

# 第17回 TOPPERS 開発者会議 開催レポート

2016年10月2日（日）、3（月）、4日（火）に、おんやど恵（神奈川県湯河原町）にて、第17回 TOPPERS 開発者会議を開催しました。TOPPERS 開発者会議は、TOPPERS プロジェクトの開発成果物の開発、利用に携わる方（非会員でも参加可能）が集まり、2泊3日の合宿形式で集中的に議論、開発する会議です。初の湯河原での開催となり、今年も活発な議論が展開されました。本レポートでは、写真を交えながら、その魅力をたっぷりお伝えします。

## ゲストトーク：



「アイデアとエレクトロニクスをつなぎ、アイデアをすぐにプロトタイピングできる」、「創りたい！から造れるへ」、「夢ある楽しいものづくりを支援する」をコンセプトに進化を続ける「がじえっとるねさす（通称：がじえるね）」プロジェクトは、アイデアを持つユーザとの直接対話、マイコンボードの製作、Maker イベントやアイデアコンテストの開催などを通じて、そのファンを増やしています。今回のゲストトークでは、がじえるねプロジェクトをゼロから立ち上げ、現在もプロジェクトリーダーを務める松山景洋氏（ルネサス エレクトロニクス株式会社マーケティングコミュニケーション統括部グローバル・セールス・マーケティング本部）をお招きし、これまでの半導体メーカの戦略とは一線を画す活動について、プロジェクトのコンセプト、立ち上げから現在に至る経緯、ファンを作り出す秘策、今後の展開等についてお話頂きました。

プロジェクトの立ち上げは2011年まで遡ります。松山氏は、マイコンボードを販売する、とある秋葉原の販売店から、ルネサス製マイコンボードの店頭売上枚数が、2008～2009年ぐらいから、Arduino や mbed に抜かれたことを知ります。この流れを何とか挽回するため、業務外活動として準備し、2012年1月、正式にプロジェクトを立ち上げました。ビジネスモデルとして、1. Free（自由）、2. Easy and Rapid（簡単に素早く）、3. Seniority（ベテラン・カリスマを巻き込む）を掲げ、従来の大口ユーザだけでなく、秋葉原、学校、出版などと積極的に連携し、一般のファンを作る活動を推進しています。2013年からは、海外でもイベントを開催し、例えば、インドでのアイデアコンテストでは、400を超える作品の応募があったそうです。自分たちで新しいモノを作り、より便利な世界を作りたいという高い意欲を強く感じたとのことです。がじえるねプロジェクトは、新しいがじえるねアイテムの創出、国内外でのイベントの開催等を進め、今後も飛躍されることでしょう。

がじえるねプロジェクトの飛躍の背景には、TOPPERS プロジェクトの活動の参考になる話題が数多くありました。例えば、がじえるねのアイテムを手にした時に誰もが思う「ピンクのマイコンボード！」という驚

きは、開発当初から、ペルソナ（理想の顧客像）を、「大学生の女性（趣味：電子工作）がバックから出してもおかしくない」として開発したものであることを聞き、そのマーケティング戦略の重要性を感じました。他にも「ゴールデンの5分（実際にボードを使えるまで5分でできるようにする）を重要視している」というお話は、新たな TOPPERS ユーザを増やすために非常に重要な観点であり、成果物の広報・配布方法の改善に参考になるお話でした。ペルソナの明確化、SNS の活用、プロデューサ（主要利用者）とのミーティング、アイデアコンテストなどの様々なアイデアを参考に、TOPPERS プロジェクトのファンが増加する仕組みを構築していきたいと考えています。がじえるねプロジェクトと TOPPERS プロジェクトは、これまでも、がじえるねアイテム（GR-SAURA と GR-PEACH）への TOPPERS OS の移植、がじえるねウェブコンパイラへのテンプレート提供、ハンズオンイベントの共催等で連携してきました。GR-PEACH に関しては、アイデアコンテストの公式プラットフォームの1つとして採用して頂いています (<http://gadget.renesas.com/ja/product/peach.html>)。今後も、積極的に連携し、組込みエンジニアの創出、啓蒙に尽力していきます。

## ヘテロ通信仕様レビュー

一つ目のテーマは「ヘテロ通信仕様レビュー」で、異なるコアの異なる OS 間での通信機構 MDCOM (Multi Domain Communication Module) についての議論が行われました。複数種類のコアを複数載せた CPU が使われるようになり、各コアで違った OS を動作させることが一般的です。たとえば、Android(Linux)+RTOS などの構成があり、使い方としては、片側のコアで画像認識ひたすら行い、もう片方のコアで通信を行うなどが考えられます。このような OS 間で通信を行うための機構として MDCOM を開発しています。先行の通信機構として OpenAMP という規格があり、Mentor Graphics では商用実装が行われています。OpenAMP は Texas Instruments の OMAP という CPU (ARM)+DSP 構成で、DSP をアクセラレータとして使用するような使い方から発展したもので、Linux 側がマスタで RTOS 側がアクセラレータとして動作するような設計を想定しています。RTOS 側をマスタにするような構成では使いにくく、また、両 CPU をリンクしないと使えないので、片側の CPU の電源を切るなどの消費電力機能の実装に支障が出て使えません。そこで独自に RTOS 側をマスタとして使用できる通信機構として MDCOM を開発しています。MDCOM は SafeG-COM を元にマルチプロセッサ向けに拡張しもので、AIC の加藤氏とで開発を行い、NXP (Freescale) i.MX6, i.MX7 で動作確認ができています。API は SafeG-COM と同じで、OpenAMP からのメリットとして、複数のタスク・プロセスで通信可能、共有メモリを用いた効率の良い通信、ハードウェア依存性が低いなどが挙げられます。デメリットとしては、ライフサイクル管理機能は用意していないことや Linux 標準でサポートされていないことが挙げられ、ウォッチドックや実行シーケンス監視などを検討しました。

二つ目のテーマは「Arduino プログラミングモデルのマルチタスク対応」で、TOPPERS/ASP と Arduino の組み合わせで公開している2つのプラットフォーム間で、使い方に違いがある点について、共通になるよう議論が行われました。Arduino M0 Pro をターゲットとした TOPPERS/R2CA と、GR-PEACH をターゲットとした asp-gr\_peach\_gcc-mbed があり、どちらも、初心者への TOPPERS の普及のために Arduino プログラミングモデルをマルチタスク拡張した環境を提供しています。両者はユーザが実装を行う部分で異なる点があり「setup と loop の関数名」「タスク数の定義の仕方」「delay の実装方法の違い」について議論が行われました。setup と loop の関数名について R2CA では、タスクごとに taskX\_setup/taskX\_loop という関数を用意する方法を取りますが、GR-PEACH では setup と loop が一つずつあり、並列動作するよう拡張するために loop1~loopN を追加する方法を取ります。この議論では、Arduino ユーザからの導入が容易であり、タスクを意識しないところから始められるのを理由に、後者の方法を採用することになりました。タスク数の定義については、両者共にヘッダーファイルにマクロ定義を用意していますが、初心者が該当箇所を変更するという使い方では、難易度が高いという議論になりソースファイルに定義することが議論されましたが、ヘッダーファイルは cfg ファイルでも使用しているので他に方法がなく、変更案はありませんでした。delay の実装方法の違いについては、並列動作をラウンドロビン方式にするか、delay の呼び出しでタスクスイッチするかで議論が行われました。ラウンドロビン方式ではユーザが意識することなく並列動作が実現するので初心者向きで、Arduino ユーザからの導入には向いています。また、cyclic\_handler 関数をソースファイルに置くことで、ラウンドロビンの実装についてユーザが気付くようにしています。一方、delay を呼ぶ方法では RTOS の動作を意識しないと実装が出来ない分、難易度が上がってしまうということで議論が行われました。R2CA では RTOS の教材を目的としているので、ラウンドロビン方式をデフォルトとするには抵抗があるということで、今後検討していくことになりました。

最後三つ目のテーマは「ARMv8-M TrustZone 向け RTOS」で、ASP3 で TrustZone を使用したプラットフォームの検討について議論が行われました。保護機能 OS としては HRP があるが、ASP3 にシンプルな保護機能を

追加することで実現したい機能についてのユースケースを募りました。非公開の商用ライブラリを保護領域で実行し、ライブラリコードの秘匿性を守る使い方や、動的ローディング機能でユーザアプリケーションの不具合から、OSの動作を保障する使い方などが提案されました。

## TOPPERS 統合仕様書のリッチテキスト化

『メイン議論1』のもう一つのセッションでは、高田光隆氏（名古屋大学組込みシステム研究センター）を中心に、TOPPERS カーネル統合仕様書のリッチテキスト化に関する検討を行いました。現在のTOPPERS 統合仕様書は、テキストファイルベースで編集作業を行ったものをPDF化して公開しています。PDFファイルへの変換時には要求仕様IDのタグ情報などが付加されていますが、目次から本文へのジャンプ機能がないなど、一般利用者からみれば使い勝手にはまだ改善の余地があるものでした。

これまでも、MS-Word や DITA などを用いた統合仕様書のリッチテキスト化が提案されてきましたが、構成管理ツールを用いた差分管理との相性が良くないなどの理由で、いずれの提案もプロジェクトとしての採用までには至っていません。そこで今回は、過去の取り組みでの問題点を踏まえ、プレーンテキストに近い形で簡潔な記述ができる Markdown 記法の一つ AsciiDoc を用いた作業プロセスを中心として、導入検討を行いました。

高田氏からは、Ruby で実装された AsciiDoc フォーマッタである asciidoctor を使い、TOPPERS 統合仕様書のテキストデータから作成された asciidoctor 形式文書を実際に PDF や HTML 形式の文書に変換した事例を紹介いただきました。その後、参加者全員が各自のノートPC上に asciidoctor などのツールチェーンを導入し、ツールの使い勝手や出力結果の品質について評価を行いました。その結果、PDF出力時の改行処理などに若干問題が残るものの、十分実用に耐える水準であるという感触が得られました。

さらに、現在のテキストファイル原稿を asciidoctor 形式へ移行させるための自動変換技術や、独自の情報を表現するための書式データ拡張に関しても検討が行われ、統合仕様書リッチテキスト化の動きは大きく前進する結果となりました。



## 組込みで (IoT) で必要なミドルウェアの議論

組み込み (IoT) で必要なミドルウェアの議論では、コンポーネント化にとどまらず、セキュリティまわりの暗号化を含むライブラリ、誤り検出・訂正、ヒープメモリの利用の有無から幅広く議論しました。

C言語は技術者を選ぶため、生産性と信頼性を確保するための JAVA, Ruby, Lua, JavaScript, C#などの言語展開も議論しました。TOPPERS では mruby の実績があり、正規表現を使いたいなどの意見がありました。視点を変えて、モデリング、形式的手法、人工知能などへの対応へも議論を広げました。ビッグデータからの処理ではなく、リアルタイムに処理するためのライブラリ、前処理・後処理、サーバレスの群制御など対応するとよい事柄がありました。また FPGA などのハードウェア対応についても、メンバに知見がある方が多いので、関連して取り組むとよいことが議論になりました。



## WGの現状が分かるプロジェクトアップデートセッション

TOPPERS では幾つかのワーキンググループがあり、開発者会議では TECS、教育、ホームネットワークの 3 グループからの発表がありました。

### ■TECS

ASP3 カーネルでデフォルト利用されることになり外部からの注目がより集まってきたと感じています。TECS の利用機会の促進のために、2016 年春より個別パッケージの定期アップデート化を実施して最新リリースを個別パッケージに取り込むことでバグフィックスや新しい機能を早く使ってもらえるようにしたいと考えています。また TECS では様々な成果物が含まれているので、バージョンを明確にして個別パッケージや簡易パッケージの内容物がはっきりとわかるリリースをしていくと報告されました。今後の予定としては、次のリリースでは動的結合が新機能として入る予定です。またその後としては、デバッグ機能の強化やドキュメントのアップデートなどを行っていく予定のようです

### ■教育

TOPPERS プロジェクトではリアルタイム OS のオープンソース活動と同時にリアルタイム OS を利用するエンジニアを育てることも活動に入っており、教育 WG が主導してきました。活動の歴史を振り返り今後の紹介をしていただきました。当初は SESSAME と連携し、座学は SESSAME 教材、実習は TOPPERS と役割分担をしながら、毎年数回、セミナーでの活動を行ってきています。現在は基礎 2、基礎 3 を改定して新基礎 2、3 とし、さらに全部入りで魅力的なシステムとしてのリファレンスシステムを作成中ということで、「TOPPERS BASE PLATFORM」の紹介がありました。RTOS だけでなく、ファイルシステムやネットワーク、GUI なども扱える意欲的なリファレンスシステムになっているようです。

### ■ホームネットワーク

TOPPERS/ECNL を中心に普及・開発を行っている WG で、今回は ECNL の周辺開発の紹介がありました。一番大きなトピックとしては mruby への取り込みで、ユーザコードを mruby で書けるようにすることで ECNL をより手軽に使ってもらえるようにしたいと考え、mruby へ対応するために ECNL をシングルタスク化など改良し、GR-SAKURA と GR-PEACH でも動作したとのこと。また ECHONET の定義をビジュアル言語でできるプロトタイピングツールの調査・検討を行っているとのこと、Scratch 風 GUI で Ruby 出力できる smalruby やブラウザベースでのビジュアルツールなどの紹介があり、ECNL だけでなく他の応用にも使えそうなど活発な意見が出てきました。

## 第 3 世代カーネルにおける TECS の位置づけとメリット

昨年公開された、第 3 世代カーネルより、SysLog 部など一部機能に TECS が標準で使われるようになりました。このセッションではさらに TOPPERS の活動のなかで、さらに重要となってきた TECS の第 3 世代カーネルにおける位置づけとメリットについて話していただきました。

TECS は、C 言語のためのコンポーネントシステムとして開発されています。TOPPERS の RTOS に最適となるように設計されており、その親和性は非常に高いものとなっています。近年では、mruby フォーラムとも連携をとり、mruby から TECS を操作するインターフェースコードを生成することも可能となっています。第 3 世代カーネルの中での TECS の位置づけとしては大きく 2 つあり、「SysLog 部のカスタマイズ手段」と「TECS によるアプリケーション開発用プラットフォーム」です。第 3 世代カーネルでは SysLog、シリアルドライバ関係が TECS を使用して再実装されています。TECS 化して、コンポーネント間の関連を図示することで、各コンポーネント間の関連を把握しやすくなります。また、TOPPERS カーネルを利用したアプリケーションを作成する上で、カーネルオブジェクトのコンポーネント化を行うことで、cfg・oil の自動生成や、API をわかりやすい名称に再定義、保護カーネルにおける保護ドメインを視覚的に表現するなど、様々なメリットがあります。

TECS は TOPPERS プロジェクトが動き始めた初期からカーネルとともに開発が続けられてきました。これらのメリットは TECS の特徴が役に立っているというよりも、このために TECS が設計されてきているからこそこのメリットといえます。

## Linux からのデバイスドライバ移植の展開

近年の組込みプラットフォームの高機能化に伴い、デバイスドライバやミドルウェアの規模が急激に拡大しており、これらの開発をどのように効率的に実施していくかが大きな課題となっています。「メイン議論 3」では、ドライバ開発効率化方策の一つとして、Linux カーネルからのデバイスドライバ移植について議論しました。ET ロボコンでも公式採用された LEGO Mindstorms EV3 用のプラットフォーム TOPPERS/EV3RT の開発者である李奕驍氏（名古屋大学情報科学研究科）から、EV3RT 開発における Linux デバイスドライバ

の再利用について報告をいただきました。EV3には標準開発環境としてLinuxが用意されており、搭載される各種デバイスのドライバ実装も含まれています。これらのうち、モータ、センサ、LCDなどEV3専用デバイスに対応するカーネルモードドライバは再利用が比較的容易であり、Linux Kernel Driver Interfaceを模倣する簡易な互換レイヤを用いることで元のドライバのソースコードをほとんど修正することなくTOPPERS/EV3RT上に移植できた、とのことでした。一方、USBやファイルシステムのような複雑な汎用ドライバ、Bluetoothのようなユーザモードで動作するドライバは、Linuxカーネルに対する依存度が高く、前記のような簡易なアプローチでの移植は難しい、とのことでした。しかし、これら複雑で大規模なドライバの実装が最も効率化が望まれる分野でもあります。議論の中では、

- TOPPERSカーネルに機能や設計に近い他のプラットフォーム（例えば Zephyr Project <https://www.zephyrproject.org/>）の開発資産を活用してはどうか。
- ほとんどのデバイスがUSB接続できる製品として、USBホストドライバ機能をしっかり整備すれば、多様なデバイスへの対応が容易になる。

などといった様々なアイデアが提案されました。

## Web ページ改定案作成

Web 頁改定案の作成について、どの程度の範囲を対象にするかの議論がありました。一つは、国内向けでは、年齢を10歳くらいまでに広げる案があがりました。根拠としては、IPAのセキュリティキャンプなどで10歳くらいからの参加者がいること、ETロボコンで取り扱っているLEGO Mindstormは、やはり10歳くらいからの利用者が多いことなどが上げられます。せっかく興味を持った子供達に、Linux, Windows, Mac OSなどの、利用するだけのOSの世界ではなく、OSを作る世界も見せるという提案です。また海外からのアクセスを増やすため、同時に英語等の外国語サイトも検討するとよいという意見がありました。特に、最初の講演のルネサスのサイトが、市場を見ながら、市場展開と並行してサイトを構築していった経験団と、その視点からのご意見をいただき、活発な討論ができました。今後は、それぞれの思いを形にしてみ、議論を進めるよいと考えます。

## TOPPERS のインフラをどうするか

TOPPERS プロジェクトの当初からサーバインフラがありますが、そのサービス内容に大きな変化はありません。会員向けには要求が足りているのか？会員外の人の情報発信をカバー（フォロー）できているのかなど、議論を行いました。議論の中で、コミュニケーションツールに特に焦点が当たっていました。現在、Webベースの議論ではメーリングリストが主に使用されています。しかし、非会員にはメーリングリストに投稿することは敷居が高いのではないかという意見や、近年ではチャットツールのようなものもあり、このようなツールを利用していくのもよいのではないかという意見が上がりました。決定したこととして



は、だれでも気負うことなく使えるTwitterを活用し、公式にTipsを流す、またこのbotにTOPPERSに関する質問があれば、気づいた人が答えていくことになりました。

また、メーリングリストで過去のメールを検索できるのかという話が上がりました。Webサイトのメーリングリストのページには検索機能はあるのですが、会員内にもはっきりと周知されていないという問題点もわかりました。



### ハッカソンや開発者会議の感想を最後に発表

クロージングでは、開発者会議へ参加した感想や、各自で取り組んだハッカソンの作業成果などについて、参加者から発表を行いました。そのうちのいくつかを紹介します。

- Blockly という Web ベースのビジュアルプログラミング言語を調査した。Arduino 用のツールをベースに、R2CA のマルチループコードを生成できるツールを開発しているので、まとまりがよいところで公開したい。
- Blockly を使ったプロトタイピングツールの実装を開始した。
- TOPPERS と mruby の連携に関する議題について、関係者と議論ができて良かった。
- 大学の研究で TINET を使うので、ホームネットワーク WG の長島さんの助言を頂きながら調査を行った。
- AsciiDoc 関連の作業を、高田先生と一緒にやった。HTML 文書出力は、2、3 の問題を除けば公開可能なレベルになった。PDF 出力は、禁則処理に不具合があるので別なフォーマット実装を探す予定。
- 初日にごじえるね松山さんから頂いた GR-CITRUS への TOPPERS カーネル移植に取り組んだ。
- R2CA セミナを受講。GE-PEACH + NCES ボード + WiFi? ボードを、mbed ライブラリを使ってすぐに動かす事ができた。

#### 開発者会議実行委員会

岡山直樹 (アイシン・コムクルーズ(株)/実行委員長)  
 伊藤良樹 ((株) ヴィッツ)  
 伊予田健敏 (創価大学)  
 小川清 (名古屋市工業研究所)  
 小南靖雄 (個人会員)  
 高瀬英希 (京都大学)  
 高田光隆 (名古屋大学)  
 長島宏明 (コアーズ)  
 平橋航 (東芝情報システム(株))  
 堀武司 (北海道立総合研究機構工業試験場)  
 松原豊 (名古屋大学)  
 山根ゆりえ (特別会員)

#### NPO 法人 TOPPERS プロジェクト

〒103-0011 東京都中央区日本橋大伝馬町 6-7 住長第 2 ビル 3F  
 (一社) 組込みシステム技術協会内

TEL & FAX: 03-5643-5166 Email: devconf@toppers.jp

Facebook : <http://www.facebook.com/toppersproject>

Copyright (C) 2000 - 2016 by TOPPERS Project, Inc. All Rights Reserved.

※ “TOPPERS” および TOPPERS プロジェクトのロゴは、TOPPERS プロジェクトの登録商標です。