

お客様各位

ZUD F35 10 0202

2011年 03月 31日

ルネサス エレクトロニクス株式会社

MCU 事業本部

ソフトウェア統括部

MCU ツール技術部

RL78 マイクロコントローラ
EEPROM エミュレーション・ライブラリ
Pack01 Ver.1.10
使用上の留意点
(暫定)

ご使用前に必ずお読みください

【 目 次 】

1	アプリケーション・ノート	3
2	動作環境	3
3	対応ツール	3
4	ソフトウェア・リソース	4
5	インストール	6
5.1	インストール方法	6
5.2	アンインストール方法	6
5.3	ファイル構成	7
6	ビルド方法	8
6.1	使用するソフトウェア	8
6.2	CubeSuite でのビルド方法	8
6.2.1	C言語の場合	8
6.2.2	アセンブリ言語の場合	13
7	デバッグ方法	17
	CubeSuite でデバッグを行う場合	17

この添付資料では、本製品のバージョンによる差分などについて記載しています。ご使用前に必ずお読みください。

- ・ Windowsは、米国Microsoft Corporation の米国およびその他の国における登録商標または商標です。
- ・ PC/ATIは、米国IBM社の商標です。

1 アプリケーション・ノート

本バージョンは下記のアプリケーション・ノートに対応しています。

アプリケーション・ノート名	資料番号
RL78 マイクロコントローラ EEPROM エミュレーション・ライブラリ Pack01	R01AN0351JJ0101_RL78

2 動作環境

IBM PC/AT™互換機版 (Windows®ベース)

OS : WindowsXP , WindowsVista, Windows7

3 対応ツール

RL78 マイクロコントローラ EEPROM エミュレーション・ライブラリ Pack01 を使用する場合には、下記のバージョンを使用してください。

ツール名	バージョン
統合開発環境 : CubeSuite	E1.50I

4 ソフトウェア・リソース

RL78 マイクロコントローラ EEPROM エミュレーション・ライブラリ Pack01 Ver1.00 を使用する場合には、該当プログラムをユーザ領域に配置する必要がありますので、そのプログラム・サイズとライブラリで使用する定数値分のプログラム領域を消費します。

また、EEPROM エミュレーション・ライブラリを動作させるためには、スタック、データ・バッファ、ライブラリ検索領域、ワーク・エリア(セルフ RAM^注、SADDR RAM)を必要とします。

各デバイスで使用するソフトウェア・リソースを表 4-1 に示します。一覧の各項目に対する詳細についてはユーザーズ・マニュアルの「2.2 ソフトウェア環境」をご参照ください。

表 4-1 EEPROM エミュレーション・ライブラリ Pack01 ソフトウェア・リソース

項目	容量(バイト)	配置制限・使用制限	
セルフ RAM ^注	0 ~ 1024	RL78/G13 : RAM 4KB, ROM 64KB 製品	FEF00H ~ FF2FFH
		RL78/G13 : RAM 20KB, ROM 256KB 製品 128pin 製品は対象外	FAF00H ~ FB2FFH
		RL78/G13 : RAM 32KB, ROM 512KB 製品	F7F00H ~ F82FFH
		RL78/F12 : RAM 4KB, ROM 64KB 製品	FEF00H ~ FF2FFH
		上記製品以外	なし
スタック	100	セルフ RAM, FFE20H-FFEFFFH 以外の RAM 領域以外の RAM 領域に展開可能	
データ・バッファ	0 ~ 262		
ライブラリ検索領域	2		
SADDR RAM ワーク・エリア	SADDR : 11 (fdl:2) (eel:9)	ショート・アドレッシング RAM 領域のみ配置可能	
ライブラリ・サイズ	8100 (fdl:1500) (eel:6600)	セルフ RAM, FFE20H-FFEFFFH 以外のプログラム領域以外の領域に展開可能	
データ・テーブル	$(n + 1) * 4$ n = 格納データ数		
固定パラメータ領域 (デフォルト値)	72 (fdl:64) (eel:8)		
EEPROM エミュレーション・ブロック	4096 ~ (4 ブロック以上)	データ・フラッシュ・メモリのみ使用可能 (コード・フラッシュ・メモリは使用不可)	

注. EEPROM エミュレーション・ライブラリ, ならびにデータ・フラッシュ・ライブラリで固定的にワーク・エリアとして使用する領域を本書, 及びアプリケーション・ノートではセルフ RAM と呼びます。マッピングはされず, EEPROM エミュレーション実行時に自動的に使用される(以前のデータは破壊される)領域のため, ユーザ設定等は必要ありません。EEPROM エミュレーションを使用していない状態の場合は, 通常の RAM 空間として使用できます。

表 4-2 EEPROM エミュレーション・ライブラリ関数のデータ・バッファ使用サイズ

関数名	バイト
FAL_Init	0
EEL_Init	0
EEL_Open	0
EEL_Close	0
EEL_Execute	6 ~ 262
EEL_Handler ^注	6 ~ 262
EEL_TimeOut_CountDown ^注	6
EEL_GetDriverStatus	6
EEL_GetSpace	2
EEL_GetVersionString	0

注. EEL_Execute 関数で指定したデータ・バッファ領域をそのまま使用します。

5 インストール

この節では、EEPROMエミュレーション・ライブラリのインストールとアンインストールの手順について説明します。

5.1 インストール方法

EEPROMエミュレーション・ライブラリのインストールは次の手順で行います。

- (1) Windowsを起動します。
- (2) EEPROMエミュレーション・ライブラリが入っている圧縮ファイルを解凍し、展開されたフォルダをユーザ任意の場所に配置します。

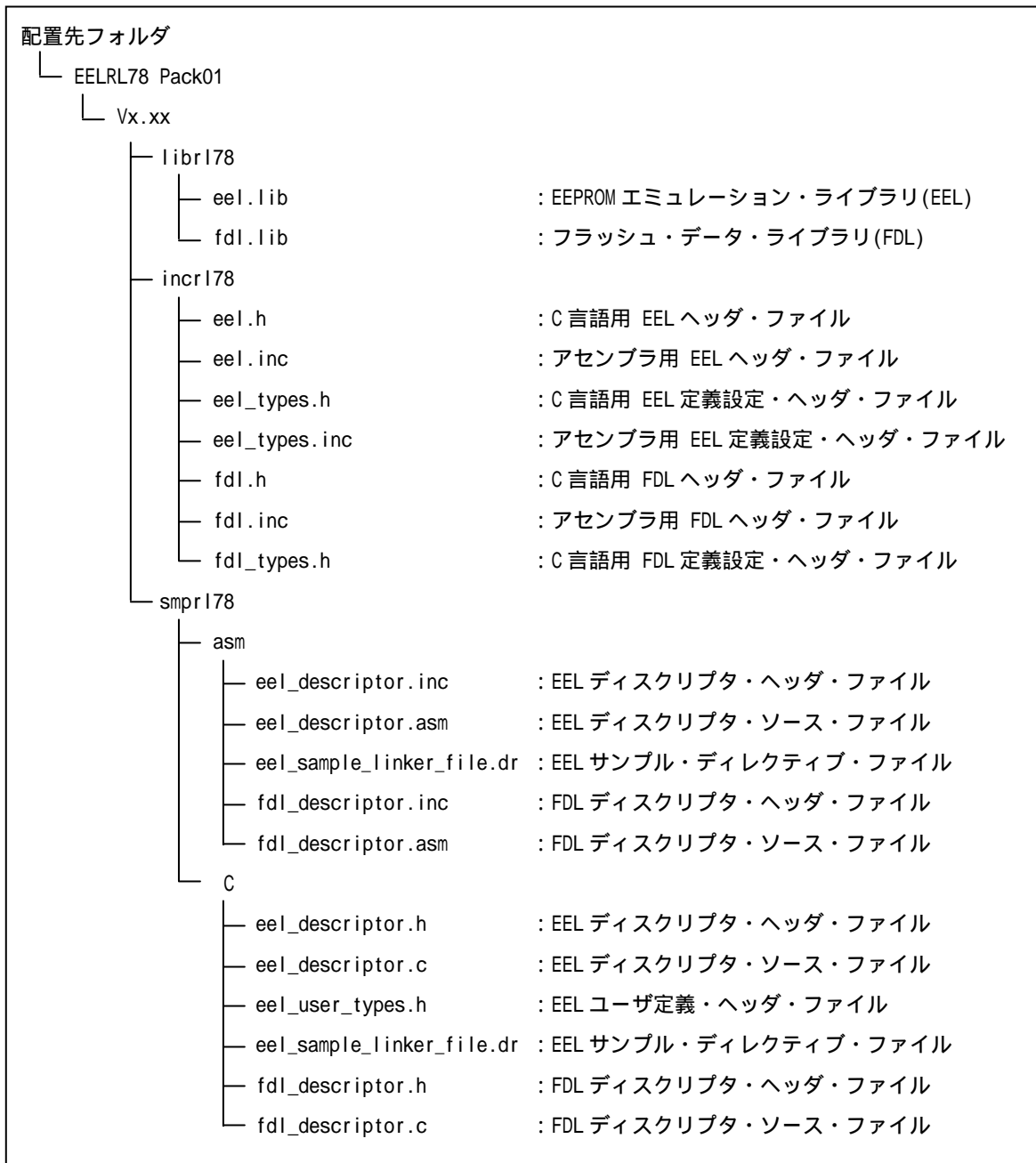
5.2 アンインストール方法

EEPROMエミュレーション・ライブラリのアンインストールは次の手順で行います。

- (1) Windowsを起動します。
- (2) ユーザ任意の場所に配置したEEPROMエミュレーション・ライブラリが入っているフォルダを削除します。

5.3 ファイル構成

EEPROMエミュレーション・ライブラリのインストールによって配置されるファイルの構成は、次のとおりです。



6 ビルド方法

この節では、EEPROM エミュレーション・ライブラリを用いたプログラムのビルドの手順を説明します。

6.1 使用するソフトウェア

EEPROM エミュレーション・ライブラリを用いたプログラムをビルドするには、次の統合開発環境が必要です。

- ・統合開発環境 CubeSuite E1.50I

6.2 CubeSuite でのビルド方法

CubeSuite を用いて EEPROM エミュレーション・ライブラリをユーザ・プログラムに組み込んで、ビルドする手順を説明します。

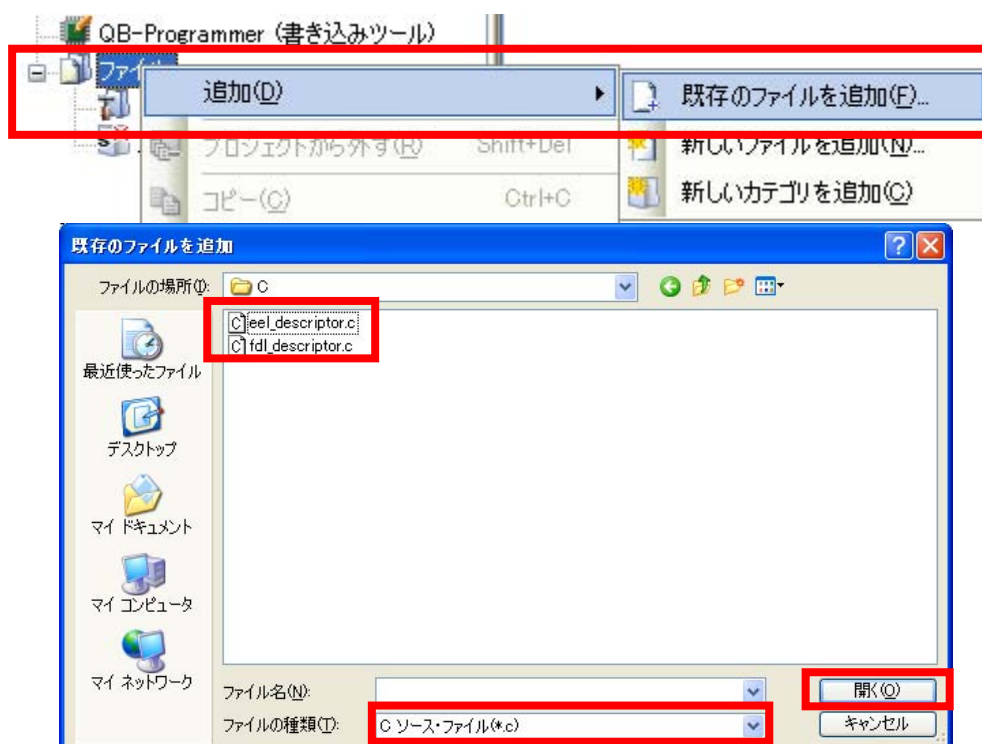
6.2.1 C 言語の場合

(1) プロジェクトの作成とソース・ファイルの設定

CubeSuite でプロジェクトを作成し、表示された画面の左側にある[プロジェクト・ツリー]から、[ファイル]を右クリックして表示されるリストの[追加]を選択し、[既存のファイルを追加]をクリックすると、[既存のファイルを追加]画面が表示されます(図6-1)。

次に画面中にある[ファイルの種類]のプルダウンメニューをクリックすると、ファイルの種類の一覧が表示されるので、その中にある[C ソース・ファイル(*.c)]を選択し、ソース・ファイルとしてユーザ・プログラムのファイル、及び EEPROM エミュレーション・ライブラリ、ならびにフラッシュ・データ・ライブラリのディスクリプタ・ファイル(eel_descriptor.c, fdl_descriptor.c)を登録してください。

図 6-1 ソース・ファイルの指定

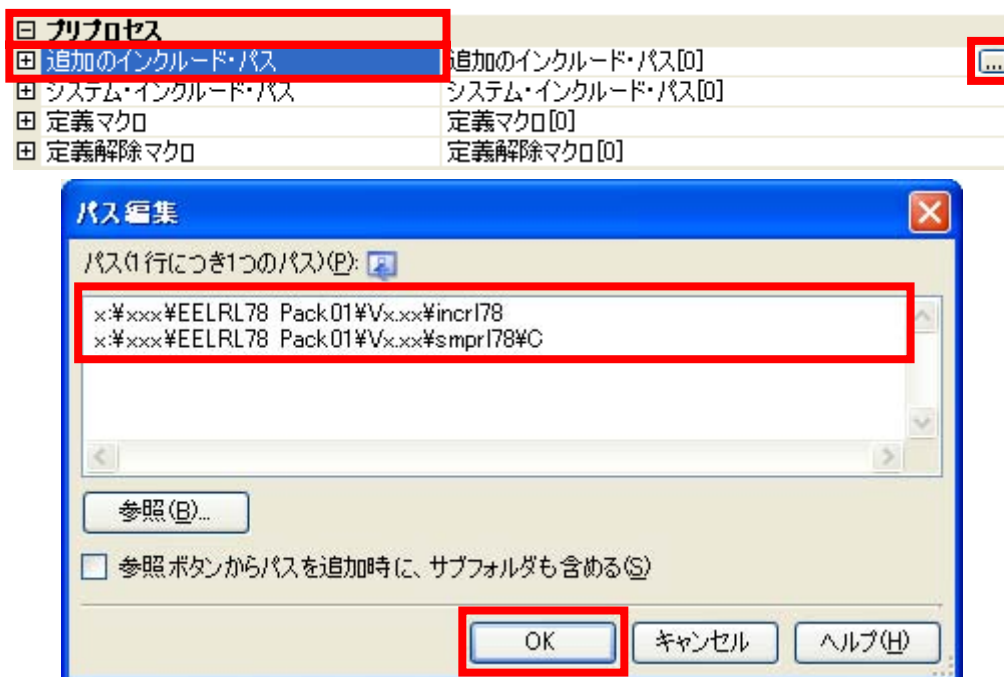


(2) インクルード・ファイル・パスの設定

CubeSuite の[プロジェクト・ツリー]にある,[CA78K0R(ビルド・ツール)]をクリックして CA78K0R のプロパティ画面を開き, 下部にある[コンパイル・オプション]タブをクリックすると, コンパイル・オプション画面が表示されます。

次に画面中にある[プリプロセス]をクリックし,項目を表示させて[追加のインクルード・パス]の欄を選択し,右に表示される[...]ボタンをクリックすると,[パス編集]ダイアログ画面が表示されるので,画面中のテキスト・ボックスに,EEPROM エミュレーション・ライブラリ用のヘッダ・ファイルがあるフォルダのパスを入力します(図 6-2)。 [参照] ボタンをクリックして,マウス操作により指定することもできます。

図 6-2 インクルード・ファイルの指定



(3) ライブラリ・ファイルの設定

ライブラリ・ファイルをプロジェクトで使用するための設定は2通りの方法があります。

・プロジェクトにライブラリ・ファイルを登録する方法

CubeSuite の[プロジェクト・ツリー]にある,[ファイル]を右クリックして表示されるリストから[追加]を選択し,[既存のファイルを追加]をクリックすると,[既存のファイルを追加]画面が表示されます(図 6-3)。

次に画面中にある[ファイルの種類]のプルダウンメニューをクリックすると,ファイルの種類の一覧が表示されるので,その中にある[ライブラリ・ファイル(*.lib)]を選択し,EEPROM エミュレーション・ライブラリ(eel.lib),及びフラッシュ・データ・ライブラリ(fd1.lib)のファイルを登録してください。

・ライブラリ・ファイルのフォルダ・パスを設定し,使用するライブラリ名を指定する方法

CubeSuite の[プロジェクト・ツリー]にある,[CA78K0R(ビルド・ツール)]をクリックして

CA78K0Rのプロパティ画面を開き、下部にある[リンク・オプション]タブをクリックすると、リンク・オプション画面が表示されます。

次に画面中の[ライブラリ]の項目にある[追加のライブラリ・パス]の欄を選択し、右に表示される[...]ボタンをクリックすると、[パス編集]ダイアログ画面が表示されるので、画面中のテキスト・ボックスに、EEPROM エミュレーション・ライブラリがあるフォルダのパスを入力します(図6-4)。[参照] ボタンをクリックして、マウス操作により指定することもできます。

フォルダ・パスの設定後、[ライブラリ]の項目にある[使用するライブラリ・ファイル]の欄を選択し、右に表示される[...]ボタンをクリックすると、[テキスト編集]ダイアログ画面が表示されるので、画面中のテキスト・ボックスに、EEPROM エミュレーション・ライブラリ(eel.lib)、及びフラッシュ・データ・ライブラリ(fdl.lib)のファイル名を設定してください(図6-5)。ファイル名はフォルダ・パスを含めて設定することも可能です。

図 6-3 ライブラリ・ファイルの指定

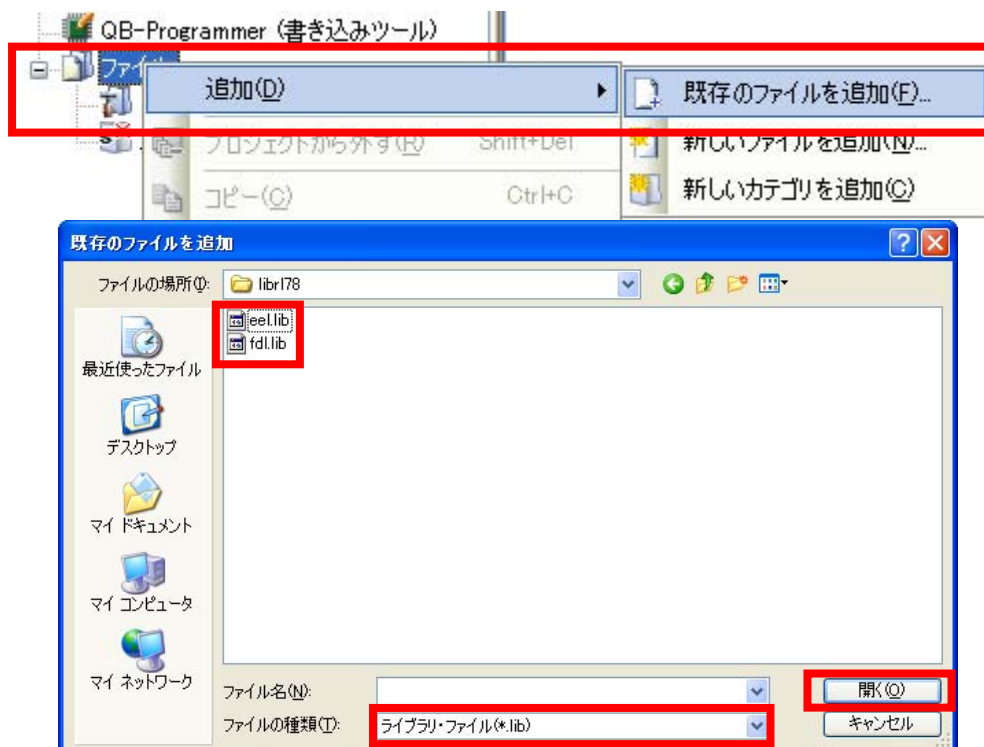


図 6-4 ライブラリ・ファイルのフォルダ・パスを指定

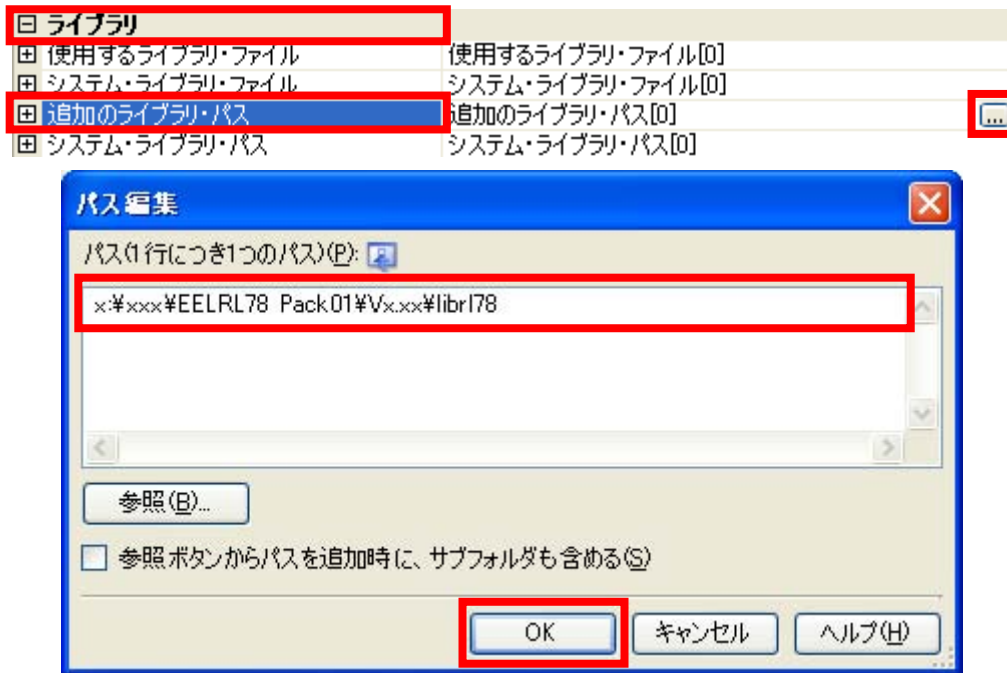
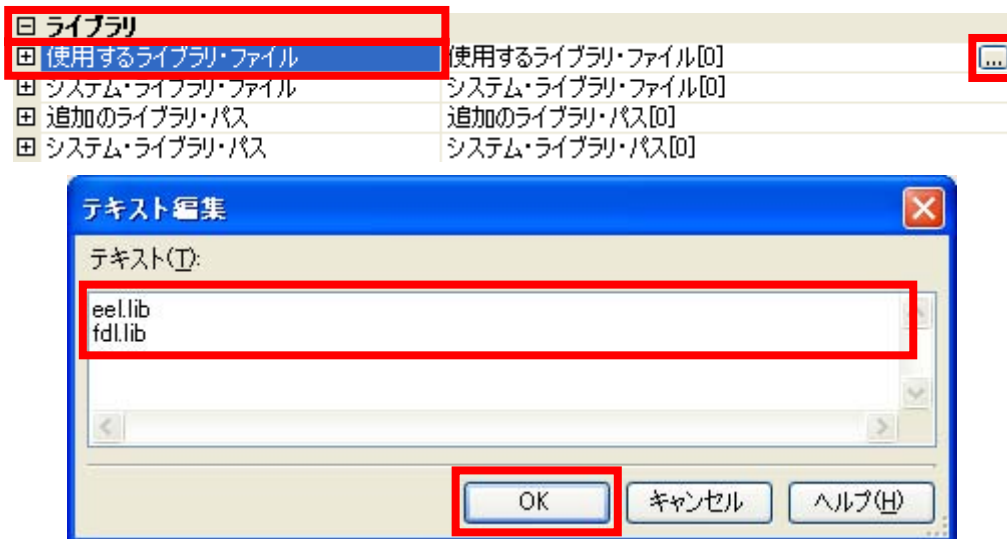


図 6-5 使用するライブラリ・ファイル名の設定

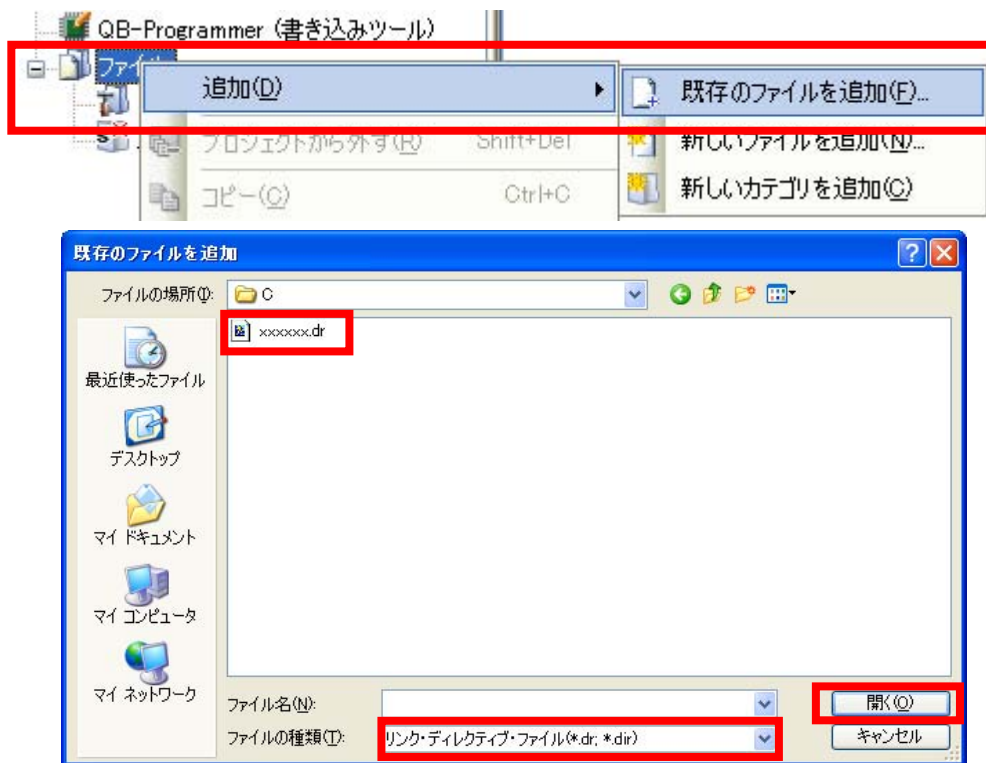


(4) リンク・ディレクティブの設定

CubeSuite の[プロジェクト・ツリー]にある,[ファイル]を右クリックして表示されるリストから[追加]を選択し,[既存のファイルを追加]をクリックすると,[既存のファイルを追加]画面が表示されます(図6-6)。

次に画面中にある[ファイルの種類]のプルダウンメニューをクリックすると,ファイルの種類の一覧が表示されるので,その中にある[リンク・ディレクティブ・ファイル(*.dr;*.dir)]を選択し,リンク・ディレクティブ・ファイル名のユーザ・ファイルを登録してください。

図 6-6 リンク・ディレクティブ・ファイルの指定



(5) ビルド

CubeSuite のメニュー [ビルド] [ビルド・プロジェクト] をクリックして,ビルドを実行します。

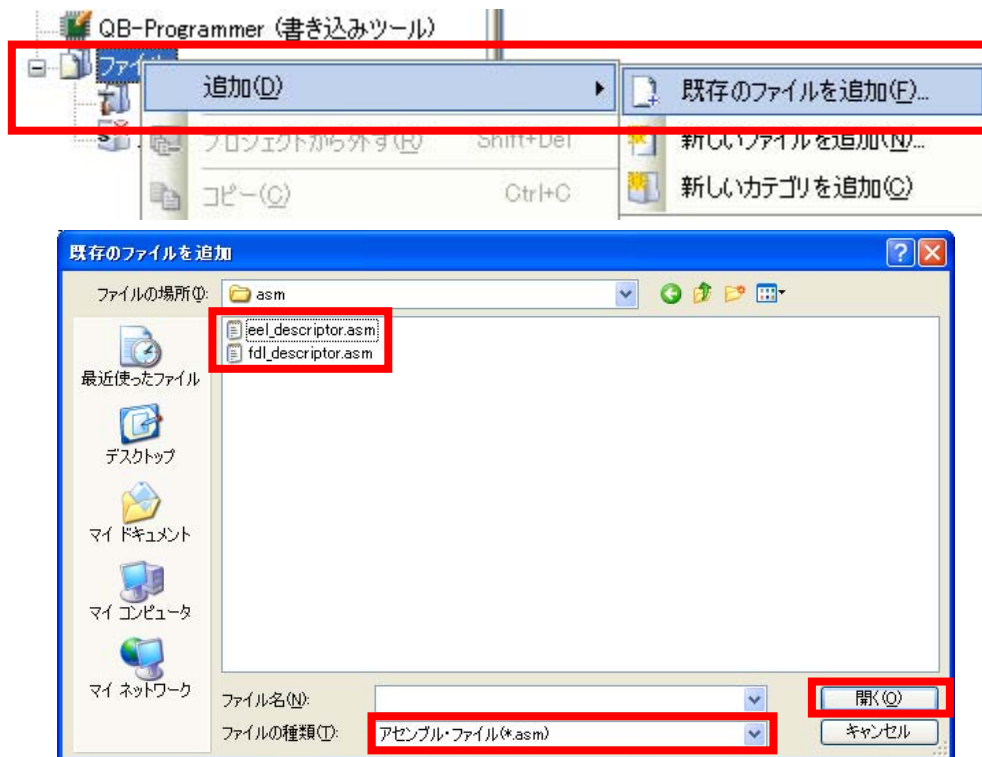
6.2.2 アセンブリ言語の場合

(1) プロジェクトの作成とソース・ファイルの設定

CubeSuite でプロジェクトを作成し、表示された画面の左側にある[プロジェクト・ツリー]から、[ファイル]を右クリックして表示されるリストの[追加]を選択し、[既存のファイルを追加]をクリックすると、[既存のファイルを追加]画面が表示されます(図6-7)。

次に画面中にある[ファイルの種類]のプルダウンメニューをクリックすると、ファイルの種類の一覧が表示されるので、その中にある[アセンブル・ファイル(*.asm)]を選択し、ソース・ファイルとしてユーザ・プログラムのファイル、及びEEPROMエミュレーション・ライブラリ、ならびにフラッシュ・データ・ライブラリのディスクリプタ・ファイル(eel_descriptor.asm, fdl_descriptor.asm)を登録してください。

図 6-7 ソース・ファイルの指定

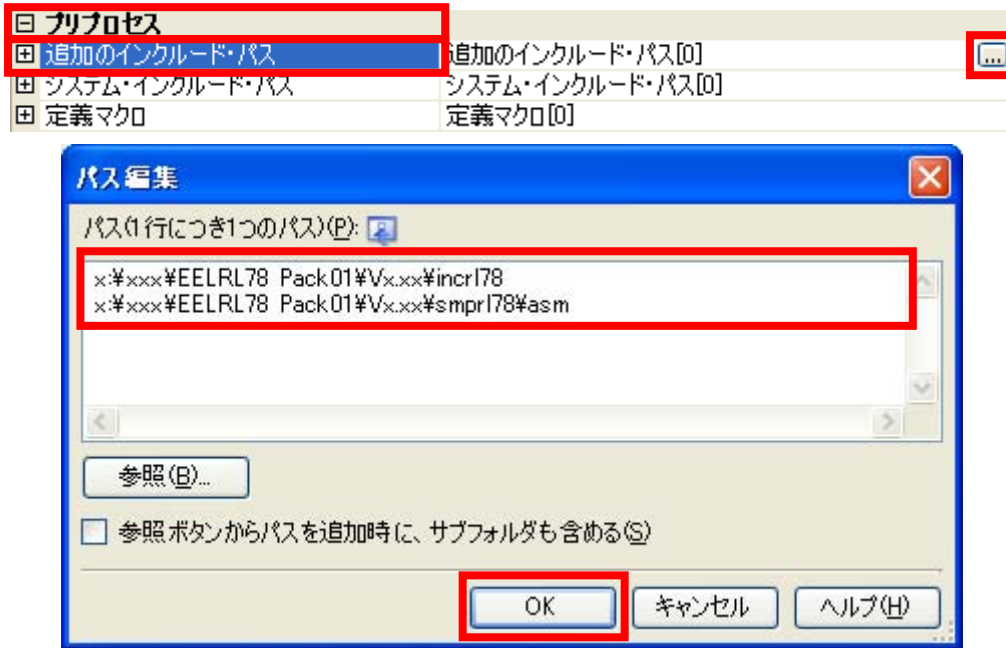


(2) インクルード・ファイル・パスの設定

CubeSuite の[プロジェクト・ツリー]にある [CA78K0R(ビルド・ツール)]をクリックして CA78K0Rのプロパティ画面を開き、下部にある[アセンブル・オプション]タブをクリックすると、アセンブル・オプション画面が表示されます。

次に画面中にある[プリプロセス]をクリックし、項目を表示させて[追加のインクルード・パス]の欄を選択し、右に表示される[...]ボタンをクリックすると、[パス編集]ダイアログ画面が表示されるので、画面中のテキスト・ボックスに、EEPROM エミュレーション・ライブラリ用のヘッダ・ファイルがあるフォルダのパスを入力します(図6-8)。 [参照] ボタンをクリックして、マウス操作により指定することもできます。

図 6-8 インクルード・ファイルの指定



(3) ライブラリ・ファイルの設定

ライブラリ・ファイルをプロジェクトで使用するための設定は2通りの方法があります。

・プロジェクトにライブラリ・ファイルを登録する方法

CubeSuite の[プロジェクト・ツリー]にある, [ファイル]を右クリックして表示されるリストから[追加]を選択し, [既存のファイルを追加]をクリックすると, [既存のファイルを追加]画面が表示されます(図 6 - 9)。

次に画面中にある[ファイルの種類]のプルダウンメニューをクリックすると, ファイルの種類の一覧が表示されるので, その中にある[ライブラリ・ファイル(*.lib)]を選択し, EEPROM エミュレーション・ライブラリ(eel.lib), 及びフラッシュ・データ・ライブラリ(fdl.lib)のファイルを登録してください。

・ライブラリ・ファイルのフォルダ・パスを設定し, 使用するライブラリ名を指定する方法

CubeSuite の[プロジェクト・ツリー]にある, [CA78K0R(ビルド・ツール)]をクリックしてCA78K0Rのプロパティ画面を開き, 下部にある[リンク・オプション]タブをクリックすると, リンク・オプション画面が表示されます。

次に画面中の[ライブラリ]の項目にある[追加のライブラリ・パス]の欄を選択し, 右に表示される[...]ボタンをクリックすると, [パス編集]ダイアログ画面が表示されるので, 画面中のテキスト・ボックスに, EEPROM エミュレーション・ライブラリがあるフォルダのパスを入力します(図 6 - 10)。
[参照] ボタンをクリックして, マウス操作により指定することもできます。

フォルダ・パスの設定後, [ライブラリ]の項目にある[使用するライブラリ・ファイル]の欄を選択し, 右に表示される[...]ボタンをクリックすると, [テキスト編集]ダイアログ画面が表示されるので, 画面中のテキスト・ボックスに, EEPROM エミュレーション・ライブラリ(eel.lib), 及びフラッシュ・データ・ライブラリ(fdl.lib)のファイル名を設定してください(図 6 - 11)。
ファイル名はフォルダ・パスを含めて設定することも可能です。

図 6-9 ライブラリ・ファイルの指定

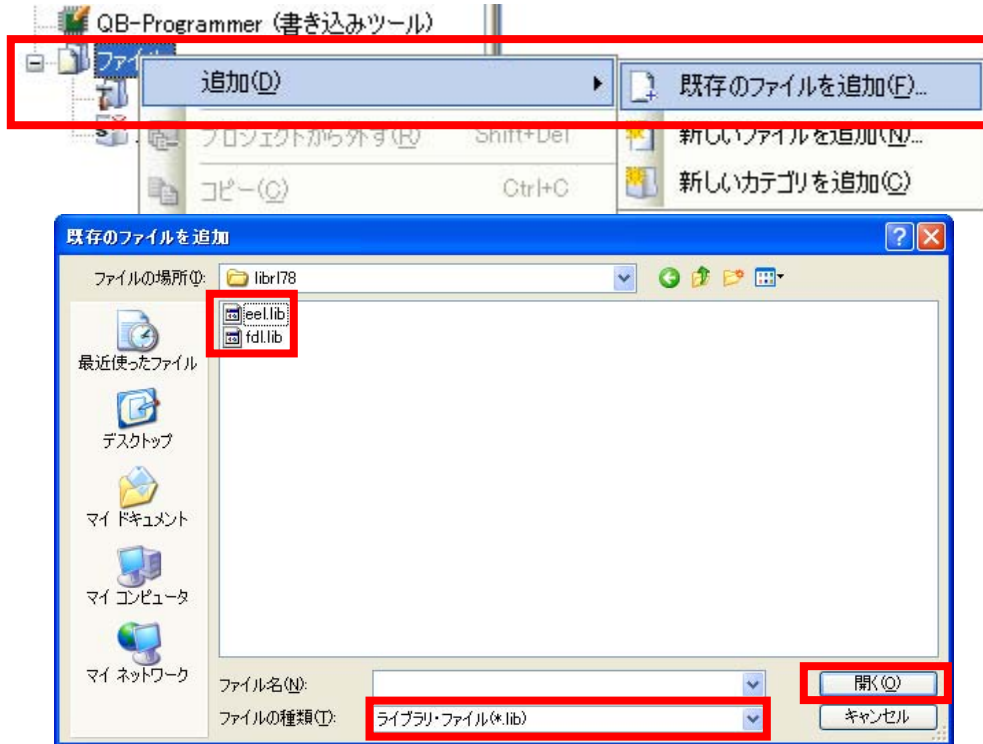


図 6-10 ライブラリ・ファイルのフォルダ・パスを指定

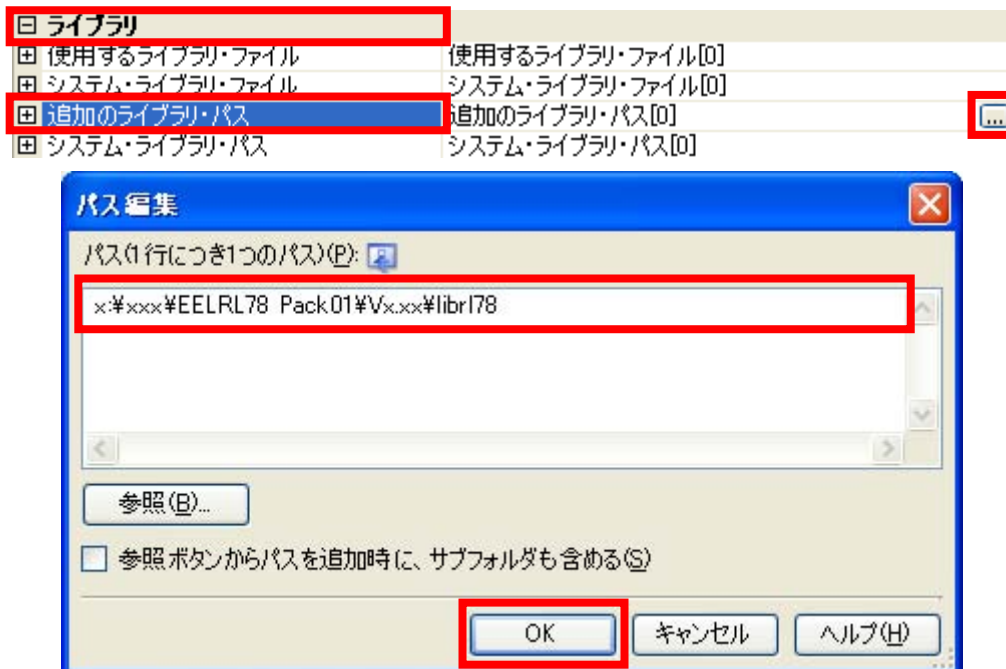
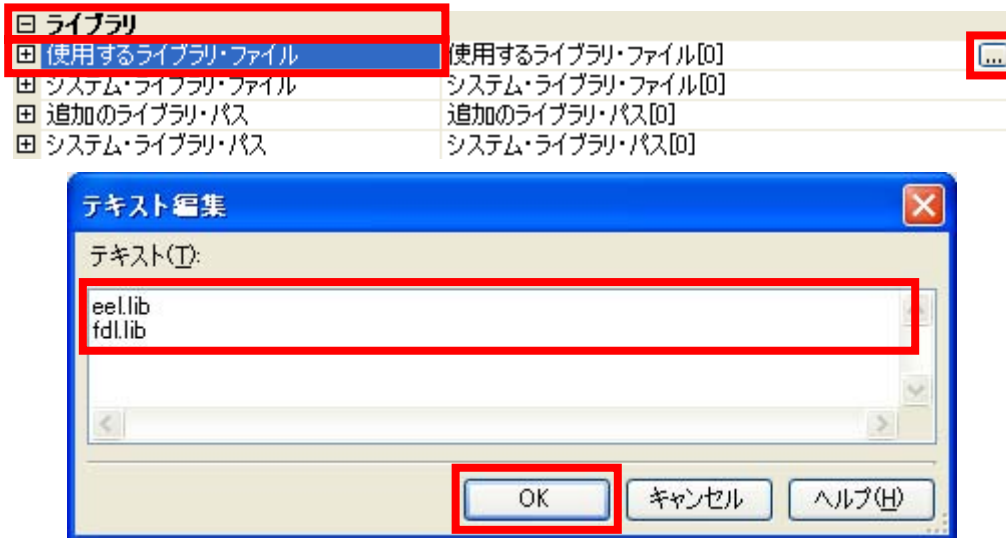


図 6-11 使用するライブラリ・ファイル名の設定

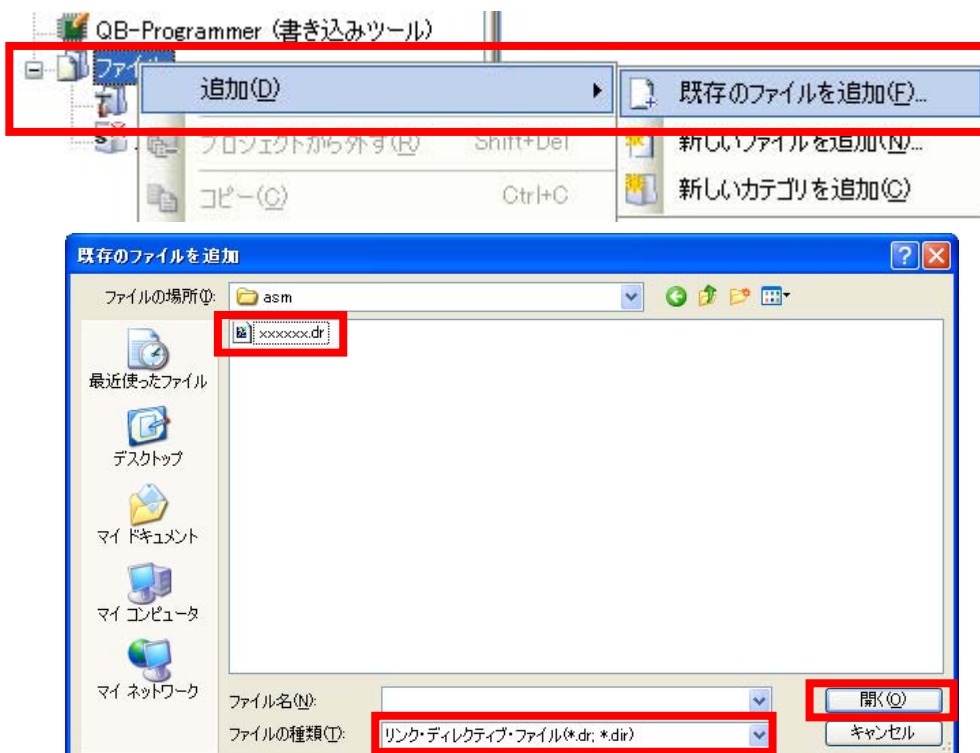


(4) リンク・ディレクティブの設定

CubeSuite の[プロジェクト・ツリー]にある,[ファイル]を右クリックして表示されるリストから[追加]を選択し,[既存のファイルを追加]をクリックすると,[既存のファイルを追加]画面が表示されます(図 6-12)。

次に画面中にある[ファイルの種類]のプルダウンメニューをクリックすると,ファイルの種類の一覧が表示されるので,その中にある[リンク・ディレクティブ・ファイル(*.dr;*.dir)]を選択し,リンク・ディレクティブ・ファイル名のユーザ・ファイルを登録してください。

図 6-12 リンク・ディレクティブ・ファイルの指定



(5) ビルド

CubeSuite のメニュー [ビルド] [ビルド・プロジェクト] をクリックして、ビルドを実行します。

7 デバッグ方法

IECUBE , または オンチップ・デバッグ・エミュレータ E1 を使用してデバッグを行う場合につきましては以下の資料のご参照ください。

CubeSuite でデバッグを行う場合

- ・ ユーザーズ・マニュアル「CubeSuite RL78,78K0R デバッグ編」