µT-Kernel 3.0 BSP2 入門ガイド

トロンフォーラム T3 WG
INIAD(東洋大学情報連携学部)
豊山 祐一

アジェンダ

- µT-Kernel 3.0 BSP2とは
- BSP2を使ったプログラミングの概要
- BSP2のサンプル・プロジェクト
- マイコンのペリフェラルの使用方法
- µT-Kernel 3.0 各種情報

- 質疑応答

µT-Kernel 3.0 BSP2とは

µT-Kernel 3.0とは

- IEEE 2050-2018国際標準仕様に準拠した軽量RTOS
  - 小規模な組込みシステムやIoTエッジノード向けの高機能RTOS
- TRONプロジェクトの最新のリアルタイムOS
  - ▶T-Kernelとの高い互換性。µITRONからの移行も容易。
- GitHubからのオープンソースの
  ライセンス(T-License)でソースコード公開
- BSPによる市販ボードへの対応

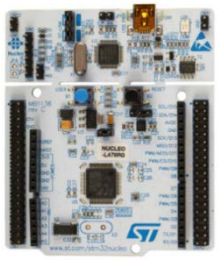
µT-Kernel 3.0

µT-Kernel 3.0 BSP (Board Support Package)

- µT-Kernel 3.0の入門や学習、評価用に市販マイコン・ボードですぐに使用可能なパッケージ
  - OS、基本デバイスドライバを実装済みのプロジェクトを提供
  - ソフトウェアはメーカのファームウェアに依存しない
  - 2021年 GitHubから一般公開

対応マイコンボード

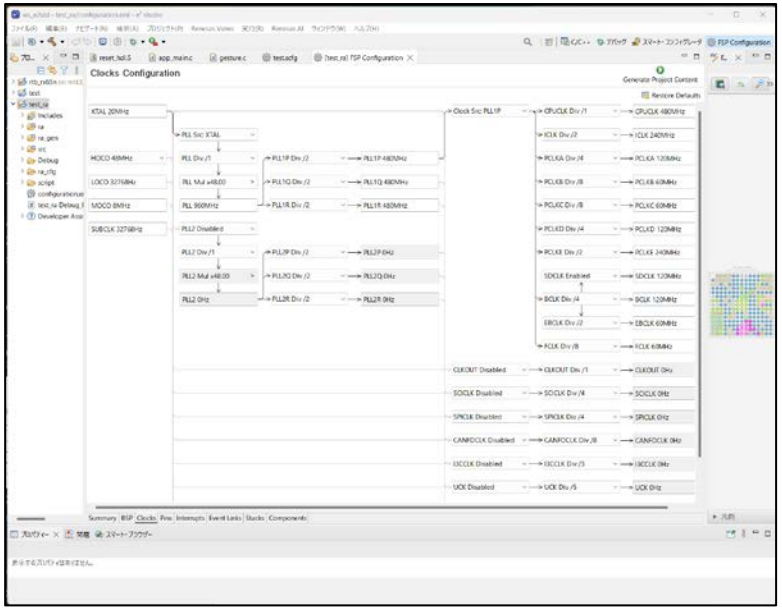
STM32L476 Nucleo-64	STM32H723 Nucleo-144	RX65N Renesas Target Board	RX65N Renesas Starter Kit+	Raspberry Pi Pico
ARM Cortex-M4	ARM Cortex-M7	RXv2	RXv2	ARM Cortex-M0+





マイコンメーカの開発環境への対応

- マイコンに内蔵されるペリフェラルの多機能化、高性能化に伴い、マイコンメーカ提供の開発ツール、ファームウェアを使用したいという要望が増えてきた
  - 今やマイコンを起動させるだけでも大変
    - ▶例えばマイコンのクロックを設定するにもメーカ提供のツール無しでは大変
- 評価、学習を目的としてBSPでは、そのニーズは大きい
  ⇒　マイコンメーカの開発環境に対応したBSP2を開発



µT-Kernel 3.0 BSP2

- マイコンメーカ提供の開発環境、ファームウェアとの親和性を強化
  - メーカ提供開発環境で作成したプロジェクトにそのまま組み込める。
  - メーカ提供のHALなどファームウェアを簡単に使用可能
  - 2023年12月から一般公開
  - 対応マイコンボード
    - ▶STマイクロエレクトロニクス
      - STM32L476 Nucleo-64
      - STM32F401 Nucleo-64
      - STM32F411 Nucleo-64
      - STM32F446 Nucleo-64
      - STM32G431 Nucleo-64
      - STM32G491 Nucleo-64
      - STM32F767 Nucleo-144
      - **STM32H723 Nucleo-144**
    - ▶ルネサス エレクトロニクス
      - EK_RA6M3
      - **EK_RA8M1**
      - Arduino UNO R4
    - ▶NXP
      - **FRDM-MCXN947**
      - LPC55S69-EVK
    - ▶インフィニオン
      - **EVK-XMC7200**

BSP2を使った
プログラミングの概要

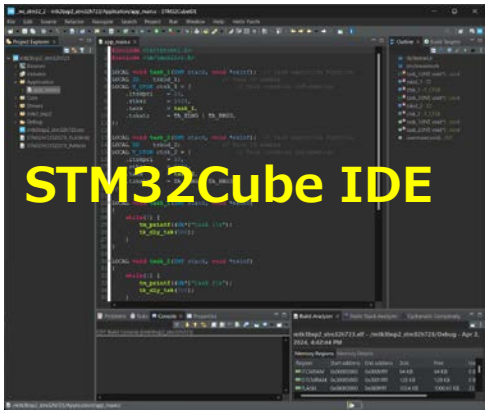
マイコンメーカの提供する開発環境・ツールを使用

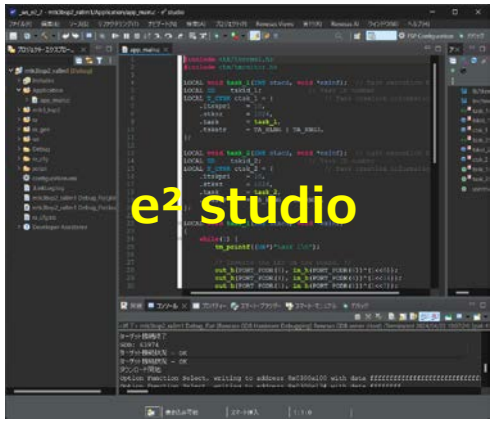
- マイコンメーカが提供するIDE(統合開発環境)、コンフィギュレータ、ファームウェアなどを使用

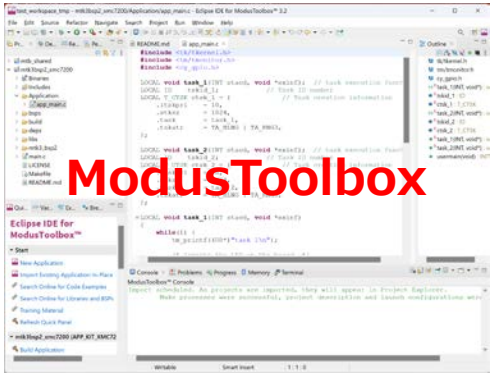
STマイクロエレクトロニクス STM32CubeIDE STM32CudeMX STM32CubeFW	ルネサス エレクトロニクス e²studio Smart Configurator Flexible Software Package (FSP)
インフィニオン テクノロジーズ ModusToolbox	NXP MCUXpresso IDE

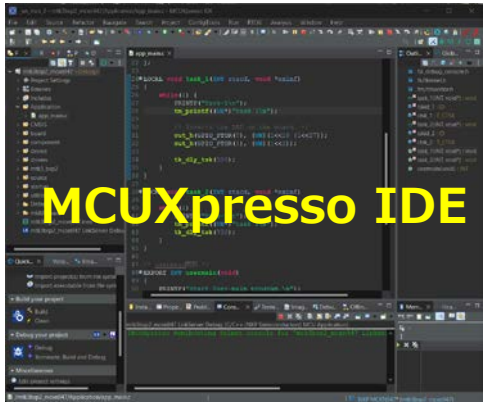
IDE（統合開発環境）

- Eclipse Embedded CDT (C/C++ Development Tools)をベースに各社のマイコンサポートツールが統合
  - 基本的なデバッグ操作などは各社共通
  - 各社のコンフィギュレータやファームウェアがIDEに統合









各社IDEの紹介ドキュメント

- IDEの使い方は各社が提供のドキュメントなどをご覧ください
- 5月にトロンフォーラムのウェビナーにて各社からの説明会を実施
  - 資料は以下からご覧いただけます
    - ▶https://www.tron.org/mt-kernel3/2024/05/30/post-82/

µT-Kernel 3.0 BSP2のソフトウェア開発手順

- IDEではソフトウェアをプロジェクト単位で開発します

IDE(コンフィギュレータ)を使用してプロジェクトを作成

①ハードウェアの選択

②クロックの設定

③端子の設定

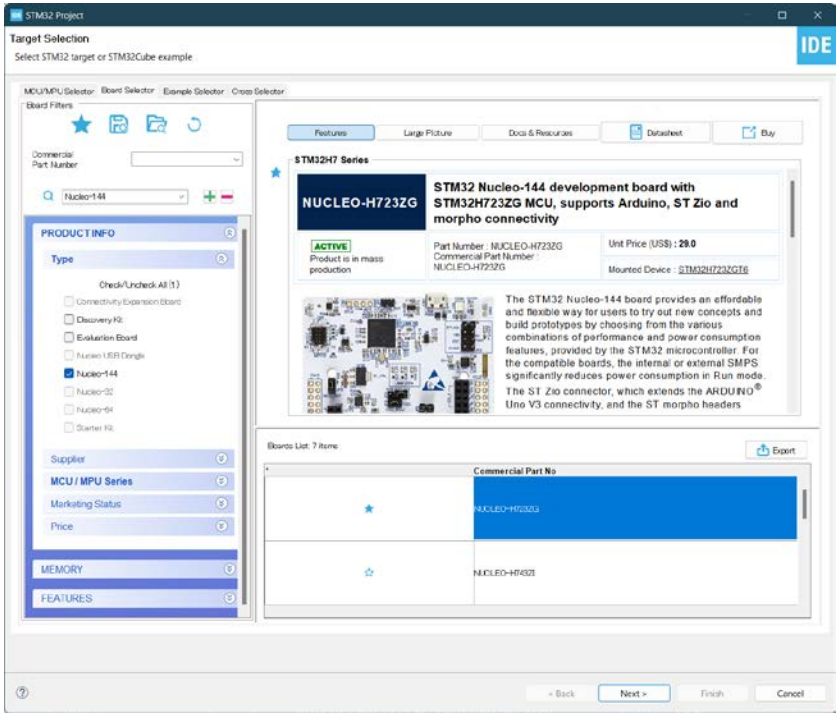
④ファームウェアの選択、設定

⑤µT-Kernel 3.0 BSP2の組込み

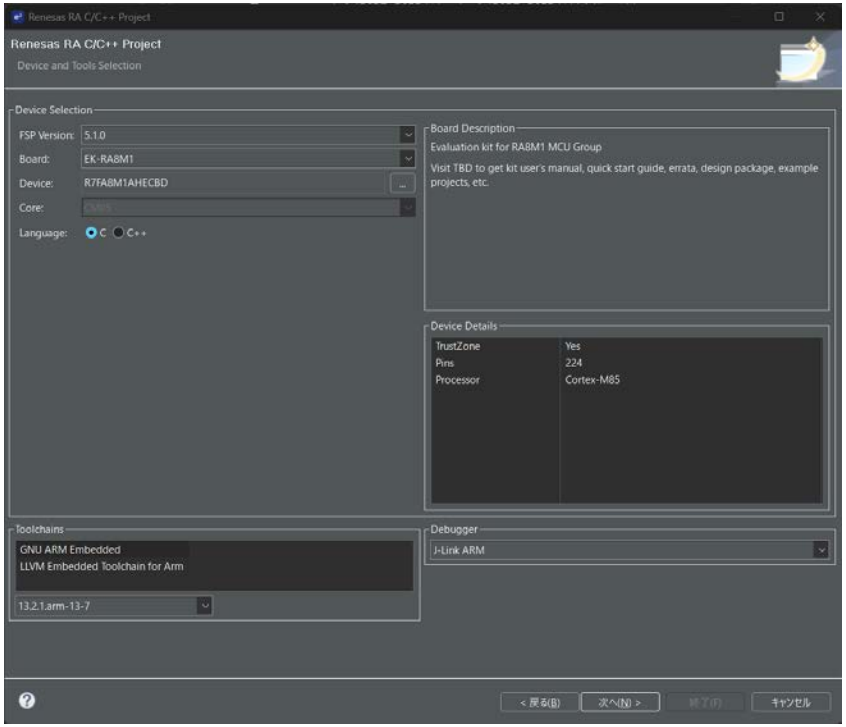
⑥アプリケーションの開発

手順(1) ハードウェアの選択

- 使用するマイコン、ボードなど対象ハードウェアを選択しプロジェクトを作成します
  - メーカ純正のボードならばIDE上で選択可能です
- 選択したハードウェアに応じた基本設定が行われます



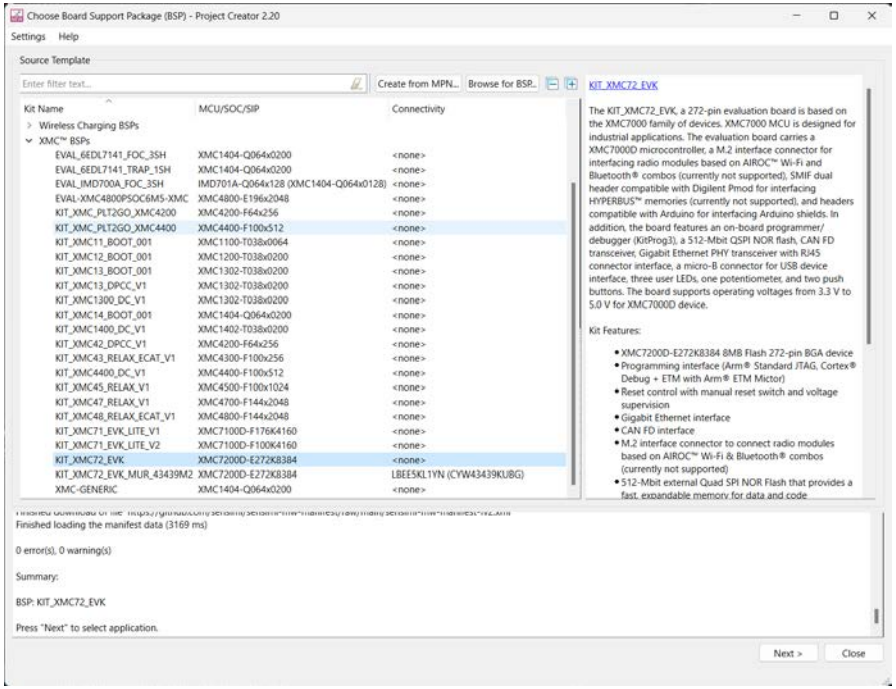
STM32Cube IDE



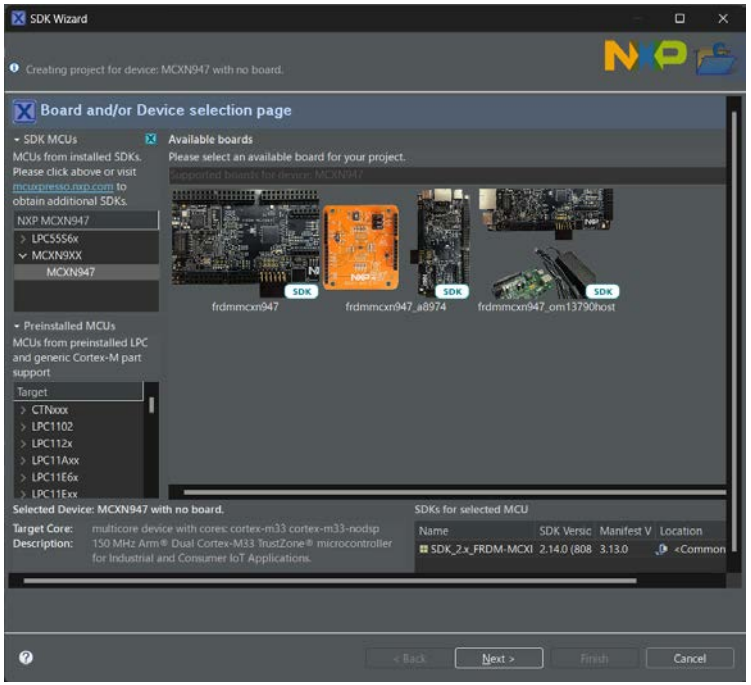
e² studio

手順(1) ハードウェアの選択

- 使用するマイコン、ボードなど対象ハードウェアを選択します
  - メーカ純正のボードならばIDE上で選択可能です
- 選択したハードウェアの基本設定が自動生成されます



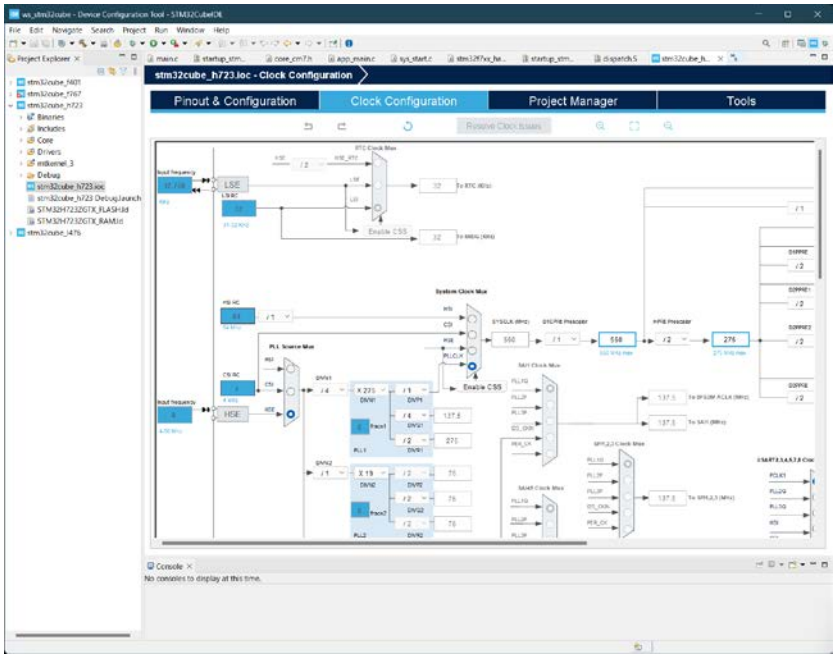
ModusToolbox



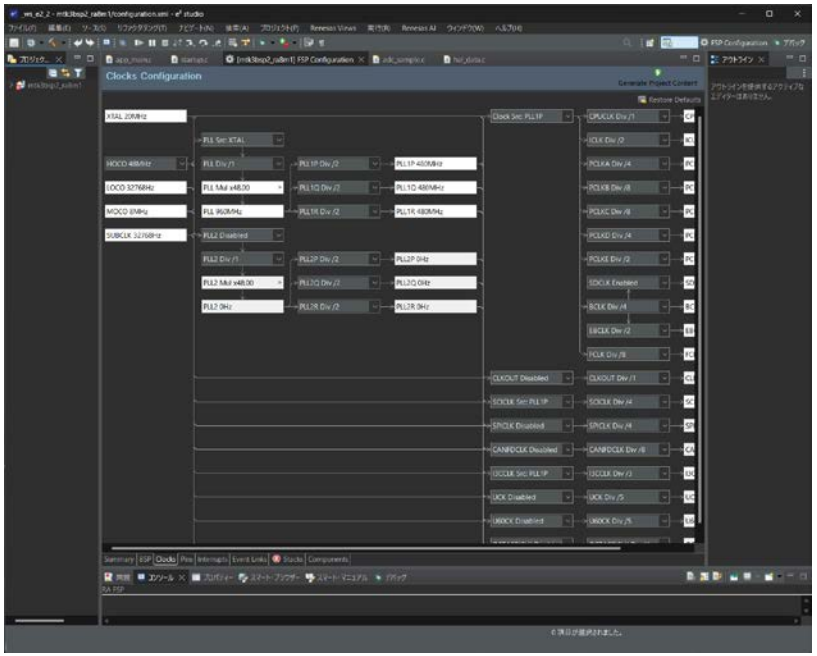
MCUXpresso IDE

手順(2) クロックの設定

- マイコンの内部クロックの設定を行います
  - ボードを選択していれば基本的な設定はできています
- ペリフェラルに関しては基本的には設定する必要があります



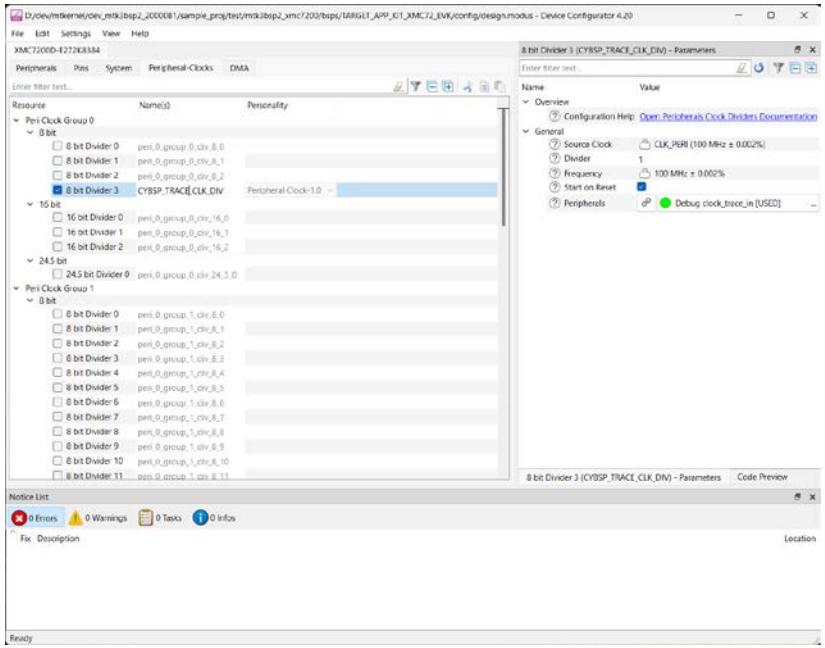
STM32Cube IDE



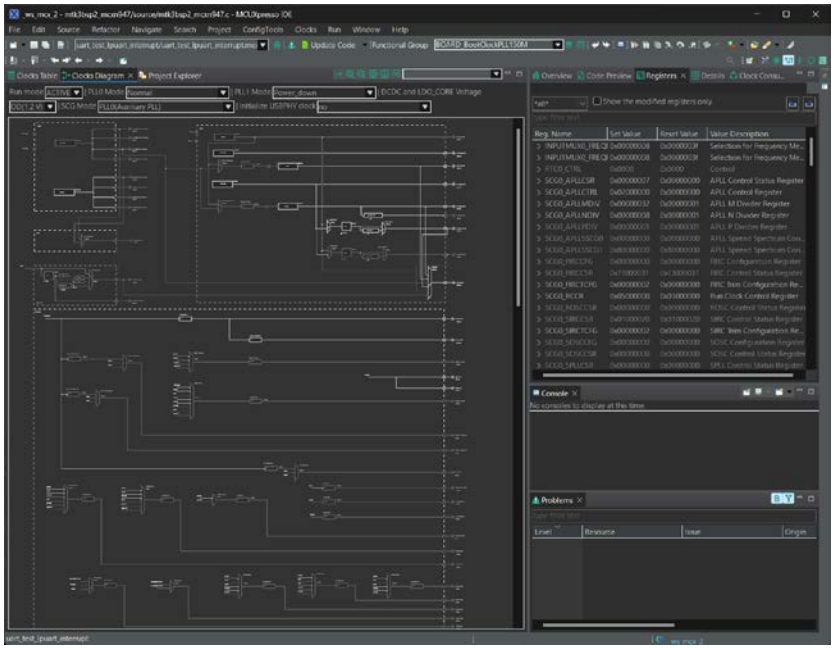
e² studio

手順(2) クロックの設定

- マイコンの内部クロックの設定を行います
  - ボードを選択していれば基本的な設定はできています
- ペリフェラルに関しては基本的には設定する必要があります



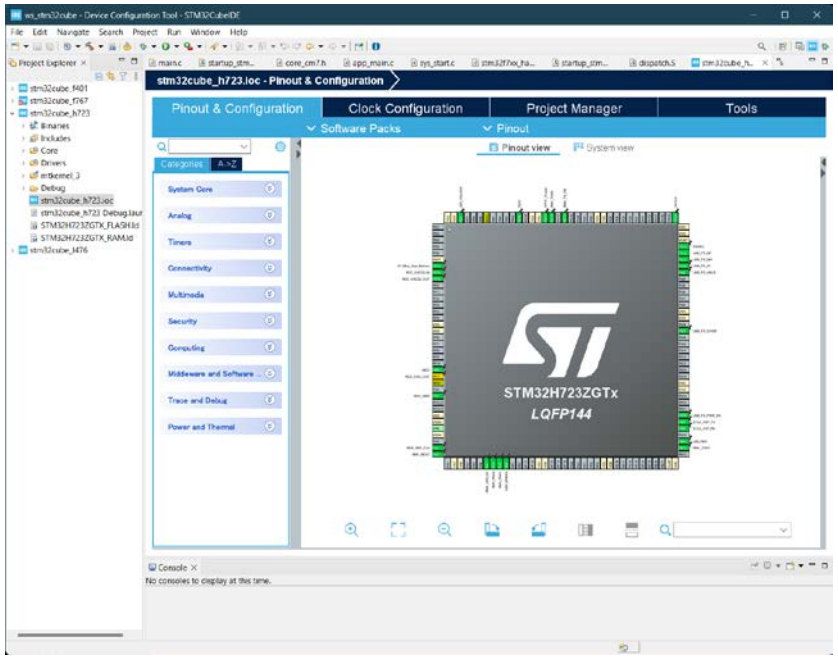
ModusToolbox



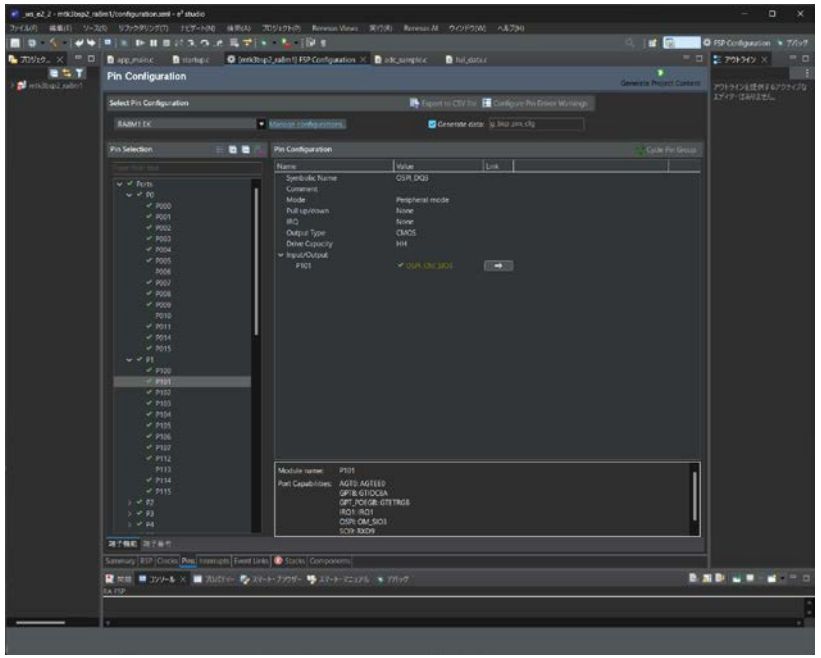
MCUXpresso IDE

手順(3) 端子の設定

- マイコンの各端子の設定を行います
  - ボードを選択していれば基本的な設定はできています
- ペリフェラルに関しては基本的には設定する必要があります



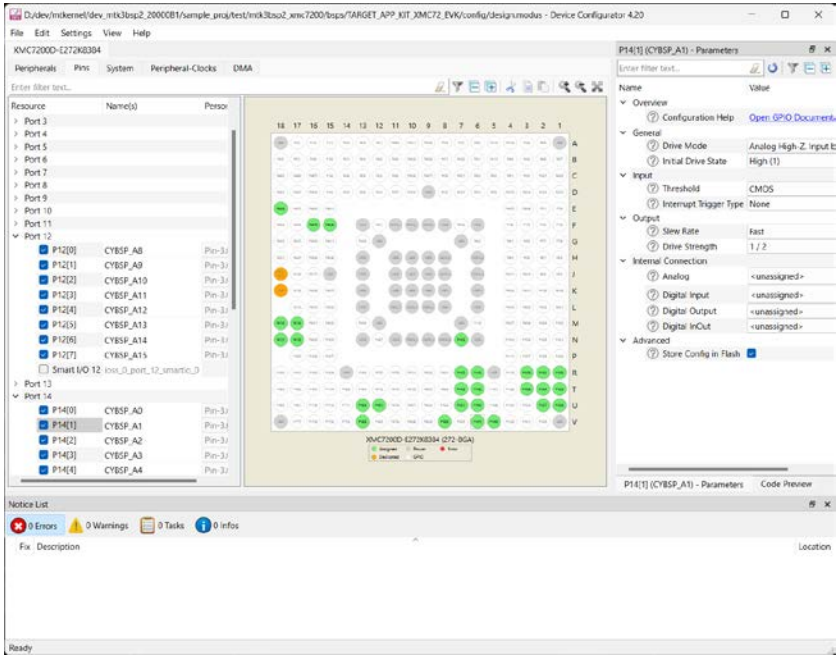
STM32Cube IDE



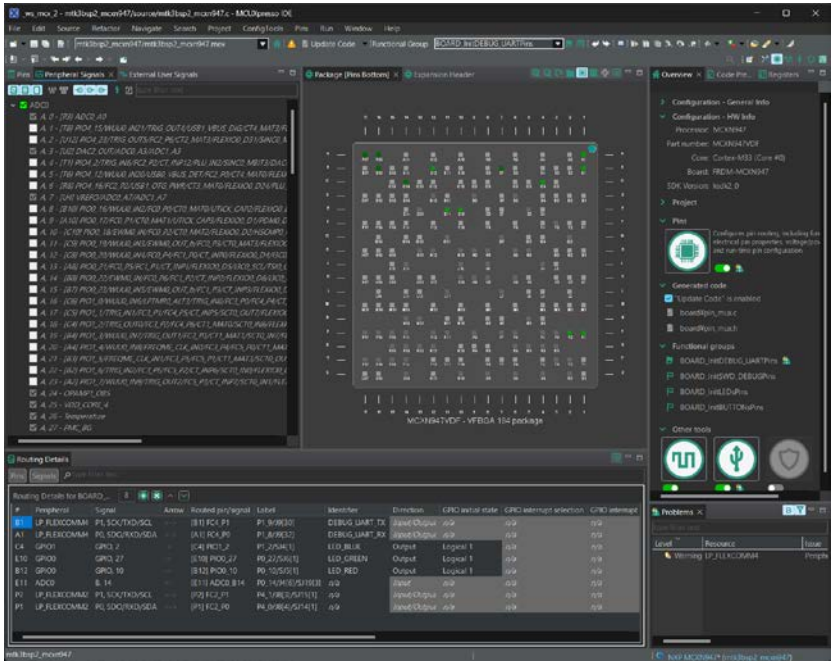
e² studio

手順(3) 端子の設定

- マイコンの各端子の設定を行います
  - ボードを選択していれば基本的な設定はできています
- ペリフェラルに関しては基本的には設定する必要があります



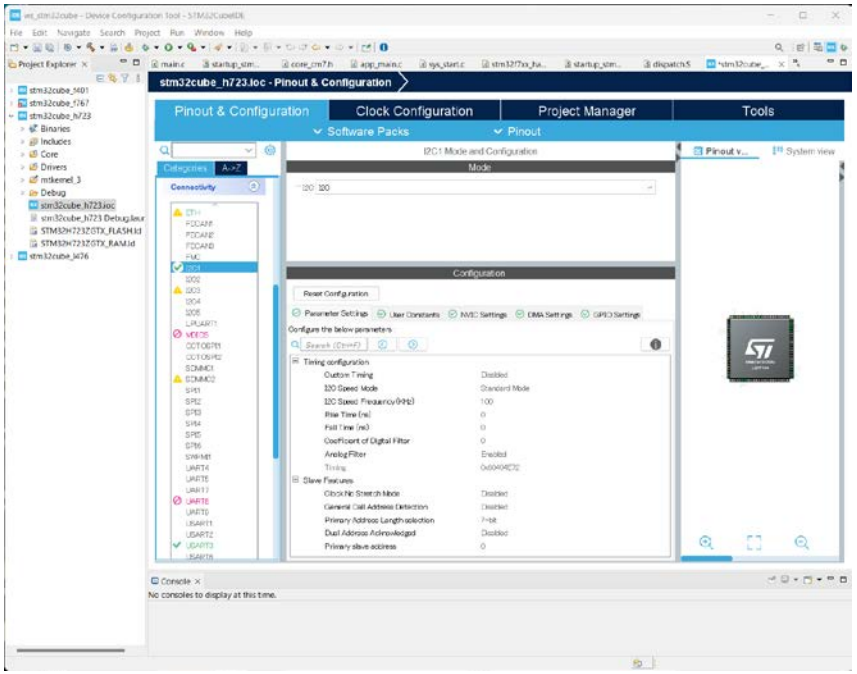
ModusToolbox



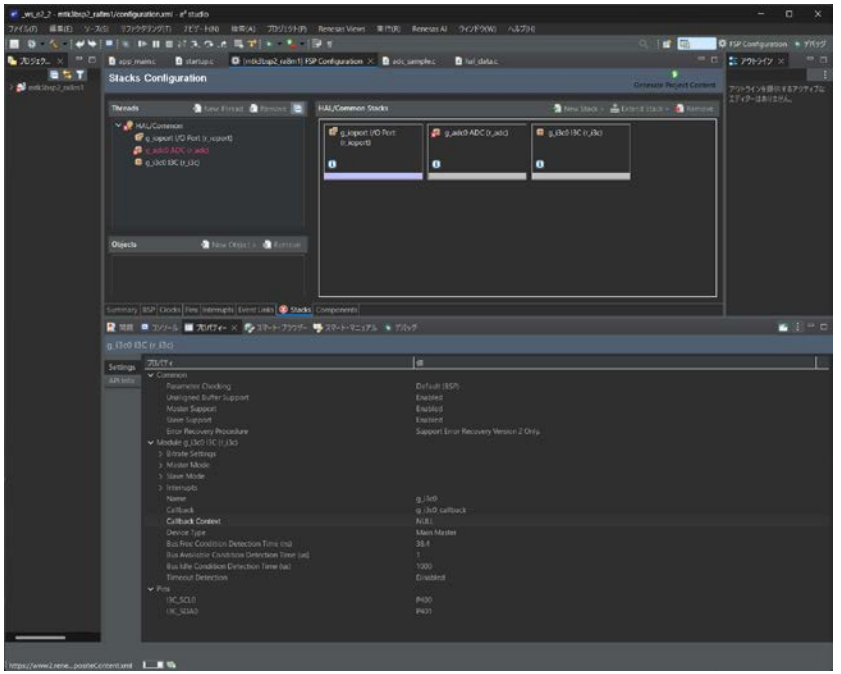
MCUXpresso IDE

手順(4) ファームウェアの選択、設定

- 使用するHALやドライバなどを選択します
- デバイスの設定などを行います



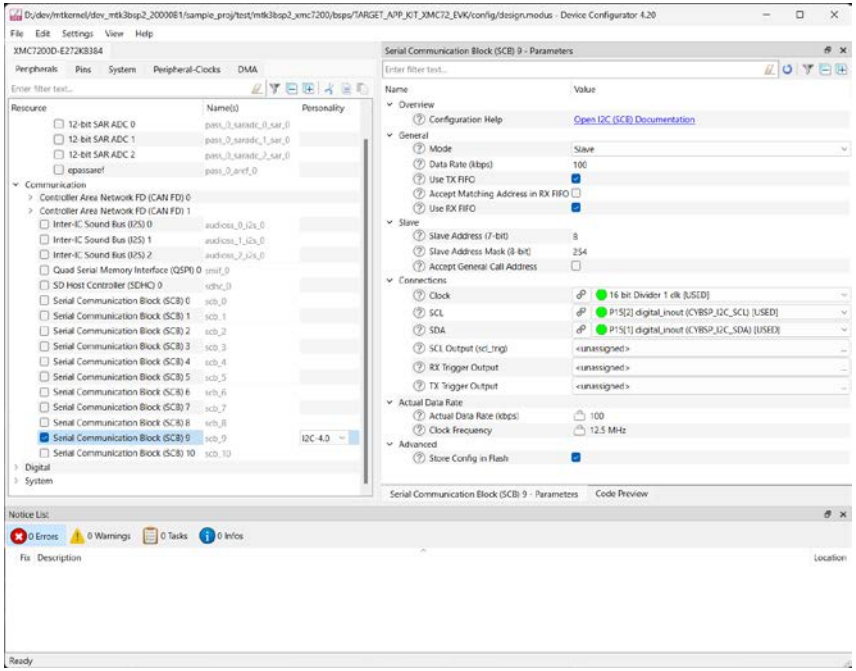
STM32Cube IDE



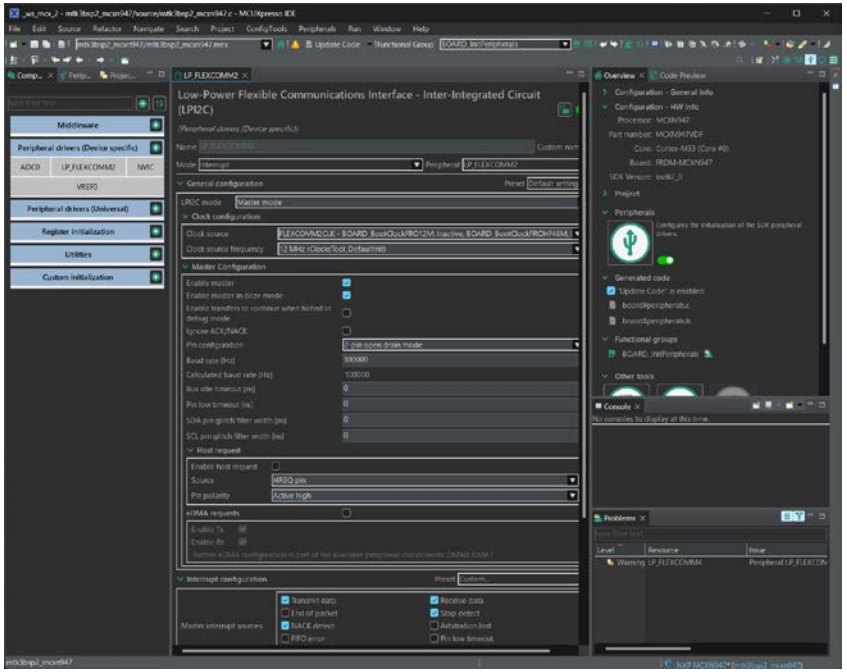
e² studio

手順(4) ファームウェアの選択、設定

- 使用するHALやドライバなどを選択します
- デバイスの設定などを行います



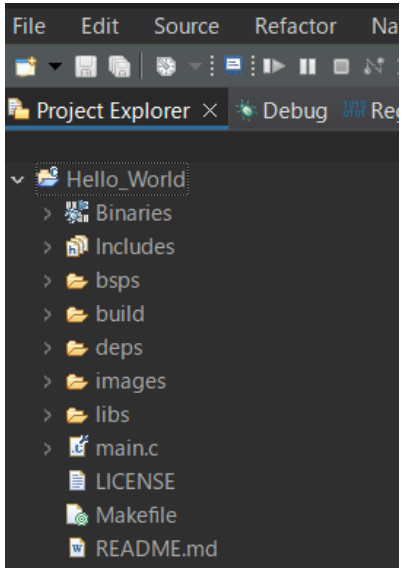
ModusToolbox



MCUXpresso IDE

いったんプロジェクトの動作確認

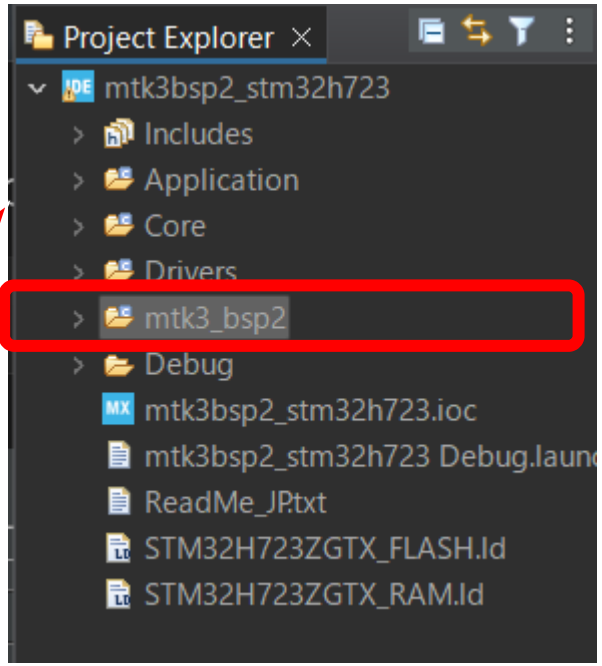
- ここまででプロジェクトの生成ができます
  - コンフィギュレータの設定に応じたソースコードが自動生成されます
- プロジェクトをビルドし、ボードで実行してみましょう
  - 何もしないアプリケーションが動作します
- 注意：自動生成されたファイルを直接編集しないように
  - 設定を変更すると上書きされてしまいます
  - アプリケーションのファイルのみ編集可能です

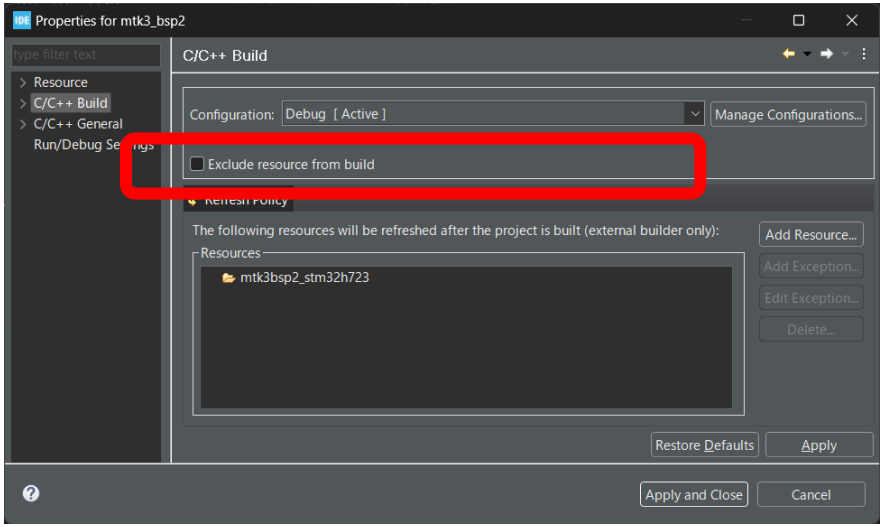


µT-Kernel 3.0 BSP2の組込み(1)
ソースコードの取り込み

- µT-Kernel 3.0 BSP2のソースコードをダウンロードします
  - https://github.com/tron-forum/mtk3_bsp2/releases
- プロジェクトにBSP2のソースコードを追加します
  - ドラッグ＆ドロップでソースコードをコピー
  - コピーした直後はビルドが無効になって場合がありますので有効にします
    - ▶ModusToolboxはこの操作は不要です

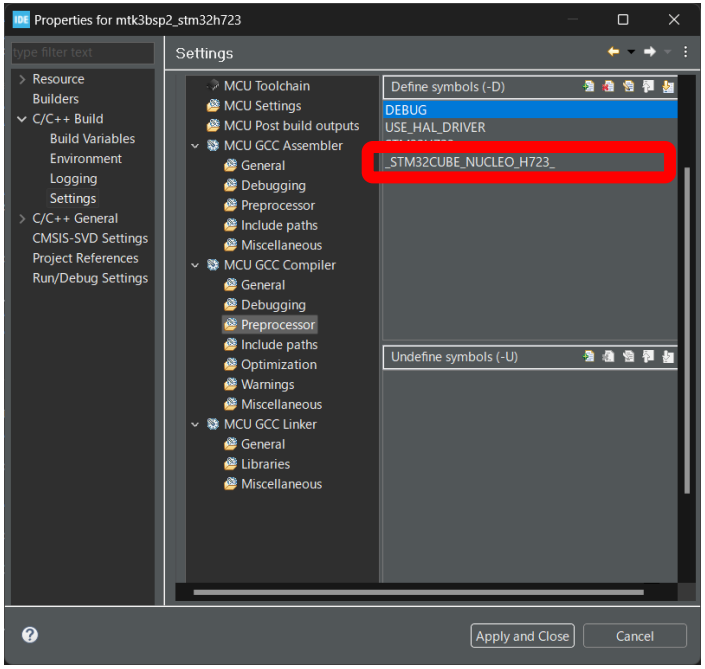
Drag&Drop

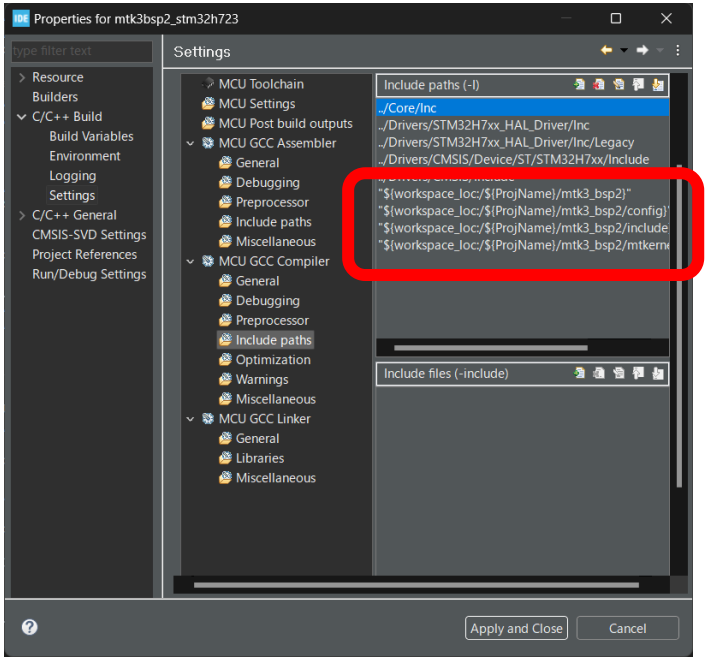




µT-Kernel 3.0 BSP2の組込み(2)
ビルドの設定

- µT-Kernel 3.0 BSP2のビルド設定を追加します
  - ターゲットボードを指定するシンボル定義
  - インクルードパスの設定
- 設定方法はµT-Kernel BSP2付属のマニュアルをご覧ください
  - ModusToolboxはmakefileを編集、それ以外のIDEはGUIで変更できます



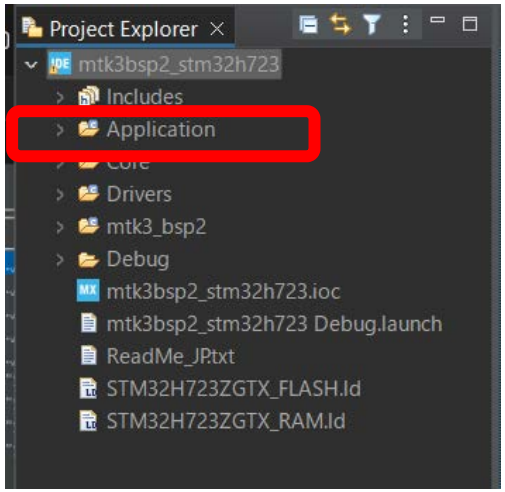


µT-Kernel 3.0 BSP2の組込み(3)
OSの起動関数の追加

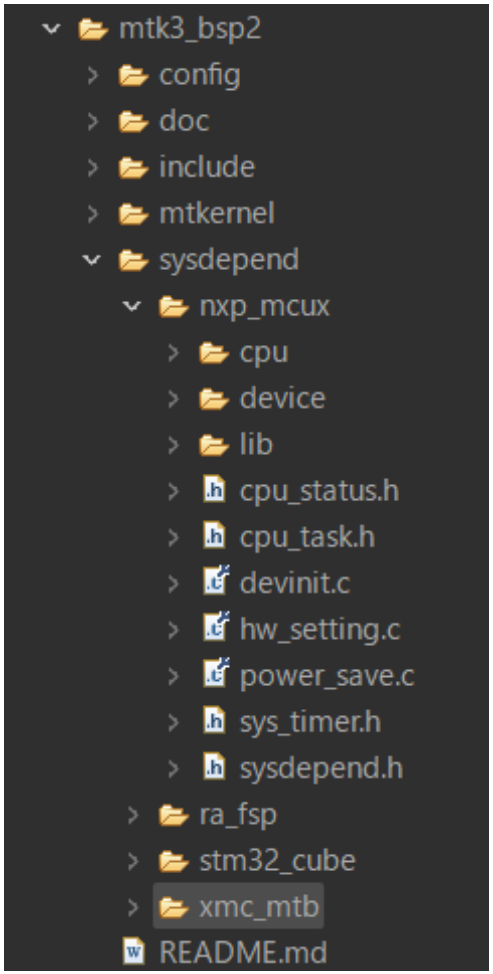
- プロジェクトのアプリケーションの先頭からµT-Kernel 3.0の実行関数を呼び出します
  - void knl_start_mtkernel(void) ;
- アプリケーションの場所はIDEにより異なります。BSP2付属のマニュアルをご覧ください。
- ここまででいったんビルドし、ボードでの実行を確認してください
  - 何もしないµT-Kernelのアプリケーションが実行されます

アプリケーションの開発

- アプリケーションをプロジェクトに追加します
  - 任意の場所に追加できますが、推奨はプロジェクトのルートディレクトリにアプリケーションのディレクトリを追加します
    ▶µT-Kernel 3.0 BSP2やIDEがバージョンアップした際に対応が楽です
- アプリケーションの中でusermain関数を定義します
  - µT-Kenrel 3.0は起動後に初期タスクからusermain関数を実行します
  - もしユーザ定義のusermain関数が無い場合はデフォルトのusermain関数を実行します。この関数は何もしません。



µT-Kernel 3.0 BSP2 ファイル構成



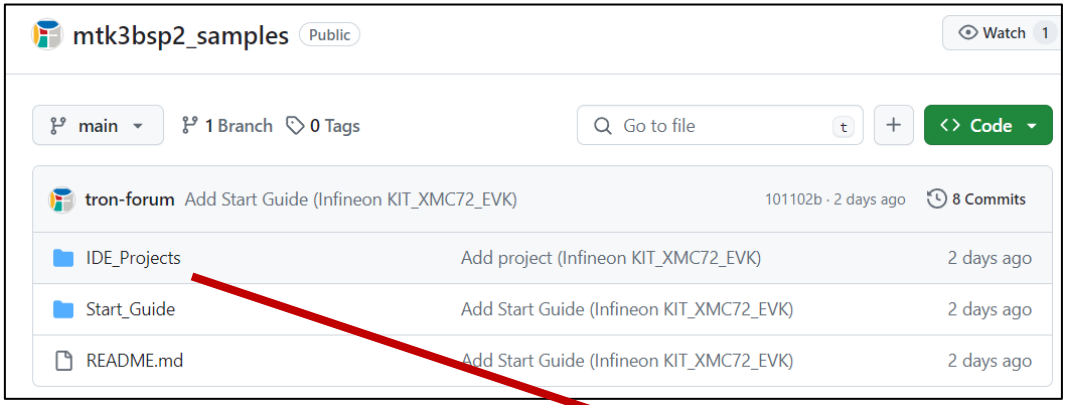
- config　コンフィギュレーション定義
- doc　ドキュメント
- include　定義ファイル
- mtkernel　µT-Kernel 3.0本体
- sysdepend　対象ボード向けH/W依存部
  - nxp_mcux　NXP
  - ra_fsp　ルネサステクノロジー
  - stm32_cube　STエレクトロニクス
  - xmc_mtb　インフィニオン

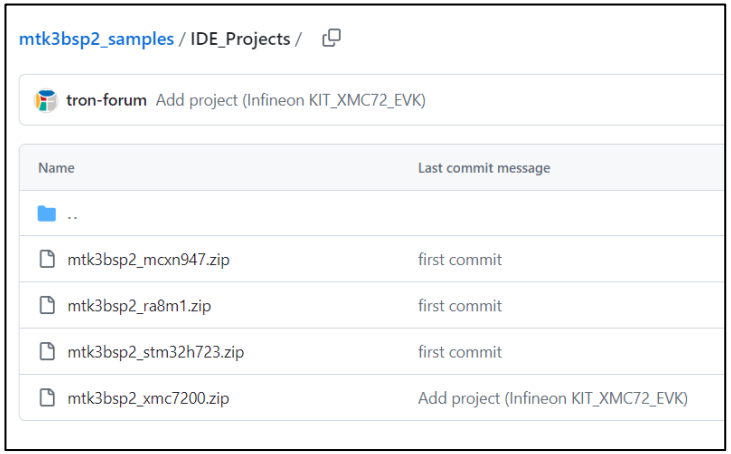
- TRONプログラミングコンテストでは、sysdependディレクトリ内のファイルは変更可能です。

BSP2の
サンプル・プロジェクト

作成済みのサンプル・プロジェクトを配布

- TRONプログラミングコンテスト用のマイコンボードは作成済みのIDEプロジェクトをGitHubから配布しています
  - https://github.com/tron-forum/mtk3bsp2_samples





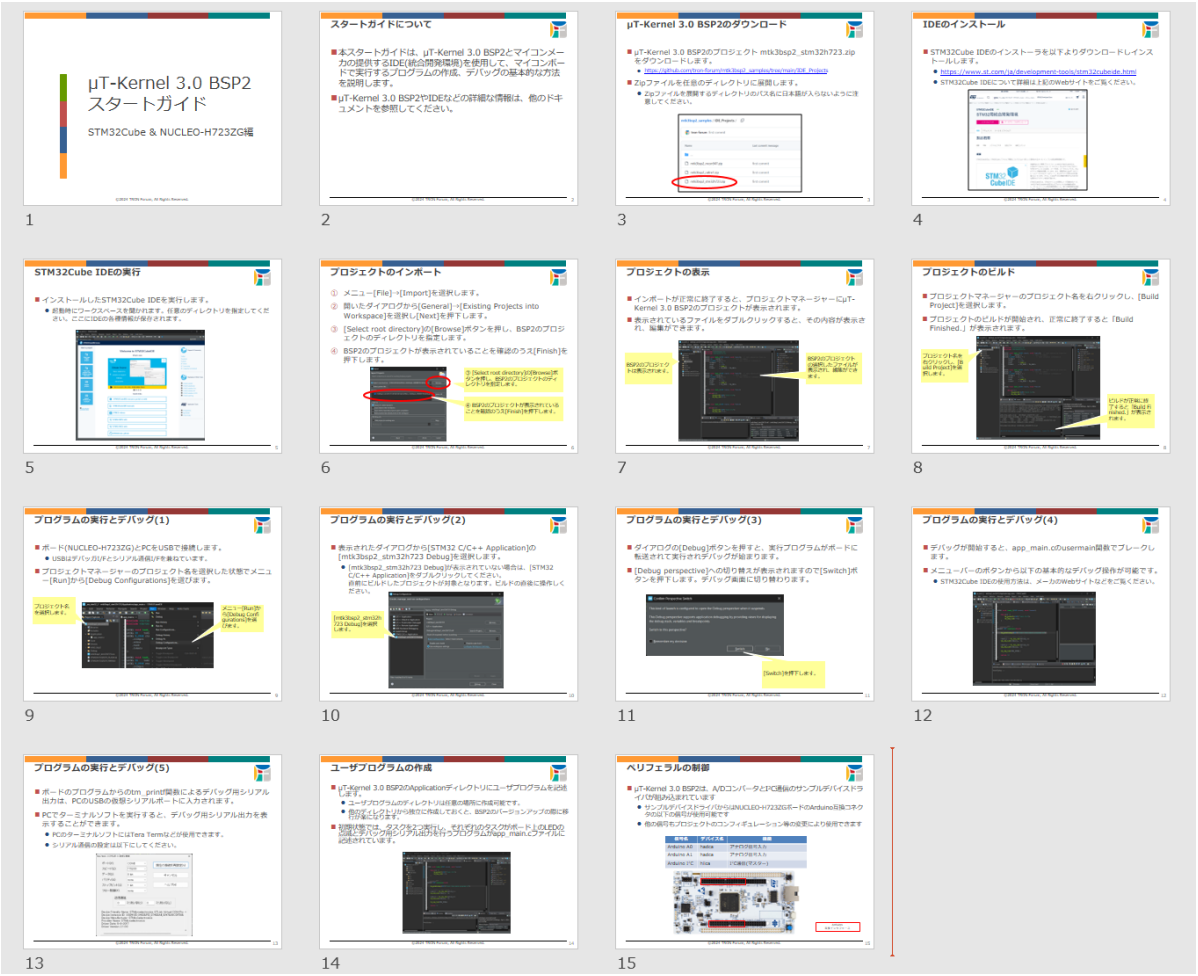
サンプル・プロジェクトの構成

- サンプル・プロジェクトは以下が組込み済みです

- µT-Kernel 3.0 BSP2
- サンプルのアプリケーション
  - ボード上のLEDをタスクから点滅
- A/Dコンバータおよび I2Cのサンプル・デバイスドライバ
  - Arduino互換I/Fが使用可能に設定済み
  - 使用可能な端子はStart Guideをご覧ください
  - 設定等を変更すれば他の信号でも使用可能です

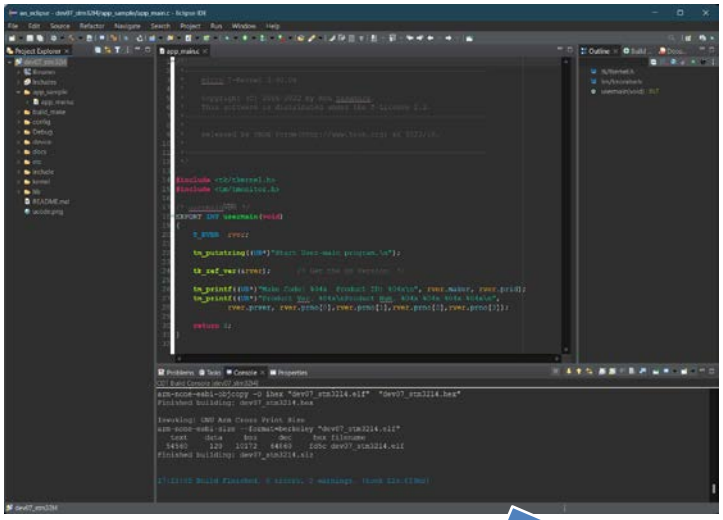
サンプル・プロジェクトのStart Guide

- サンプル・プロジェクトについてはStart Guideをご覧ください

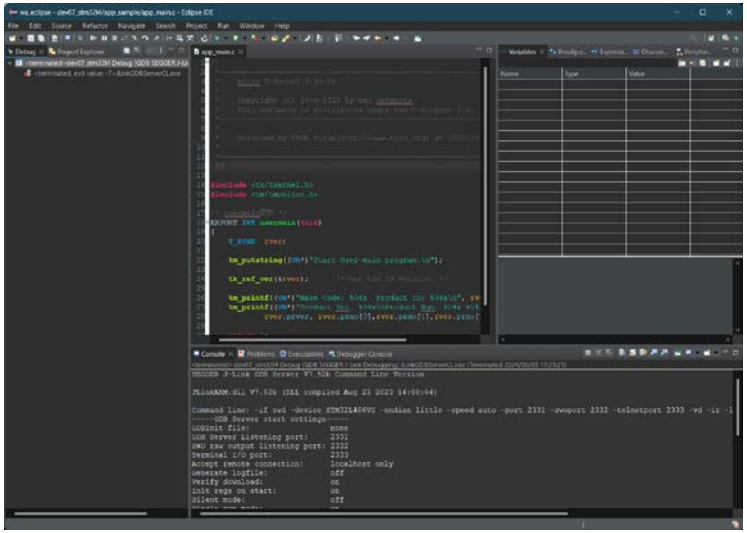


Eclipse系IDEのポイント(1)

- 複数の画面（パースペクティブ）を行き来します。
  - Toolbarのボタンで移動できます



C/C++

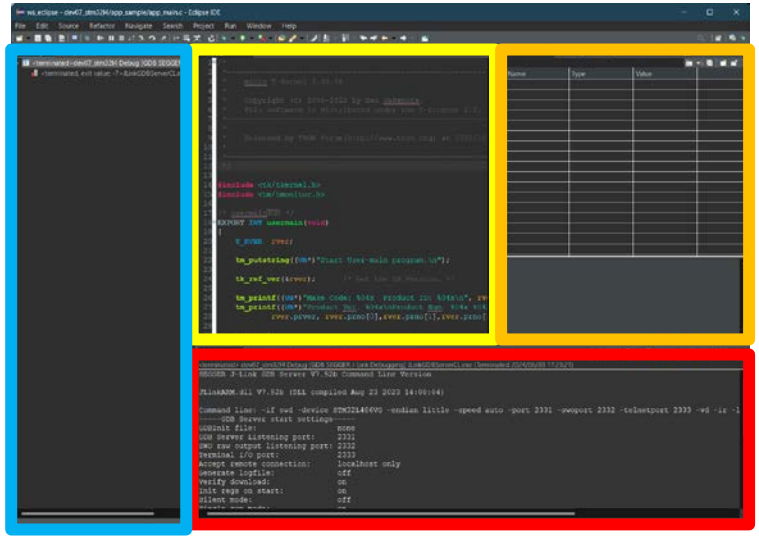


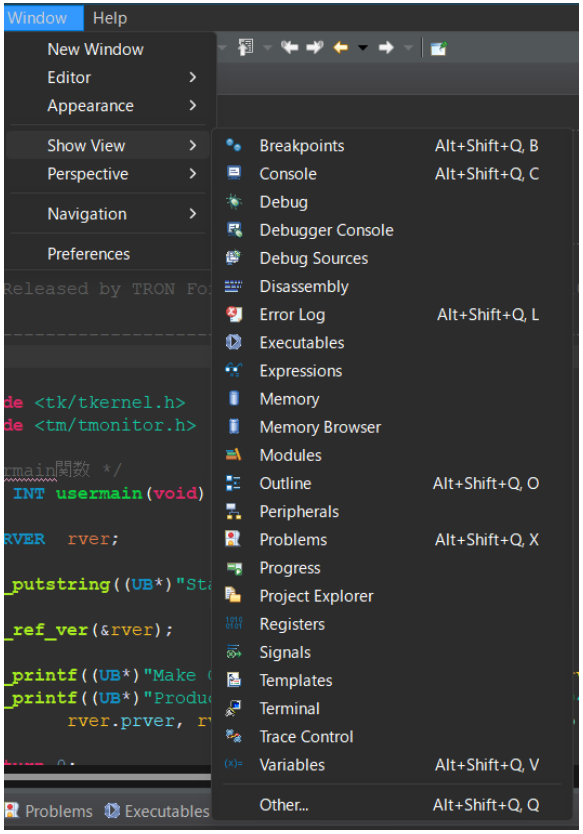
Debug

メーカ独自機能
（主にコンフィギュレーション関係）

Eclipse系IDEのポイント(2)

- パースペクティブ上のビューはカスタマイズできます
  - 画面分割、タブ化、非表示が自由にできます
- ビューはメニュー [Windows]>[Show View]で選択できます





Eclipse系IDEのポイント(3)

- Eclipse は外部のソースコード・エディタが使用できます
  - 特に設定は不要
  - 外部のエディタでソースコードを変更するとEclipse に反映されます
    - ▶外部のエディタでソースコードを編集し、Eclipseでプロジェクトを選択して、ビルドするだけ
    - ▶外部エディタでソースコードのセーブを忘れずに
    - ▶プロジェクトを選択せずにビルドすると反映されないときがあるので注意

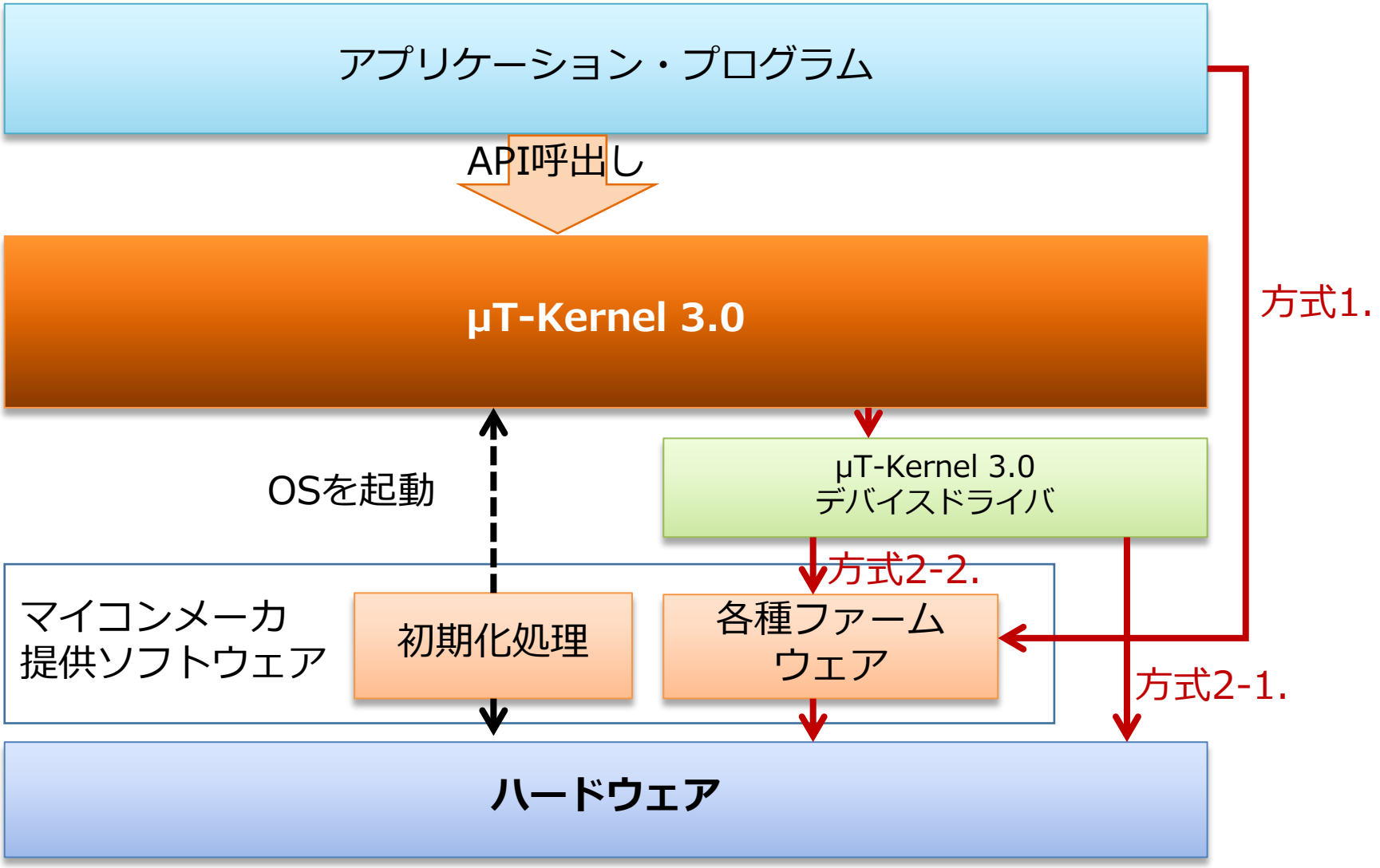
ペリフェラルの使用方法

ペリフェラルの使用方法

- 以下の3種類の方法があります

- 1. マイコンメーカ提供のファームウェア(HALなど)を直接制御する
- 2. µT-Kernel 3.0のデバイスドライバを作成する
  - 2-1. デバイスドライバからハードウェアを制御する
  - 2-2.デバイスドライバからメーカのファームウェアを使用してハードウェアを制御する **推奨**

µT-Kernel 3.0 BSPのシステム構成



ペリフェラルを操作する各方式の違い

方式	メリット	デメリット
1. メーカ提供のファームウェアを直接制御	• ハードウェアのほぼ全ての機能を利用可能 • コンフィギュレータで設定の変更が可能	• 操作方法やAPIの標準化はあまりなされていない • RTOSのマルチタスクでの使用には注意が必要
2-1. µT-Kernel 3.0デバイスドライバからハードウェアを制御	• 操作方法やAPIの標準化されている • RTOSから使用するにあたり制約のない効率の良い実装が可能	• 基本的に個々に開発する必要がある
2-2. µT-Kernel 3.0デバイスドライバからメーカのファームウェアを使用してハードウェアを制御	• 操作方法やAPIの標準化されている • コンフィギュレータで設定の変更が可能 • RTOSから使用するにあたり制約のない実装が可能	• 基本的に個々に開発する必要がある（ただし、2-1に比べれば容易）

メーカ提供のファームウェアを使用する場合

- アプリケーションから、メーカ提供のファームウェアを使用することができます
  - 使用方法については各メーカからの情報を参照ください
- リアルタイムOS上で実行するために以下の注意が必要です
  - 複数のタスクから同時に使用する場合は、何らかの排他制御が必要です
  - ポーリング（ビジーループ）は他のタスクの動作を阻害しますので、割り込みの使用を推奨します
  - 割り込みからタスクへのイベントの通知はOSのAPIの使用を推奨します
    ▶グローバル変数のポーリングを行うと他のタスクの動作を阻害します

µT-Kernel 3.0 BSP2
サンプル・デバイスドライバ

- µT-Kernel 3.0 BSP2には2-2方式のサンプル・デバイスドライバが付属しています（A/DC、I^2C通信）
- サンプル・デバイスドライバはHALを使用しています
  - ペリフェラルの基本機能のみをサポートしています
  - デバイスドライバを開発するためのサンプルの位置づけです

A/Dコンバータの制御プログラム

- センサーからのアナログ入力を1秒毎に読み取り、UART(デバッグ)出力に表示するタスク・プログラム

```
LOCAL void task_1(INT stacd, void *exinf)
{
  UW        adc_val;
            ID          dd;         // デバイスディスクリプタ
            ER          err;

            dd = tk_opn_dev((UB*)"hadca", TD_UPDATE);       // デバイスのオープン
            if(dd <= E_OK) tm_printf((UB*)"ERR-0  %d¥n", dd);

            while(1) {
                        err = tk_srea_dev(dd, 0, &adc_val, 1, NULL);    // A/DC チャンネル0からデータを取得
                        if(err == E_OK) {
                                    tm_printf((UB*)"A/DC A0 =%06d¥n", adc_val);     // デバッグ出力
                        } else {
                                    tm_printf((UB*)"ERR-ADC¥n");
                        }
                        tk_dly_tsk(1000);       // 1000ms待ち
            }
}
```

I²C通信の制御プログラム

- センサーの値をI²C通信を使って1秒毎に読み取り、UART（デバッグ）出力に表示するタスク・プログラム
  - 温度・湿度センサー(SHT35)の例

```
LOCAL void task_2(INT stacd, void *exinf)
{
            UW          tmp0, hum0;
            UB          cmd[] = {0x2C, 0x06};
            UB          rcv[6];
            ID          dd;         // デバイスディスクリプタ
            ER          err;

            dd = tk_opn_dev((UB*)"htiica", TD_UPDATE);      // デバイスのオープン
            if(dd <= E_OK) tm_printf((UB*)"ERR-0  %d¥n", dd);

            while(1) {
                        err = tk_swri_dev(dd, 0x45, cmd, 2, NULL);      // コマンドの送信
                        if(err != E_OK) tm_printf((UB*)"ERR-1¥n");
                        tk_dly_tsk(1);
                        err = tk_srea_dev(dd, 0x45, rcv, 6, NULL);      // データの受信
                        if(err != E_OK) tm_printf((UB*)"ERR-2¥n");

                        tmp0 = rcv[0]<<8 | rcv[1]; hum0 = rcv[3]<<8 | rcv[4];
                        tmp0 = (tmp0*175)/65535 - 45; hum0 = (hum0*100)/65535;
                        tm_printf((UB*)"温度 %d ℃  湿度 %d %¥n¥n", tmp0, hum0);    // デバッグ出力

                        tk_dly_tsk(1000);       // 1000ms待ち
            }
}
```

µT-Kernel 3.0 情報

トロンフォーラムセミナー

トロンフォーラムではμT-Kernel 3.0の各種セミナーを実施しています

- **【実習】μT-Kernel 3.0 組込みプログラミング入門**
  - 市販のマイコンボードを使用した１日のハンズオン実習
  - 使用したマイコンボードなど教材を持ち帰れます。
  - 予定
    - ▶6月28日 RXマイコン編 (Target Board for RX65N)
      NIAD(東洋大学情報連携学部) 東洋大学赤羽台キャンパス
    - ▶以降、各種マイコンについて実施していきます

- その他、中級プログラミング演習、ネットワーク演習などを実施
- 今後の予定はトロンフォーラムのWebサイトに掲載します
  - https://www.tron.org/ja/seminar/schedule2024/

INIAD (東洋大学情報連携学部)
Open IoT 教育プログラム

- 高度なIoT技術を身に付けたい社会人の方を対象
- トロンフォーラムと連携し、リアルタイムOS μT-Kernel 3.0を用いたシステム開発を中心に、Webアプリケーション開発やAI等の IoTと関連が深い分野の最新技術について学習します。
- **2024年度　'24/9月～'25/1月開催予定**
  - **受講生募集は2024年6月中旬開始を予定**
  - **https://enpit.iniad.org/**

µT-Kernel 3.0 解説書

- リアルタイムOSの基礎から
  デバイスドライバの開発、
  OSの移植まで説明
- µT-Kernel 3.0 BSPを使用し、市販のマイコンボードでプログラムを作成し実行
- 実習セミナーのテキストにも使用

トロンフォーラムWeb

- トロンフォーラムのWebからも技術情報を提供しています

- リアルタイムOS入門「μT-Kernel 3.0で学ぶリアルタイムOSプログラミング」
  - https://www.tron.org/ja/page-722/rtos01/
  - リアルタイムOSの基本と、リアルタイムOSを使用したプログラミングについて説明します

- 「新しいBSP（ボードサポートパッケージ）を使いこなす」
  - https://www.tron.org/ja/programming_contest/bsp/
  - μT-Kernel 3.0 BSP2の紹介記事（TRONWARE誌掲載）

µT-Kernel 3.0 仕様書

- µT-Kernel 3.0仕様書はWebから閲覧可能です。
  - https://tron-forum.github.io/mtk3_spec_jp/index.html

µT-Kenrel 技術情報サイト

- µT-Kernel 技術情報サイトではµT-Kernel 3.0に関する各種情報、関連リンクなどを掲載しています
  - https://www.tron.org/mt-kernel3/

ご清聴ありがとうございました

- TRONプログラミングコンテスト開催中です
  - ボードをご自身で用意いただければ参加できます
  - 応募締め切り 9月30日　まだ間に合います
  - 詳しくはWebを
    ▶https://www.tron.org/ja/programming_contest/

- ご質問があればお答えします