Rev 1.00.00

Renesas RZ/A1H 用サンプル(IARC ベアメタル版)の説明

(EV-RX/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

サンプルプロジェクト名			
IAR_debug_rz	MCU 基板(MP-RZA1H/FPGA-*)	ソース公開	
	単体サンプル		
EVrxRZ_Sample	MCU 基板(MP-RZA1H/FPGA-*)	ソース公開	
	評価用基板(EV-RX/RZ-*)用サンプル		
EVrxRZ_Sample_USB	MCU 基板(MP-RZA1H/FPGA-*)	実行ファイ	
	評価用基板(EV-RX/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
EWARM(バージョン 7.50.2)	iccarm(バージョン 7.50.2)

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。		
注*1 x = DEFnano を使用[1]する・[0]しない。		
	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>			
USED DEFnano =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				
EVrxRZ_Sample				
EVrxRZ_Sample_USB	ITF_LIB			

注*****1

「_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。

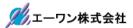


2-1.「IAR_debug_rz」プロジェクトの説明

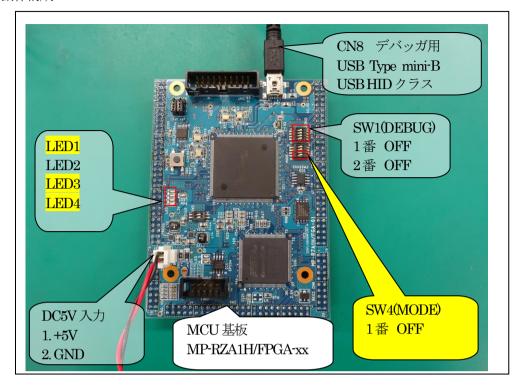
- 1) IAR_debug_rzの動作
 - ・デバッグツール「DEFnano」等にて「IAR_debug_rz.out」をダウンロードして実行させる。
 - ・基板上の LED をメインループ 200 回ごとに LED3 を点滅
 - ・基板上のLEDをOSタイマー割り込みによりLED4を20msec毎に点滅
 - ・電源 DWN 検出 LED1 を点灯、FRAM に内蔵 RAM[0x2000_0000]から Size[0x8000]書き込む

2) フォルダ構成とファイル名

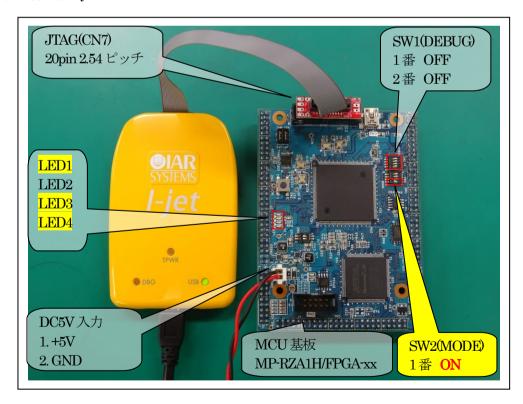
Sample_IAR¥IAR_debug_rz				
	Debug	Exe	IAR_debug_rz.out #ABSファイル	
			IAR_debug_rz.mot #Hex ファイル	
		List	IAR_debug_rz.map // MAPファイル	
		Obj	*.cout	
			*.0	
			*.pdi	
	config	smpza1h.icf	ロケート定義用ファイル	
	src_app	inc	src_app のインクルード用ディレクトリ	
		main.c	メイン処理	
		board_d.c	LED・SW 等の処理ソフト	
		ostm_d.c	OSタイマー処理ソフト	
		sfram.c	FRAM の初期化と read/write 処理	
		spibsc.c	SPIBSC の初期化と read 処理	
	src_sys	inc	src_sys のインクルード用ディレクトリ	
		_vector_table_s.s	リセットベクターテーブル	
		_init_handler_s.s	割り込みハンドラー処理	
		_rst_handler_s.s	リセット時の ARM 初期化処理	
		resetprg.c	システム初期化と CPG 初期化処理	
		intc_table.c	IRQベクターテーブル	
		interrupt.c	割り込みコントロール処理	
		Umonitor.c	デバッグモニター処理	



3)動作構成



4)動作構成(I-jet)



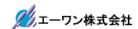


2-2.「EVrxRZ_Sample」プロジェクトの説明

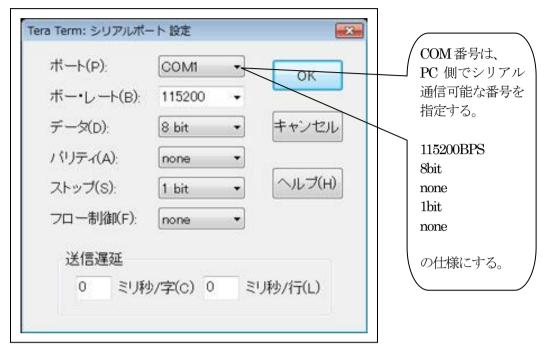
- 1)動作説明
 - ・Tera Term からのコマンド指示により各デバイスを動作させる。
 - ・各コマンド体系は後記にて説明します。

2) フォルダ構成とファイル名

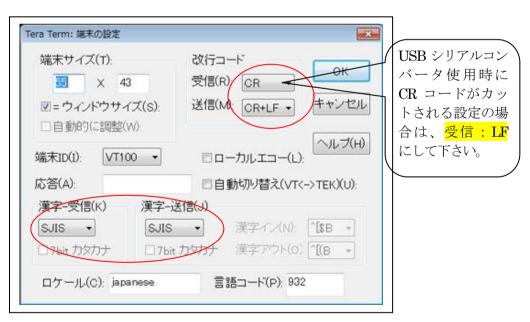
Sample_IAR¥EV-RXRZ¥EVrxRZ_Sample			
	Debug	Exe	EVrxRZ_Sample.out #ABSファイル
			EVrxRZ_Sample.mot #Hexファイル
		List	EVrxRZ_Sample.map //MAPファイル
		Obj	*.cout
			*.O
			*.pdi
	config	smpza1h.icf	ロケート定義用ファイル
	src_app	inc	src_app のインクルード用ディレクトリ
		main_s.c	メイン処理
		board.c	LED・SW 等の処理ソフト
		bsc.c	BSC初期化処理
		ostm_s.c	OSタイマー処理ソフト
		rtc.c	RTCの初期化と処理ソフト
		sfram.c	FRAM の初期化と read/write 処理
		spibsc.c	SPIBSC の初期化と read 処理
		command.c	コマンド処理
	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_sys	inc	src_sys のインクルード用ディレクトリ
		_vector_table_s.s	リセットベクターテーブル
		_init_handler_s.s	割り込みハンドラー処理
		_rst_handler_s.s	リセット時の ARM 初期化処理
		resetprg.c	システム初期化と CPG 初期化処理
		intc_table.c	IRQベクターテーブル
		interrupt.c	割り込みコントロール処理
		Umonitor.c	デバッグモニター処理
	l		



- 3) 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

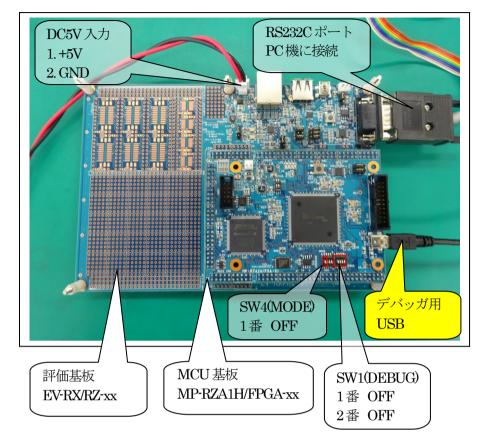


④端末の設定

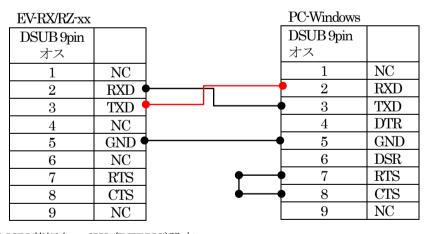


赤丸の設定にする。

4)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。



④MCU 基板上の SW1(DEBUG)設定



SW1-1 OFF ベアメタル版では影響なし

SW1-2 OFF MP-RZA1H/FPGA-xx 用サンプルを指定

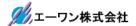
⑤EV-RX/RZ-01のJP設定

JP1/JP2

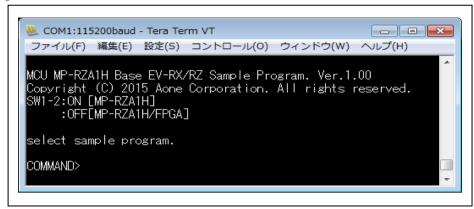
<mark>2側</mark>

JP4

open(1/2/3)

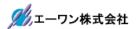


5)「EVrxRZ_Sample」プロジェクトのプログラムを MCU 基板にダウンロードして実行させます。



TeraTerm pro の画面にオープニングメッセージが表示されます。





2-2-1. 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 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。

TIME コマンド

MCU 内蔵の RTC に時間を設定します。

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}



{Read Only Memory アドレス}

・シリアルフラッシュ ROM エリア
 ・内蔵 RAM エリア
 {0x1800_0000 ~0x18FF_FFFF}
 {0x2002_0000 ~0x209F_FFFF}

{Read/Write Memory アドレス}

MCU内蔵RAMエリア
 FPGA側I/Oエリア
 FPGA内蔵RAMエリア
 MCU内蔵周辺モジュール
 (0x2000_0000~0x2001_FFFF)
 (0x4800_0000~0x4800_7FFF)
 (0x4800_8000~0x4800_BFFF)
 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 $7 \mid V \mid x$ {0x0 \sim 0x7FFF}

Store アドレス {0x2000_0000 ~0x2001_FFFF}
 Memory アドレス {0x2000_0000 ~0x209F_FFFF}

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 1usec タイマーの精度を計るため LED1 を点滅させます。

DELAY _{Time 値}usec / DELAY 10 10usec の精度

- ①LED1{time 値} 点灯
- ②LED1{time 値} 消灯
- ③LED1{time 値} 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_{01}_{01}
 // LEDF {LED5} {LED6} 0:消灯 1:点灯

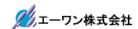
RSCAN コマンド

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}_サイズ心

 $\{READ\}$

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、(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}

・内蔵 RAM エリア {0x2002_0000 ~0x209F_FFFF}

{Write Memory アドレス}

・EEPROM エリア $\{0x0 \sim 0x7F\}$

・内蔵 RAM エリア {0x2000_0000 ~0x2001_FFFF}

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



2-3.「EVrxRZ_Sample_USB」プロジェクトの説明

1)動作説明

- ・EVrxRZ_Sample 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。

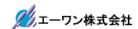
2) フォルダ構成とファイル名(評価用)

S	Sample_IAR¥_				
	EV-RXRZ\(\frac{1}{2}\)EVrxRZ_USB	EVrxRZ_Sample.USB.mot	実行用 Hex ファイル		
	_PC_Test	ITF_USB_TESTEXE	PC用テストプログラム		
		DRIVER¥ITFUSBLib	PC側 USB ドライバー		

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_IAR¥EV-RXRZ¥EVrxRZ_Sample_USB				
	Debug	Exe	EVrxRZ Sample.out #ABSファイル	
			EVrxRZ_Sample.mot #Hexファイル	
		List	EVrxRZ_Sample.map // MAPファイル	
		Obj	*.cout	
			*.0	
			*.pdi	
	ITF_LIB	空		
		ReadMe.txt	ITF_LIB オリジナル CD からのインポート	
			手順書	
	config	smpza1h.icf	ロケート定義用ファイル	
	lnk_app	空	EVrxRZ_Sample¥src_app にリンク	
	lnk_eva	空	EVrxRZ_Sample¥src_eva にリンク	
	lnk_sys	空	EVrxRZ_Sample¥src_sys にリンク	
	I			

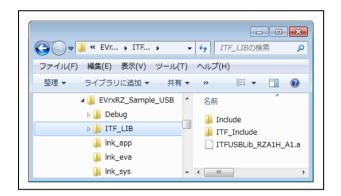
無償評価用「EVrxRZ_Sample_USB」に黄色部分(有償)をインポートします。



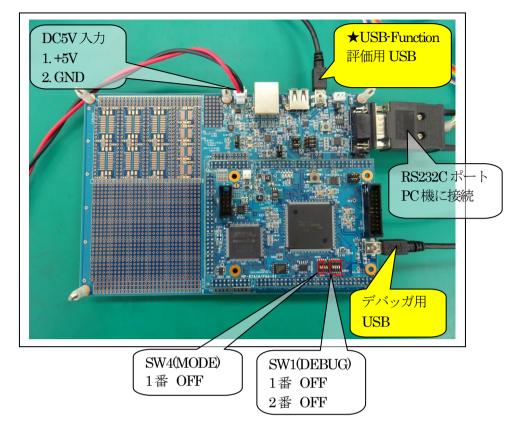
- 4) ITF_LIBオリジナル CD (有償) からサンプル CD にインポートする手順
 - a. サンプル CD「EVrxRZ_Sample_USB」を PC 機の「EVrxRZ_Sample」と同じフォルダに 全 Copy します。または、「Sample_IAR.zip」を適当なフォルダで解凍します。
 - b. 2) 表の黄色部は、空ディレクトリになっていますので、ITF_LIBオリジナルCDから必要なファイルをCopy します。

ITF_LIBオリジナル CD		サンプル(EVrxRZ_Sample_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「EVrxRZ_Sample_USB¥ITF_LIB」の空 ディレクトリに Copy して下さい。

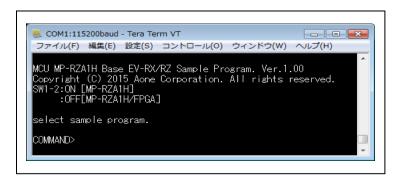


5)動作構成(電源OFF)

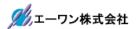


6)動作手順

- 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「EVrxRZ_Sample_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。



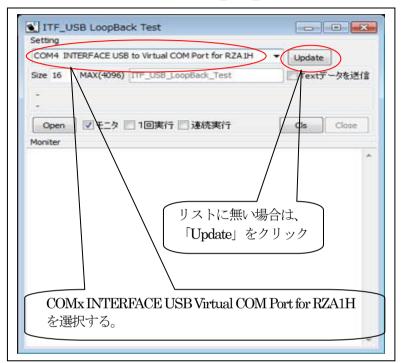
TeraTerm pro の起動画面

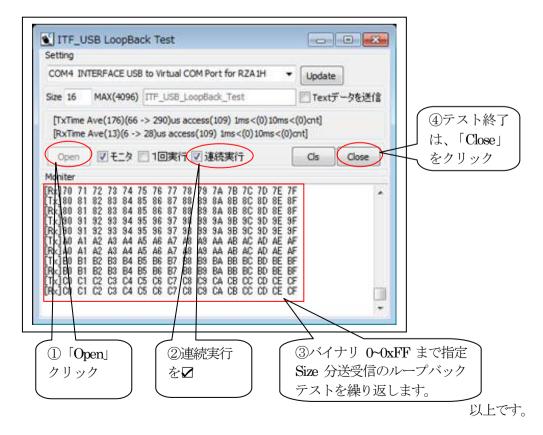


1. Windows が、USB ドラーバーのインストールを要求しますので USB-Driver をインストールする。

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

- m. TeraTerm pro 画面でコマンド「USBFや」を入力する。
- n. Windows 側のテストプログラム「ITF_USB_TEST.EXE」を起動する。





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、IAR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。
- ・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

- ・EWARMは、IAR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
- ・「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」 IAR社
- ・「IAR アセンブラリファレンスガイド」 IAR社
- ・「IARデバッグプローブガイドIARprobes-2-j」 IAR社
- その他

T486-0852

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