



TOPPERS BASE PLATFORM で組み込みソフトを学ぶ準備をしよう 14:00～14:45



2020年6月27日

TOPPERSプロジェクト
開発者会議実行委員会



TOPPERSプロジェクトとは？

TOPPERS = Toyohashi OPen Patform for
Embedded and Real-time Systems

- プロジェクトの活動内容
 - ITRON仕様の技術開発成果を出発点として、組込みシステム構築の基盤となる各種の高品質なオープンソースソフトウェアを開発するとともに、その利用技術を提供

**組込システム分野において、Linuxのように広く使われる
オープンソースOSの構築を目指す！**

- プロジェクトの推進主体
 - 産学官の団体と個人が参加する産学官民連携プロジェクト
 - 2003年9月にNPO法人として組織化



開発者実行委員会とは

- 「開発者会議」という、毎年秋ごろに開催されるTOPPERSの開発者や利用者が集まる会議の実行委員会
 - 近年は会員以外も参加できます
 - 温泉宿に宿泊して、開発者と直接議論できます
- ゲストトークでRTOS関連の状況など共有します
 - 2017年 「マルチコア通信について」
<https://www.toppers.jp/devconf2017.html>
 - 2018年 「AWS 及び Amazon FreeRTOSのご紹介」
<https://www.toppers.jp/devconf2018.html>
 - 2019年 「ソニーSpresense技術解説とハンズオン」
<https://www.toppers.jp/devconf2019.html>
- 開発者会議の実行委員会ですが、OSCへの出展など、TOPPERSの広報活動も行っています

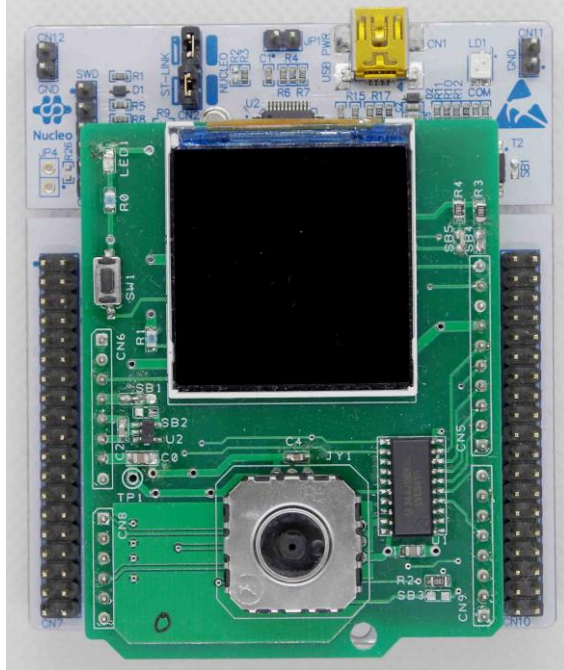
TOPPERS BASE PLATFORMとは



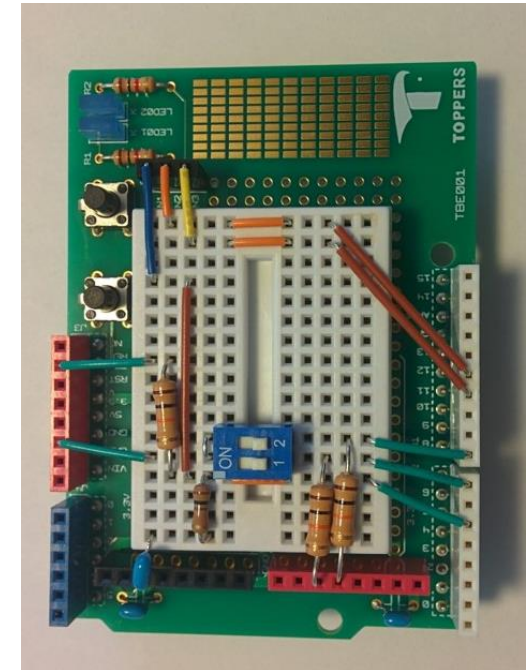
- TOPPERSのワーキンググループの一つ
教育WGで作成した教育コンテンツで使う
ソフトウェアプラットフォーム
- 教育コンテンツは組込み技術者養成用
 - 基礎 1、 2、 3 のコースがありそれぞれ 2 日間
 - 基礎1：ハードウェア、開発環境、デバドラの開発実習
 - 基礎2：RTOSの導入、RTOS運用、TOPPERS BASE
PLATFORM(ST)環境セットアップ
 - 基礎3：SPI,ADCデバイスドライバ、グラフィックLCDの表
示方法、SDカードを使ったFATファイルの理解と学習
 - とっても詳しいコンテンツです！



セミナー用基板



去年までのセミナー用
フラットーク社製TEB001



今年のセミナー用
IWHI-LCDシールド



対応ハードウェア

- コンテンツではSTマイクロの「NUCLEO-F401RE」または「NUCLEO-F446RE」を使用
- その他のSTM32やRISC-V、 Intel Cyclone V SoC向けのソフトウェアが公開されている
- 一部紹介

CPU	ボード
ARM Cortex-A9	Cyclone V SoC DE10-Nano
ARM Cortex-M7	STM32F7 Discovery
ARM Cortex-M4	STM32F401 Nucleo-64
ARM Cortex-M4/M0	STM32WB55 Nucleo
ARM Cortex-M0+	STM32G071 Nucleo-64
RISC-V K210	Maixシリーズ



3コース×2日間までは…。

興味はあるけど、セミナーの参加には抵抗を感じる。
資料を元に自分で試してみたい。

とりあえず、「TOPPERS BASE PLATFORM」を
手元で使えるようにしたい方向けに
ビルド環境を作るところを実演します。



ターゲット・開発環境の情報

- ターゲットボード
 - NUCLEO-F401RE
 - CPU : Cortex-M4
- ビルド環境
 - コンパイラ : GCC (arm-none-eabi-gcc)
 - ビルド : make (msys2)
 - IDE : Visual Studio Code + C++拡張機能
- デバッグ
 - 接続 : ST-Link
 - ソフト : GDB + OpenOCD
 - IDE : Visual Studio Code + Cortex Debug



- BASE PLATFORMに付属する資料

BaseTrainingSeminar_environment-020010004.pdf

- TOPPERS/ASP3 + GR-PEACH向けの資料

- GR-PEACHはarm Cortex-Aだが、使うものは同じ
- Windows編、MAC編、Ubuntu編がある！

<https://qiita.com/takasehideki/items/59e3d179d462142a8633>

<https://qiita.com/takasehideki/items/f897e6e063ceb05a9f35>

<https://qiita.com/takasehideki/items/6119d50dbd8e0a34d64d>

- サンプル1 動作までの資料

https://qiita.com/Yukiya_Ishioka/items/9ecbe080939600c323c6

ダウンロード



● TOPPERSプロジェクトのホームページ

- + 会員向けページ
- + プロジェクトについて
- + 取り組み
- 開発成果物
 - + リアルタイムカーネル
 - + AUTOSAR関連
 - TOPPERS/EV3RT(EV3 Platform)
 - TECS(コンポーネントシステム)
 - TINET(TCP/IPスタック)
 - TOPPERS/ECNL
 - CAN/LIN通信ミドルウェア
 - FatFs for TOPPERS
 - SafeG
 - SafeG64
 - MDCOM
 - TOPPERS BASE PLATFORM
 - TOPPERS Test Suite Package(TOPPERS新世代カーネル)
 - TLV(トレースログ可視化ツール)
 - TOPPERS Builder Bootable CD-ROM イメージ
 - TOPPERSカーネル向けシミュレーション環境
 - TOPPERS新世代カーネル用コンフィギュレータ
 - + その他
- + 教育コンテンツ
- + ドキュメント
- + ダウンロード

The screenshot shows the TOPPERS Project Home Page. The browser address bar displays 'https://www.toppers.jp/index.html'. The page features a navigation menu with links: Topics, About Project, ASP3 Kernel, Documents, Download, Community, Report, Contacts, and FAQ. A sidebar on the left lists various project components, with '開発成果物' (Development Results) highlighted. The main content area includes a large banner for 'TOPPERSカンファレンス 2020' (TOPPERS Conference 2020) held on June 12th from 10:00 to 18:00. Below the banner, there are sections for 'TOPPERS教育セミナーで使用中!' (Using TOPPERS Education Seminar!), '会員の皆様へ' (To our members), and 'TOPPERSの最新情報が満載!' (Full of the latest TOPPERS information!). The footer mentions the 'TOPPERS Project Newsletter' and provides a link to the newsletter's back number.

ダウンロード



- ファイルへのリンクは下の方にあります

TOPPERS BASE PLATFORMとは x +

← → ↺ 家 🔒 https://www.toppers.jp/edu-baseplatform.html ☆ ⚙️ ☆ 🗂️ 👤 ...

開発構築の説明として教育教材中の「TOPPERS実装ゼミナー(STM32F4-Disc/Nucleo:DE10-Nano版:基本)開発環境編」を参照してください。
インストール手順等は、「TOPPERS BASE PLATFORM対応版 基礎2ゼミナーコンテンツ」の1日目の4章 TOPPERS/ASPの導入以降、または、「基礎3ゼミナーコンテンツ」の1日目の2章 ASPプラットフォーム環境構築以降を参照してください。

対応教育教材

TOPPERS BASE PLATFORM(ST)を使用した組込み実装教材は以下の通りです。

1. 基礎1、基礎2、基礎3

開発成果物のダウンロード

TOPPERS BASE PLATFORM(ST) : 担当TOPPERS教育WG		
パッケージ	マニュアル	リリース日
asp_baseplatformv1.4.1_051120.tar.gz	V1.4.1 Reference_manual(ST) V1.4.1 Reference_manual(RV)	2020-05-11
asp_baseplatformv1.4.0_051120.tar.gz	V1.4.0 Reference_manual(ST)	2019-05-06
asp_baseplatformv1.3.0_052018.tar.gz	V1.3.0 Reference_manual(ST)	2018-06-04

TOPPERS BASE PLATFORM(CV) : 担当TOPPERS教育WG		
パッケージ	マニュアル	リリース日
fmp_1.4.0_baseplatformv1.1.1_041820.tar.gz	V1.1.1 Reference_manual(CV)	2020-04-18
fmp_1.4.0_baseplatformv1.1.0_030419.tar.gz	V1.1.0 Reference_manual(CV)	2019-03-04
fmp_1.4.0_baseplatformv1.0.0_052218.tar.gz	V1.0.0 Reference_manual(CV)	2018-06-04

ROMモニタ : 担当TOPPERS教育WG

下の方に
スクロール

2. 教材は「T…ツ」
をクリック

1. 最新版はここ
(Ver.1.4.1)



教材のダウンロード

● セミナーで使用する教材のスライドをダウンロード

TOPPERSプロジェクト／教育コンテンツ

https://www.toppers.jp/edu-download.html#edu-basebp-download

Google カスタム検索 検索

言語を選択 Powered by Google 翻訳

Topics | About Project | ASP3 Kernel | Documents | Download | Community | Report | Contacts | FAQ

会員向けページ
プロジェクトについて
取り組み
開発成果物
教育コンテンツ
ドキュメント
ダウンロード
関連製品
利用事例
コミュニティ情報
その他の情報

TOPPERS教育セミナーで使用中！
Arduino互換 拡張ボード **TEB001**
詳しくはこちら(基礎セミナー紹介ページ)

会員の皆様へ
関連製品情報掲載のお誘い

TOPPERSの最新情報が満載！
TOPPERS Project Newsletter
最新号は[こちら](#)をクリック！
Newsletterのバックナンバーは[こちら](#)

Toyohashi Open Platform for Embedded Real-time Systems

TOPPERS BASE PLATFORM対応基礎1、2、3セミナー教材のダウンロード

基礎1、2、3セミナーの教材を公開しています。コンテンツの内容は、基礎1、2、3セミナーでの使用後、問題点等の修正を行ったものです。教材には以下のものが含まれます。

- セミナーのテキスト
- セミナーに必要なプログラムとサンプル回答プログラム

開発環境の構築方法は、コンテンツ内の「TOPPERS基礎実装セミナー：開発環境編」に記載しております。

リリース名	バージョン	タイプ	サイズ	リリース日
TOPPERS BASE PLATFORM対応版 基礎1、2、3実装セミナーの教材	3.4	zip	16MB	2018-11-10
TOPPERS BASE PLATFORM対応版 基礎1、2、3実装セミナーの教材	3.3	zip	9.1MB	2018-07-19

TOPPERS BASE PLATFORM対応版基礎1、2セミナー教材のダウンロード

基礎1、2セミナーの教材を公開しています。コンテンツの内容は、基礎1、2セミナーでの使用後、問題点等の修正を行ったものです。教材には以下のものが含まれます。

- セミナーのテキスト
- セミナーに必要なプログラムとサンプル回答プログラム

開発環境の構築方法は、コンテンツ内の「TOPPERS基礎実装セミナー：開発環境編」に記載しております。

この教材を使用したセミナーの詳細は[こちら](#)です。ぜひご覧ください。

最新リリース

教材をダウンロード

上の方に
スクロール

実はRTOS本体も必要です



● TOPPERS/ASPをダウンロード

TOPPERSプロジェクト/ASPカーネル x +

← → ↺ 🏠 🔒 https://www.toppers.jp/asp-kernel.html ☆ ⚙️ ☆ 🗂️ 👤 ...

Google カスタム検索 🔍 検索 x

言語を選択 ▼ Powered by Google 翻訳

Topics | About Project | ASP3 Kernel | Documents | Download | Community | Report | Contacts | FAQ

+ 会員向けページ
+ プロジェクトについて
+ 取り組み
- 開発成果物
 - リアルタイムカーネル
 - TOPPERS/ASP3カーネル
 - TOPPERS/HRP3カーネル
 - TOPPERS/FMP3カーネル
 - TOPPERS/HRMP3カーネル
 - **TOPPERS/ASPカーネル**
 - TOPPERS/JSPカーネル
 - TOPPERS/FMPカーネル
 - TOPPERS/HRP2カーネル
 - TOPPERS/SSPカーネル
 - TOPPERS/PARK
 - TOPPERS/FDMPカーネル
 - TOPPERS/FI4カーネル
 - TOPPERS/HRPカーネル
 - TOPPERS/ATK1
 + AUTOSAR関連
 - TOPPERS/EV3RT(EV3 Platform)
 - TECS(コンポーネントシステム)
 - TINET(TCP/IPスタック)
 - TOPPERS/ECNL
 - CAN/LIN通信ミドルウェア
 - FatFs for TOPPERS
 - SafeG
 - SafeG64

TOPPERS/ASPカーネルとは

TOPPERS/ASPカーネル（以下、ASPカーネル）は、TOPPERS新世代カーネルの基盤（出発点）となるリアルタイムカーネルで、[TOPPERS新世代カーネル統合仕様](#)に準拠しています。ASP（Advanced Standard Profile）の名前が示す通り、[μITRON4.0仕様](#)のスタンダードプロファイル準拠のリアルタイムカーネルであるTOPPERS/JSPカーネルを拡張・改良する形で開発しました。

TOPPERS/ASPカーネルのダウンロード方法

ASPカーネルは、ターゲットシステムごとに必要なソースコードを一つにまとめた簡易パッケージと、簡易パッケージの内容を分割した個別パッケージの2つの形態で配布しています。簡易パッケージは[こちら](#)から、個別パッケージは[こちら](#)から、それぞれダウンロードできます。

また、ASPカーネルの仕様を記述したTOPPERS新世代カーネル統合仕様書は[こちら](#)から、ASPカーネルを使用する際に必要となるコンフィギュレータのソースコードは[こちら](#)から、それぞれダウンロードできます。

TOPPERS/ASPカーネルの英語/中国語パッケージ

ASP 1.7.0のコードとドキュメントを英語/中国語化したパッケージです。公募事業の開発成果に、修正を加えたものです。

- ソースコードのコメント：英語化
- docディレクトリ以下のドキュメント：中文化

[こちら](#)からダウンロードできます。

TOPPERS/ASPカーネルの適用対象領域と仕様設計方針

ASPカーネルは、20年以上に渡るITRON仕様の技術開発成果をベースとして、完成度の高いリアルタイムカーネルを実現したものです。完成度を高めるという観点から、カーネル本体の仕様については、枯れた技術で実装できる範囲に留めています。保護機能を持ったカーネルやマルチ

今回は
個別パッケージを
使います
「ここ」をクリック



個別パッケージ

- ターゲット非依存部をダウンロード

TOPPERSプロジェクト/ASPカーネル

https://www.toppers.jp/asp-d-download.html

Google カスタム検索

言語を選択

Powered by Google 翻訳

Topics | About Project | ASP3 Kernel | Documents | Download | Community | Report | Contacts | FAQ

TOPPERS教育セミナーで使用中!

Arduino互換
拡張ボード **TEB001**

詳しくはこちら(基礎セミナー紹介ページ)

会員の皆様へ

関連製品情報掲載のお誘い

TOPPERSの最新情報が満載!

TOPPERS Project Newsletter

最新号は[こちら](#)をクリック!

Newsletterのバックナンバーは[こちら](#)

Toyohashi Open Platform
for Embedded Real-time Systems

TOPPERS/ASPカーネル個別パッケージのダウンロード

TOPPERS/ASPカーネルの最新リリースの個別パッケージを配布しています。各ターゲットに対応するカーネル非依存部パッケージ、依存部パッケージ（アーキテクチャ依存部、ターゲット依存部）、シリアルドライバ依存部（PDIC）パッケージを個別にダウンロードできます。

ターゲットごとに必要なソースコードを一つにまとめた簡易パッケージは、[こちら](#)からダウンロードできます。

一般公開以前のバージョン（Release 1.3.0以前）に対応した個別パッケージは、会員向け早期リリースとしての扱いますが、[こちら](#)からダウンロードできます。

ターゲット非依存部

TOPPERS/ASPカーネル ターゲット非依存部パッケージ（担当：名古屋大学）	
バージョン	ダウンロード
asp-1.9.3.tar.gz	2017-04-29 release 1.9.2との違い
asp-1.9.2.tar.gz	2015-05-30 release 1.9.1との違い
asp-1.9.1.tar.gz	2014-11-17 Release 1.9.0との違い
asp-1.9.0.tar.gz	2014-01-06 Release 1.8.0との違い
asp-1.8.0.tar.gz	2012-12-26 Release 1.7.0との違い
asp-1.7.0.tar.gz	2011-05-09

TOPPERS/ASP
Ver.1.9.3
を使います

次に
下の方に
スクロール



個別パッケージ

- ターゲット依存部をダウンロード

TOPPERSプロジェクト / ASPカーネル x +

← → ↺ 🏠 📁 https://www.toppers.jp/asp-d-download.html ☆ ⚙️ ☆ 🗑️ 👤 ...

asp_arch_arm_m4_gcc-1.9.2.tar.gz	target/stm32f746nucleo144_gcc tools/rommon	1.9.*	2016-07-23
ARM Cortex-M4アーキテクチャ・GCC依存部パッケージ (担当: TOPPERSプロジェクト教育WG)			
パッケージ	ディレクトリ	対応する非依存部のバージョン	リリース日
asp_arch_arm_m4_gcc-1.9.6.tar.gz	arch/arm_m_gcc target/stm32e407_gcc target/stm32f4discovery_gcc target/stm32f401nucleo_gcc target/stm32f429board_gcc target/stm32f446nucleo64_gcc target/stm32f446nucleo144_gcc target/stm324r5nucleo144_gcc target/stm32l476nucleo64_gcc target/stm32l476discovery_gcc target/stm32g474nucleo64_gcc target/stm32g431nucleo64_gcc target/stm32wb55nucleo_gcc tools/rommon tools/TrueSTUDIO	1.9.*	2020-03-10
asp_arch_arm_m4_gcc-1.9.5.tar.gz	arch/arm_m_gcc target/stm32e407_gcc target/stm32f4discovery_gcc target/stm32f401nucleo_gcc target/stm32f429board_gcc target/stm32f446nucleo64_gcc target/stm32f446nucleo144_gcc target/stm324r5nucleo144_gcc target/stm32l476nucleo64_gcc target/stm32l476discovery_gcc target/stm32g474nucleo64_gcc target/stm32g431nucleo64_gcc target/stm32wb55nucleo_gcc tools/rommon tools/TrueSTUDIO	1.9.*	2019-02-04

ARM Cortex-M4アーキテクチャ・GCC
を使います



コンフィギュレータ

- コンパイル前の処理を行うコマンドをダウンロード

The screenshot shows the TOPPERS website with a sidebar menu on the left. The menu items include: 会員向けページ, プロジェクトについて, 取り組み, 開発成果物, リアルタイムカーネル, AUTOSAR関連, TOPPERS/EV3RT(EV3 Platform), TECS(コンポーネントシステム), TINET(TCP/IPスタック), TOPPERS/ECNL, CAN/LIN通信ミドルウェア, FatFs for TOPPERS, SafeG, SafeG64, MDCOM, TOPPERS BASE PLATFORM, TOPPERS Test Suite, Package(TOPPERS新世代カーネル), TLV(トレースログ可視化ツール), TOPPERS Builder Bootable CD-ROM イメージ, TOPPERSカーネル向けシミュレーション環境, TOPPERS新世代カーネル用コンフィギュレータ (highlighted in blue), その他, 教育コンテンツ, ドキュメント, and ダウンロード. The main content area is titled 'TOPPERS新世代カーネル用コンフィギュレータとは' and contains text about the tool. A blue box highlights the link 'こちら' in the text 'コンフィギュレータの最新リリースは[こちら](#)からダウンロードできます。'. A blue arrow points from a callout box to this link.

「ここ」をクリックします

名前	所属	時期
高木信尚	個人特別会員	2006年～



コンフィギュレータ

- Windows用バイナリをダウンロードします

TOPPERSプロジェクト / ダウンロード

https://www.toppers.jp/cfg-download.html

Google カスタム検索

TOPPERS
Toyohashi Open Platform
for Embedded Real-time Systems

Topics | About Project | ASP3 Kernel | Documents | Download | Community | Report | Contacts | FAQ

TOPPERS新世代カーネル用コンフィギュレータ

TOPPERS新世代カーネル用コンフィギュレータの最新リリースを配布しています。コンフィギュレータ自体については[こちら](#)を、構築方法等についてはアーカイブに含まれる README.txt をご覧ください。

下記32bit Linux用バイナリは、次の環境でライブラリを静的リンクしてビルドしたものです。

- Ubuntu 12.04 LTS 32bit
- boost 1.46.1
- Xerces C++ 3.1.1

64bitのLinux用のboostライブラリに現状問題があるため、64bitのLinuxでビルドしたcfgは正しく動作しません。64bitのLinuxでcfgを使用する場合は、下記の32bit Linux用バイナリを使用するか、32bitのLinuxでソースからビルドしてスタティックリンクしたバイナリを使用下さい。

リリース名	タイプ	サイズ	リリース日
コンフィギュレータ Release 1.9.6	tar.gz(EUC,LF)	150KB	2017-03-31
コンフィギュレータ Release 1.9.6 (Windows用バイナリ)	zip	10.5MB	2017-03-31
コンフィギュレータ Release 1.9.5 (Windows用バイナリ)	gz	7.0MB	2017-03-31

過去のリリース

Windows用バイナリ



ファイルを展開し統合

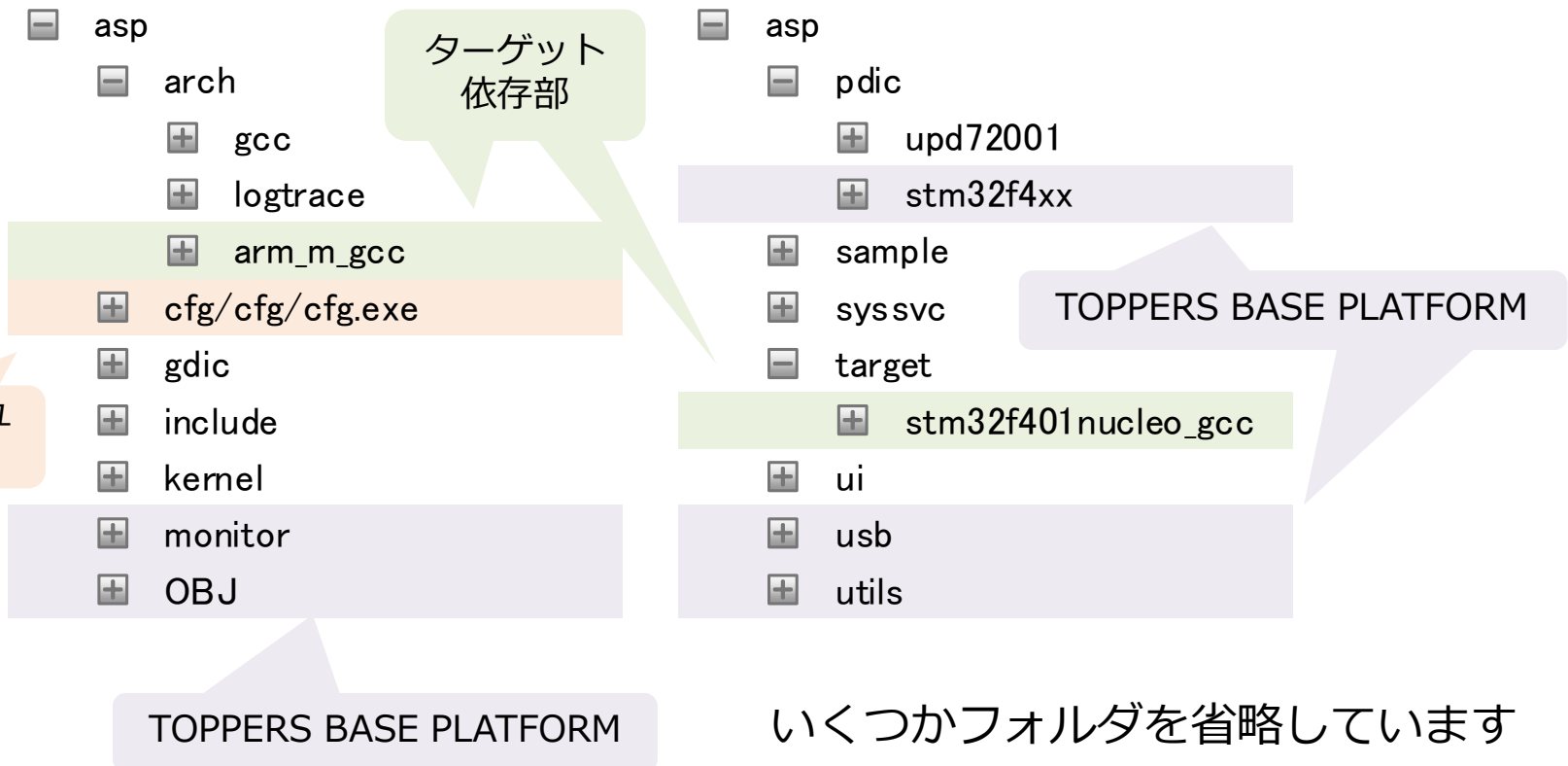
- Windowsでの「*.tar.gz」の解凍
 - 7zipを使うとGUIで解凍できます
 - ただし2段階「*.tar.gz」→「*.tar」→フォルダ
 - 別のフォルダに解凍した方が吉
- ダウンロードしたファイルと展開後

種類	圧縮ファイル名	展開後
ターゲット非依存部	asp-1.9.3	asp
ターゲット依存部	asp_arch_arm_m4_gcc-1.9.6	asp
コンフィギュレータ	cfg-mingw-static-1_9_6	cfg.exe
BASE PLATFORM	asp_baseplatformv1.4.1_051120	asp
セミナー教材	base123-111018	base123-111018



統合後のフォルダ構成

- ダウンロードした4つを展開しフォルダを統合する
 - コンフィギュレータは「cfg¥cfg¥cfg.exe」





開発環境のインストール

- Windowsで作業します
 - WindowsのPOSIX環境としてmsys2を使います
(WSLでもできますが、ひと手間あるのと、
BASE PLATFORMの資料にあわせています)
 - msys2のコマンドラインの代わりに
Visual Studio Codeを使います
(BASE PLATFORMの説明にはないです)
 - makeはmsys2環境のコマンドを使います
- gccとOpenOCDはmsys2とは別にインストール
- ST-Linkのドライバをインストール
- TeraTermをインストール



ツールのダウンロードURL

- 下記のURLからダウンロードしてインストール
 - msys2はパッケージマネージャ「pacman」で更新とインストール

アプリ	サイト
VSCode	https://code.visualstudio.com
msys2	https://www.msys2.org \$ pacman -Syu \$ pacman -Su
make	\$ pacman -S make
arm-none-eabi-gcc (6-2017-q2-update)	https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads/6-2017-q2-update
OpenOCD (v0.10.0-12 20190422)	https://github.com/gnu-mcu-eclipse/openocd/releases/tag/v0.10.0-12-20190422
ST-Link	https://www.st.com/ja/development-tools/stsw-link009.html
Tera Term	https://ja.osdn.net/projects/ttssh2/

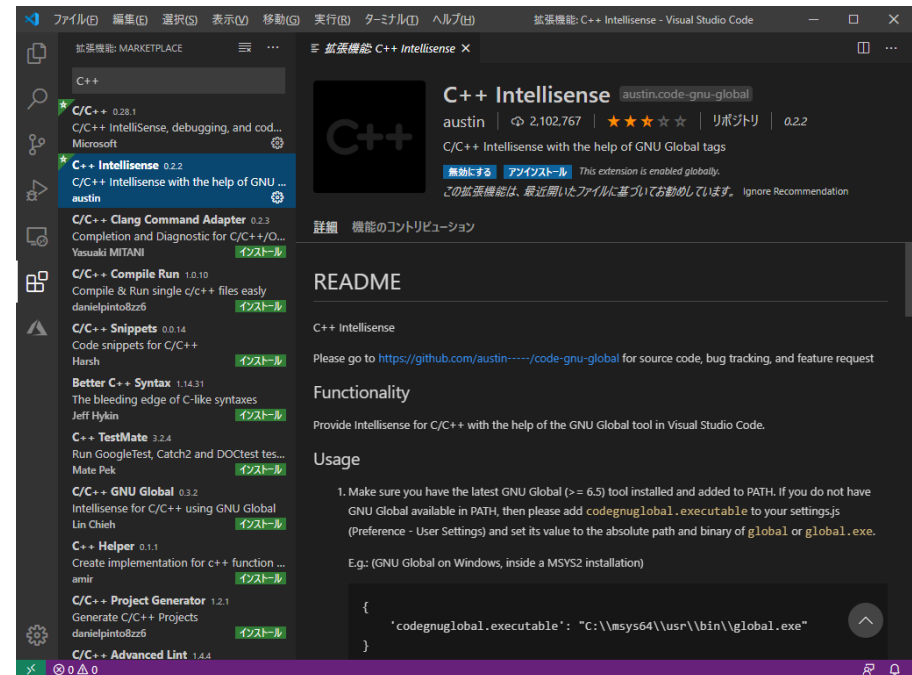
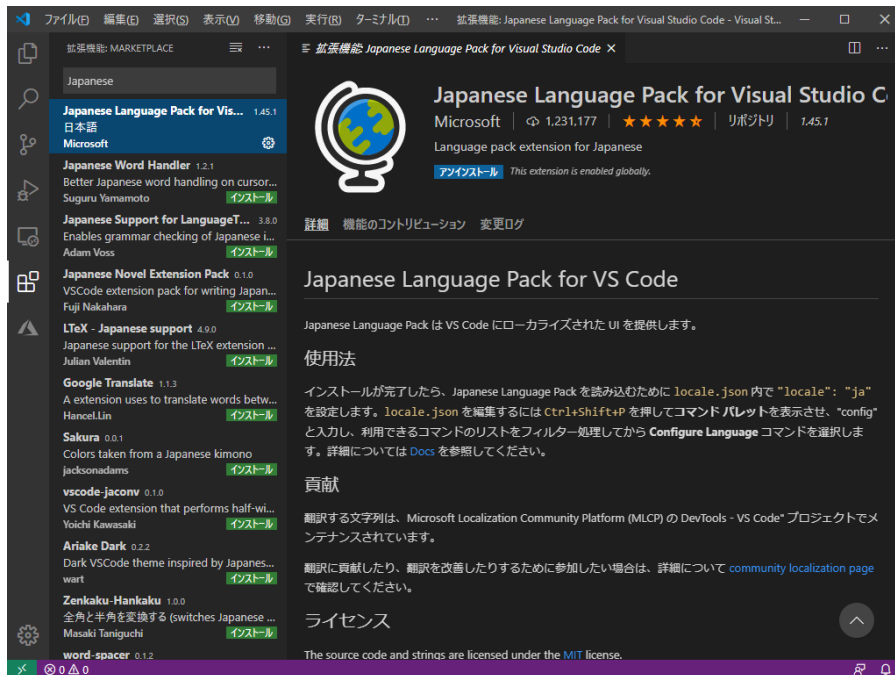


-



VSCodeの拡張機能

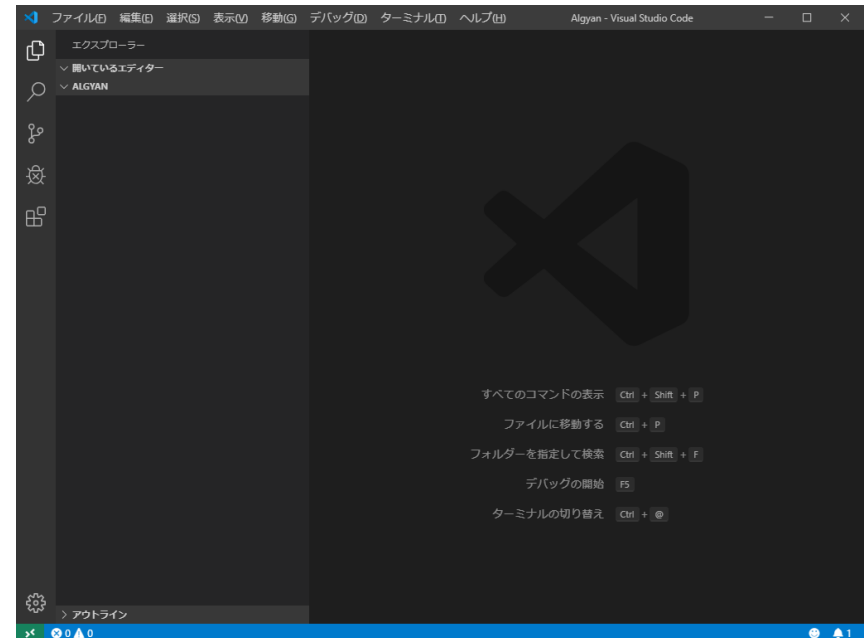
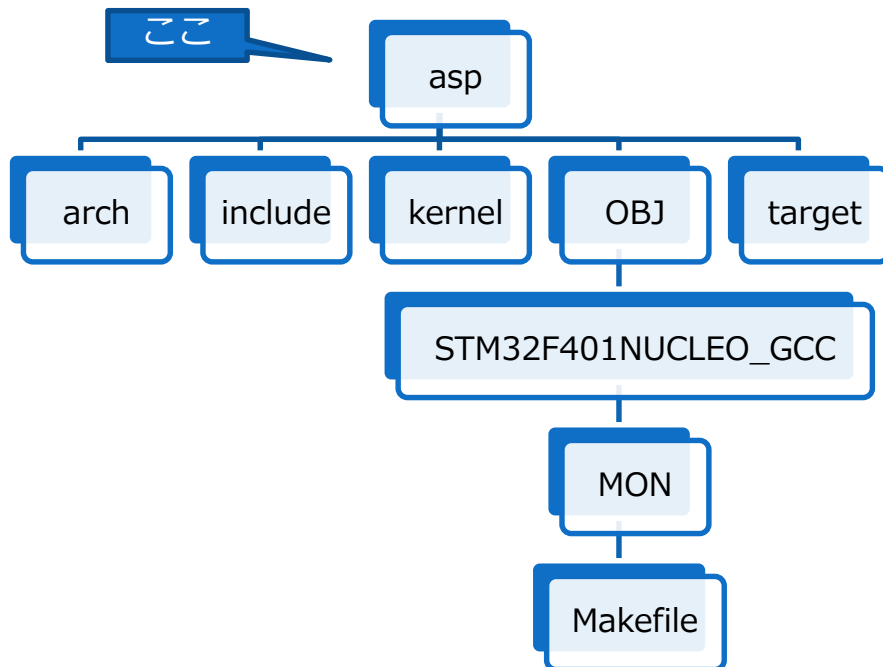
- お好みで下記もどうぞ
 - Japanese Language Pack
 - C++ Intellisense





展開したフォルダを開く

- 展開して統合したaspフォルダをVSCodeで開く





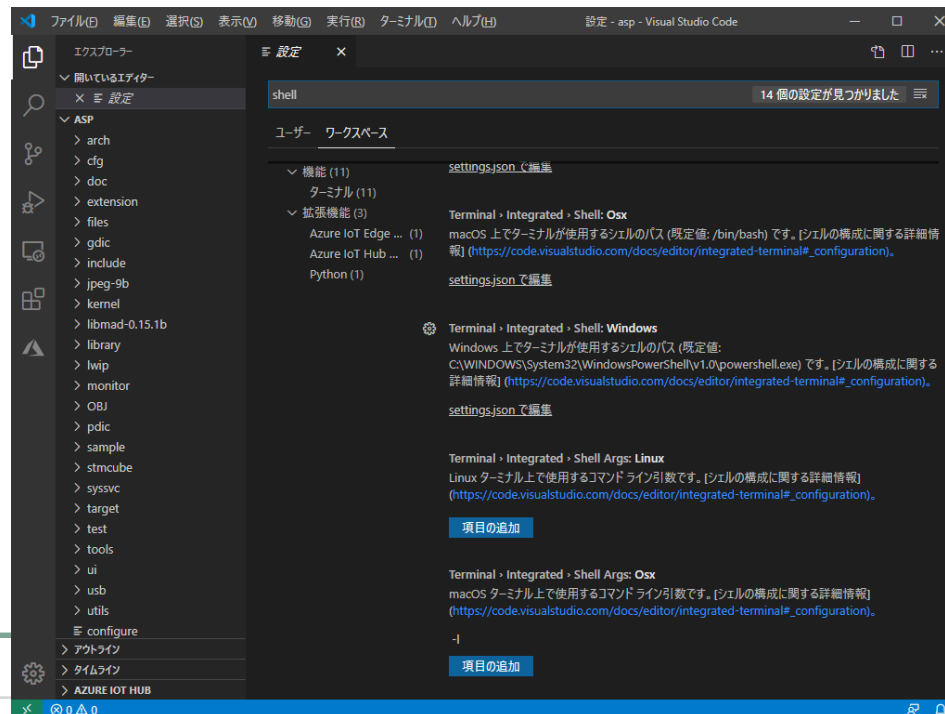
VSCodeの操作手順

1. VSCodeのターミナルの設定
 - settings.json
2. ビルドタスクを作成
 - tasks.json
3. コンパイラの出力（警告やエラー）の処理を設定
 - tasks.json/problemMatcher
4. ビルド/実行
5. デバッグの設定
 - launch.json
6. デバッグ実行



1.VSCodeのターミナルの設定

- Ctrl+,で「設定」画面を開き「ワークスペース」を選択
- 「Terminal > Integrated > Shell: Windows」を探して「settings.jsonで編集」を選択
 - 「shell」で検索すると早い





settings.jsonを編集

- msysのbashへのパスを設定
- bashの引数として「--login -i」を追加

```
{  
  ...  
  "terminal.integrated.shell.windows": "C:¥¥msys64¥¥usr¥¥bin¥¥bash.exe",  
  "terminal.integrated.shellArgs.windows": [  
    "--login",  
    "-i"  
  ]  
  ...  
}
```



環境変数の設定

- msysの動作を設定するための環境変数を設定
 - 「settings.json」に追記
 - 「PATH」にGCCへのパス
 - 「MSYSTEM」に32bit版なら「MINGW32」、64bit版なら「MINGW64」
 - 「CHERE_INVOKING」にmsysの作業ディレクトリ関連の動作設定
 - 「MSYS2_PATH_TYPE」にmsysのPATH関連の動作設定



環境変数の設定

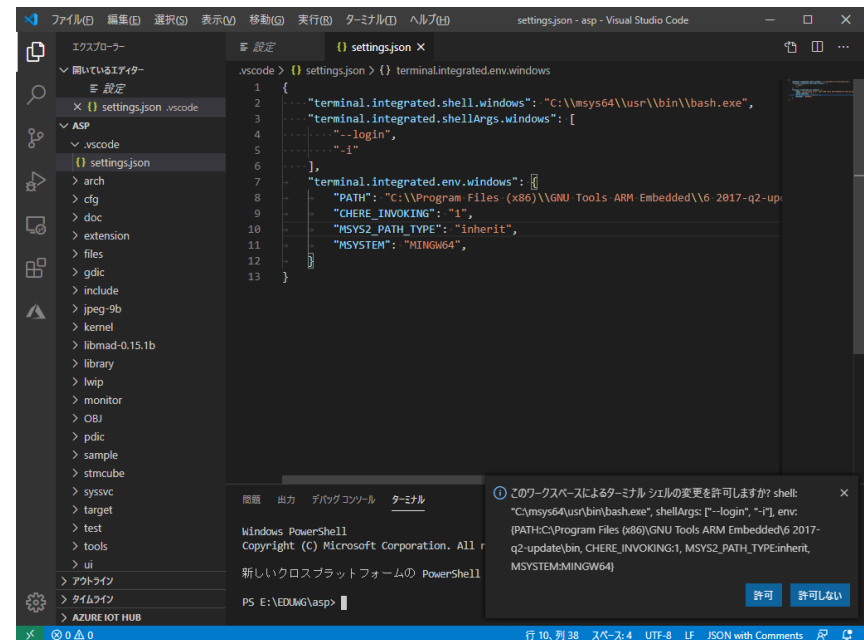
- 設定例

```
{  
  "terminal.integrated.shell.windows": "C:¥¥msys64¥¥usr¥¥bin¥¥bash.exe",  
  "terminal.integrated.shellArgs.windows": [  
    "--login",  
    "-i"  
  ],  
  "terminal.integrated.env.windows": {  
    "PATH": "C:¥¥Program Files (x86)¥¥GNU Tools ARM Embedded¥¥6 2017-q2-update¥¥bin",  
    "CHERE_INVOKING": "1",  
    "MSYS2_PATH_TYPE": "inherit",  
    "MSYSTEM": "MINGW64",  
  },  
  ...  
}
```



ターミナルを表示

- Ctrl+@でターミナルを表示
- ターミナルを初めて表示したときに通知出る
- 必ず「許可」を選択
- 一度「許可しない」としてしまおうと二度と使えません！



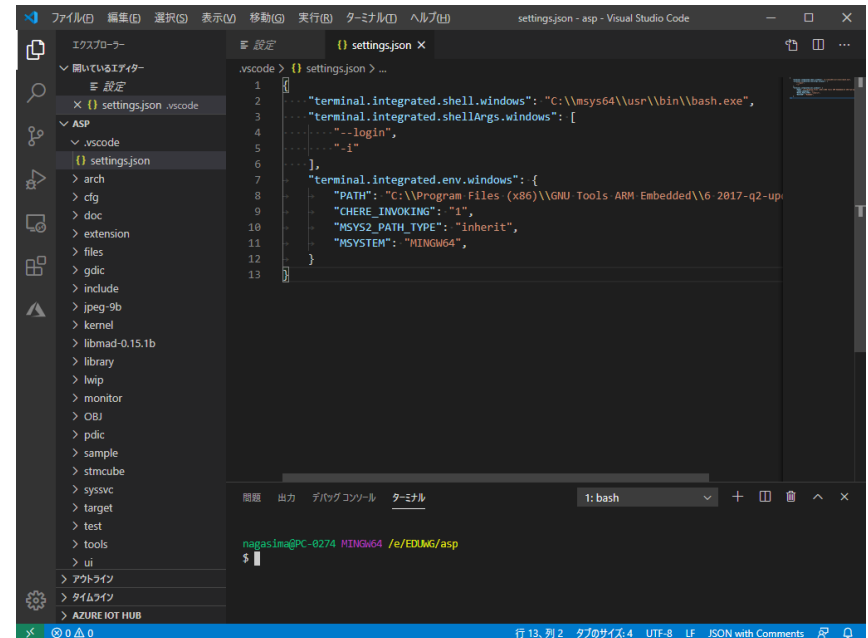


ターミナルを再表示

- Ctrl+@でターミナルを再表示

```
user@PC-NAME MINGW64 /d/TBP/asp  
$
```

のような表示が出れば
成功
(少し時間がかかります)





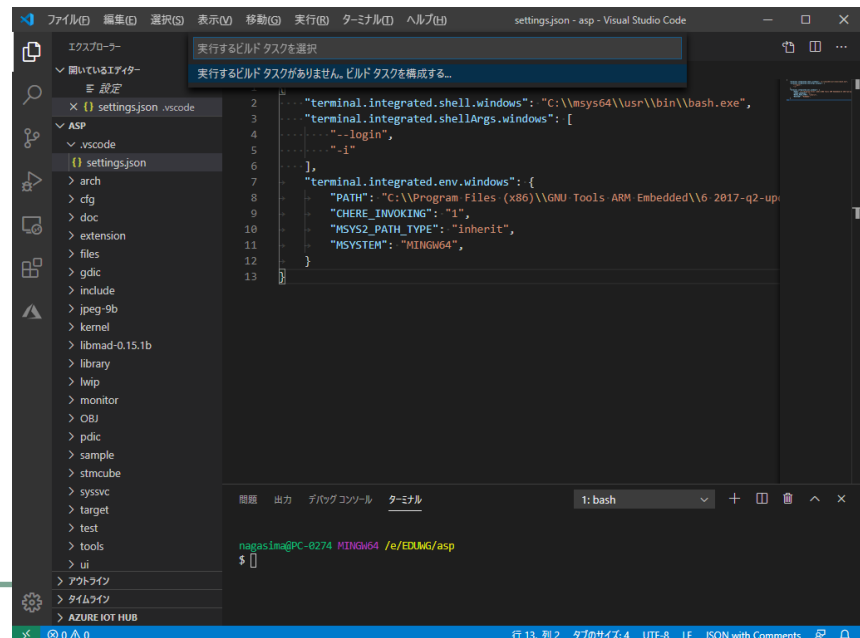
2.ビルドタスクを作成

- makeコマンドを呼び出すビルドタスクを作る
 - ビルドと出力ファイルを削除する動作2つを作る
 - ビルドはターミナルで「make -j all」と入力するのと同じ動作とする
 - 「build app」という名前とする
 - 出力ファイルを削除するにはターミナルで「make -j clear」と入力するのと同じ動作とする
 - 「clear app」という名前とする
 - ちなみに「-j」はCPUのコア数分で並列実行するための指定
 - マルチコアなPCでは、makeが高速になる
 - 弊害として、並列に実行したコンパイラの出力が混ざってしまい、ファイルパスが特定できないことがある



タスクの作成

- Ctrl+Shift+Bで「ビルド タスクの実行」を行う
 - 「tasks.json」ファイルがないと、「実行するビルド タスクがありません。ビルド タスクを構成する」が表示されるので、選択
 - 「テンプレートから tasks.json を生成」が表示されるので、選択
 - テンプレートがいくつか表示されるので、「Others 任意の外部コマンドを実行する例」を選択





ビルドタスク設定

- 「.vscode」フォルダに「tasks.json」ファイルが生成され、開かれる
- 例は「echo Hello」を実行するタスク

```
1 // See https://go.microsoft.com/fwlink/?LinkId=733558
2 // for the documentation about the tasks.json format
3 "version": "2.0.0",
4 "tasks": [
5   {
6     "label": "echo",
7     "type": "shell",
8     "command": "echo Hello"
9   }
10 ]
11
12
```



ビルドのためのタスク追加

- ビルド タスク
 - 「make -j 1 all」 コマンドを実行するタスク
 - 「group」の「kind」に「build」を指定し、「isDefault」を「true」とすると「ビルド タスク」になる

```
{
...
  "tasks": [
    {
      "label": "build all",
      "type": "shell",
      "command": "make.exe",
      "args": [ "-j", "1", "-C", "OBJ/STM32F401NUCLEO_GCC/MON", "all" ],
      "group": {
        "kind": "build",
        "isDefault": true
      }
    },
    ...
  ],
  ...
}
```

makeコマンドを指定

makeコマンドの引数を指定

Ctrl+Shift+Bで実行できる



出力ファイルを削除のためのタスク追加

- 出力ファイルを削除するタスク
 - 「make -j clean」コマンドを実行するタスク
 - ビルドタスクのコピーでコマンド引数を変える

```
{  
...  
  "tasks": [  
    {  
      "label": "clean all",  
      "type": "shell",  
      "command": "make.exe",  
      "args": [ "-j", "-C", "OBJ/STM32F401NUCLEO_GCC/MON", "clean" ],  
      "group": "build",  
    },  
    ...  
  ],  
  ...  
}
```

3.コンパイラの出力の設定



- コンパイルエラーなどの出力を解析する
 - 「problemMatcher」を「\$gcc」にすればGCCの出力を処理できる
 - 少し違った動作をさせたいときは、下記のように記述（出力に表示されるパスとファイルまでのパスとの調整が必要）

```
...
"problemMatcher": {
  "owner": "cpp",
  "fileLocation": [
    "relative",
    "${workspaceFolder}/OBJ/STM32F401NUCLEO_GCC/MON"
  ],
  "pattern": {
    "regexp": "^(.*):(\\d+):(\\d+):\\d+(warning|error|fatal error):\\d+(.*)$",
    "file": 1, "line": 2, "column": 3, "severity": 4, "message": 5
  }
},
...
```

Makefileのあるフォルダ

Makefileからの相対パス



Makefileの変更

- ROM版を出力するにはMakefileの以下の部分を変更
 - 79行目付近

```
#  
# 実行環境の定義（ターゲット依存に上書きされる場合がある）  
#  
DBGENV :=
```



```
#  
# 実行環境の定義（ターゲット依存に上書きされる場合がある）  
#  
DBGENV := ROM
```



Makefileの変更

- asp.binを出力するにはMakefileの以下の部分を変更
 - 222目付近

```
#  
# ターゲットファイル（複数を同時に選択してはならない）  
#  
all: $(OBJNAME)  
#all: $(OBJFILE).bin  
#all: $(OBJNAME).srec
```



```
#  
# ターゲットファイル（複数を同時に選択してはならない）  
#  
#all: $(OBJFILE)  
all: $(OBJNAME).bin  
#all: $(OBJNAME).srec
```



4.ビルド・実行

- Ctrl+Shift+Bでビルドタスク実行
 - 「libkernel.a」がないというエラー
- セミナー資料「基礎 2 編： 2 日目」の10ページについての作業が必要
- Ctrl+Shift+@でターミナルを表示
 - 下記の操作で「libkernel/libkernel.a」を作成

```
user@PC-NAME MINGW64 /d/TBP/asp  
$ cd OBJ/STM32F401NUCLEO_GCC/MON
```

```
user@PC-NAME MINGW64 /d/TBP/asp/OBJ/STM32F401NUCLEO_GCC/MON  
$ make libkernel.a
```

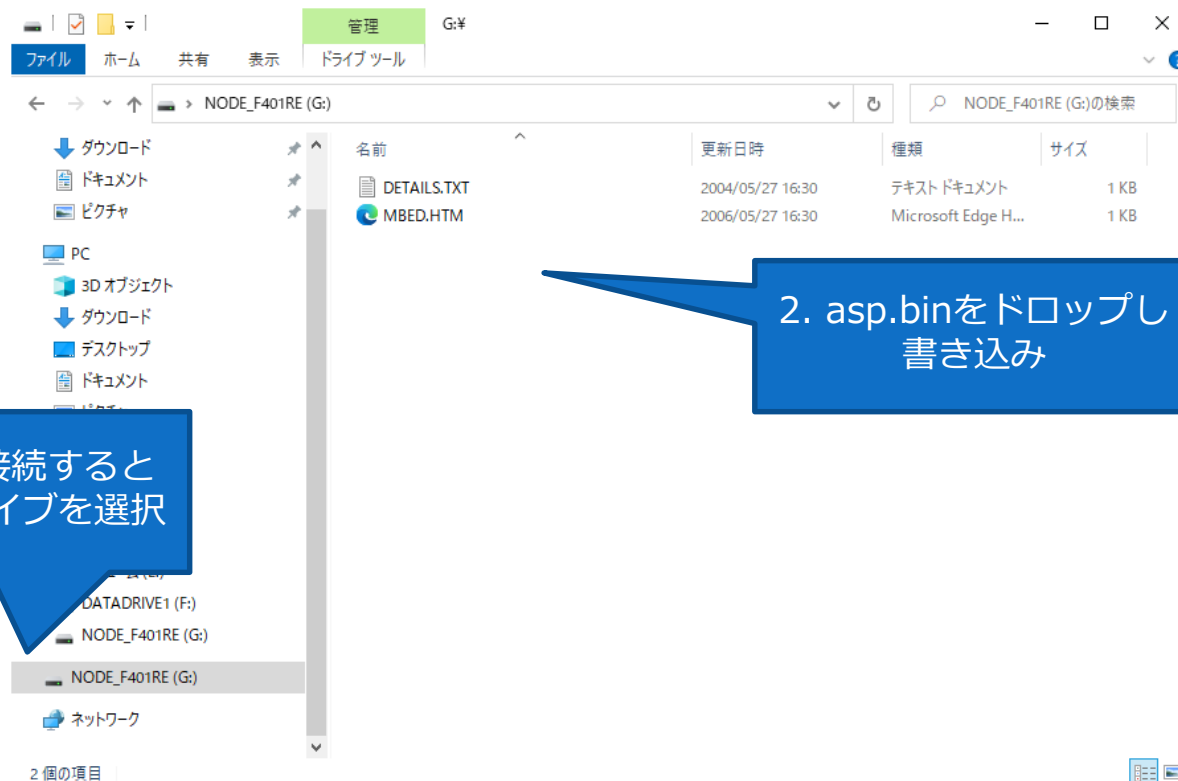
```
user@PC-NAME MINGW64 /d/TBP/asp/OBJ/STM32F401NUCLEO_GCC/MON  
$ mkdir ../libkernel
```

```
user@PC-NAME MINGW64 /d/TBP/asp/OBJ/STM32F401NUCLEO_GCC/MON  
$ cp libkernel.a ../libkernel/
```




プログラムの書き込み

- プログラムは教材とは違う方法で書き込み
 - 「asp.bin」を「NODE_F401RE」ドライブにコピー



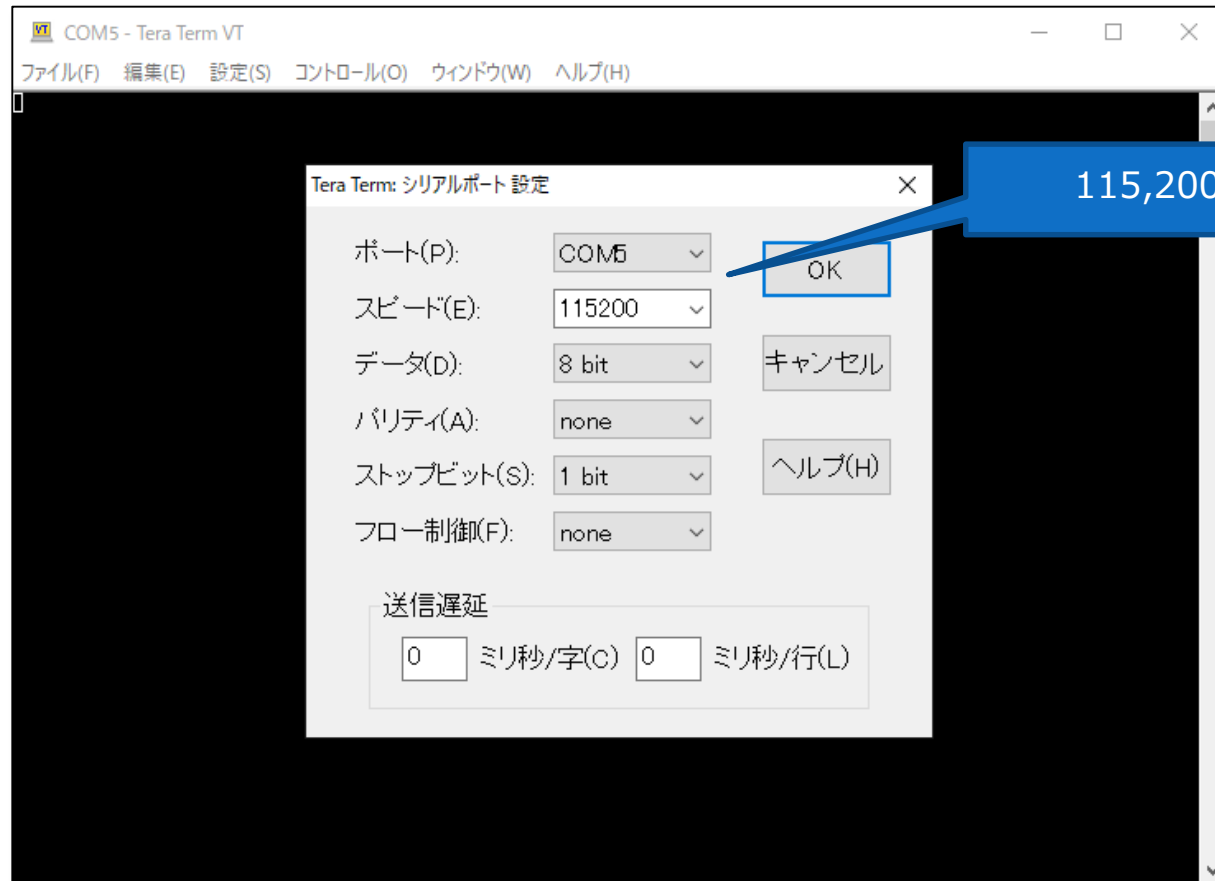
1. NUCLEOを接続すると
表示されるドライブを選択

2. asp.binをドロップし
書き込み



TeraTermでモニタ

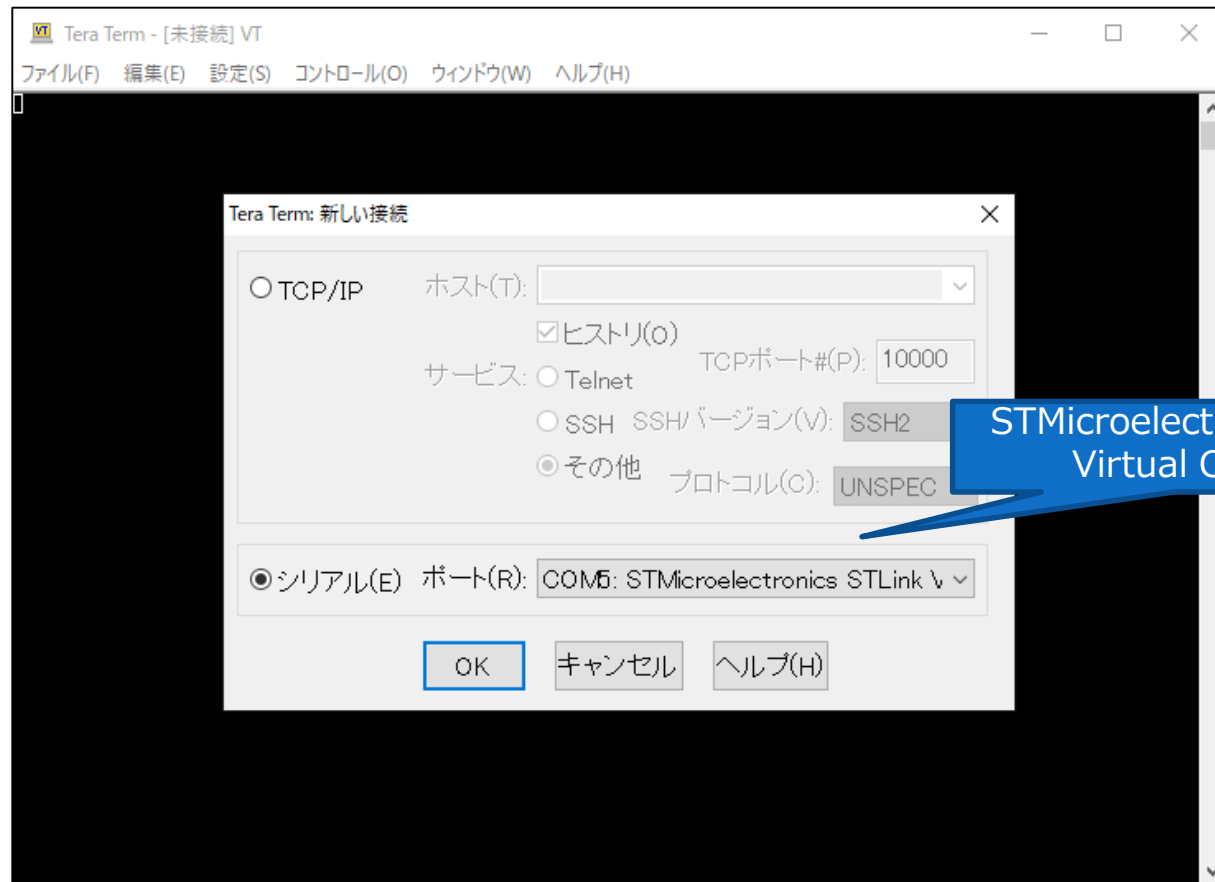
- シリアルポートの設定





TeraTermでモニタ

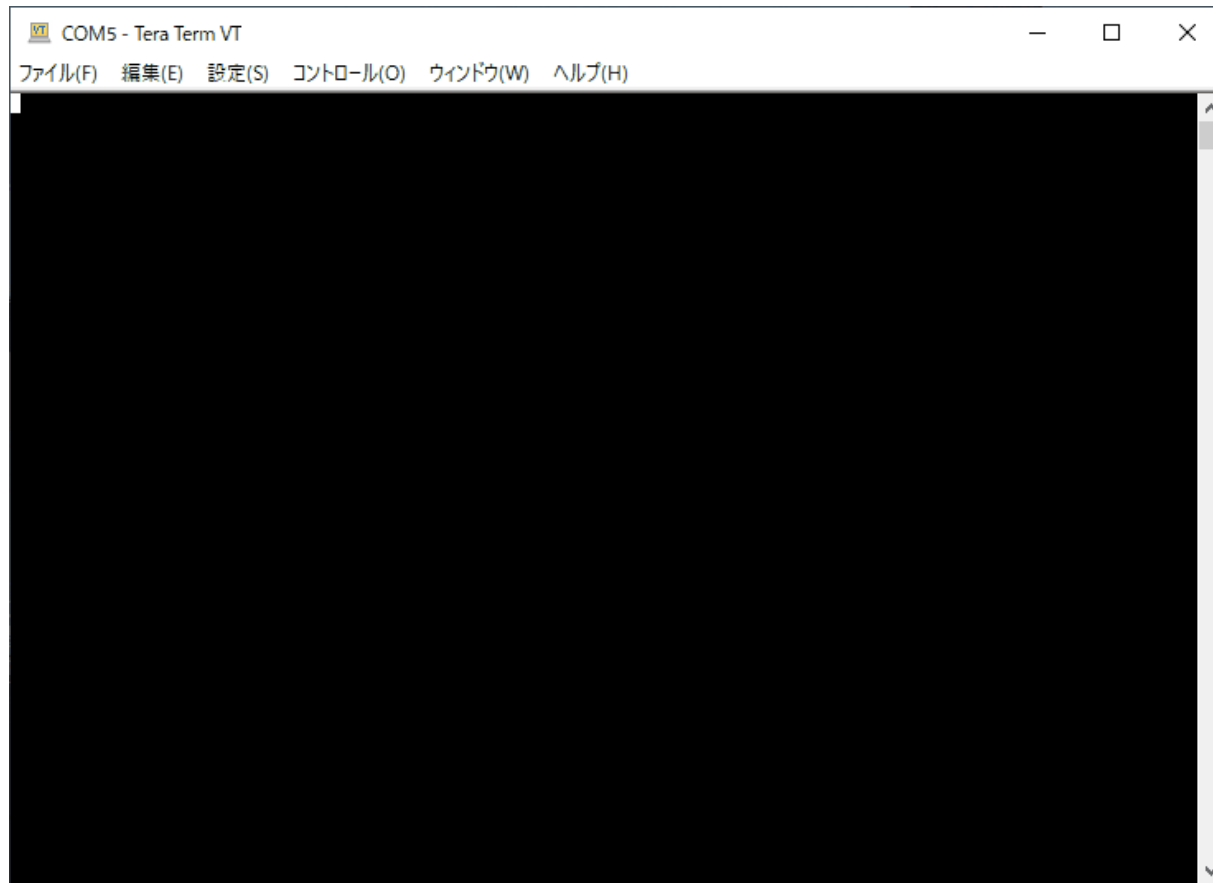
- シリアルポートの設定



実行



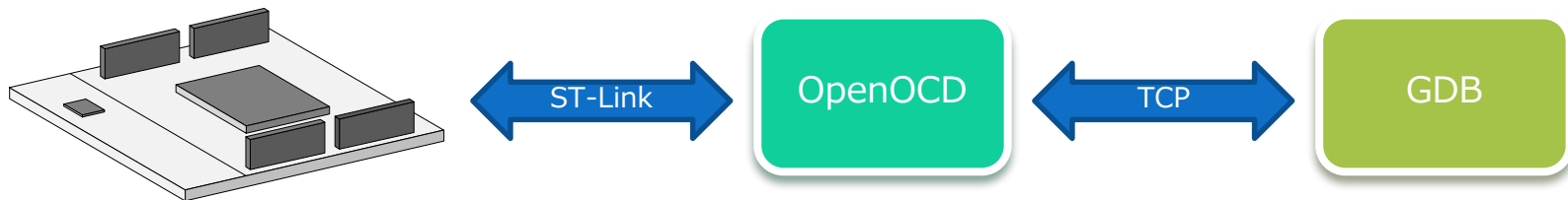
- リセットボタンを押して実行





5.デバッグの設定

- GDBとOpenOCDを使ってデバッグする設定をする
 - GDBはarm-none-eabi-gccに付属する
arm-none-eabi-gdb
 - NUCLEOとはOpenOCDで繋げる
 - VSCodeはGDBと対話してソースコードデバッグができる
 - GDBはデバッグサーバーとしてOpenOCDと接続する





デバッグ画面の表示

- サイドバーのデバッグアイコンを押し「デバッグ」画面を表示
 - 「実行とデバッグをカスタマイズするには、launch.jsonを作成します。」を選択、「Cortex Debug」を選択と「.vscode¥launch.json」が作成される

The screenshot shows the Visual Studio Code interface. The left sidebar has the 'Run and Debug' icon selected. The main editor displays the `tasks.json` file with the following content:

```
{} settings.json {} tasks.json x
.vscode > {} tasks.json > {} tasks > {} 0 > {} args
1
2 // See https://go.microsoft.com/fwlink/?LinkId=733558
3 // for the documentation about the tasks.json format
4
5 "version": "2.0.0",
6 "problemMatcher": "$gcc",
7 "tasks": [
8   {
9     "label": "build_all",
10    "type": "shell",
11    "command": "make",
12    "args": [
13      "-j",
14      "-C",
15      "OBJ/STM32F401NUCLEO_GCC/MON",
16      "all"
17    ],
18    "group": {
19      "kind": "build",
20      "isDefault": true
21    },
22  },
23  {
24    "label": "clean_all"
25  }
26 ]
```

The bottom panel shows the terminal output for the task 'build_all':

```
3: タスク - build all
--rom-image asp.srec --symbol-table asp.syms \
-I ../../../../target/stm32f401nuclco_gcc/target check.tf --api-table ../../../../kern
el/kernel_api.csv --cfg1-def-table ../../../../kernel/kernel_def.csv --cfg1-def-table ../../../../arch/arm_m
gcc/common/core_def.csv sample1.cfg
check complete
arm-none-eabi-objcopy -O binary -S asp.exe asp.bin
make: ディレクトリ '/e/EDUMG/asp/OBJ/STM32F401NUCLEO_GCC/MON' から出ます
ターミナルはタスクで再利用されます。閉じるには任意のキーを押してください。
```



launch.jsonを書き加え

```
{
  // IntelliSense を使用して利用可能な属性を学べます。
  // 既存の属性の説明をホバーして表示します。
  // 詳細情報は次を確認してください: https://go.microsoft.com/fwlink/?linkid=830387
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Cortex Debug",
      "armToolchainPath": "C:\\Program Files (x86)\\GNU Tools ARM Embedded\\6-2017-q2-update\\bin",
      "serverpath": "C:\\Program Files\\GNU MCU Eclipse\\OpenOCD\\0.10.0-12-20190422-
2015\\bin\\openocd.exe",
      "cwd": "${workspaceRoot}\\OBJ\\STM32F401NUCLEO_GCC\\MON",
      "executable": "${workspaceRoot}\\OBJ\\STM32F401NUCLEO_GCC\\MON\\${workspaceFolderBasename}.exe",
      "request": "launch",
      "type": "cortex-debug",
      "servertype": "openocd",
      "interface": "swd",
      "svdFile": "${workspaceRoot}\\STM32F401xE.svd",
      "configFiles": [
        "board/st_nucleo_f4.cfg"
      ],
      "openOCDLaunchCommands": [
        "gdb_breakpoint_override hard",
        "gdb_flash_program disable",
        "gdb_memory_map disable"
      ],
      "preLaunchCommands": [
        "-gdb-set breakpoint pending on"
      ]
    }
  ]
}
```

OpenOCD

ペリフェラルレジスタの定義ファイル
<https://github.com/xpacks/stm32f4-cmsis/tree/xpack/CMSIS/SVD>
から入手



6.デバッグ実行

- 左側上のボタンでデバッグ実行

1. デバッグ実行

3. 継続実行

2. スタートコードで停止する

```
arch > arm_m_gcc > common > start.S
57 /*
58 *--スタートアップルーチン
59 *
60 *--Threadモードで呼び出されることを前提とする。
61 */
62 ATEXT
63 ABALIGN(4)
64 ATHUMB(_start)
65 AGLOBAL(_start)
66 ALABEL(_start)
67 cpsid f...../* 割り込みロック状態へ*/
68
69 #ifdef INIT_MSP
70 /*
71 * MSPの設定
72 */
73 mov r0, #CONTROL_MSP /* MSPを有効に */
74 msr control, r0
75 isb /* controlの操作後に必要 */
76 ldr r0, =kernel_istkpt
77 ldr r1, [r0]
78 msr msp, r1
79 #endif /* INIT_MSP */
```

問題 出力 デバッグコンソール ターミナル

2: タスク - build all

ターミナルはタスクで再利用されます、閉じるには任意のキーを押してください。

> Executing task: make -j 1 -C OBJ/STM32F401NUCLEO_GCC/MON all <

make: ディレクトリ '/e/EDUNG/asp/OBJ/STM32F401NUCLEO_GCC/MON' に入ります
make: 'all' に対して行うべき事はありません。
make: ディレクトリ '/e/EDUNG/asp/OBJ/STM32F401NUCLEO_GCC/MON' から出ます

ターミナルはタスクで再利用されます、閉じるには任意のキーを押してください。

行 73, 列 1 タブのサイズ: 4 EUC-JP LF ARM

完了！



- TOPPERS BASE PLATFORMを試す環境が完成！





あとはセミナーで

● 去年の情報

TOPPERS基礎実装セミナー募集要項

◆主 催

特定非営利活動法人TOPPERSプロジェクト

◆協 賛

STマイクロエレクトロニクス社

◆日 時

基礎1セミナー：2019年9月28日(土)、10月5日(土)

基礎2セミナー：2019年10月19日(土)、26日(土)

基礎3セミナー：2019年11月9日(土)、23日(土)

いずれも10:00～17:00を予定

◆定 員

各セミナー 5名程度

◆場 所

東京都中央区日本橋大伝馬町6-7住長第2ビル3階会議室

◆参加費

各セミナー 会員は無料、非会員は6,000円

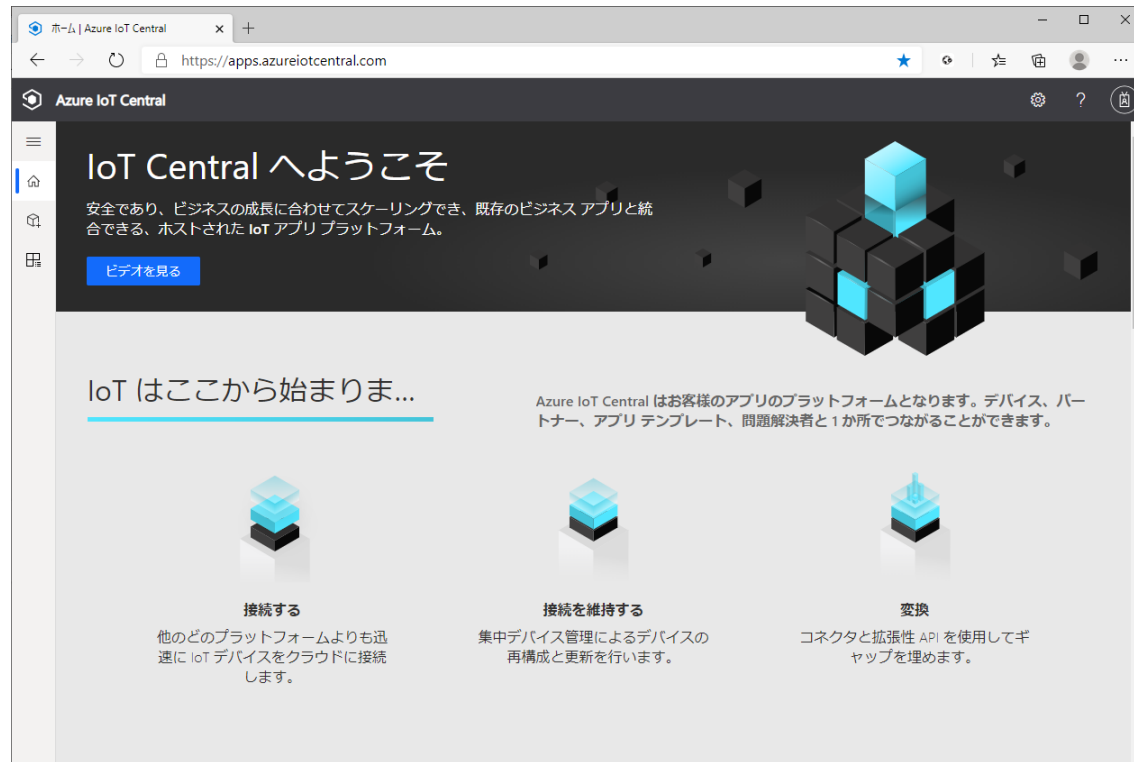


<https://www.toppers.jp/edu-basebp.html>



Azure IoT 接続サンプル

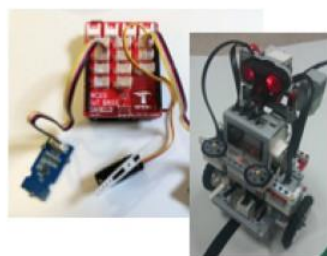
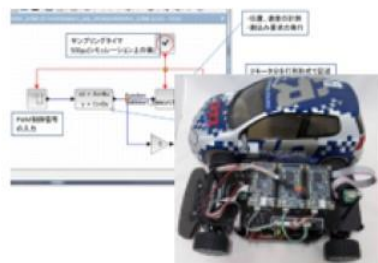
- TOPPERS BASE PLATFORM ver.1.4.1
の NUCLEO-F767ZI で
Azure IoT に接続する展示を16:15から行います



第10回 TOPPERS活用アイデア・アプリケーション開発コンテスト

アプリケーション開発部門

TOPPERS開発成果物を利用した
楽しい・独創的なアプリケーション
実用的なアプリケーション
わかりやすいサンプルアプリケーション
を募集します



実用的なアプリケーション
や開発支援ツールから

たのしい
趣味の電子工作まで！

Arduino on TOPPERS
TOPPERS/EV3RTも
こちらの開発部門へ！

活用アイデア部門

TOPPERS開発成果物の利用者にとって
有益な活用アイデアや
TOPPERSをより良くするアイデア
を募集します（アイデアだけでもOK！）



TOPPERSの教材開発、
ビルドツールや文書管理
の改善案なども！

構想段階の企画案や
ソースコード非公開作品
でもOK

応募資格：国内に在住の方（グループ、法人の場合にはその代表者）なら誰でも
（※TOPPERSプロジェクトの会員でない方もご応募頂けます）

応募期間：**2020年6月12日（金）～ 2020年8月31日（月）**

表彰：＜活用アイデア部門＞ 金賞 銀賞 銅賞 各1万円！
＜アプリケーション部門＞ 金賞 5万円！！ 銀賞 3万円！！ 銅賞 1万円！
＜学生賞＞ 学生の応募作品の中から優秀なものに学生賞（1万円）を授与！

協賛社からの
特別賞も予定
しています

学生や組込み
初心者からベテラン
エンジニアまで、
幅広い応募をお待ち
しています！

募集要項、応募規約、FAQ、技術講習会の案内等は以下のページで公開していきます！

<http://www.toppers.jp/contest.html>



TOPPERS
Toyohashi Open Platform for Embedded Real-time Systems

前回コンテスト(第9回) 受賞作品の紹介

活用アイデア部門: 金賞

クラウドIDEとエミュレータを利用した実機レス開発環境

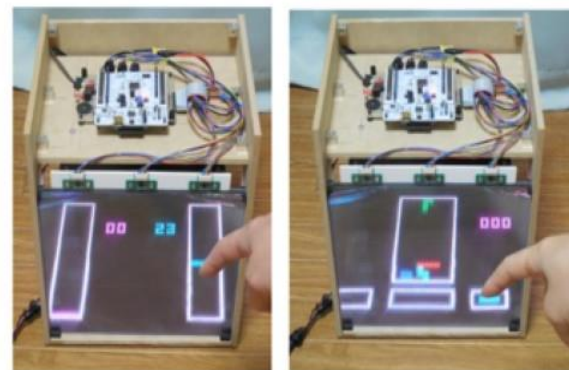
庭野正義 (アイコムシステック株式会社)



アプリケーション開発部門: 銀賞

3Dパネルを使った非接触操作デモ機

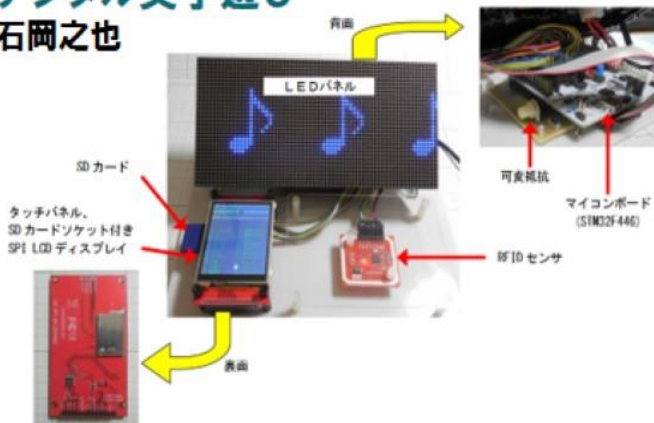
木下浩彰 (ファルコン電子株式会社)



アプリケーション開発部門: 銀賞

デジタル文字遊び

石岡之也

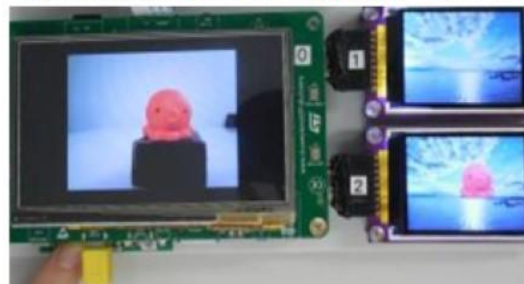


アプリケーション開発部門: 銀賞

ワンタッチマイコンによる画像処理システム

羽佐田 理恵

(FCT組込みシステム研究サークル)



アプリケーション開発部門: 銀賞

TOPPERS Configuration Generator

嶋原一人 (富士ソフト株式会社)



第10回 TOPPERS活用アイデア・アプリケーション開発コンテスト

主催: NPO法人TOPPERSプロジェクト

協賛: CQ出版社、(株)アフレル

ウェブサイト: <http://www.toppers.jp/contest.html>
Facebook <http://www.facebook.com/toppersproject>



TOPPERS

ありがとうございました