RX 開発環境の使用方法

3.1版 2023年10月02日

1. 概要

1. 1 概要

本アプリケーションノートでは、RX シリーズで使用する開発環境についての解説を行います。 解説を行う開発環境は以下の3つになります。

- 1. RX ファミリ用 C/C++コンパイラパッケージ
- 2. フラッシュ開発ツールキット (FDT)
- 3. E1エミュレータ

開発環境の使用方法は、弊社サンプルプログラムを使用して解説しています。 各 CPU ボードのサンプルプログラムと CPU ボードの固有の設定値は、対応する各 CPU ボードのアプリケーションノート 「サンプルプログラム解説」を参照してください。

1. 2 開発環境について

本サンプルプログラムは統合開発環境 High-performance Embedded Workshop (本書では以下 HEW と記述します)を用いて 開発されています。

名称	バージョン	用途	備考
統合開発環境 HEW	V.4.08.00	統合開発環境	
RX ファミリ用 C/C++コンパイラパ ッケージ	V.1.01 Release 00	コンパイル、ビルド	評価版使用可能 V.1.00 Release 01 不可 [※]
フラッシュ開発ツールキット	V.4.06 Release 01	FlashROM 書き込み	評価版使用可能
E1エミュレータ	V1.01	デバッグ、FlashROM 書き込み	

※ RX ファミリ用 C/C++コンパイラパッケージは、V.1.00 Release01 から V.1.01 Release00 にアップデートされるにあたり、 プロジェクトの構成が変更されているため V.1.01 Release00 用のプロジェクトをそのまま V.1.00 Release01 でコンパイル することは出来ません。

1.3 本書の読み方について

本アプリケーションの記述について説明します。

本アプリケーションの記述のうち、赤い四角で囲まれた内容はその操作における選択箇所を示しています。 Fig 1.3-1の例では、「新規ワークスペースの作成」を選択し、「OK」を選択するという操作を意味します。

<u>ک</u>
選択項目
ニストレーション(<u>A</u>)

Fig 1.3-1 選択箇所の例

本アプリケーションノートの記述のうち、青い下線が引かれ番号が付記された項目は、別途対応する CPU ボードの アプリケーションノート内の「4. 開発環境使用時の各設定値」を参照してください。

Fig 1.3-2の例は、①の操作で指示されている読み込み対象のワークスペースの名称が、別途対応する CPU ボードの アプリケーションノートの該当箇所に記述されていることを意味します。

1)HEW を起動し、サ	トンプルフ	プログラムの <u>ワークスペース</u> を読み <u>込みます。</u> 2-1 参照項目
2)最初の読み込みを	行なった。	ときに、「ワークスペース(Workspace)が移動しま
		本アプリ	ケーションノート内の記述例
4	. 開発環境使用時の名	設定值。	
4 4	 開発環境使用時の名 開発環境を使用する際の アプリケーションノート ビルド・動作確認方法。 	ト設定値。 AP-RX62T-OA ・「RX開発環境。	▲固有の設定を以下に示します。↓ の使用方法」と合わせて使用してください。↓
4 ب	開発環境使用時の名 開発環境を使用する際の アプリケーションノート ビルド・動作確認方法。 項目名。	 設定値。 AP-RX62T-0A 「RX開発環境。 「RX開発環境。 項目番号。 	8 圏有の設定を以下に示します。~ の使用方法」と合わせて使用してください。~ 設定値~
4 4	開発環境使用時の名 開発環境を使用する際の アプリケーションノート ビルド・動作確認方法。 項目名。 ワークスペースファイル名。	 ・設定値。 ・ ・ ・	A 圏有の設定を以下に示します。 ~ の使用方法」と合わせて使用してください。 ~ 設定値。 ¥Samp Le¥ap_rx62t_0a.hws。 参照先
4 با	 開発環境使用時の名 開発環境を使用する際の アプリケーションノート ビルド・動作確認方法。 項目名。 ワークスペースファイル名。 出力フォルダ。 	 ・ 設定値 ・ ・ ・	A 固有の設定を以下に示します。~ の使用方法」と合わせて使用してください。~ 設定値。 ¥Samp le¥ap_rx62t_0a.hws。 ¥Samp le¥ap_rx62t_0a/bws。

Fig 1.3-2 外部参照の例(本文)

また、Fig1.3-3のように図示中に青い四角で囲まれ数字が付記されている場合も、同じように別途対応する CPUボードのアプリケーションノート内の「4.開発環境使用時の各設定値」を参照してください。

フラッシュ関発ツート	したっトはモスサス型フラッシュ	マイマンをサポートします。		
プロジェクトで使用す	なデバイス名を下記リストが	いてコンをリホートしょう。 ら選択して下さい。 、"を避知して下ざい		
0.100mae0000/978 c	Cideneric BOOT Device	ZABINUC FOUS		
フィルタ				Other
Туре	Full Name	Kernel Version	Info	
R32C	R5F64601	1_0_00	4	
R32C R32C	R5F64601_ECC R5F6460F	1_0_00	<i>i</i>	
R32C	R5F6461F	1_0_00	参照項目	
R32C R32C	R5F64610 R5F64610 FCC	1_0_00		
R32C	R5F64611	1_0_00	۲ ۲	
R32C R32C	R5F64611_ECC R5F6461F	1_0_00		
Prior Generic De	vice RX600 Series	N/Ã	<u>3-2</u>	
Generic BOOT D	evice Generic BOOT De	vi N/A		
				<u>.</u>
		< 戻る)(B) 次へ(N) >	キャンセル
	 本アプリケ	ーションノートロ	内の図示例	
	本アプリケ	ーションノートロ	内の図示例	
フラッシュ闘発ツ-	本アプリケ		内の図示例 	
フラッシュ開発ツ- 項目名	本アプリケ -ルきっとを使用した Flast 項目番号	· ーションノート 1書き込み方法 設定値	内の図示例 	
フラッシュ開発ツ- 項目名 ボード設定 (Flash	本アプリケ -ルきっとを使用した Flash 項目番号 書き込み) 3-1	ーションノート 1書き込み方法 設定値 Fig 41を参照	内の図示例 	
フラッシュ開発ツ- 項目名 ボード設定 (Flash デバイス名	本アプリケ -ルきっとを使用した Flast 項目番号 書き込み) 3-1 3-2	- ションノート 1 書き込み方法 設定値 Fig 41 を参照 Generic BOOT Device	内の図示例	

Fig 1.3-3 外部参照の例(図示)

2. ビルド・動作確認方法

2.1 ビルド方法

HEW を使用してサンプルプログラムをビルドする方法を説明します。

① HEW を起動すると以下のようなウィンドウが表示されますので、「キャンセル」を選択します。

ようこ て !	? 🛛
デ 新規プロジェクトワークスペースの作成(C)	OK
	アドミニストレーション(A)
□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	

② HEW のメニューから「ファイル(<u>F</u>)」→「ワークスペースを開く(<u>R</u>)...」を選択し、サンプルプログラムの ワークスペースファイルを読み込みます。

> 2-1 🛞 High-performance Embedded Workshop ファイル(E) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッ □ 新規作成(N) Ctrl+N motion (2)... Ctrl+O 閉じる(<u>C</u>) Ctrl+F4 新規ワークスペース(E). ワークスペースを開く(<u>R</u>)。 フークスペースの保存(型) ワークスペースを開く ? 🗙 🚽 🗢 🖻 💣 📰 ファイルの場所仰 🚞 ap_rx62t_0a 🛅 ap_rx62t_0a 🛛 **2**-🖓 ap_rx62t_0a.hws 2-1 ap_rx62t_0a.hws ファイル名(N): 選択 ファイルの種類(①: HEW Workspaces (*.hws) キャンセル -

③ 最初の読み込みを行なったときには、「ワークスペース (Workspace) が移動しました」という内容の 確認メッセージが表示されますので「はい」を選択します。 ④ 最初の読み込みを行なったときに、コンパイラバージョンによって、バージョンの選択を行なうダイアログが表示されることがあります。表示された場合には、使用するコンパイラバージョンを選択してください。

⑤ メニューの「ビルド (<u>B</u>)」→「ビルド (<u>B</u>)」を実行してください。

🏟 ap_rx62t_0a - High-performance En	nbedded Workshop
ファイル(E) 編集(E) 表示(V) プロジェクト(P)	ビルド(B) デバッグ(D) 基本設定(U) ツール(I
D 🚅 8 🖉 & X 🖻 8 0	RX Standard Toolchain
X	🕸 コンパイル(F) Ctrl+F7
⊡	🏙 ビルド(<u>B</u>) F7
⊟	ゴベ(をヒルド(A)

⑥ 出力フォルダにモトローラファイル (.mot)、アブソリュートファイル (.abs)、マップファイル (.map) が

2-2	2-3	2–4	2–5
出力されます。			

2.2 動作確認方法

(1) FlashROM 書き込み

フラッシュ開発ツールキットを使用して FlashROM に書き込みを行い、動作を確認します。 詳細は、「3.フラッシュ開発ツールキットを使用した FlashROM 書き込み方法」を参照してください。

(2) E1 エミュレータを使用したデバッグ

E1 エミュレータを使用し、HEW 上でのデバッグを行います。 詳細は、「4.E1 エミュレータを使用したデバッグ方法」を参照してください。

3. フラッシュ開発ツールキットを使用した FlashROM 書き込み方法

弊社サンプルプログラムを、フラッシュ開発ツールキットを使用して FlashROM に書き込む方法を説明します。

- ① CPU ボードを Flash 書き込み時のボード設定に変更し、CPU ボードと PC をシリアルケーブルで接続します。
- ② フラッシュ開発ツールキット(以下、FDTと記述します)を起動します。

3-1

③ FDT を起動すると、以下のようなウィンドウが表示されますので、「新規プロジェクトワークスペースの作成」を選択し、「OK」ボタンをクリックします。

ようこそ!	<u>? ×</u>
新規プロジェクトワークスペースの作成(©)	OK
○ 最近使用したブロジェクトワークスペースを開く(@):	
ーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーー	

 ④ 新規プロジェクトワークスペースの設定ウィンドウが表示されますので、ワークスペース名及びプロジェクト名を 「ap_rx_fdt」と設定します。

その後、プロジェクトスペースを保存するディレクトリを選択します。「参照」ボタンを押し、「C:¥work」ディレクトリを選択します。

(※ ワークスペース名、プロジェクト名およびプロジェクトスペースの保存先は、お客様の環境に合わせて 任意で設定可能です)

全ての設定が完了したら、「OK」ボタンを押します。

プロジェクト プロジェクト ap_rx_fdt プロジェクト名(P): ap_rx_fdt ディレクトリ(D): C¥work¥ap_rx_fdt CPU種別(C): All Flash Devices ツールチェイン(T):	×
None ד ספולדי	
 OK キャンセル	

⑤ プロジェクトスペースが作成されると、デバイスとカーネル選択のウィンドウが表示されます。
 サンプルプログラムの対応 CPU に合わせてデバイス名を選択します。

3–2

デバイス名を選択したら、「次へ」ボタンを押します。

rルタ					Other
Type	Full Name	Kernel V	ersion	Info	
R32C	R5F64601	1000		-4	
R32C	R5F64601 ECC	1 0 00			
R32C	R5F6460E	1 0 00			
R32C	R5F6461F	1000			
R32C	R5F64610	1_0_00			
R32C	R5F64610_ECC	1_0_00			
R32C	R5F64611	1_0_00			
R32C	R5F64611_ECC	1_0_00			
R32C	R5F6461E	1_0_00	2 0		
Prior Generic Device	RX600 Series	N/A	3-2		
Seneric BOOT Device	e Generic BOOT Devi	N/A			-

⑥ 通信ポートのウィンドウが表示されますので、CPUボードで使用する通信ポートに合わせて通信ポートを選択します。
 通信ポートを選択したら、「次へ」ボタンを押します。

通信ポート		×
	フラッシュ開発ツールキットは、標準のPCシリアルポートま ドの接続をサポートします。こ使用の通信ポートを選択	はびUSBアダプタボー して下さい。
	Select port: COM1	
	E8Direct,E8aDirect,E1Direct,E20Directを選択した場 す。COM×,USB Direct,E8,E8aを選択した場合はDirec す。	合はブランクになりま t Connectionになりま
	Select Interface: Direct Connection	×
	■ デバイスとの切断時にResetピンをLowレベルに設定	する
	〈戻る(B) (次へ(1))	**>セル



⑦ 以下の注意ウィンドウが表示されたら、CPUボードの電源を投入します。電源を投入したら、「OK」ボタンを押します。

FLASH De	evelopment Toolkit
<u>.</u>	FDT will now attempt to connect to your generic device. Please ensure the board is connected, powered and in Boot mode.
	OK キャンセル

⑧ デバイス確認が開始されると、デバイスの確認ウィンドウが表示されます。

確認が完了すると、「OK」ボタンが押せるようになりますので、「OK」ボタンを押します。 デバイスの確認中にエラーが発生した場合は、ボードの設定や通信ポートの接続などを確認し、 再度デバイスの確認を行ってください。

汎用デバイン	への確認	×
\checkmark	Booting Device	
\checkmark	Sending Supported Device	es Inquiry
\checkmark	Selecting Device	RX600 Series
~	Sending Clock Mode Inqu	iry
\checkmark	Selecting Clock Mode	0
~	Sending Other Inquiries	
		OK キャンセル

⑨ デバイス確認の後、デバイス設定ウィンドウが表示されます。

<u>入力クロック</u>、メインクロックの逓倍比、周辺クロックの逓倍比を入力してください。

設定値の入力が完了したら、「次へ」ボタンを押します。

Worker to be as as as to the	デバイス設定用の値を入力してください。	
T AN DA CLEMA AD SD 19 AL	[RX600 Series] using [Protocol C]	
Workspace 'Industrial Ce	外部クロックまたは内部クロックを選択し External Clock てください:	<u>.</u>
Target files on	入力クロック: 3-3 12 M	Hz
0 55 65 F9 91 91 91 LSD.mob)F 1 4 75 E0 81 T SjoKeyboard.m 1 55 84 33 4 SjoComms.mot	クロックモード: 0 🗾	
Device Image	メインクロックの逓倍比 (CKM): 3-4 8 I	
8 52 14 20 6 20 00 00 00 00 00 00 00 00 00 00 00 00	周辺クロックの逓倍比 (CKP): 3-5 4 I	
0 47 EY 84 80 83 67 00 1E		

⑩ 次に、接続タイプウィンドウが表示されます。

「ボーレート(推奨)」を選択し、「Use Default」チェックボックスを選択します。 選択が完了したら、「次へ」ボタンを押します。

Workspace at DA FF 55	フラッシュ開発ツールキットは多くの異なる方法でユーザのデバイスに とができます。	接続するこ
Workspace 'Industrial Co	Select Connection: © BOOT Mode © USER Program Mode	
Target files LCD.mob Keyboard.m Comms.mot Device Image Target files	ブートモードで接続した場合は、フラッシュメモリのデータはあらかじゅ ています。(デバイスの種類に依存します) 通信は調歩同期式で行っております。そのため通信速度はデバイス ロックおよび接続機器により設定できる通信速度が異なります。	の動作ク
27 91 08 1 91 Data mos 70 56 70 5 19 10 10 10 10 10 10 10 10 10 10 10 10 10	ボーレート(ユーザ入力): ボーレート(ユーザ入力):	ult

次に、書き込みオプションウィンドウが表示されます。

書き込みオプションを、それぞれ「Protection」を「Automatic」、「Messaging」を「Advanced」、 「Readback Verification」を「No」に選択します。 選択が完了したら、「完了」ボタンを押します。

Water	フラッシュ書き込み時の保護レベルはどれにしますか?
TANK WAT TO TO TO THE T	Automatic C Interactive None
Display	書き込み済みブロックへの書き込みを認識した場合は書き込み前に自動的(ブロック消去します。
Target files	出力メッセージレベルはどれにしますか? Monoraging
1 SE 8A 33 - Si Comme mot	C Standard C Advanced
Device Image	詳細なメッセージを出力します。
6 F0 56 F0 51 A DE AS 55 64 85 97	書き込み完了後、リードバックベリファイを実行しますか? Readback Verificat <u>ion</u>
0 14 04 40 75 54 AD 20 F6 3 8 47 EF 84 80 83 57 90 12 1	O Yes 📀 No

12 設定が完了すると、デバイスへの接続が開始されます。

正常に接続が完了すると、FDT のログに「接続が成功しました」と表示されます。

⁽¹³⁾ 接続が成功したら、FlashROM に書き込むファイルを選択します。

メニューの「ファイル(<u>F</u>)」→「データファイルを開く(<u>L</u>)...」を選択し、書き込むファイルを指定します。 FlashROM に書き込むファイルは、サンプルプログラムの<u>モトローラファイル</u>を指定します。

				3–6		
ファイル(E) 編集(E) 表示(Y)	プロジェクト	H(P)	ツール(I)	ウィンドウ(<u>W</u>)	デバイス	Z(D) ^
閉じる(_)	Ctrl+F4	T		• #h	8 R	
新規ワークスペース(E) ワークスペースを開く(W) ワークスペースの(呆存(V) ワークスペースを閉じる(K)		4	id id			
	Ctrl+R					
□ 上書き保存(5) 名前を付けて保存(点)	Ctrl+S					
最近使ったワークスペース(P)		F				
最近使ったデータファイル(E)		F				
終了(X)						

④ 書き込みファイルの指定が完了すると、ファイルの内容が FDT の画面に表示されます。
 画面の表示を確認したら、メニューの「デバイス(<u>D</u>)」→「対象ファイルのダウンロード(<u>L</u>)」を選択します。

ウィンドウ(<u>W</u>)	デバイス(ロ) ヘルプ(田)		
_ № 8	デバイスとの接続(⊆) タ。デバイスとの切断(□)	Ctrl+Alt+C	
<u> </u>	🥭 ブロック消去(E)	Ctrl+Alt+E	
	ジェブランクチェック(B)	Ctrl+Alt+B	
	<u> 🎻 アップロード(1)</u>	Ctrl+Alt+U	
F FF FF	🔊 対象ファイルのダウンロード(山)	Ctrl+Alt+P	FF FF F
F FF FF F	· Z ノラッシュのチェックサム(<u>5</u>)	Ctrl+Alt+5	FF FF F
F FF FF F	😫 アドレスジャンプ(⑤)	Ctrl+Alt+G	FF FF F
	▶ 実行(R)	Ctrl+Alt+R	FF FF F
F FF FF F	💱 ブロックのロック(L)	Ctrl+Alt+L	FF FF F
'F FF FF F 'F FF FF F	ID第 IDコードの設定(I)	Ctrl+Alt+T	FF FF F FF FF F
'F FF FF F 'F FF FF F	● 操作の中止(№)	Ctrl+Break	FF FF F FF FF F
'F FF FF F 'F FF FF F	が プロジェクトの設定(P)	Alt+Shift+R	FF FF F FF FF F
F FF FF F F FF FF F	`F FF FF FF FF FF FF FF FF FF FF FF FF FF	FF FF FF FF	FF FF F FF FF F

- ⑤ 「対象ファイルのダウンロード(L)」を選択すると、書き込みが開始されます。 正常に書き込みが完了すると、FDTのログに「書き込みが完了しました」と表示されます。
- ⑩ 書き込みが完了したら、メニューの「デバイス(<u>D</u>)」→「デバイスとの切断(<u>D</u>)」を選択します。

- ₩ K		Ctrl+Alt+C	
16 7 AN An	グブロック消去(E)	Ctrl+Alt+E	
F FF FF F	(1) クランクテェック(回)	Ctrl+Alt+U	FF
4 77 77 77 7 7 77 77 7 7 77 77 7 7 77 77	 Σ フラッシュのチェックサム(5) 2 アラッシュのチェックサム(5) 	Ctrl+Alt+S	FF I FF I FF I
F FF FF F F FF FF F F FF FF F F FF FF F	★ アドレスクィンク・ハリン・ハリン・トレスクインク・ハリン・ハリン・ハリン・ハリン・ハリン・ハリン・ハリン・ハリン・ハリン・ハリン	Ctrl+Alt+R	FF I FF I
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	「 IDコードの設定…(D)	Ctrl+Alt+T	FF I
F FF F		Ctrl+Break	FF 1 FF 1
F FF FF F	♪ ノロンIクFの読え走(ピ) F FF F	AIC+SOIFC+R	FF I

⑦「デバイスとの切断(<u>D</u>)」を選択すると、ブロックのロックウィンドウが表示されます。
 「何もしない」を選択し、「0K」ボタンを押します。

赛続時のロック状態	切断時のロック状態		What should FDT do with the
RX600 Series			block locking settings now?
😑 🛄 User Flash			
EB0	Unlocked		
🛄 EB1	Unlocked		○ 設定する
EB2	Unlocked	-	
EB3	Unlocked		 一般定画面を再表示する。
🛄 EB4	Unlocked		
EB5	Unlocked		● 何もしない
EB6	Unlocked		
	Unlocked		
	Unlocked		□ 編集の許可
- () EB9	Unlocked		
	Unlocked	-	

- 「OK」ボタンを押すと、デバイスとの切断が開始されます。
 正常に切断が完了すると、FDTのログに「Disconnected」と表示されます。
 正常に切断されたことを確認したら、CPUボードの電源を切ります。その後、FDTを終了します。
 以上で、FlashROMへの書き込みは完了です。
- (1) CPUボードを<u>サンプルプログラム動作時のボード設定</u>に変更し、再度電源を投入してサンプルプログラムが 3-7

動作することを確認します。

再度同じ設定でファイルを書き込む際は、同じプロジェクトワークスペースを開くことで一部設定を省略できます。 まず、プロジェクトワークスペースを開いた後、メニューの「デバイス(D)」→「デバイスとの接続(C)」を選択します。 その後は、⑪以降の手順に従いファイルの書き込みを行うことができます。 その他の機能など、FDTの詳細につきましては、FDTのユーザーズマニュアルを参照してください。

4. E1 エミュレータを使用したデバッグ方法

弊社サンプルプログラムを、E1 エミュレータを使用してデバッグする方法を説明します。

E1 エミュレータと CPU ボードを接続する場合は、CPU ボードの電源が投入されていないことを確認した上で接続を 行ってください。

- (※ 画像及び画像で使用している設定値は全て弊社製 CPU ボード「AP-RX62T-0A」のものです)
- ① CPU ボードを E1 エミュレータデバッグ時のボード設定にした上で CPU ボードと E1 エミュレータを接続し、HEW を 4-1

起動した上でサンプルプログラムの<u>ワークスペースファイル</u>を開きます。

4-2

② メニューの「デバッグ(D)」→「デバッグの設定(D)...」を選択します。
 デバッグの設定ウィンドウが表示されますので、「ターゲット」と「デバッグフォーマット」を設定します。
 4-3 4-4

		<u> </u>
efaultSession	▼ ターゲット オプション	
🕞 ap Jx62t Da	ターゲット(T): 4-3 FX600 E1/E20 SYSTEM コア(2): Single Core Target デパッグフォーママト(F): 4-4 Elf/Dwarf2 ダウンロードモジュール(2): Filename Offset Address Format	▼ ▼ ▼ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

③ デバッグの設定ウィンドウの「ダウンロードモジュール」を設定するため、「追加」をクリックします。

デバッグの設定				<u>? ×</u>
DefaultSession	ターゲット オプション	1		
	ターゲット(工):			
der aboxostoo	RX600 E1/E20 SYSTEM			
	⊐7(<u>C</u>):			
	Single Core Tare	ret .	*	
	デバッグフォーマット(<u>E):</u>		
	Elt/Dwart2		<u> </u>	
	ダウンロードモジュー	·µ@:		The second second
	Filename	Uttset Address Format		追加密
				変更(<u>M</u>)
				削除(配)
			1	上へ(凹)
				下へ(2)
1	<u>1:</u>			

④ ダウンロードモジュールウィンドウが表示されますので、「オフセット」「ファイルフォーマット」を設定し
 4-5
 4-6

「ファイル名(N)」の「参照(B)...」をクリックします。

ダウンロードモジュール ? 🔀						
オフセット(0):	0000000	4–5	•	OK		
ファイルフォーマット(<u>E</u>):	Elf/Dwarf2	4-6	•	キャンセル		
ファイル名(N):						
アクセスサイズ(<u>A</u>):	1					
「デバッグ情報のみのダウンロード(D)						
「 ダウンロード時のメモリベリファイ(P)						
🔲 ターゲット接続時にタ	『ウンロード(工)					

⑤ ダウンロードモジュールの選択ウィンドウが表示されますので、サンプルプログラムの実行ファイルを選択します。

				4-7
ダウンロードモジュー	ルの選択			? 🛛
ファイルの場所型:	🗀 Debug		- + 🗈	* 💷
👼 ap_rx62t_0a.abs				
				[]
ファイル名(<u>N</u>):	ap_rx62t_0a.abs	4–7		選択
ファイルの種類(工):	ELF/DWARF2 File	⊧s (*.elf;*.abs)	•	キャンセル

⑥ ダウンロードモジュールウィンドウに戻りますので、「OK」を選択します。

ダウンロードモジュール		? 🛛	
オフセット②:	00000000	ОК	
ファイルフォーマット(E):	Elf/Dwarf2	キャンセル	
ファイル名(N):	a¥ap_rx62t_0a¥Debug¥ap_rx62t_0a.abs	(●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●	
アクセスサイズ(<u>A</u>):	1		
□ デバッグ情報のみのダウンロード(D)			
□ ダウンロード時のメモリベリファイ(P)			
□ ターゲット接続時にダウンロード(1)			

⑦ デバッグの設定ウィンドウに戻りますので、「OK」を選択します。

デバッグの設定		? 🛽
DefaultSession ▼	ターゲット オブション ターゲット(T): Rx600 E1/E20 SYSTEM コア©: Sincle Core Target デバッグフォーマット(E): Elf/Dwart2]
	ダウンロードモジュール(型): Fil. Offset Address Format C¥ 00000000 Elf/Dwarf2	追加(台) 。 変更(M)。 削除(名) 上へ(U) 下へ(Q)
	 ОК	キャンセル

⑧ 警告ウィンドウが表示されますので、「はい(<u>Y</u>)」を選択します。

Debugger		
🧘 ବ-ゲット	を変更するとオブションは	空になります。続行しますか?
	いいえ(<u>N</u>)	キャンセル

- ⑨ デバイスウィンドウが表示されますので、「<u>MCU グループ:」</u>および「<u>デバイス名(V)</u>」を設定し「次へ」を
 4-8
 4-9
 - 選択します。

デバイス		Σ	<
<u>M</u> CUグループ:	RX62T Group	4-8	
デバイス名(V):	R5F562TA	4-9	
┌動作モード──			
⊙ デバッグモ			
- ホット3 状態で	アラクイン(ユーザジステム) 「、エミュレータの電源を入	ニエミエレータの接続を外した れてOKを押してくだざい)(H)	
C 75%91	ライタモード(E)		
デバッ:	ガ終了時にユーザーブログ	ラムを実行&	
- 電源供給設定		· 200 A)	
1 121/-	タから電源1共結1日/1歳へ 2277 - こちの7	, 200mA)	
	200 0 200		
_通信設定			
エミュレータ シリアルNo:	E1: 0HS002172	▼ 更新(<u>R</u>)	
		〈 戻る(B) (次へ()) キャンセル	
		〈戻る(日) (次へいの) キャンセル	

⑩ 通信ウィンドウが表示されますので、「JTAG クロック:」を設定し「完了」を選択します。

	4–10		
通信			
JTAG <u>J</u> TAGクロック:	16.5 4−10 - MHz		
	< 戻る(B)	完了	キャンセル

① 接続が始まります。E20RX ウィンドウが表示されますので、CPUボードの電源を投入し「OK」を選択します。

E20RX	\mathbf{X}
ユーザシステムの電源を入	れてからOKを押してください。
COK	キャンセル

4-12

 ② コンフィグレーションプロパティウィンドウが表示されますので、「入力クロック(EXTAL)」と

 「ワーク RAM 開始アドレス」を入力し「OK」を選択します。

 4-11

コンフィジ	ブレーションプロパティ
MCU	システム 内蔵フラッシュメモリの上書き 外部フラッシュメモリ
⊢ MC	
đ	b/作モード(<u>M</u>): シングルチップモード ▼
I	- , <u></u>
7	、 カクロック(EXTAL): 12 4–11 MHz
5	フークRAM開始アドレス (0x400 パ) 「1使用):

 ③ 以上で CPU ボードとの接続は完了です。続いて、サンプルプログラムのダウンロードを行うため ワークスペースウィンドウの「Download modules」に登録されているファイル(⑤で選択したファイル)を 右クリックし、「ダウンロード(<u>D</u>)」を選択します。

⊡ — 当 vecttbl.c ⊡ — 🔄 Download modules ap_rx62t_0a.abs - 0		
Dependencies BordDepend.h Common.h	<mark>ダウンロード(©)</mark> ダウンロード(debug情報のみ)(©) アンロード(U)	
Proj	ダウンロードモジュールの追加(<u>N</u>) 削除(<u>R</u>)	
L OT AL AT 21 21 0	デバッグの設定(S) 	
	 ✓ ドッキングビュー 非表示 	
	プロパティ(<u>P</u>)	

④ 以上でサンプルプログラムのダウンロードが完了し、E1 エミュレータを使用したデバッグの用意が整いました。
 必要に応じてプログラムのデバッグを行ってください。

E1 エミュレータのデバッグ機能などにつきましては、E1/E20 エミュレータユーザーズマニュアルを参照してください。

⑤ E1エミュレータを使用したデバッグを終了する場合は、HEWと CPUボードの接続を解除する必要があります。
 HEW と CPU の接続を切るためには、HEW の「接続解除」ボタンをクリックします。



¹⁰ 以上で HEW と CPU ボードの接続解除は完了です。CPU ボードの電源を切り、E1 エミュレータとの接続を 解除してください。

再度同じソースのデバッグを行う際には、同じプロジェクトのワークスペースを開くことで一部設定を省略することが 可能となっています。

接続解除後に再び CPU ボードとの接続を行うためには、①の操作を行った後に HEW の接続ボタンをクリックします。



これにより E1 エミュレータと CPU ボードの接続が開始されますので、⑨以降の手順に従い接続を確立してください。 接続が確立された後は、⑬以降の手順に従い対象のサンプルプログラムをダウンロードし、デバッグを行ってください。

ご注意

- ・本文書の著作権は株式会社アルファプロジェクトが保有します。
- ・本文書の内容を無断で転載することは一切禁止します。
- 本文書に記載されているサンプルプログラムの著作権は株式会社アルファプロジェクトが保有します。
- ・本文書に記載されている内容およびサンプルプログラムについてのサポートは一切受け付けておりません。
- ・本文書の内容については、万全を期して作成いたしましたが、万一ご不審な点、誤りなどお気付きの点がありましたら弊社までご連絡 下さい。
- ・本文書の内容およびサンプルプログラムに基づき、アプリケーションを運用した結果、万一損害が発生しても、弊社では一切責任を負いませんのでご了承下さい。
- ・本文書の内容は、将来予告なしに変更されることがあります。

商標について

・RX はルネサス エレクトロニクス株式会社の登録商標、商標または商品名称です。

- ・High-performance Embedded Workshop はルネサス エレクトロニクス株式会社の登録商標、商標または商品名称です。
- ・E1 エミュレータはルネサス エレクトロニクス株式会社の登録商標、商標または商品名称です。
- ・フラッシュ開発ツールキットはルネサス エレクトロニクス株式会社の登録商標、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。



株式会社アルファプロジェクト

〒431-3114 静岡県浜松市中央区積志町 834 https://www.apnet.co.jp E-Mail: query@apnet.co.jp