

mruby Bytecode Loader Using Bluetooth in Multi-VM Environment (Bluetooth を用いたマルチ VM 対応 mruby バイトコードローダ)

学籍番号：09C12174 潮 研究室 山本 拓朗

1 まえがき

近年、組込みシステムは複雑化・大規模化しているため、組込みソフトウェアの生産性が問題になっている。組込みソフトウェア開発の生産性の向上を目的として、組込みシステム向けのスクリプト言語である mruby (軽量 Ruby) を適用させたコンポーネントベース開発が可能なフレームワークである mruby on TECS が提案されている [1]。

現状の mruby on TECS では、プラットフォームに mruby バイトコードを組み込んでいるため、mruby プログラムを修正する度にコンパイル・リンクし直す必要がある。さらに、マルチ VM を提供しているが、複数の mruby アプリケーションを効率よく並行動作させるには開発者がリアルタイム OS の機能を熟知している必要がある。

本研究では、mruby on TECS の拡張として、mruby アプリケーションのバイトコードを Bluetooth で転送することで開発効率を向上させる。さらに、複数の mruby アプリケーションを協調動作できるフレームワークを提案する。

2 既存フレームワークの問題点

mruby on TECS は、mruby と、組込みシステムに適したコンポーネントシステムである TECS (TOPPERS Embedded Component System) を組み合わせたフレームワークである。

スクリプト言語は生産性が高い反面、C 言語に比べると実行速度が遅いため、組込みシステムに適用するのは難しい。mruby on TECS では、mruby プログラムから C 言語の関数を呼ぶ機能を提供しており、mruby に比べて、アプリケーションを約 100 倍速く実行できる。

mruby on TECS では、mruby プログラムを修正する度にターゲットデバイスで SD カードを抜き差しし、OS を再起動する必要がある。さらに、複数の mruby アプリケーションを並行実行させる場合、開発者がタスクを待ち状態へ遷移させる OS の機能を呼び出さなければならない。

3 提案フレームワーク

提案フレームワークは、mruby on TECS を拡張して、Bluetooth を用いた mruby バイトコードローダと、実用的なマルチタスク処理の実装を行った。システムモデルを図 1 に示す。

提案フレームワークでは、プラットフォーム (図 1 の下部) をコンパイル・リンクし、ターゲットデバイス上で起動する。ホスト PC 側では、mruby アプリケーション (.rb) をバイトコード (.mrb) にコンパイルし、Bluetooth を通じてターゲットデバイスにバイトコードを転送する。ターゲットデバイス側では、RiteVM に実装されたローダが転送されたバイトコードを受信し、すでにプラットフォームに含まれている mruby ライブラリと合わせてアプリケーションを実行する。これによって、SD カードを抜き差しする手間や OS を再起動する時間が省けるため、作業効率を上げることができる。

さらに、各 VM の処理を平等に実行する RiteVM スケ

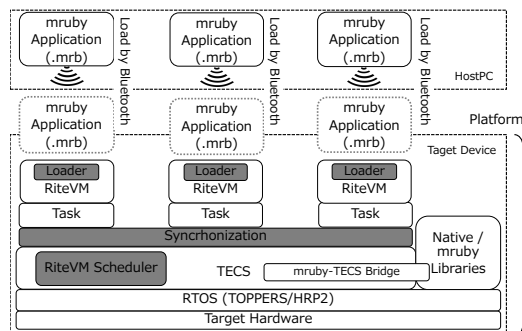


図 1 System Model

表 1 Comparison of the proposed and previous work

	Bluetooth Loader	VM Management	VM Scheduler	Synchronization of Application
eLua				
Owl system				
mruby on TECS		✓		
Proposed framework	✓	✓	✓	✓

ジューラを提供することで、マルチタスク処理を効率的に利用できる。複数タスク間の同期にはイベントフラグ処理を適用し、すべてのタスクが同時に起動するように実装した。イベントフラグのセットパターンや待ちパターンは、各コンポーネントの属性として定義する。これによって、VM の数に関わらず、同じ C ファイルを利用できる。

提案フレームワークでは、RiteVM や RiteVM スケジューラ、イベントフラグはすべて TECS のコンポーネントとして実装されているため、開発者の必要に応じて容易に機能の取り外しができる。

表 1 に、提案フレームワークと既存フレームワークの比較を示す。提案フレームワークでは、Bluetooth を用いたバイトコードローダに加え、VM スケジューラおよびアプリケーションの同期が可能になった。

4 結論

Bluetooth を用いた mruby バイトコードローダによって、mruby on TECS でのソフトウェア開発の作業効率を向上させ、RiteVM スケジューラは複数の mruby アプリケーションを効率良く実行する。さらに、提案フレームワークで提供される機能はコンポーネントであるため、機能の取り外しや再利用が容易になる。

実験評価では、ローダによる作業効率の向上やオーバヘッドの少ないマルチタスク設計の有用性、コンポーネントベース開発の利点を示した。

参考文献

- [1] T. Azumi, and Y. Nagahara, and H. Oyama, and N. Nishio, “mruby on TECS: Component-Based Framework for Running Script Program,” Proceedings of the 18th IEEE International Symposium on Real-Time Distributed Computing (ISORC), pp.252-259, 2015.