μT-Kernel3.0
RX231 IoT-Engine 向け
開発環境マニュアル

- e² Studio 編 -

Version.01.00.00

2021.05.18

更新履歴

版数(日付)	内　容
1.00.00 (2021.05.18)	● 初版

目次

1. 概要 4
1.1 対象とするOSおよびハードウェア 4
1.2 対象とする開発環境 4
2. 開発環境の準備 5
2.1 e² Studioのインストール 5
3. プロジェクトの作成 6
3.1 GitHubからのプロジェクトの入手 6
3.2 プロジェクトのファイル構成 7
3.3 プロジェクトのインポート 7
3.4 プロジェクトのビルドの確認 8
4. ユーザ・プログラムの作成 8
4.1 ユーザ・プログラムのソースファイル 8
4.2 ワーキング・セットの選択 8
4.3 実行プログラムのビルド 9
5. 実機でのプログラム実行 10
5.1 E² studioによるプログラムの実行 10

1. 概要

本書は以下の対象においてμT-Kernel 3.0のアプリケーション開発を行うまでの手順を説明する。

以下に本書の対象を記す。

1.1 対象とするOSおよびハードウェア

開発対象のマイコンおよびOSは以下である。

分類	名称	備考
マイコン	RX200シリーズ RX231グループ R5F52318ADFL	ルネサス エレクトロニクス製
OS	μT-Kernel3.00.04	TRONフォーラム
実機	RX231 IoT-Engine	UCテクノロジー製

1.2 対象とする開発環境

分類	名称	備考
開発環境	e² studio Version: 2021-04 (21.4.0)	ルネサス エレクトロニクス https://www.segger.com/

※ バージョン番号は確認した最新版のバージョンを示している。

2. 開発環境の準備

μT-Kernel 3.0 のアプリケーション開発を行うにあたり、以下の手順で開発環境の準備を行う。

(1) e² Studio のインストール

以降、それぞれについて説明をしていく。なお、すでに各ソフトがインストール済みの場合はこの手順は不要である。

2.1 e² Studio のインストール

e² studio は、オープンソースの"Eclipse"をベースとした、ルネサス製マイコン用の統合開発環境である。

e² studio は以下の e² studio のホームページからインストーラが入手可能である。なお、ダウンロードにはユーザ登録が必要である。

https://www.renesas.com/jp/ja/products/software-tools/tools/ide/e2studio.html

インストーラによる e² studio のインストールの際には、対象デバイスとして RX マイコンを選択する。
e² studio のインストールや操作については、上記のホームページを参照のこと。

3. プロジェクトの作成

開発するμT-Kernel 3.0のアプリケーションについて、e² Studioのプロジェクトを作成する。
TRONフォーラムでは、基本となるプロジェクトをGitHubにて公開している。このプロジェクトを使用することにより、容易にアプリケーション開発のプロジェクトを作成することができる。

3.1 GitHubからのプロジェクトの入手

TRONフォーラムのGitHubの以下のレポジトリにて、各開発環境のプロジェクトが公開されている。

https://github.com/tron-forum/mtk3_devenv

プロジェクト一式は、以下のいずれかの方法で取得できる。

- プロジェクト一式をZipファイルとして入手する場合

masterブランチのarchivesディレクトリに各プロジェクトのzipファイルが格納されている。本プロジェクトのZipファイル名は「mtk3_e2studio_rx231.zip」である、このZipをダウンロードし、任意の場所に展開する。

- プロジェクト一式をGitのクローンとして入手する場合

プロジェクト毎にブランチが設けられている。本書の対象開発環境のブランチは「e2studio/rx231」である。

以下のGitコマンドを実行することにより、レポジトリのクローンが取得できる。

```
git clone -b e2studio/rx231 --recursive https://github.com/tron-forum/mtk3_devenv.git
```

なお、レポジトリ中のμT-Kernel 3.0のソースコードはサブモジュールとして、TRONフォーラムのGitHubの以下のURLのμT-Kernel 3.0レポジトリに紐づけられている。

https://github.com/tron-forum/mtkernel_3

3.2 プロジェクトのファイル構成

プロジェクトのディレクトリおよびファイルは以下のように構成される。

ディレクトリまたはファイル名	内容
mtkernel_3	μT-Kernel 3.0 ソースコードのディレクトリ
user_program	ユーザ・プログラム用のディレクトリ
.settings	e² Studio の設定ファイル
.cproject	e² Studio のプロジェクトファイル
その他のファイル	e² Studio の各種ファイル

ディレクトリ「mtkernel_3」の内容は、TRON フォーラムから公開されているμT-Kernel 3.0 のソースコードと同一である。

ディレクトリ「user_program」に作成するユーザ・プログラムを格納する。初期状態では、サンプルコードを記述した user_main.c ファイルが格納されている。

その他のやファイルは、e² Studio の各種ファイルである。ユーザが直接これらのファイルにアクセスする必要はない。

3.3 プロジェクトのインポート

入手したプロジェクトを e² Studio の開発環境に以下の手順で取り込む。

(1) メニュー「インポート」を選択する。
(2) インポートのダイアログから「一般」→「既存プロジェクトをワークスペースへ」を選択する。
(3) 「ルート・ディレクトリの選択」で「参照」ボタンを押し、前項で入手したプロジェクトのディレクトリを指定する。
(4) 「プロジェクト」に「mtk3_rx231」が表示されるので、それをチェックする。
(5) 「完了」ボタンを押下すると、プロジェクトがワークスペースに取り込まれる。

プロジェクトのインポートが完了すると、Eclipse のプロジェクト・エクスプローラに「mtk3_rx231」が表示される。

3.4 プロジェクトのビルドの確認

プロジェクトのμT-Kernel 3.0のプログラムがビルド出来ることを確認する。
プロジェクト「mtk3_rx231」が選択されている状態で、メニュー「プロジェクト」→「プロジェクトのビルド」を選択する。
プログラムのビルドが実行され、「コンソール」に「Build Finished. 0 errors」が表示されれば、正常にビルドが完了している。

4. ユーザ・プログラムの作成

4.1 ユーザ・プログラムのソースファイル

本プロジェクトでは、ユーザ・プログラムのソースファイルは、プロジェクトのディレクトリ下の「user_program」ディレクトリに置くことを前提としている。
初期状態では、サンプルコードを記述したuser_main.cファイルが置かれている。ユーザは本ファイルの内容を変更することにより、プログラムを記述することができる。また、複数のファイルが存在する場合は、このディレクトリ下に置くことができる。

4.2 ワーキング・セットの選択

e² Studioではプログラムの各種設定をワーキング・セットとしてプロジェクトに対して作成する。ワーキング・セットは、一つのプロジェクトに複数作ることができる。
本プロジェクトは初期状態では、「Release」と「Debug」の二つのワーキング・セットが作られている。
ワーキング・セットは、メニュー「プロジェクト」→「ビルド構成」→「アクティブにする」から選択することができる。

「Release」と「Debug」の違いは最適化である。通常、デバッグ時は「Debug」を使用する。なお、コンフィギュレーションの各種設定は、ユーザ・プログラムの必要に応じて設定・変更を行う。

項目	Debug	Release
Debugging Level	-g2	-g1
Optimization Level	-O0	-O2

4.3 実行プログラムのビルド

プロジェクトが選択されている状態で、メニュー「プロジェクト」→「プロジェクトのビルド」を選択すると、プログラムがビルド（コンパイル、リンク）され、実行コード・ファイル「mtk3_rx231.elf」が生成される。

実行コード・ファイルは、プロジェクトのディレクトリ下のワーキング・セットと同名のディレクトリに生成される。

5. 実機でのプログラム実行

ビルドし生成した実行コードを実機上で実行する方法は、実機環境により異なる。
本プロジェクトでは、JTAG エミュレータ Renesas E1 を使用したデバッグ実行環境が設定されている。以下に動作を確認した実機環境を記す。

項目	内容
実機ハードウェア（実行ボード）	IoT-Engine Starter Board
デバッガ（JTAG エミュレータ）	Renesas E1

5.1 E² studio によるプログラムの実行

(1) E² studio のメニューからメニュー「実行」→「デバッグの構成」を選択し、開いたダイアログから項目「Renesas GDB Hardware Debugging」を選択する。

(2) 「新規構成」ボタンを押し、「Renesas GDB Hardware Debugging」に構成を追加する。

(3) 追加した構成を選択し、「構成の作成、管理、実行」画面にて以下の設定を行う。

「メイン」タブ

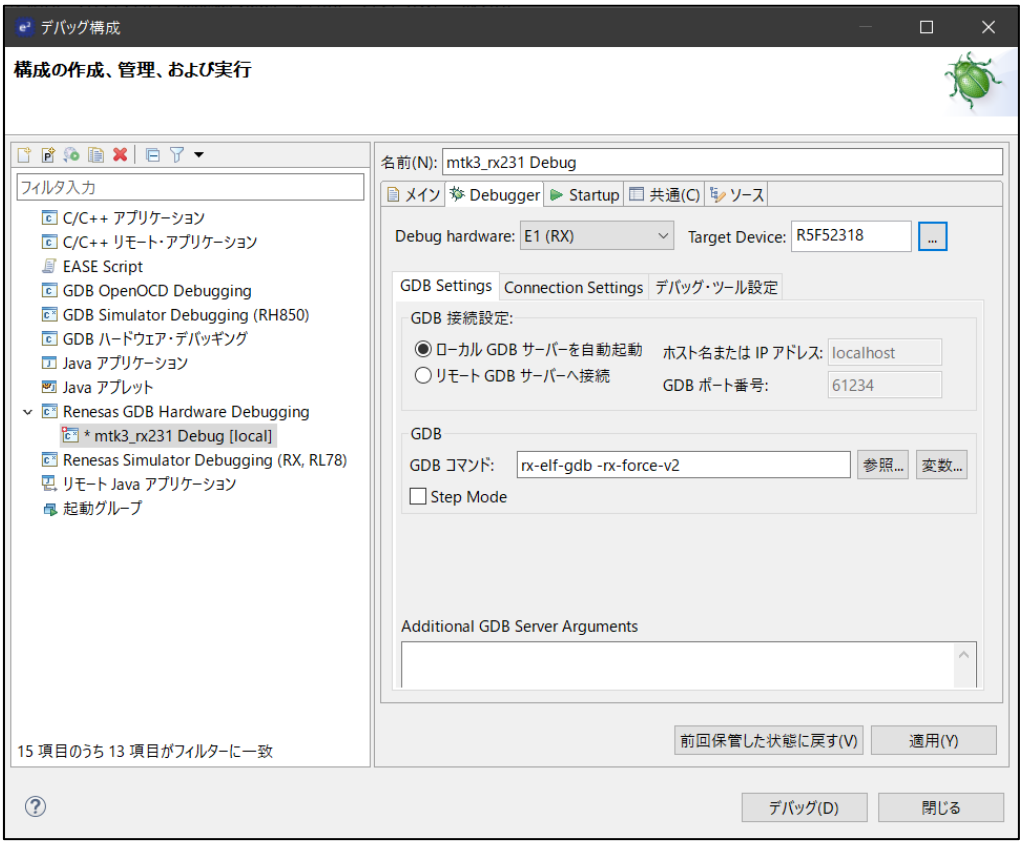
名前：（任意）を入力

C/C++アプリケーション：ビルドした ELF ファイル

「Debugger」タブ → 「GDB Settings」

Debug Hardware：「E1 (RX)」を選択

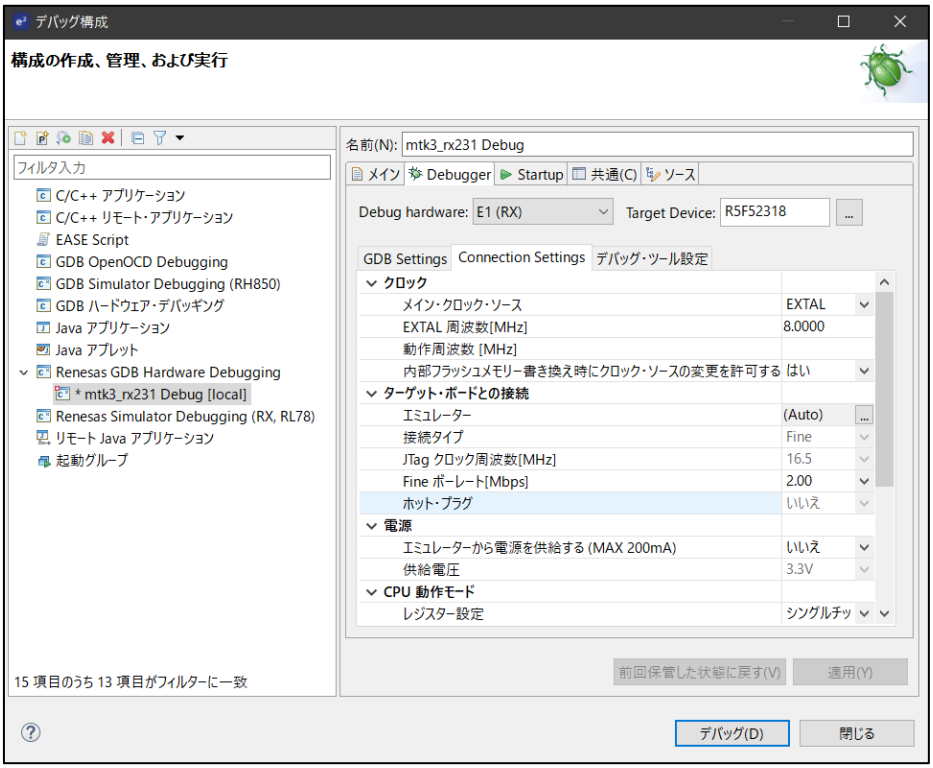
Target Device：「**R5F52318**」を選択



「Debugger」タブ→　「Connection Setting」

「クロック」→「EXTAL 周波数」8.0000 を入力

「エミュレータから電源を供給する」いいえを選択



「Startup」タブ

ブレークポイント設定先：「user_main」を入力

(4) デバッグ開始

「デバッグ」ボタンを押すとプログラムが実機に転送され、ROM に書き込まれたのち、実行される。

プログラムは実行すると、OS 起動後にユーザのアプリケーションプログラムを実行し、usermain 関数にてブレークする。

以上