μT-Kernel3.0
STM32L4 IoT-Engine 向け
開発環境マニュアル

- SEGGER Embedded 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

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	STM32L486VG (ARM Cortex-M4コア)	STマイクロエレクトロニクス製
OS	μT-Kernel3.00.04	TRONフォーラム
実機	STM32L4 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	Embedded Studio for ARM Release 5.32a	SEGGER 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) Embedded Studio のインストール
(2) CPU サポートパッケージのインストール

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 Embedded Studio のインストール

以下から使用する PC の OS に合わせて、ARM 向けの Embedded Studio をダウンロードする。

Embedded Studio のダウンロードページ
https://www.segger.com/downloads/embedded-studio

インストーラがダウンロードされるので、それを実行し、指示に従ってインストールを進める。特に問題が無ければデフォルトの設定でインストールを行えばよい。

2.2 CPU サポートパッケージのインストール

Embedded Studio を実行し、開発対象とする CPU に応じたサポートパッケージを以下に手順でインストールする。

(1) メニュー「Tools」→「Package Manager」を選択する。
(2) ダイアログの検索欄に「STM32L4」と入力すると、検索結果に「STM32L4xx CPU Support Package」が表示される。
(3) 「STM32L4xx CPU Support Package」を右クリックし、ポッポアップしたメニューから「Install Selected Packages」を選択する。以降、指示に従いインストールを進める。

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、Embedded 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_seggeres_stm32l4.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「segger_es/stm32l4」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 segger_es/stm32l4 --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
mtk3_stm32l4.emProject	Embedded Studio 用プロジェクト・ファイル
その他のファイル	Embedded 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、Embedded 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 プロジェクトのオープン

Windows の場合は、プロジェクトのディレクトリ中の「mtk3_stm32l4.emProject」ファイルをエクスプローラから開くと、Embedded Studio が起動し本プロジェクトがオープンされる。

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_stm32l4」が選択されている状態で、メニュー「Build」→「Build mtk3_stm32l4」を選択する。
プログラムのビルドが実行され、以下が表示されれば、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 コンフィギュレーションの選択

Embedded Studio は初期状態では、「Release」と「Debug」の二つのコンフィギュレーションが作られている。
コンフィギュレーションは、メニュー「Build」→「Set Active Build Configuration」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	Level 3	Level 1
Optimization Level	None	Level2 balance

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「Build」→「Build mtk3_stm32l4」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_stm32l4.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に「Output」ディレクトリが生作られ、その中の以下に生成される。

Debug コンフィギュレーションの場合
<プロジェクトのディレクトリ>¥Output¥Debug¥Exe¥mtk3_stm32l4.elf

Release コンフィギュレーションの場合

<プロジェクトのディレクトリ>¥Output¥Release¥Exe¥mtk3_stm32l4.elf

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 SEGGER J-Link 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）	SEGGER J-Link

メニュー「Build」→「Build and Debug」を選択すると、プログラムは実機に転送され、デバッグ実行が開始する。

以上