

TOPPERS 活用アイデア・アプリケーション開発 コンテスト

部門 : 活用アイデア部門

作品のタイトル : 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

作成者 : アイコムシステック株式会社 庭野正義

共同作業 :

対象者 : TOPPERS RTOS カーネル学習者・開発者

使用する開発成果物 : TOPPERS/ATK2(V850)
TOPPERS/ATK2(Zynq UltraScale+ MPSoC Cortex-R5)
TOPPERS/ASP3(ZYBO-Z7, ZYBO)
TOPPERS/HRP3(ZYBO-Z7, ZYBO)
TOPPERS/FMP(Zynq UltraScale+ MPSoC Cortex-A53)
TOPPERS/FMP(Zynq UltraScale+ MPSoC Cortex-R5)
TOPPERS 新世代カーネル用コンフィギュレータ
TECS

目的・狙い

・目的

クラウド IDE とエミュレータを利用することで、実機レスの開発環境を手軽に構築・共有・破棄できる環境を整える。

具体的には、以下の機能を実現する。

- 開発環境一式のテンプレート^{注1}をクラウド上で管理
- 開発環境一式のテンプレートから各開発者専用のワークスペースを生成
- 生成したワークスペースの共有
- ブラウザ上で、コード編集・コードジェネレート・コンパイル・リンク・エミュレータによる動作確認とデバッグを完結

・狙い

「目的」で記載した機能を実現することで、以下の利点を享受する。

- 環境構築工数の削減
- 環境差分(ツールのバージョン違い等)を無くすことによるトラブルシューティングの簡単化
- ワークスペース共有による遠隔コードレビューや遠隔ペアプロ・モブプロの支援

また、ユーザー専用のワークスペース作成・削除が簡単にできることから、セミナーや講義の演習環境として活用することができると考えている。

注 1: どの RTOS, コードジェネレータ, コンパイラ, リンカ, エミュレータを組み合わせ
て利用するかを定義

アイデア/アプリケーションの概要

Eclipse が開発しているクラウド IDE プロダクトである Eclipse Che(<https://github.com/eclipse/che>) を利用し、そこに TOPPERS カーネル開発環境一式のテンプレートを登録することで、開発環境の作成・共有・破棄とコード編集が簡易に実施できるようになる。また、コードジェネレート・コンパイル・リンク・エミュレータによる動作確認とデバッグを、ブラウザ上で完結することができる。

アプリケーション概要

はじめに、ざっくりとどんなアプリケーションなのかを理解してもらうために、アプリケーション利用の流れを説明する。

アプリケーション利用手順

1. ユーザーは、ウェブブラウザに本アプリケーションが公開する URL を入力し、クラウド IDE のウェブページを開く
 - クラウド IDE のログイン画面が表示される
2. ログイン画面でユーザー名とパスワードを入力し、ログインする
 - クラウド IDE のワークスペース作成画面(図 1)が表示される
3. ワークスペース作成画面で、クラウド IDE が提供するワークスペーステンプレートを選択し、**CREATE** ボタンを押下
 - 選択した **TOPPERS** カーネルの開発に必要な環境が整ったワークスペースが、クラウド上に生成される
 - ワークスペースが生成されると、ワークスペース画面(図 2)に遷移する
4. ワークスペース画面上で、エディット・コード生成・コンパイル・エミュレータによる動作確認とデバッグを行う
 - ここで行う作業は、UI 以外はすべてクラウド上で実行される

このように、ログイン後数クリックで各種 **TOPPERS** カーネル開発環境を整えられる。

提案するクラウド IDE では、以下の環境をワークスペーステンプレートとして提供する。

- TOPPERS/ATK2(V850)
- TOPPERS/ATK2(Zynq UltraScale+ MPSoC Cortex-R5)
- TOPPERS/ASP3(ZYBO-Z7, ZYBO)
- TOPPERS/HRP3(ZYBO-Z7, ZYBO)
- TOPPERS/FMP(Zynq UltraScale+ MPSoC Cortex-A53)
- TOPPERS/FMP(Zynq UltraScale+ MPSoC Cortex-R5)

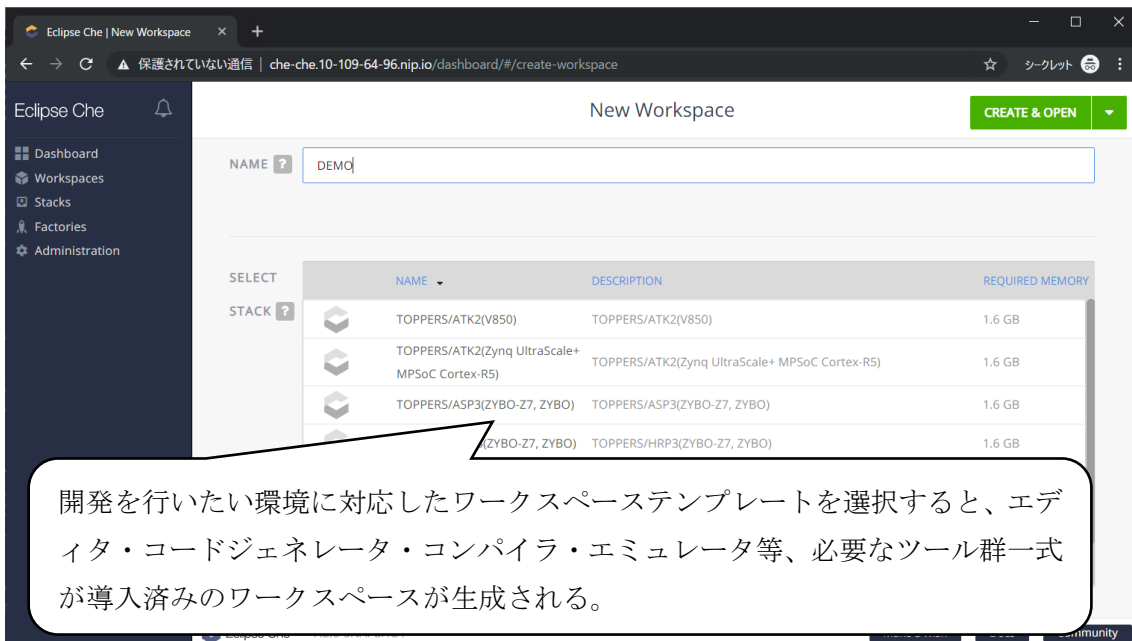


図 1: ワークスペース作成画面。作成するワークスペースのテンプレートを選択する

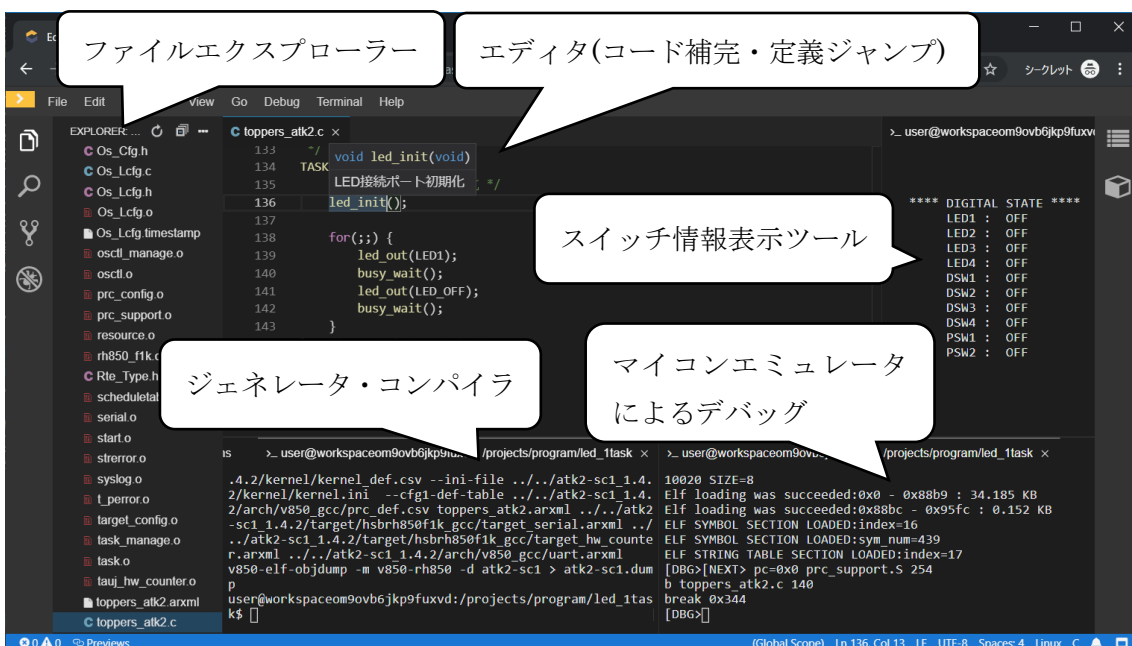


図 2: ワークスペース画面。エディット・ビルド・動作確認を行っているところ

背景

TOPPERS カーネルを用いて RTOS の教育を行う際に、手作業による環境構築を行っているため、図 3 のような問題が発生した。

クラウド IDE を利用し、ワークスペーステンプレートから開発環境を生成することで、

- 環境構築ミスの削減
- 意図せぬ環境差分によるトラブルシューティングの難易度上昇の予防

が期待できる。

また、クラウド環境を使用し、簡単に生成・破棄ができるようになることにより、

- 初学者が本質(カーネルの使い方・仕組みの学習)に集中できる
- 「数時間から数日だけクラウド上に構築する」などが簡単なことから、研修やセミナー、講座の実習環境といった短期間だけ必要な開発環境として使える

といった、教育方面への活用が期待できる。

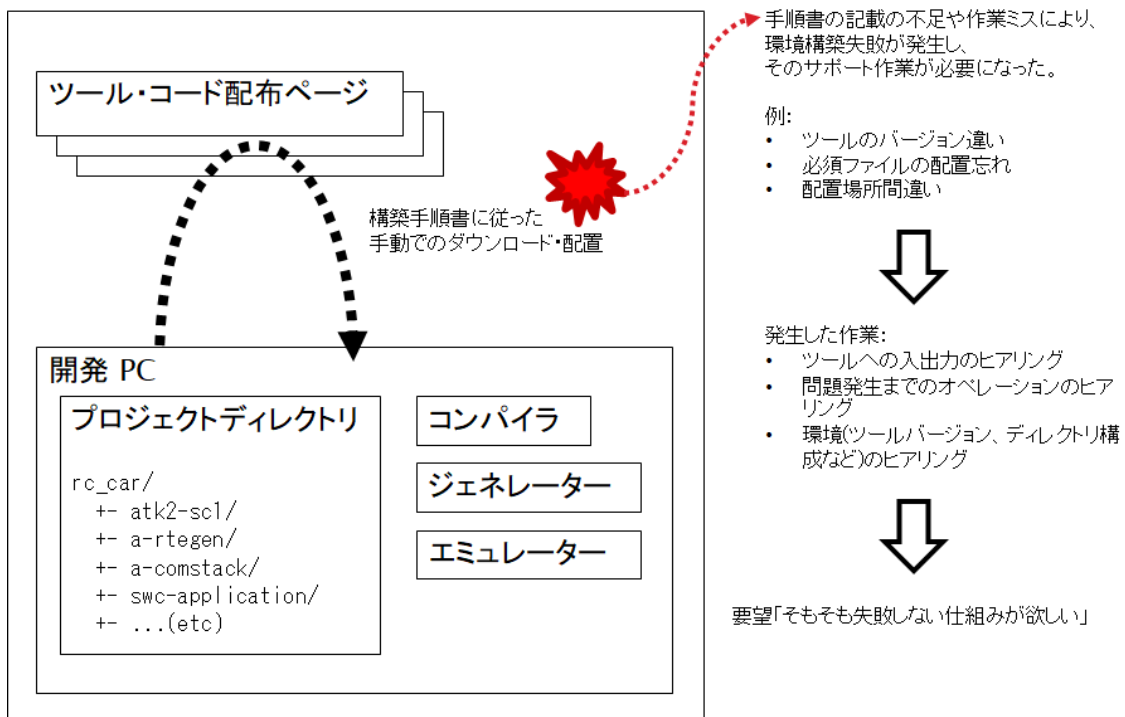


図 3: 開発環境構築時によくある問題

アプリケーション構成

オーバービュー

提案アプリケーションの構成要素を大きく分けると、以下の 5 つとなる。

1. ブラウザ
2. UI サーバー
3. 認証サーバー
4. ワークスペースサーバー(図 4 は、リポジトリの位置との兼ね合いで不正確になってしまっているが、ワークスペースサーバーはワークスペーステンプレート・ワークスペースを含む)
5. リポジトリ

この中で、ローカルマシンにインストールしておかなければならないアプリケーションは「1.」のブラウザのみ(図 4 赤枠)。「2.~4.」は Kubernetes 上に構築された Eclipse Che が管理する(図 4 青枠)。

「5.」のリポジトリは、クラウド環境からアクセスできるリポジトリであればどこで管理してもよい(例: GitHub)。

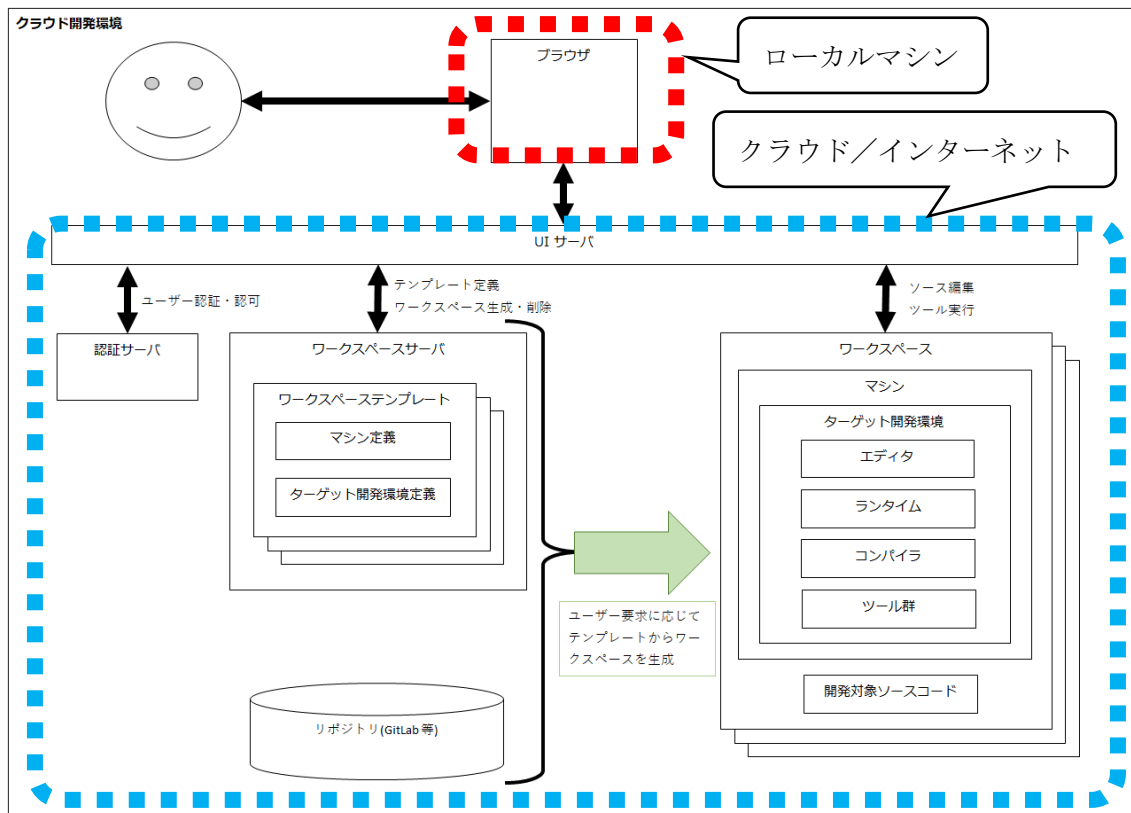


図 4: アプリケーションオーバービュー

「アプリケーション概要」で示したスクリーンショット(図 1, 図 2)と、図 4 の構成要素の対応付けを行うと、図 5 のようになる。

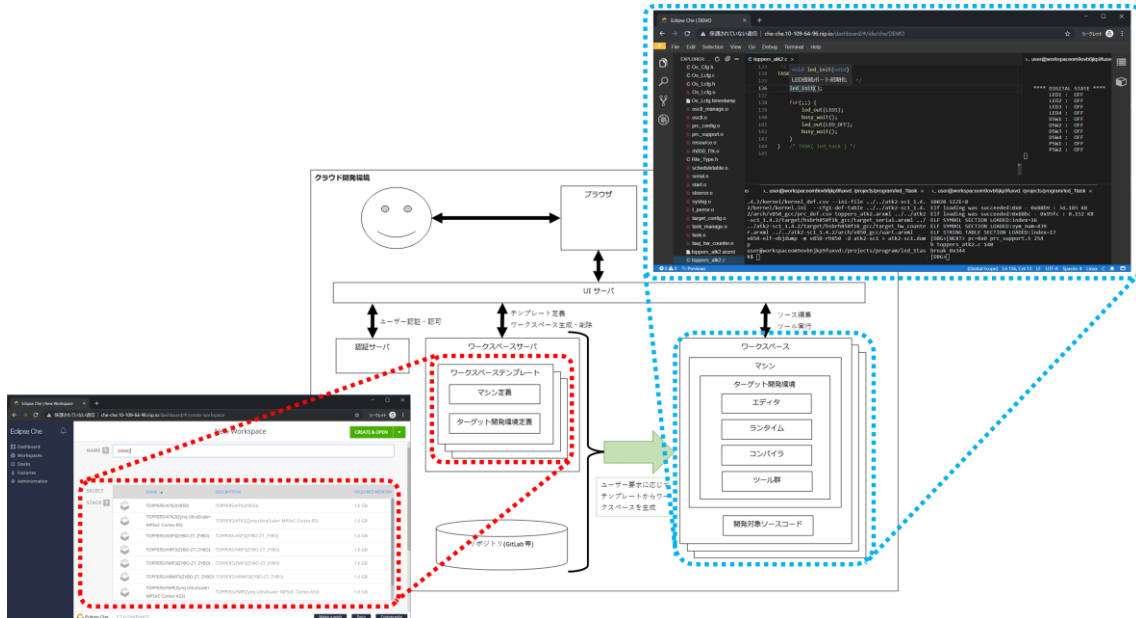


図 5: ワークスペーステンプレートとワークスペースの関係

ワークスペーステンプレートの構成

提案するアプリケーションは、大きく分けて V850 系と ARM 系の開発環境を提供する。それぞれに導入するツール群は以下の通り。

V850 系 (Renesas V850)

種別	導入ツール
Docker ベースイメージ	eclipse/stack-base:ubuntu
カーネル	ATK2
エミュレータ	tmori/athrill
コンパイラツールチェーン	Renesas V850 Toolchain
コンフィギュレータ	mikoto2000/cfg-1
その他	上記ツール群の必須環境

ARM 系 (ZYBO, ZYBO-Z7, Zynq UltraScale+)

種別	導入ツール
Docker ベースイメージ	eclipse/stack-base:ubuntu
カーネル	以下のカーネルのうちのいずれか。 ATK2, ASP3, HRP3, FMP
エミュレータ	Xilinx/qemu
コンパイラツールチェーン	GNU Arm Embedded Toolchain (GNU-A Aarch64 , GNU-RM)
コンフィギュレータ	mikoto2000/cfg-1
その他	上記ツール群の必須環境

構築にあたっての苦労話

Eclipse Che を利用し本アプリケーションを構築するにあたり、いくつかの苦労があったので、そのうちの 3 つを紹介する。

コンフィギュレータの Linux 64bit 対応

Eclipse Che のワークスペースの生成・破棄は、Linux コンテナ技術によって実現されている。そのため、ワークスペースで使用するツールは、64bit Linux 環境で動作する必要がある。

TOPPERS で公開されているコンフィギュレータ(<https://toppers.jp/cfg.html>)は、64bit Linux 環境に対応していないため、今回は TOPPERS/cfg に 64bit Linux 対応の修正を行い、それを利用した([mikoto2000/cfg-1 v1.9.7.2](#))。

実装にドキュメントが追いついていない

「機能としては存在するが、ドキュメントでの言及が全くないもの」が存在する。

例えば今回の検証では、「オンプレ+自己証明書+HTTPS」の組み合わせで Eclipse Che を構築したが、Eclipse Che のドキュメント(<https://www.eclipse.org/che/docs/che-7/introduction-to-eclipse-che/>)には「自己証明書を用いた HTTPS 通信の実現」に関するものが存在しない。今回は、構築時のエラーメッセージから「自己証明書に関する問題である」とあたりをつけたうえで issue, commit, merge request の検索を行い、関連情報を集め、自己証明書を使用する際にのみ必要なファイルやコマンドラインオプションを把握した。

図 4 ではわかりやすさを優先するために省略しているが、ワークスペース管理サーバーはさらに 3 つに分かれており(`devfile-registry`, `plugin-registry`, `che-server`)、ワークスペーステンプレートを追加するには、その役割を理解し、3 つのサーバーの設定ファイルの整合性を保ちながら修正を行う必要があるが、慣れないと非常に難しい。

今後の展望

「背景」であげた問題点の解決方法として、本アプリケーションのクラウド IDE を用いることができる。以下、本アプリケーションをより発展させていくために検討していきたいことを記述する。

1. CI との連携

コンテナベースの CI サーバーを使用すれば、Eclipse Che 用に作成した Docker コンテナを流用し、それをそのまま使って CI を行うことができるはず。

CI サーバーとの連携のノウハウ・ガイドラインを整備すれば、より効率よく CI を実施することができる。そうすれば、教育だけでなく、開発にも本格的に導入していくことができるかもしれない。

2. ワークスペーステンプレート作成支援の仕組み検討

前述の通り、ワークスペーステンプレートの追加は `devfile-registry`, `plugin-registry`, `che-server` の 3 つのサーバーの設定を意識しながら行うことになる。これらが、誰でも追加できるほどに簡単になれば、ワークスペーステンプレートの種類が増えて便利になり、それにより利用が増えるという好循環が作れると思われる。