample/cRTOS/inc}"
"${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}/${ProjName}/src/sample/ev/WiViCom/Driver/Driver_TCP/API_Lib/wps/utlis}"
"${workspace_loc}/${ProjName}/src/sample/ev/WiViCom/Driver/Driver_TCP/API_Lib/wps/tls}"
"${workspace_loc}/${ProjName}/src/sample/ev/WiViCom/Driver/Driver_TCP/API_Lib/wps/wps}"
"${workspace_loc}/${ProjName}/src/sample/ev/WiViCom/Driver/Driver_TCP/API_Lib/wps/eap_common}"
"${workspace_loc}/${ProjName}/src/sample/ev/WiViCom/Driver/Driver_TCP/API_Lib/wps/eap_peer}"
"${workspace_loc}/${ProjName}/src/sample/ev/WiViCom/Driver/Driver_TCP/API_Lib/wps/common}"
"${workspace_loc}/${ProjName}/src/sample/ev/WiViCom/Driver/Driver_TCP/API_Lib/wps/crypto}"
"${workspace_loc}/${ProjName}/src/sample/ff12b/inc}"
"${workspace_loc}/${ProjName}/src/sample/msd/inc}"
"${workspace_loc}/${ProjName}/hif_inc}"
"${workspace_loc}/${ProjName}/NORTI/INC}"
"${workspace_loc}/${ProjName}/NORTI_smp/NETSMP/inc}"
"${workspace_loc}/${ProjName}/NORTI_smp/SMP/inc}"
"${workspace_loc}/${ProjName}/src/monitor}"
  
```

4) 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。

5) 共有メモリマップ

- 3-1-1. コア **【M3】** 側 3) 共有メモリマップを参照

6) 定義説明

シンボル	説明	宣言 FILE
OSLESS_M3	<p>【M3】側の HWRRTOS を使用レスにするかの定義</p> <p>[true] HWRRTOS を使用しない。 [false] HWRRTOS を使用する。</p> <p>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>HWRRTOS を使用しない。</u></p>	ecat_demo.h
ECAT_LOOPBACK	<p>EtherCAT からの受信データを Loopback 送信する。 (PDO-utput->PDO-Input) (CoE-Download->CoE-Upload)</p> <p>[1] Loopback する。 [0] しない。</p>	ecat_demo.h
ECAT_INCDATA	<p>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)</p> <p>[1] データインクリメント送信 [0] しない。</p>	ecat_demo.h
Console_M3	<p>【M3】側の事象ごとの推移をコンソール(SCI-CH2)に出力する。</p> <p>[true] コンソール(SCI-CH2)に出力する。 [false] しない。</p>	main_s.c
Console_R4F	<p>【R4F】側の事象ごとの推移をコンソール(SCI-CH0)に出力する。</p> <p>[true] コンソール(SCI-CH0)に出力する。 [false] しない。</p>	main_s.c

7) 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

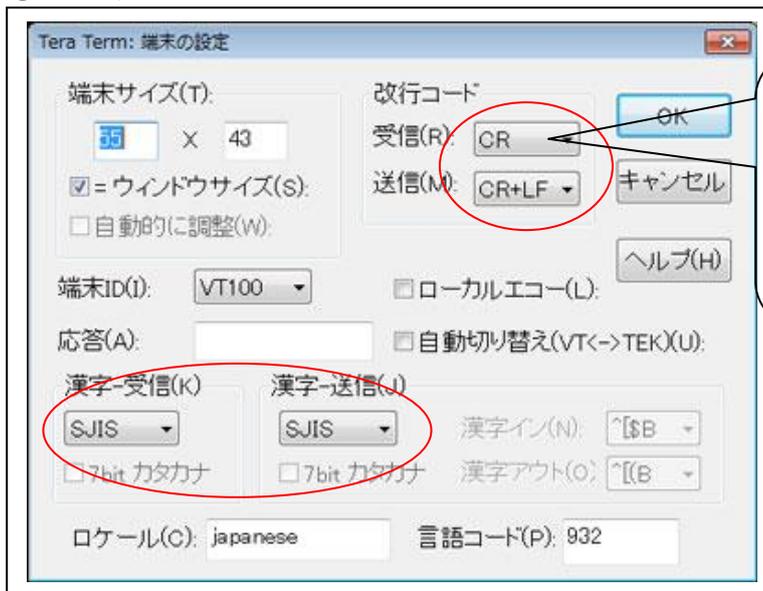


COM 番号は、
PC 側でシリアル
通信可能な番号を
指定する。

115200BPS
8bit
none
1bit
none

の仕様にする。

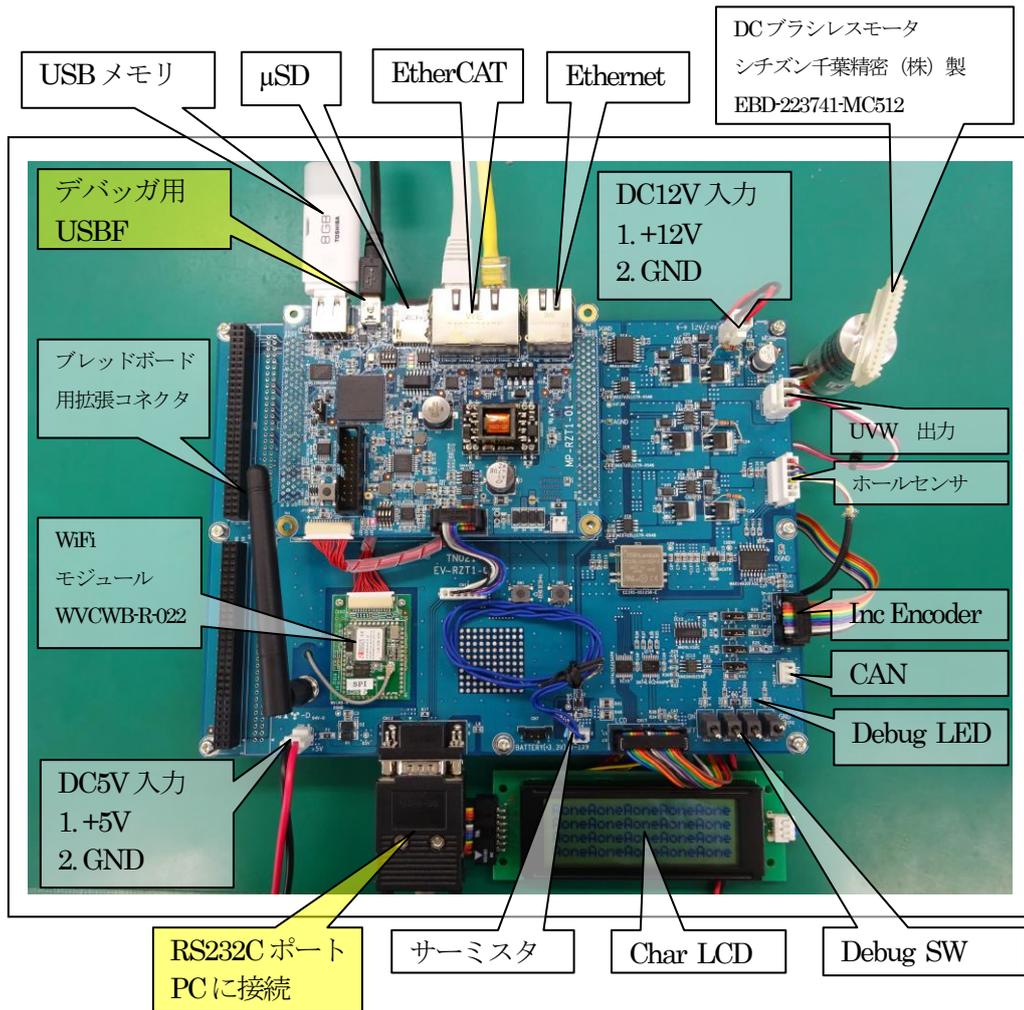
④端末の設定



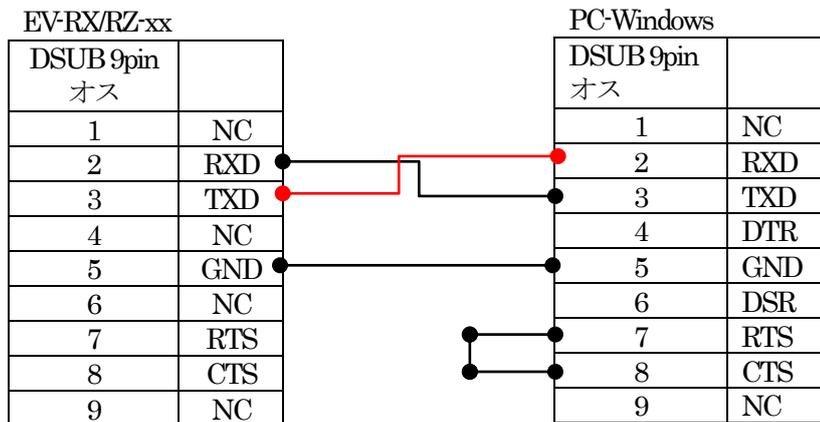
USB シリアルコン
バータ使用時に
CR コードがカット
される設定の場合
は、受信 : LF
にして下さい。

赤丸の設定にする。

3-1-3. 動作構成

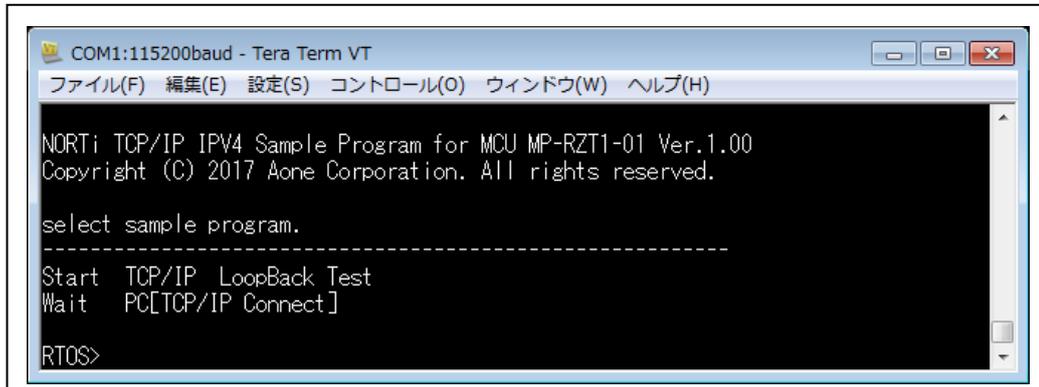


- ①PC機と接続するRS232Cケーブルは、市販「クロスケーブル」でも可能です。
- ②USB-シリアル変換ケーブルを使用される場合は、「StarTech.com社 ICUSB232FTN」推奨
- ③自作する場合は、下記の配線になります。



1) 「Demo_NORTi_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



```

COM1:115200baud - 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)

NORTi TCP/IP IPV4 Sample Program for MCU MP-RZT1-01 Ver.1.00
Copyright (C) 2017 Aone Corporation. All rights reserved.

select sample program.
-----
Start TCP/IP LoopBack Test
Wait PC[TCP/IP Connect]

RTOS>
  
```

TeraTerm pro の画面にオープニングメッセージが表示されます。



```

COM1:115200baud - 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)

?
---MP-RZT1---
HELP      : HELP [Command]
LOOP      : LOOP ON/OFF
DATE      : RTC Set/Get Date
TIME      : RTC Set/Get Time
MEM1      : Memory Access ( 8bit)
MEM2      : Memory Access (16bit)
MEM4      : Memory Access (32bit)
E2P0      : EEPROM Access used EtherCAT
E2P1      : EEPROM Access used Ethernet MAC
E2P2      : EEPROM Access used WiFi/etc.
FRAM      : FRAM Access
MSD       : μSD Read Sample
SWM       : MCU Get DIP-SW
LEDM      : MCU Set LED
DELAY     : Software Delay Test
CM3START  : Cortex-M3 is execute & EtherCAT Demo Start.
WDOG      : WatchDogTimer Test
PowerDown : NMI->Mem[0x2001_0000]->FRAM[0x0]Size[0x8000] Write
           : FRAM Writing LED3[ON]
---EV-RZT1---
SWB       : BaseBord Get SW
LEDB      : BsideBord Set LED
AN003     : AN003 12bitAD Sampling
CTEXT     : Text Write [X:0->19] [Y:0->3] ["Text"]
RSCAN     : RSCAN-0 External Loop Back Test
WIFIS     : Start WiFi SPI-mode Loopback Test
ENC       : Input Encoder
PWM3      : MTU3a to Complementary PWM Mode(6 phase)
---TCP/IP[NORTi]---
IP        : display IP configuration
PING      : ping command
DHCP      : DHCP client test
SNTP      : SNTP client test & RTC Set Date/Time
DNS       : DNS client test
GMDATE    : UTC time
-----
KEY[?]    : ? [Command][HELP]
KEY[*]    : Recall Command
KEY[<]    : Left Move
KEY[>]    : Right Move
KEY[BSP]  : Back Space
KEY[ESC]  : Abort
[.]       : Parameter
  
```

「?」または「HELP」で
HELP画面を表示する。
「? 各コマンド」で個別表示

3-1-4. 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 記述はスペース表現とします。以下省略

LOOP 0 ↵ //LOOP 解除 ↵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 ↵ //DATE2015_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 ↵

//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 ↵ //TIME9_0_0 9時0分0秒

TIME ↵

//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/W/F} 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 0 ↵ //0x2000_0000から0x100要素分(0)8bitFILL

MEM2 F 0x2000_0000 0x100 0 ↵ //0x2000_0000から0x100要素分(0)16bitFILL

MEM4 F 0x2000_0000 0x100 0 ↵ //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 ↵ //0x2000_0000から0x100要素分(0++)16bitFILL

MEM4 I 0x2000_0000 0x100 0 ↵ //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}
- ・ 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_RZsamp1↵

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 【GNU-RTOS 版は未対応】

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 基板上的 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 基板上的 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 $\{0\sim 19\}$ $\{0\sim 3\}$ $\{Text\}$ ␣ // 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

//ESC キー入力で終了します。

ex)

It is connecting SPI-Mode

rsi_bootloader()

.

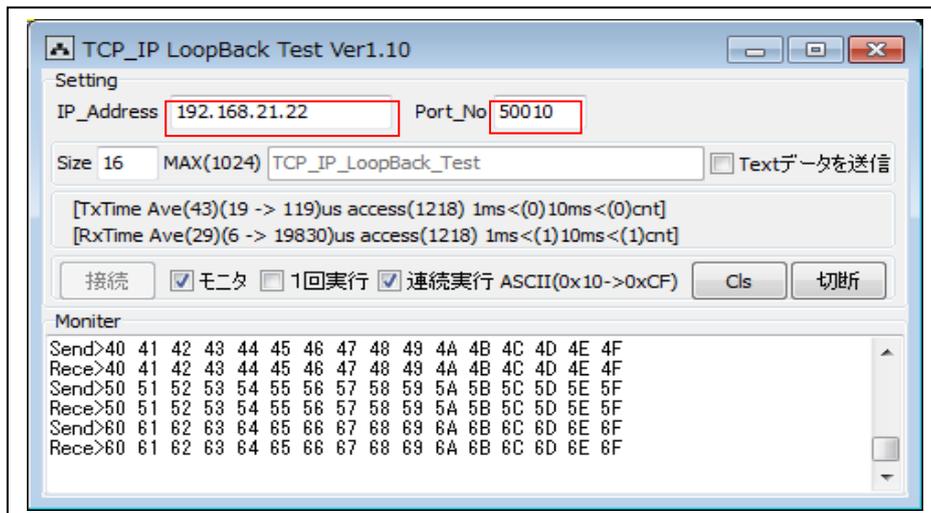
rsi_socket()

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 【RTOS 版は未対応】

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

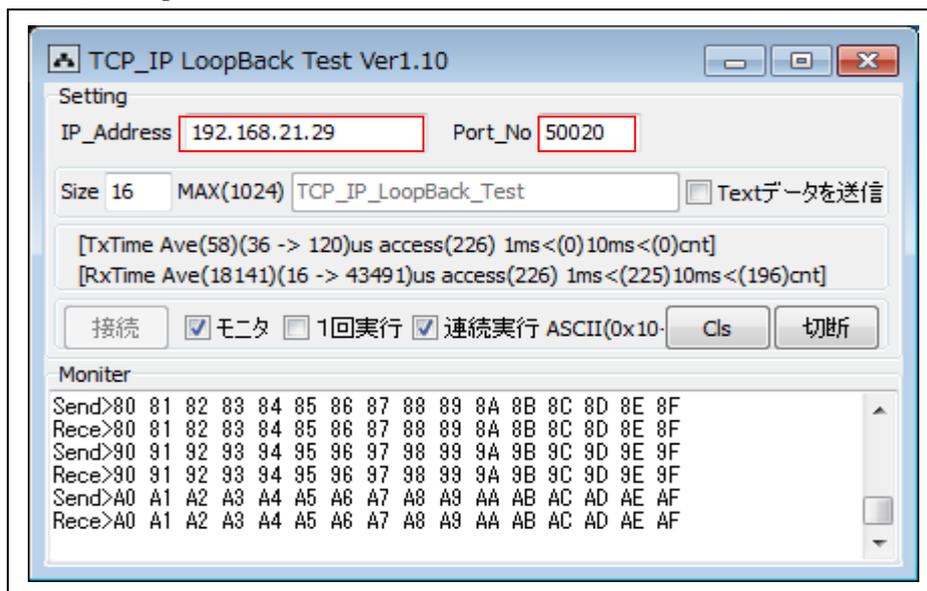
```
at+rsi_band=0
at+rsi_init
at+rsi_numscan=4
at+rsi_authmode=3
at+rsi_network=INFRASTRUCTURE
at+rsi_scan=0
at+rsi_psk=7kimcrdfssk34
at+rsi_join=WHR-G301N,0,2
at+rsi_ipconf=1,0,0,0
at+rsi_ltcp=50020
```

 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 相カウンタ、ホール素子状態、各相ドライバの AD 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 AD 値]
- ⑥V[V 相ホール素子状態[0/1] V 相ドライバ AD 値]
- ⑦U[U 相ホール素子状態[0/1] U 相ドライバ AD 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、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]

DHCP コマンド

DHCP 処理を実行

DHCP↵

ex)

Successfully assigned by DHCP server (eth0)

```
[Ethernet Address ]:[D8-80-39-91-79-A9]
[My IP Address   ]:[192.168.21.10]
[Default IP Address]:[192.168.21.19]
[Default Gateway ]:[192.168.21.126]
[Subnet Mask     ]:[255.255.255.0]
[DHCP            ]:[192.168.21.63]
[DNS             ]:[192.168.21.126]
[PORT Number    ]:[50000]
```

IP コマンド

DHCP 等により内部登録された、ネットワークインターフェイス名、IP アドレス、ポート番号を表示する。

```
IP<↵
```

ex)

```
RZsamp1 : 192.168.21.10 : 50000
```

PING コマンド

PING コマンドを発行する。

```
PING_<xxx.xxx.xxx.xxx>↵
```

ex)

```
PING_<192.168.21.22>↵
```

```
32 bytes from 192.168.21.22: icmp_seq= 1, time= 2 ms
```

```
32 bytes from 192.168.21.22: icmp_seq= 2, time= 1 ms
```

```
32 bytes from 192.168.21.22: icmp_seq= 3, time= 1 ms
```

SNTP コマンド

SNTP サーバーにアクセスして標準時間を取得し、RTC に年月日曜日と時分秒を設定する。

```
SNTP<↵
```

ex)

```
Tue Mar 31 17:49:29 2017
```

```
DATE 2017/03/31[Tue]
```

```
TIME 17:49:29
```

DNS コマンド

DNS サーバーにアクセスして IP アドレスを取得する。

```
DNS_<{ドメイン名}>↵
```

ex)

```
DNS_<www.yahoo.co.jp>↵
```

```
www.yahoo.co.jp has address 182.22.31.252
```

GMDATE コマンド

SNTP コマンドにより取得したグリニッジ標準時間を表示する。

GMDATE↵

ex)

Tue Mar 31 08:53:18 2017

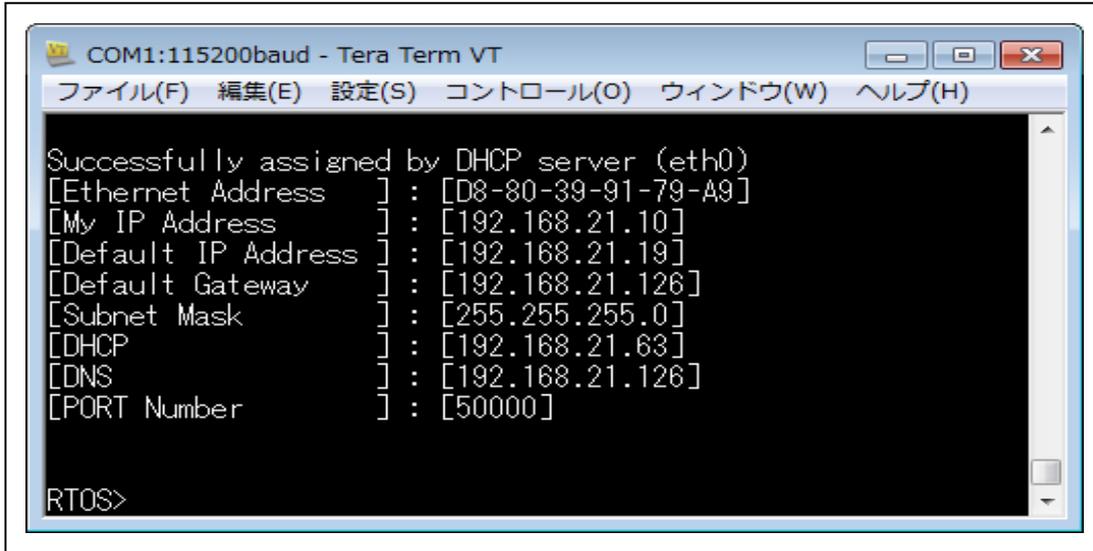
KEY 操作

簡単な 1 ラインエディタ機能を入れてあります。

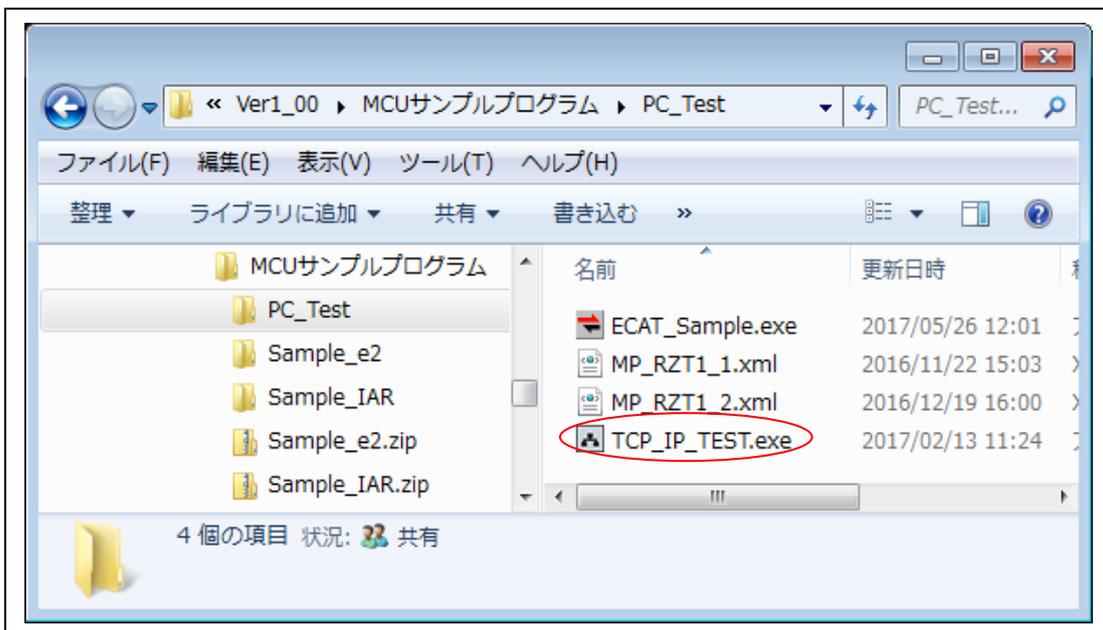
- BS バックスペース
- ← 左にカーソル移動
- → 右にカーソル移動
- ↑ 1回前に入力した内容のリコール
- ESC コマンド処理中の中断

4. TCP_IP Loopback テストツール

- 1) 「DHCP」 コマンドにより IP アドレスを取得する。

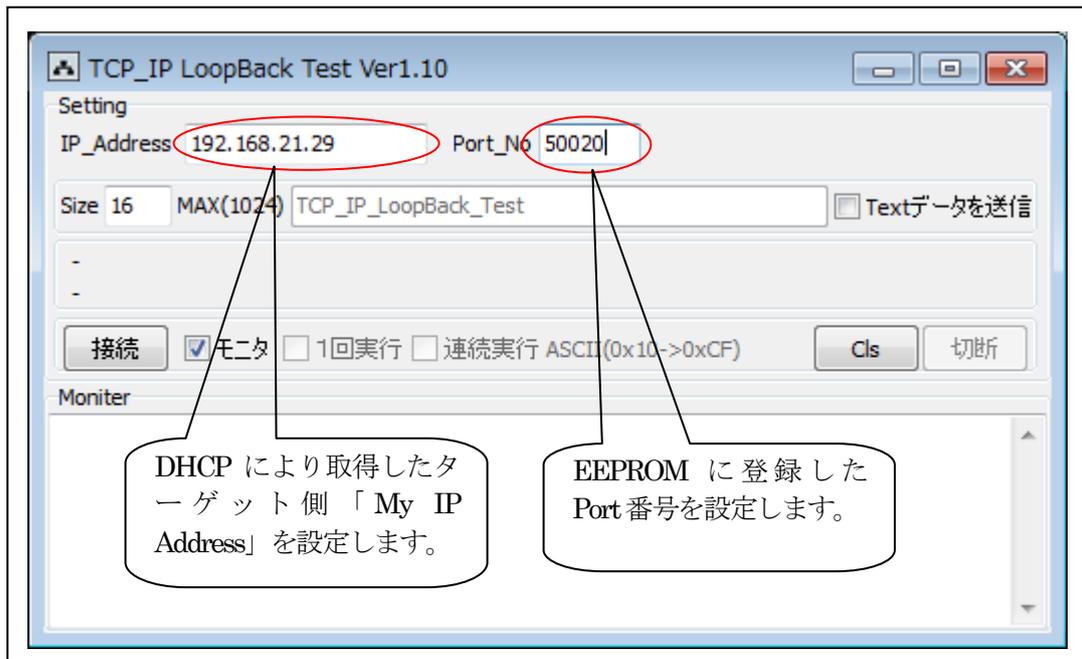


- 2) TCP_IP Loopback テストツールの起動

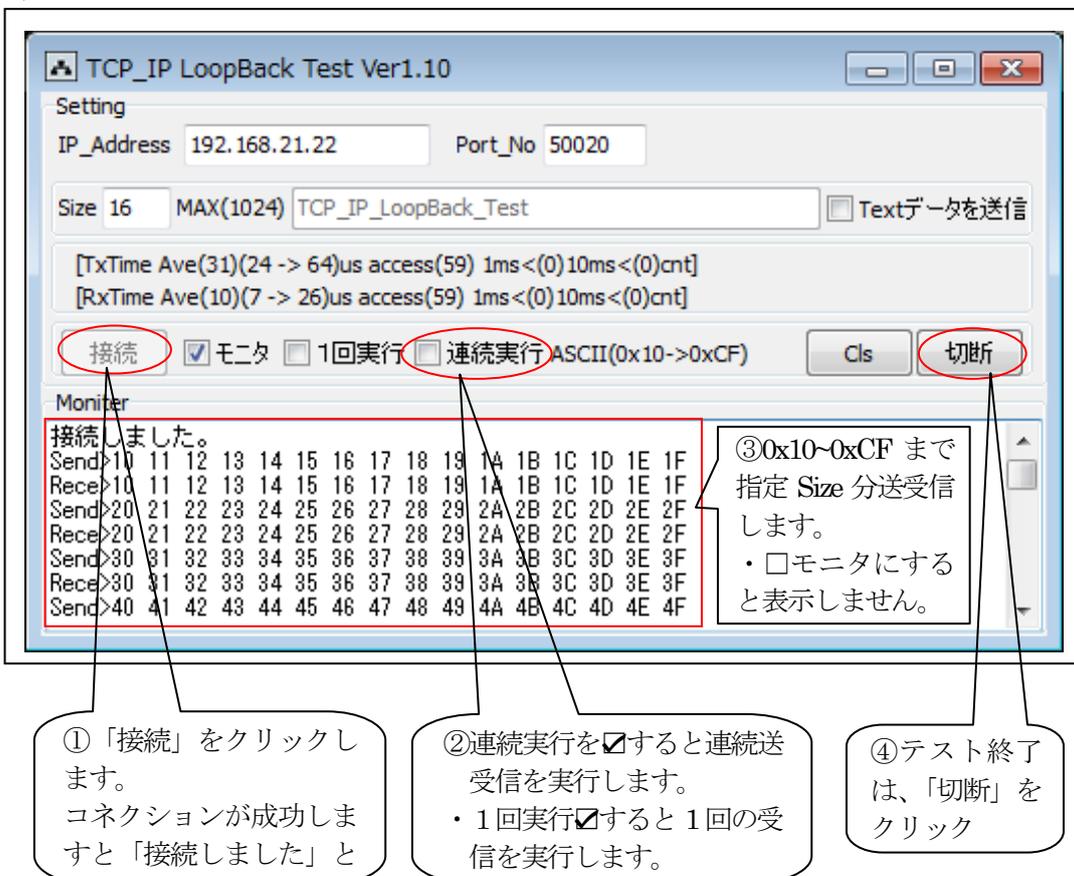


- <MCU サンプルプログラム> – <_PC_Test> の 「TCP_IP_TEST.exe」 を直接クリック
- インストール操作不要

3)



4)



- 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ミスポ社および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。
- ・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を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
- ・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。
- ・NORTiは、ミスポ社の登録商標、または商品名称です。
- ・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
- ・「J-Link User Guide」 SEGGER社
- ・「J-Flash User Guide」 SEGGER社
- ・「Flasher User Guide」 SEGGER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>