

Rev 1.40.00

Renesas RZ/A1H 用サンプル(ARMCC RTOS-NORTi版)の説明

(EV-RZ-xx+MP-RZA1H/FPGA-xx 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

サンプルプロジェクト名				
EVRZ_NORTi MCU 基板(MP-RZA1H/FPGA-*)		ソース公開		
	評価用基板(EV-RZ-*)用サンプル	(有償)		
EVRZ_NORTi_USB	MCU 基板(MP-RZA1H/FPGA-*)	実行ファイ		
	評価用基板(EV-RZ-*)	ルのみ添付		
	USB-Function 機能を追加したサンプル			

統合開発環境	コンパイラー
DS-5(バージョン 5.20.2)	armcc(バージョン 5.05)

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。			
<u>注*1</u>			
USED_DEFnano=x	x=DEFnanoを使用[1]する・[0]しない。		
RTOS	NORTi 使用時に定義		
CH2	NORTi 使用時に定義		
ITF_LIB	USB-Function 使用時に定義		
ARMC	USB-Function 使用時に定義		

ASM ソースに IF等のマクロ定義している場合に使用します。		
<mark>注*1</mark>		
pd='USED_DEFnanoEQU x''	x = DEFnanoを使用[1]する・[0]しない。	

サンプルプロジェクト別に必要なマクロ定義例				
EVRZ_NORTi	RTOS	CH2		
EVRZ_NORTi_USB	RTOS	CH2	ITF_LIB	ARMC

<mark>注*1</mark>

「_USED_DEFnano_=0」と使用しない側に定義しても内蔵 RAM へのダウンロードとシリア ルフラッシュ ROM への書き込み操作は可能です。ただし、再操作する場合はターゲット側のリ セット操作が必要になります。

1

- 2-1.「EVRZ_NORTi」プロジェクトの説明
 - 1)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。

2) フォルダ構成とファイル名

Sample_ARMC¥EV-RZ¥EVRZ_NORTi					
	Debug	ビルドにより生	により生成された実行ファイル等が格納される場所		
	NORTi	空	インストール済み NORTi オリジナルからの ンポート手順書		
		ReadMe.txt			
	NORTi_smp	NETSMP	inc	NETSMP のインクルード	
			aonettcp.c	TCP/IP ループバックテスト	
			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	
			nonshel.c	コマンド shell	
			nonsntp.c	SNTP処理	
			nontelc.c	TELNET サンプル	
			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		
			ethza1h.c	LANドライバーの初期化	
			n4iza1h.c	NORTi 使用の MCU 依存部	
			nonecfg.c	TCP/IP サンプルコンフィグ	
			(1		
			nonethw.c	ETHER 割り込み登録 CIO 用いりマルドライバ	
			nosrzaln.c		
			phyza11.c		
			mmuav /a.s	MIMU 処理 ナリンナル CMD の亦更層度	
	coattor filo	conttorisant	Readivie.txt	スリンノル SIVIP の変更履歴	
	scatter_file	dwl Sample de		ンロード用スクリプト	
	Seript	ini Sample.ds		化用スクリプト	
		rst Sample.ds	ULINK OB-	ゲットリセット用スクリプト	
	src app	inc	src app のイン	クルード用ディレクトリ	
	- 11	main_r.c	メイン処理 RZ/A1Hの初期化処理 LED・SW等の処理ソフト BSC 初期化処理 OS タイマー処理ソフト RTC の初期化と処理ソフト FRAM の初期化と read/write 処理		
		cevrza1h.c			
		board.c			
		bsc.c			
		ostm_r.c			
		rtc.c			
		sfram.c			
		spibsc.c	SPIBSC の初期	化と read 処理	
command.c		コマンド処理			

r	1		
	src_eva	inc	src_eva のインクルード用ディレクトリ
		e2p.c	EEPROMの read/write 処理
		riic_comm.c	RIICの初期化と read/write 処理
		rscan.c	RSCAN の初期化と read/write 処理
		sci_comm.c	SCIの初期化と read/write 処理
		usb_func.c	ITF_USBLibの使用サンプル
	src_evb	inc	src_evb のインクルード用ディレクトリ
		c_lcd_fpga.c	キャラクタ LCD 表示処理
		m_lcd_fpga.c	モノクロLCD表示処理
		pwm1.c	DCモータ(PWM出力)制御処理
		tp_mode.c	タッチパネル(RIIC)制御処理
	src_vdc	inc	src_vdcのインクルード用ディレクトリ
		c_font.c	半角英数字のフォントテーブル(5x7)
		k_font12.c	漢字フォントテーブル(12x12)
		dvdec.c	DVDEC 初期化・コントロール処理
		vdc5.c	VDC5初期化・コントロール処理
		vfont.c	グラフィック LCD 文字出力・表示処理
		video.c	ビデオコマンド管理処理
		vin.c	VINコントロール処理
		image.c	色識別(赤・緑・青)コントロール処理
		vram.c	ビデオ RAM 定義
	src_sys	inc	src_sysのインクルード用ディレクトリ
		_vector_table.s	リセットベクターテーブル
		_init_handler.s	割り込みハンドラー処理
		_rst_handler.s	リセット時の ARM 初期化処理
		profile.c	プロファイル処理
		Nmonitor.c	デバッグモニター処理
1	1	1	

- 3) 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
 - a. サンプル「EVRZ_NORTi」を PC 機の適当なフォルダに全 Copy します。または、 「Sample_ARMC.zip」の圧縮ファイルを適当なフォルダで解凍します。
 - b. 2) 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	\rightarrow	NORTi¥INC
NORTi¥LIB	\rightarrow	NORTi¥LIB
NORTi¥SRC¥nosio.c	\rightarrow	NORTi¥nosio.c

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



- c. 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。
- d. アプリケーションノート「DS5_Inport.pdf」を参考にして DS-5 にインポートして下さい。

4) 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



④端末の設定

Tera Term: 端末の設定	36/1*	× USB シルアル-
端木サイス(T):		I USB 2 5 / 10 / 10 / 10 / 10 / 10 / 10 / 10 /
■ X 43		CR コードがオ
図='フィント'フワイス(S):	KIBIMI CR+LF - H + > CA	トされる設定の
	ヘルプ(H)	● 合は、 <mark>受信:</mark> ● にして下さい
端末ID(I): VT100 -	□ローカルエコー(L):	
応答(A):	自動切り替え(VT<->TEK)(U):	
漢字-受信(K) 漢字-道		
SJIS • SJIS	★ 漢字1ノ(N) 【\$B ★ 】	
	-)3/)) /∉+/ /(0, <u>[KB ▼</u>]	
ロケール(C) japanese	言語コート(P) 932	

赤丸の設定にする。

5) 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」を推奨
 ③自作する場合は、下記の配線になります。

EV-RZ-xx

DSUB 9pin		
オス		
1	NC	
2	RXD •	
3	TXD	
4	NC	
5	GND •	
6	NC	
7	RTS	
8	CTS	
9	NC	

	PC-Windows	
	DSUB 9pin	
	オス	
	1	NC
	2	RXD
	3	TXD
	4	DTR
	5	GND
	6	DSR
•	7	RTS
••	8	CTS
	9	NC

④MCU 基板上の SW1 設定



SW1-1 ON 起動時に DHCP サーバーへのアクセスを禁止する。 SW1-2 OFF MP-RZA1H/FPGA-xx 用サンプルを指定 SW1-3 OFF 7inchAM-800480SBTMQW-T00 (ON) 5inch AM-800480L1TMQW-TNAH-I 7inchAM-800480SETMQW-T00H-I 接続時

6)「EVRZ_NORTi」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。



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 - Tara 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 LOOP HELP LOOMMANN HELP 画面を表示する。
DATE : RTC Set/Get Date 「2 タコマン」、「「田田」
TIME : RTC Set/Get Time 「? 谷ユマノト」 C1個別衣小
MEMI : Memory Access (8b) MEM2 : Memory Access (16b;t)
MEMAZ : Memory Access (10011) MEM4 : Memory Access (32bit)
FRAM : FRAM Access
SWM : MCU Get DIP-SW
LEDM : MCU Set LED WDOG : WatchDogTimer Test
DELAY : Software Delay Test
STB : Software Standby Mode Test
PowerDown : NMI->MemL0x2000_0000J->FRAML0x0JSizeL0x8000J Write
MP-R7A1H/FPGA
SWF : FPGA/CPLD Get DIP-SW
LEDF : FPGA Set LED
F2P · FEPROM Access
RSCAN : RSCAN-3 External Loop Back Test
IP : display IP configuration
PING : ping command DHCP : DHCP alignst test
SNTP : SNTP client test & RTC Set Date/Time
DNS : DNS client test
GMDATE : UTC time
SWB : BaseBord Get SW
LEDB : BsdeBord Set LED
PWM : PWM1 Output Test
UIEXI : Text Write [X:U->T9] [Y:U->3] [Text] MTEXT : Text Write [X:0] [Y:0->3] [″Text]
EV-RZ-01[Graphic]
VMODE : Disp Mode [1(全画面) 1(1A)/2(2A)/3(1B)/4(2B)]
: L2(2分割) I(IA:IB)/2(2A:2B)」 ・ 「A(A公室II) 1(1A・1B)/2(2A・2B)]
VPAINT : 全面面Paint[0:全消去 Color:1->5]
VFONT : Font Property Set [Color:1->5]
: [倍率:1->10 1(12x12)] MTEVT : Tt Weite [Y:0 \700] [Y:0 \700] ["Tt"]
MTEXT : Text WFIte LX:U-2700」LT:U-2470」LText 」 TMAGE : 白詞識 「A:stop 1:RED 2:GREEN 3:RITE]
KEY[?] : ? [Command]{HELP}
KEY[<] : Kecall Command KEY[<] : Left Move
KEY[>] : Right Move
KEY[BS] : Back Space
KEYLESCJ : Abort
L] . rarameter [] : Comment

2-1-1. 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
 必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。	//TIME_9_0_0	9時0分0秒
TIME	現設定データを表示	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ジ // 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ジ // 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_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	{0x1800_0000~0x18FF_FFFF}
・内蔵 RAM エリア	{0x2002_0000~0x209F_FFFF}

{Read/Write Memory $\mathcal{T} \vdash \mathcal{V} \mathcal{X}$ }

・MCU 内蔵 RAM エリア	{0x2000_0000~0x2001_FFFF}
・FPGA側I/Oエリア	{0x4800_0000 ~0x4800_7FFF}
・FPGA 内蔵 RAM エリア	{0x4800_8000~0x4800_BFFF}
・MCU内蔵周辺モジュール	{周辺モジュールの仕様による

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_0000~0x2001_FFFF}
• Memoryアドレス	{0x2000_0000~0x209F_FFF}

SWM コマンド

MCU側が制御している DIP-SW1の状態を表示します。

SWM<┚

ex)

MCU DIP-SW1_1[ON/OFF] SW1_2[ON/OFF] SW1_3[ON/OFF] SW1_4[ON/OFF]

LEDM コマンド

MCU側で制御している LED1/2/3を点灯・消灯します。

LEDM_{0/1}_{0/1}_{0/1}// // LEDM {LED1} {LED2} {LED3} 0:消灯 1:点灯

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。

DELAY コマンド

MCU内部で利用している lusec タイマーの精度を計るため LED1 を点滅させます。

DELAY_{Time 値}usecc // DELAY 10 10 usec の精度

①LED1{time 值》点灯
 ②LED1{time 值》消灯
 ③LED1{time 值》点灯
 ④LED1{10msec} 消灯

STB コマンド

ソフトウェア・スタンバイ・モードに移行させます。 STB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。

LED1 点灯

FRAM(0x0)から内蔵 RAM(0x2000_0000)の内容を 32Kbyte 分 Write する。 LED1 消灯

LED1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。

SWF コマンド

FPGA側が制御している DIP-SW2 の状態を表示します。

SWF☆

ex)

FPGA/CPLD DIP-SW2_1[ON/OFF] SW2_2[ON/OFF]

LEDF コマンド

FPGA 側で制御している LED5/6 を点灯・消灯します。

LEDF_{0/1}_{0/1}/2 // LEDF {LED5} {LED6} 0:消灯 1:点灯

RSCAN コマンド

RSCAN-3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 E2P コマンド EEPROMのRead/Write 処理をします。 E2P {R/W} EEPROM アドレス {メモリアドレス} サイズ {READ} E2P_R_EEPROMアドレス」サイズ ex) E2P, 0x0, 0x100 // EEPROM の 0x0 番地から 0x100 サイズ分ダンプ表示 {WRITE} E2P、W、EEPROMアドレス、メモリアドレス、サイズ ex) E2P_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。 $\{ Read Only Memory \mathcal{T} \vdash \mathcal{V} \mathcal{X} \}$ ・EEPROMエリア $\{0x80 \sim 0xFF\}$ 内蔵 RAM エリア {0x2002 0000 ~0x209F FFFF}

{Write Memoryアドレス} ・EEPROMエリア ・内蔵 RAM エリア

{0x0~0x7F} {0x2000_0000~0x2001_FFFF}

E2P コマンドには LAN 用データ設定コマンドが用意されています。

①E2P_MAC {Read Only} MAC の表示

②E2P_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} E2P_NAME

{Write} E2P_NAME_ (名前) // MAX 7 ASCII 文字 E2P_NAME_RZsamp1 ジ

2015-102株式会社

③E2P_PORT {Read/Write} PORT番号の表示と設定 {Read} E2P_PORT& {Write} E2P_PORT_{xxxx} // E2P_PORT_50000 4E2P_IP {Read/Write} デフォルトIPアドレスの表示と設定 {Read} E2P_IP¢┚ {Write} E2P_IP_{xx.xx.xx}公 // E2P_IP_192.168.21.12 5E2P_SUB {Read/Write} サブネットマスクの表示と設定 {Read} E2P_SUB¢┚ {Write} E2P_SUB_{xx.xx.xx,xx} // E2P_SUB_255.255.0 / 6 E2P_GATE {Read/Write} ゲートウェイアドレスの表示と設定 {Read} E2P_GATE {Write} E2P_GATE_{xx.xx.xx.xx} // E2P __GATE __ 192.168.21.126 쉬 ⑦E2P_DHCP {Read/Write} DHCPアドレスの表示と設定 {Read} E2P_DHCP⇔ {Write} E2P_DHCP_{xx.xx.xx.xx} // E2P __GATE __ 192.168.21.63 쉬

⑧E2P_DNS {Read/Write}DNSアドレスの表示と設定

{Read}

E2P_DNS☆

{Write}

E2P_DNS_{xx.xx.xx,xx}// // E2P_GATE_192.168.21.126 /

<mark>注記</mark>

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。

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 : 192.168.21.21 : 50000

PING コマンド

PING コマンドを発行する。

PING_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

DHCP コマンド

DHCP処理を実行

DHCP↔

ex)

 Successfully assigned by DHCP server (RZsamp1)

 [Ethernet Address
]: [00-1E-C0-F0-E8-7F]

 [My IP Address
]: [192.168.21.21]

 [Default IP Address
]: [192.168.21.12]

 [Default Gateway
]: [192.168.21.126]

 [Subnet Mask
]: [255.255.255.0]

 [DHCP
]: [192.168.21.126]

 [DNS
]: [192.168.21.126]

 [PORT Number
]: [50000]

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 2015 DATE 2015/03/31[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}の

ex)

DNS_www.yahoo.co.jp

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 2015

SWB コマンド

EV-RZ-xx 側の SW2_2~5の状態を表示します。

SWB¢⊐

ex)

BaseBord-SW2_2[ON/OFF] SW2_3[ON/OFF] SW2_4[ON/OFF] SW2_5[ON/OFF]

LEDB コマンド

EV-RZ-xx 側の LED2~5 を点灯・消灯します。

LEDB_{0/1}_{0/1}_{0/1}_{0/1}// // LEDB {LED2} {LED3} {LED4} {LED5} 0:消灯 1:点灯

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}

MTEXT コマンド

モノクロ LCD に英数文字/漢字を表示します。

MTEXT_{0}_{0~3}_{文字}/2 //MTEXT {X:0 固定}{Y行}{文字}

PWM コマンド

DCモータの回転とデモ運転します。

PWM // SWB [5:++duty 4:-duty 3:demo 2:exit]

キャラクタ LCD とモノクロ LCD にデモ表示します。



VMODE コマンド

カラーグラフィック LCD の表示モードとカメラ入力信号の番号を指定します。

VMOD_{画面モード}_{1:全画面 // +{VIN 指定}---1:全画面 // +{VIN 指定}---1:1A/2:2A/3:1B/4:2B // {画面モード}-2:2分割画面 // +{VIN 指定}---1:(1A:1B)/2:(2A:2B) // {画面モード}-4:4分割画面 // +{VIN 指定}---1:(1A:1B)/2:(2A:2B)



VPAINT コマンド

カラーグラフィック LCD の全画面を指定色でペイントします。

VPAINT_{色番号}/J // 色番号}- 0:消去 1:Red 2:Green 3:Blue 4:Black 5:White

VFONT コマンド

カラーグラフィック LCD に描画する文字の色と倍率を指定します。

VFONT_{後番号}_{倍率}// 《色番号}-- 0:消去 1:Red 2:Green 3:Blue 4:Black 5:White // {倍率}---- 1~10倍

VTEXT コマンド

カラーグラフィック LCD に {VFONT} コマンドで指定した色と倍率で文字を描画します。

VTEXT_{0~799}_{0~479}_{文字}/2 //VTEXT {X:列ドット}{Y:行ドット} 英数漢字}



IMAGE コマンド

CCD カメラにより入力した画像(カラーグラフィック LCD)から指定色を認識します。

<内部パラメータ変更>

IMAGE _4_{RESOL 値} _ {DEAD 値} _ {2 値化の検出間隔 2

// (RESOL 値) 1->16 分析する BOX 値
// 仮に4とした場合 4x4の BOX
// (DEAD 値) 1->16 連続判断の不干渉値
// 仮に6とした場合、6BOX 以上
// 判定位置終了とする。
// 検出間隔) 1->8 2 値化判定の Y 方向間隔
// 仮に2とした場合、指定 BOX の
// の Y 方向を1つ飛びに2値化判定

・終了は、IMAGE_0⁴⁹を入力します。



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

- 2-2. 「EVRZ_NORTi_USB」 プロジェクトの説明
 - 1) 動作説明
 - ・EVRZ_NORTiに USB-Function 機能を追加したプロジェクトになります。
 - ・各コマンド体系は、2-2項を参照して下さい。
 - ・USB-Function ライブラリーは、別途有償にて提供しております。ご購入前の評価用として実行用ファイルは添付しております。

2) フォルダ構成とファイル名(評価用)

Sam	ple_ARMC¥_		
	EV-RZ¥_EVRZ_USB	EVRZ_NORTi_USB.mot	実行用 Hex ファイル
	_PC_Test	ITF_USB_TESTEXE	PC用 USB テストプログラム
		DRIVER¥ITFUSBLib	PC側 USB ドライバー
		TCP_IP_TEST.EXE	PC用 TCP/IP テストプログラム

3) フォルダ構成とファイル名(有償用) ご購入 ITFUSBLib_RZA1H_xx を添付

Sample_	Sample_ARMC¥EV-RZ¥EVRZ_NORTi_USB		
	Debug	ビルドにより生成された実行ファイル等が格納される場所	
	ITF_LIB	空	
		ReadMe.txt	ITF_LIB オリジナル CD からのインポート手
			順書
	lnk_NORTi	空	EVRZ_NORTi¥NORTiにリンク
	lnk_NORTi_smp	空	EVRZ_NORTi¥NORTi_smpにリンク
	scatter_file	scatter.scat	ロケート定義用スキャッタファイル
	script	dwl_Sample.ds	ULINKのダウンロード用スクリプト
		ini_Sample.ds	ULINKの初期化用スクリプト
		rst_Sample.ds	ULINKのターゲットリセット用スクリプト
	lnk_app	空	EVRZ_NORTi¥src_appにリンク
	lnk_eva	空	EVRZ_NORTi¥src_evaにリンク
	lnk_evb	空	EVRZ_NORTi¥src_evbにリンク
	lnk_vdc	空	EVRZ_NORTi¥src_vdc にリンク
	lnk_sys	空	EVRZ_NORTi¥src_sys にリンク

「EVRZ_NORTi_USB」に黄色部分(有償)をインポートします。

- 4) ITF_LIBオリジナル CD (有償) からサンプル CD にインポートする手順
 - a. 3)表の黄色部は、空ディレクトリになっていますので、ITF_LIBオリジナル CD から必要なファイルを Copy します。

ITF_LIBオリジナル CD		サンプル(EVRZ_NORTi_USB)
ITF_LIBi¥Include	\rightarrow	ITF_LIB¥Include
ITF_LIB¥ITF_Include	\rightarrow	ITF_LIB¥ITF_Include
ITF_LIB¥ITFUSBLIB_xx.a	\rightarrow	ITF_LIB¥ITFUSBLIB_xx.a

上記のように ITF_LIB オリジナル CD から、サンプル「EVRZ_NORTi_USB¥ITF_LIB」の空デ ィレクトリに Copy して下さい。



5)動作構成(電源OFF)



- 6) 動作手順
 - a. ターゲット基板側の電源 OFF の状態で上図の★USB-Function 評価用 USB ケーブル以外 を接続する。
 - b. ターゲット基板側の電源をONにする。
 - c. デバッガ「DEFnano」を立ち上げる。
 - d. デバッガ「DEFnano」画面の左下隅の「Start」をクリックする。
 - e. デバッガ「DEFnano」の【オプション】- 【フラッシュ ROM ライター】を起動する。
 - f. 無償評価用 Hex ファイル「EVRZ_NORTi_USB.mot」をシリアルフラッシュ ROM へ書き 込みをする。
 - g. ターゲット側の電源をOFFにする。
 - h. デバッガ用 USB ケーブルを抜き取る。
 - i. ★USB-Function 評価用 USB ケーブル を PC 機に接続する。
 - j. RS232C ケーブルが PC 機に接続されているのを確認後、「TeraTerm pro」を起動する。
 - k. ターゲット基板側の電源を ON にする。



TeraTerm pro の起動画面

- 7) USBファンクションの動作確認
 - a. Windows が、USB ドラーバーのインストールを要求しますので USB-Driver をインスト ールする。

「Sample_ARMC¥_PC_Test¥ DRIVER¥ITFUSBLib」 にドライバーがあります。

b. Windows 側のテストプログラム「ITF_USB_TEST. EXE」を起動する。





- 8) TCP/IP プロトコル通信の動作確認
 - a. TCP/IP 通信に必要な情報を EEPROM 登録する。(ルータの設定データ) 2-1 「EVRZ_NORTi」プロジェクトの説明
 - 6) 各コマンドの説明・E2P コマンド を参照

ex)

 Successfully assigned by DHCP server (RZsamp1)

 [Ethernet Address
]: [00-1E-C0-F0-E8-7F]

 [My IP Address
]: [192.168.21.21]

 [Default IP Address
]: [192.168.21.12]

 [Default Gateway
]: [192.168.21.126]

 [Subnet Mask
]: [255.255.255.0]

 [DHCP
]: [192.168.21.126]

 [DNS
]: [192.168.21.126]

 [PORT Number
]: [50000]

上記の全情報を登録しないと TCP/IP 通信は出来ません。

b. MCU 基板上の SW1 設定



SW1-1OFF[ON]-起動時に DHCP サーバーへのアクセスを禁止する。[OFF]起動時に DHCP サーバーへのアクセスを許可する。SW1-2OFFMP-RZA1H/FPGA-xx 用サンプルを指定

- c. ターゲット基板側の電源を入れ直します。
 - ・EEPROMに情報が全て登録されていない場合は、 TeraTerm 画面に「There is no LAN data in EEPROM」と表示します。
 - ・サンプルソフトは電源 ON 時にルータに対して DHCP 処理(IP アドレスの取得処理)を 実行します。



DHCP 処理が成功すると 「My IP Address」に取得し た IP アドレスを表示します。

🌽 エーワン株式会社

d. Windows 側のテストプログラム「TCP_IP_TEST.EXE」を起動する。





以上です。

- 3. 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。
 - ・本サンプルプログラムに関して、インターフェイス社、ミスポ社、ARM社、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。

🌽 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。
- 4. 商標
 - ・ARM DS-5は、ARM 社の登録商標、または商品名称です。
 - ・RZ および RZ/A1H 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。

5. 参考文献

- ・「RZ/A1H 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
- ・「armcc ユーザガイド DUI 0472JJ」 ARM 社
- ・「アセンブラの使用 DUI 0473GJ」 ARM 社
- ・「リンカの使用 DUI 0474GJ」 ARM 社
- ・「コンパイラリファレンスガイド DUI 0328BJ」 ARM社
- ・「アセンブラリファレンス DUI 0489GJ」 ARM社
- ・「armkink リファレンスガイド DUI 0804AJ」 ARM 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