

μ T-Kernel 3.0 構築手順書 (SBK-M4MN)

Rev 3.00.00

April, 2023



μ T-Kernel 3.0

目次

1. はじめに	2
1.1. 本書について	2
1.2. 表記について	2
2. 概要	3
2.1. 対象とするハードウェアと OS	3
2.2. 対象とする開発環境	3
2.3. デバイスドライバ	4
2.4. 関連ドキュメント	4
3. 開発環境の準備	5
3.1. IAR Embedded Workbench for Arm のインストール	5
3.2. SBK-M4MN の設定と接続	5
3.3. ソースコードの展開	6
3.4. Tera Term の設定	6
4. ビルドと実行	7
4.1. サンプルプロジェクト	7
4.2. 実行手順	7
4.2.1. EWARM の起動	8
4.2.2. ビルド構成の選択	8
4.2.3. ビルド	9
4.2.4. ターゲットボードの接続	9
4.2.5. デバッグ開始	9
4.2.6. エントリポイントで停止	10
4.2.7. 実行	10
4.2.8. 動作確認	10
5. μ T-Kernel 3.0 のディレクトリ／ファイル構成	12
5.1. μ T-Kernel 3.0 のソースコード	12
5.2. μ T-Kernel 3.0 BSP のディレクトリとファイル	12
5.3. プロジェクトファイル	13
5.4. サンプルアプリケーション	13

1. はじめに

本品はユーシーテクノロジー(株)が開発した SBK-M4MN(ES)ボード(以下、SBK-M4MN)対応の μT-Kernel 3.0 である。

本品は、トロンフォーラムの配布する μT-Kernel 3.0 をベースに、イーエスピー企画製の評価ボード SBK-M4MN で動作させるための機種依存部を追加してある。

- ◆ 本書のターゲットボードである SBK-M4MN は実在しない。ターゲットボードとしては「SBK-M4KN(ES)ボード」(SBK-M4KN)を使用する必要がある。このため、本書の説明にあるターゲットボード名称は、SBK-M4MN を SBK-M4KN と読み替えること。

ただし、μT-Kernel 3.0 の対象となる CPU としては TPM4MNFYAFG 用と TPM4KNFYAFG 用を別に実装してある。ターゲットボードも SBK-M4KN と同一の構成で CPU のみが TPM4MNFYAFG に変更されているターゲットボード(SBK-M4MN)が別に存在するものと仮定して実装してある。よって、μT-Kernel のディレクトリ名やファイル名を読み替える必要はない。

1.1. 本書について

本書は SBK-M4MN 向けの μT-Kernel 3.0 の構築手順について記載した構築手順書である。本書の対象となる実装は、ユーシーテクノロジー(株)が公開している μT-Kernel 3.0 BSP(Board Support Package)に含まれている。

以降、単に OS や RTOS と称する場合は μT-Kernel 3.0 を示し、**本実装**と称する場合は SBK-M4MN 向けのソースコードの実装を示すものとする。

1.2. 表記について

表記	説明
[]	[]はソフトウェア画面のボタンやメニューを表す。
「 」	「 」はソフトウェア画面に表示された項目などを表す。
◆	注意が必要な内容の場合に記述する。
i	補足やヒントなどの内容の場合に記述する。
<TARGET>	ターゲットボード用のディレクトリ名を表す。
<CPU>	CPU 用のディレクトリ名を表す。
<CORE>	CPU コア用のディレクトリ名を表す。

2. 概要

本書では、μT-Kernel 3.0 BSP の使用方法について説明する。

μT-Kernel 3.0 BSP は、特定のマイコンボード等のハードウェアに対して移植した μT-Kernel 3.0 の開発および実行環境一式を提供するものである。

2.1. 対象とするハードウェアと OS

開発対象のハードウェアおよび OS は以下である。

表 2-1 開発対象のハードウェアと OS

分類	名称	備考
マイコン	TMPM4MNFYAFG	東芝デバイス & ストレージ株式会社
OS	μT-Kernel 3.00.06	トロンフォーラム
実機 (マイコンボード)	SBK-M4MN(ES)ボード	株式会社 イーエスピー企画

- ① μT-Kernel 3.0 の最新版は以下の GitHub リポジトリにて公開されている。

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

- ① 対象マイコンボード(SBK-M4MN)に関しては株式会社 イーエスピー企画のサイトを参照のこと。

<http://www.esp.jp/>

2.2. 対象とする開発環境

対象とする開発環境は以下である。

開発を行うホスト PC の OS は Windows とする。動作確認は Windows 10 にて行った。

表 2-2 開発環境

分類	名称	備考
開発環境	IAR Embedded Workbench for Arm 9.32.2	IAR Systems

- ① バージョンは動作確認に使用したバージョンを示している。

2.3. デバイスドライバ

μT-Kernel 3.0 BSP では、トロンフォーラムが提供する μT-Kernel 3.0 のサンプル・デバイスドライバを、対象となる実機に移植して実装している。

以下に本実装に含まれるデバイスドライバを示す。

表 2-3 本実装に含まれるデバイスドライバ

種別	デバイス名	デバイス	IO ピン	コネクタ
UART	sera	UART0	PC[0:1]	USB2

2.4. 関連ドキュメント

表 2-4 関連ドキュメント一覧

分類	名称	発行
OS 仕様	μT-Kernel 3.0 仕様書 (Ver.3.00.01)	トロンフォーラム TEF020-S004-3.00.00
デバイスドライバ	μT-Kernel 3.0 デバイスドライバ 説明書 (Ver.1.00.2)	トロンフォーラム TEF033-W007-210331
実装仕様書	μT-Kernel 3.0 実装仕様書 (SBK-M4MN)	ユーシーテクノロジー(株)
構築手順書 (本書)	μT-Kernel 3.0 構築手順書 (SBK-M4MN)	ユーシーテクノロジー(株)
ターゲットボード	SBK-M4MN 取り扱い説明書 SBK-M4MN 回路図	株式会社 イーエスピー 企画

- ❶ トロンフォーラムが発行するドキュメントは、トロンフォーラムの Web ページ、または GitHub で公開する μT-Kernel 3.0 のソースコードに含まれている。
<https://www.tron.org/ja/specifications/>
https://github.com/tron-forum/mtkernel_3
- ❷ ユーシーテクノロジー(株)が発行するドキュメントは、ユーシーテクノロジー(株)の GitHub で公開する μT-Kernel 3.0 のソースコードに含まれている。
https://github.com/UCTechnology/mtk3_bsp

3. 開発環境の準備

μ T-Kernel 3.0 BSP を使用するにあたり、以下の手順で開発環境の準備を行う。

3.1. IAR Embedded Workbench for Arm のインストール

IAR の Web サイト(下記)から IAR Embedded Workbench for Arm(EWARM)をダウンロードする。

<https://www.iar.com/jp/products/architectures/arm/iar-embedded-workbench-for-arm/>

[無償の評価版]→[ダウンロード]を選択すると「評価用ライセンスのユーザ登録」が表示される、そこに情報を入力し、[登録を送信]をクリックすると記入したメールアドレスにダウンロードのリンクが記載されたメールが届くので、そこからインストーラをダウンロードする。

インストーラを実行し、指示に従って EWARM のインストールを進める。

💡 2023/03/22 時点での EWARM の最新バージョンは、Version 9.32.2 である。
本資料では移植作業に使用した Version 9.32.2 を基に説明する。

3.2. SBK-M4MN の設定と接続

SBK-M4MN は、USB1(CMSIS-DAP 接続 USB コネクタ)を PC に接続することで CMSIS-DAP を利用したデバッグが可能となる。

また、SBK-M4MN では、ターゲットマイコン(TMPM4MNFYAFG)の UART0 を USB2 経由で PC にシリアルポートとして接続することができる。そこで、本実装では UART0 を T-Monitor のコンソールとして利用している。

開発時にはターゲットボードの USB1 を CMSIS-DAP、USB2 を UART0 として PC と接続する(下図参照)。

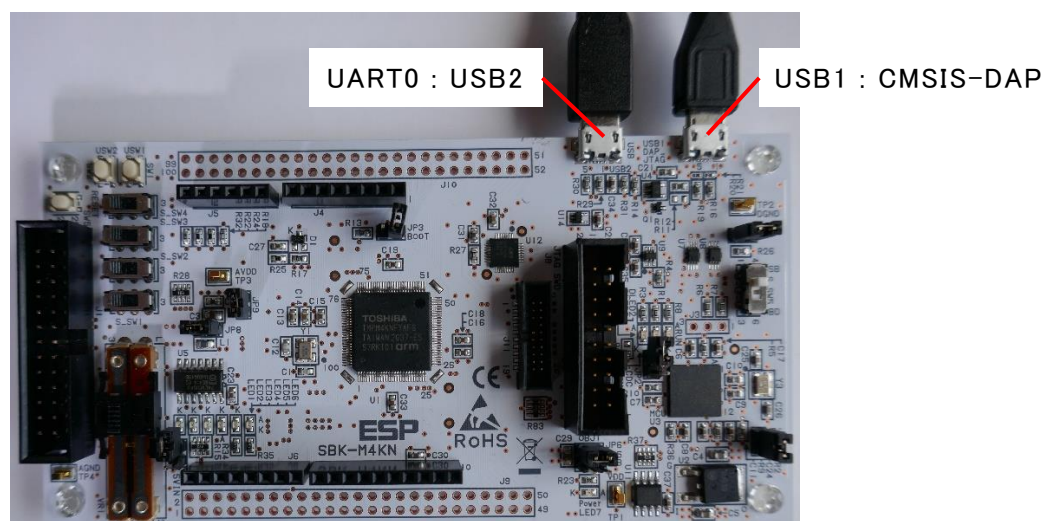


図 1 UART0 と CMSIS-DAP を USB で接続

3.3. ソースコードの展開

ソースコードを C:\UCT に展開する。

- ❶ 本書では、C:\UCT\utkernel_source にインストールされているとして説明する。
- ❷ ユーシーテクノロジー(株)の GitHub から本実装のソースコードを入手した場合は、ブランチを ewarm-sbk_m4mn に切り替える必要がある。

3.4. Tera Term の設定

Tera Term を起動し、メニューの[設定]→[シリアルポート]を選択する。

[ポート]で SBK-M4MN が接続されている USB シリアルポート(下図では COM3)を選択し、通信パラメータとして下図のように設定して、[新規オープン]をクリックする。



図 2 シリアルポートの選択と通信パラメータの設定

- ❶ 設定方法は Tera Term のバージョンによって異なる場合がある。
詳細は Tera Term のマニュアルを参照のこと。
(上図は Tera Term Version 4.106 の画面である。)
- ❷ 「スピード」(通信速度)と通信パラメータ(データ長、パリティビット、ストップビット)については SBK-M4MN のコンソール設定に合わせる必要がある。
SBK-M4MN のコンソール設定については、「μ T-Kernel 3.0 実装仕様書(SBK-M4MN)」の「9.2.2. コンソール入出力」を参照のこと。

4. ビルドと実行

4.1. サンプルプロジェクト

本実装では以下のサンプルプロジェクトを用意してある。

(1) 基本構成

コンソールにメッセージを表示してキー入力待ちになる。

任意のキーを入力するとシステムを終了する。

- ・ サンプルアプリケーションのソースコード :

`app_program/app_main.c`

- ・ プロジェクトファイル :

`ide/iar/sbk_m4mn/sbk_m4mn.eww`

(2) シリアル通信ドライバサンプル

シリアル通信ドライバの設定の取得、再設定、入力のエコーを行う。

- ・ サンプルアプリケーションのソースコード :

`app_serial/app_main.c`

- ・ プロジェクトファイル :

`ide/iar/sbk_m4mn_serial/sbk_m4mn.eww`

サンプルアプリケーションの詳細については、各アプリケーションのソースコードを参照のこと。

4.2. 実行手順

EWARM を利用した μT-Kernel 3.0 BSP プロジェクトでのビルドから実行までの手順は以下のとおりである。

- (1) EWARM の起動
- (2) ビルド構成の選択
- (3) ビルド
- (4) ターゲットボードの接続
- (5) デバッグ開始
- (6) エントリポイントで停止
- (7) 実行
- (8) 動作確認

以下、基本構成を利用した実行手順の詳細について説明する。

4.2.1. EWARM の起動

ide/iar/sbk_m4mn/sbk_m4mn.eww をダブルクリックして EWARM 開発環境を起動する。
EWARM が起動すると以下の画面が表示される。

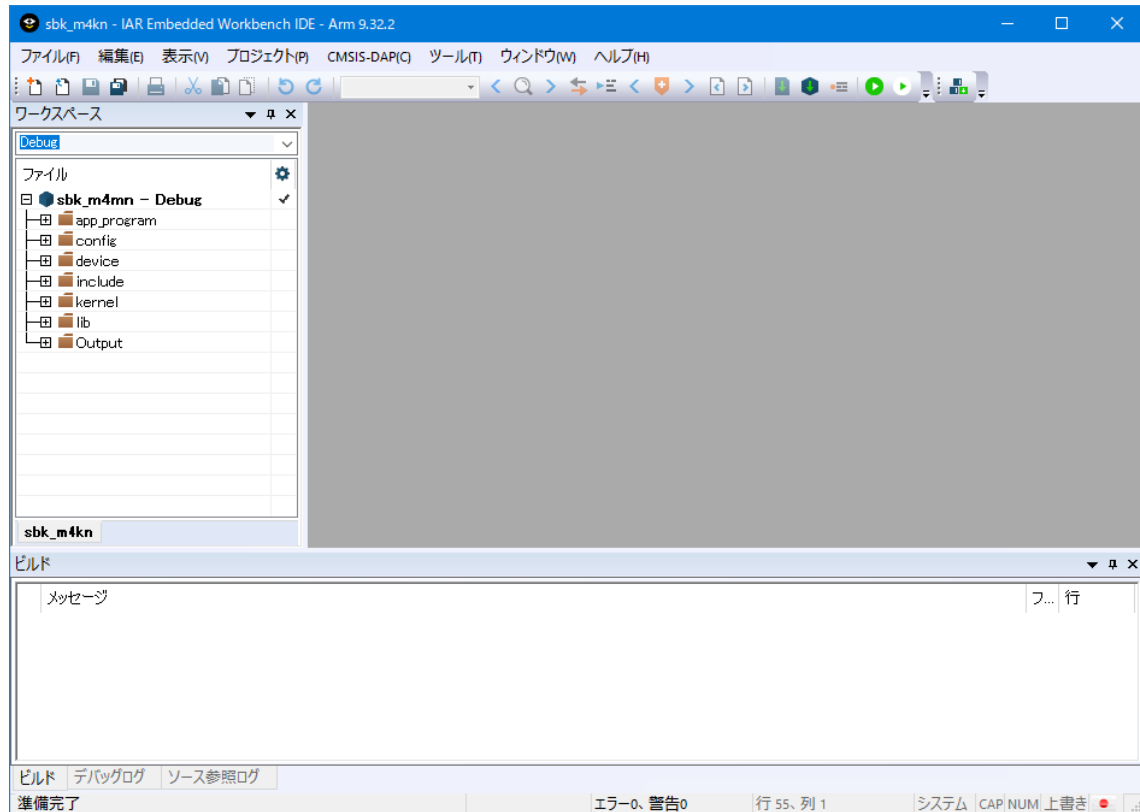


図 3 EWARM の起動

4.2.2. ビルド構成の選択

ビルド構成のプルダウンメニューで使用するターゲットビルドとして [Debug] が選択されていることを確認する。

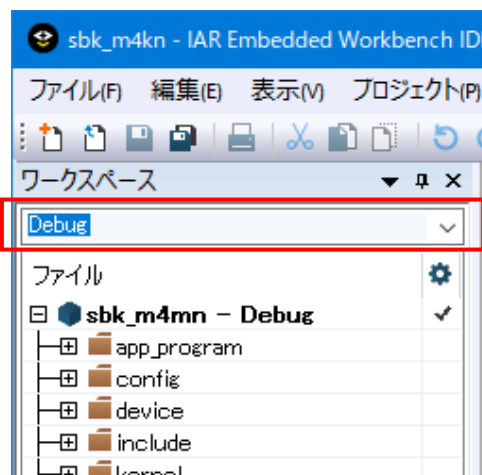


図 4 ビルド構成の選択

4.2.3. ビルド

[メイク]をクリックしてビルドする。

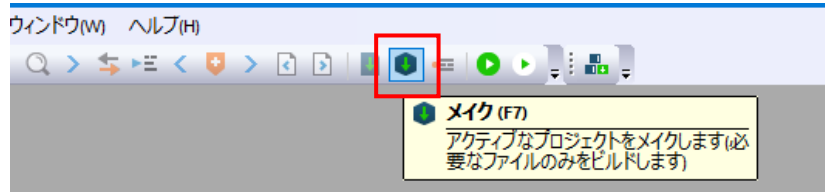


図 5 ビルド

以下の警告が発生するが、ここでは無視して構わない。

サンプルプロジェクトビルド時に発生するワーニング

```

ワーニング[Go003]: Optimization can only be lowered from the default, not raised.
                  C:¥UCT¥utkernel_source¥kernel¥sysdepend¥sbk_m4mn¥cpu_clock.c 55
ワーニング[Pe815]: type qualifier on return type is meaningless
                  C:¥UCT¥utkernel_source¥kernel¥sysdepend¥cpu¥core¥armv7m¥reset_hd
                  l.c 32
ワーニング[Pe111]: statement is unreachable
                  C:¥UCT¥utkernel_source¥kernel¥sysinit¥sysinit.c 102
ワーニング[Pe111]: statement is unreachable
                  C:¥UCT¥utkernel_source¥app_program¥app_main.c 63
ワーニング[Pe111]: statement is unreachable
                  C:¥UCT¥utkernel_source¥app_program¥app_main.c 79
ワーニング[Pe111]: statement is unreachable
                  C:¥UCT¥utkernel_source¥kernel¥tkernel¥task_manage.c 237

```

❗ 警告の内容に関しては、ソースコードの当該箇所を参照のこと。

4.2.4. ターゲットボードの接続

「3.2. SBK-M4MN の設定と接続」の説明に従ってターゲットボード(SBK-M4MN)と PC を接続する。

「3.4. Tera Term の設定」の説明を参考にしてコンソール用の通信ソフトを起動して、シリアルポートの設定を行う。

4.2.5. デバッグ開始

ビルド完了後、[ダウンロードしてデバック]をクリックしてデバッグを開始する。

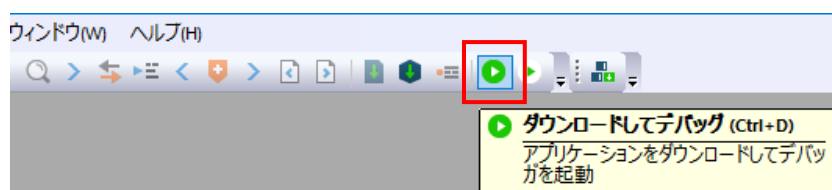


図 6 デバッグ開始

4.2.6. エントリポイントで停止

エントリポイント(kernel/sysdepend/cpu/core/armv7m/reset_hdl.c の Reset_Handler)で停止する(下図)。

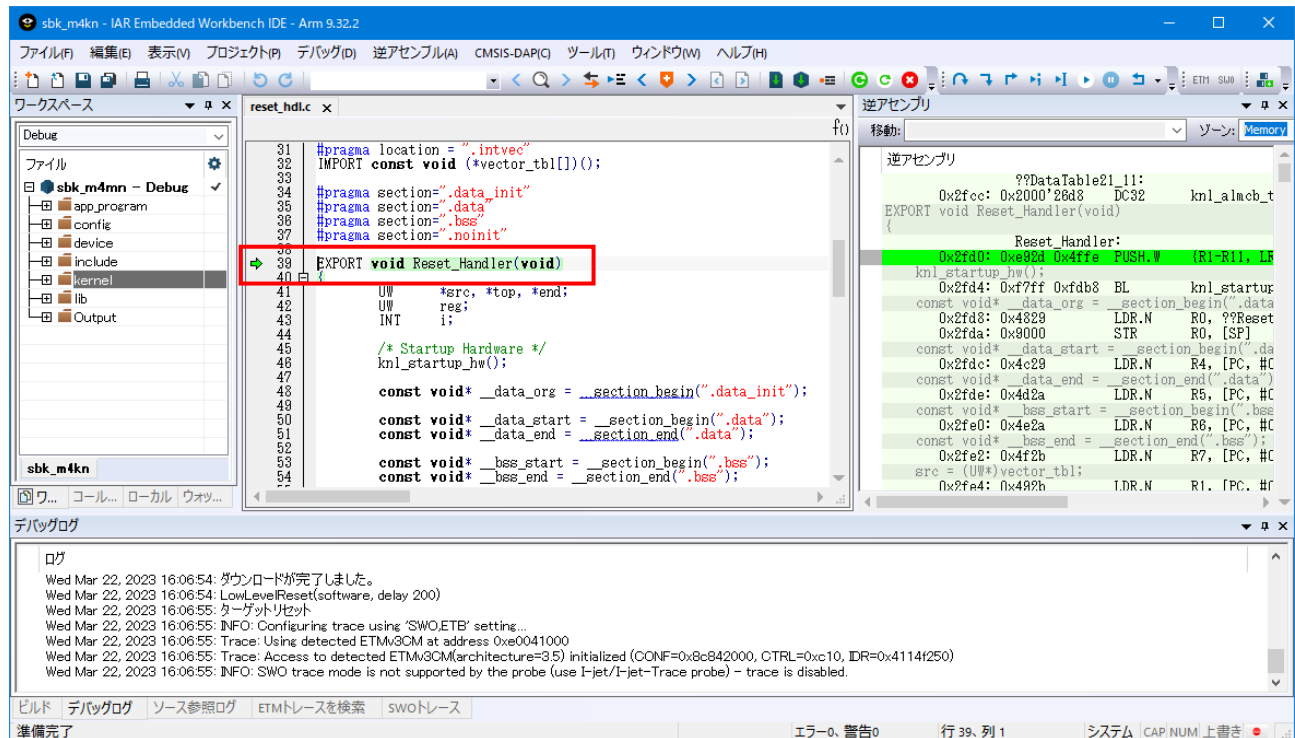


図 7 エントリポイントで停止

4.2.7. 実行

[実行]をクリックして実行する。

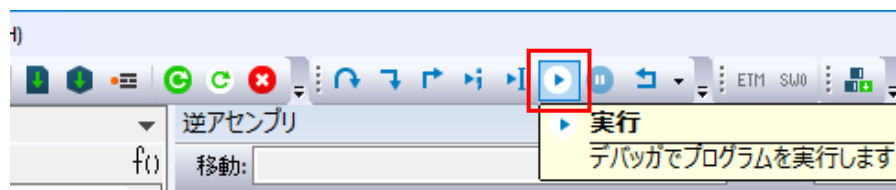


図 8 実行

4.2.8. 動作確認

通信ソフトの画面には、図 9 のようなメッセージが表示される。

ターゲットボードでは、約 1 秒の間隔で LED1~5 が点滅を繰り返す。

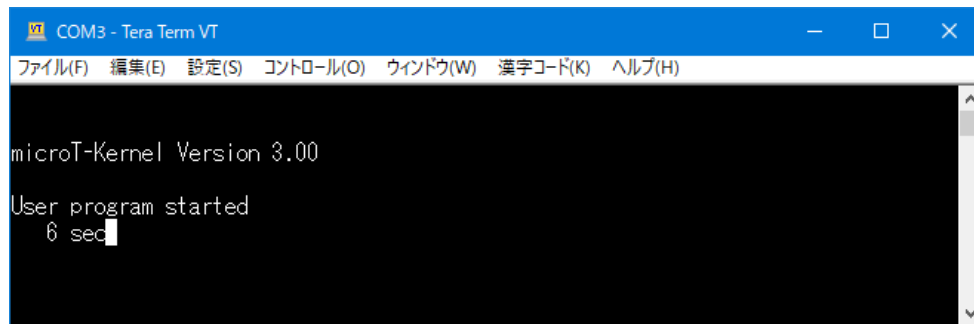


図 9 コンソールの表示

サンプルのアプリケーションタスクは以下に実装されている。

app_program/app_main.c

このサンプルアプリケーションではタスクを 1 つ生成・起動し、その中で LED を点滅させている。

また、初期タスクである `usermain()` 関数では、タスクを生成・起動した後、「1 秒間の待ち」と「コンソールへの経過秒数(カウント数)」の表示を繰り返し実行している。

詳細については、サンプルアプリケーションのソースコードを参照のこと。

5. μT-Kernel 3.0 のディレクトリ／ファイル構成

SBK-M4MN 版の μT-Kernel 3.0 のディレクトリおよびファイルの構成は、μT-Kernel 3.0 の正式リリース版に準じて以下のように構成してある。

表 5-1 プロジェクトのファイル構成

ディレクトリ名、ファイル名	内容
app_program/	μT-Kernel 3.0 用サンプルアプリケーション
app_serial/	シリアル通信ドライバのサンプルアプリケーション
config/	コンフィギュレーション
device/	デバイスドライバ
include/	各種定義ファイル
kernel/	μT-Kernel 3.0 本体
lib/	ライブラリ
ide/	プロジェクトファイル
docs/	ドキュメント
README.md	概要説明
ucode.png	μT-Kernel 3.0 の ucode
build_make/	Make 構築ディレクトリ(本実装では未使用)
etc/	リンカファイル等(本実装では未使用)

5.1. μT-Kernel 3.0 のソースコード

config/、device/、docs/、include/、kernel/、lib/ の各ディレクトリには μT-Kernel 3.0 のソースコードが含まれる。

サンプルプログラム程度であれば特に変更する必要はないが、タスクやセマフォの最大数などの調整が必要な場合は config/以下のファイルで調整することになる。

- ④ μT-Kernel 3.0 のコンフィギュレーションについては「μT-Kernel 3.0 共通実装仕様書」を参照のこと

5.2. μT-Kernel 3.0 BSP のディレクトリとファイル

README.md は μT-Kernel 3.0 BSP に関する概要説明である。μT-Kernel 3.0 BSP のソースコードを GitHub から入手する前提で説明されているので、ターゲットボードのブランチに関する説明が含まれている。

ucode.png は μT-Kernel 3.0 の ucode である(オリジナルから変更していない)。

docs/には μT-Kernel 3.0 BSP 関連のドキュメントが含まれている。

build_make/、etc/は gcc を用いて開発する場合に利用する。本実装では EWARM を利

用するので、これらのディレクトリは利用しない。

5.3. プロジェクトファイル

`ide/`は本実装で追加したディレクトリであり、プロジェクトファイルが含まれる。

本実装では `ide/iar/`以下に EWARM のプロジェクトファイルが含まれている。

5.4. サンプルアプリケーション

サンプルアプリケーションのベースディレクトリは任意であるが、本実装では以下を利用している。

- `app_program/app_main.c`
- `app_serial/app_main.c`

アプリケーションプログラムを開発する場合は、このディレクトリをコピーして改造するか、または `app_program/`に直接アプリケーションプログラムを追加していく。

- ❗ ディレクトリやファイルの構成を変更した場合は、EWARM においてグループやファイルの追加／削除を行う必要がある。

μ T-Kernel 3.0

構築手順書 (SBK-M4MN)

Rev 3.00.00 (April, 2023)

ユーシーテクノロジー株式会社

141-0031 東京都品川区西五反田 2-12-3 第一誠実ビル 9F

©2023 Ubiquitous Computing Technology Corporation All Rights Reserved.