

Component-Based Software Development Framework for Embedded IoT Devices

(組込みIoTデバイス向けコンポーネントベースソフトウェア開発フレームワーク)

学籍番号：29C16100 潮 研究室 山本 拓朗

1 緒論

近年, IoT (Internet of Things) 市場の拡大により, 組込み機器をネットワークに繋げて操作するために, ソフトウェア開発の高い生産性が要求されている. 組込みソフトウェア開発の生産性向上のため, スクリプト言語 mruby (軽量 Ruby) を用いたコンポーネントベース開発が可能なフレームワーク mruby on TECS [1] が提案されている.

本研究では, IoT 機器に対応するため, mruby on TECS の拡張として組込み向け TCP/IP プロトコルスタック TINET の機能を mruby プログラムから実行できるフレームワークを提案する. 提案フレームワークでは, TINET のコンポーネント化を行うことで拡張性やコンフィグナビリティが向上し, プロトコルを容易に設定できるようになる. さらに, 動的メモリアロケータ TLSF (Two-Level Segregate Fit) のコンポーネント化も行い, スレッドセーフに複数のスレッドを動作可能なメモリアロケータもフレームワークに組み込んでいる.

2 mruby on TECS

mruby on TECS は, スクリプト言語 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. スクリプト言語はその使いやすさから生産性が高い反面, 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ことは難しい. mruby on TECS では, mruby ブリッジという mruby プログラムから C 言語の関数を呼び出す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.

さらに, TECS によってコンポーネントベースで開発されているため, ソフトウェアの再利用が高く, 機能の追加や取り外しが容易である.

3 提案フレームワーク

提案フレームワークでは, mruby on TECS を拡張して, TCP/IP プロトコルスタック TINET+TECS と動的メモリアロケータ TLSF+TECS の設計・実装を行った. 図 1 に提案フレームワークのシステムモデルを示す.

TINET+TECS は, TECS により TINET をコンポーネント化した組込みシステム向け TCP/IP プロトコルスタックである. TINET は組込みシステムに適したコンパクトな TCP/IP プロトコルであるが, 多くの複雑なソースコードやマクロのせいでコンフィグナビリティが低く, メンテナンスや拡張・検証が難しい. TINET+TECS では, プロトコルの各層をコンポーネントとして実装しているため, TCP 層や UDP 層の付け外し, IPv4 と IPv6 の共存, 通信バッファのサイズ変更など, プロトコルの設定が既存の TINET に比べて容易になっている.

TLSF+TECS は, TECS により TLSF をコンポーネント化した組込みシステム向け動的メモリアロケータであ

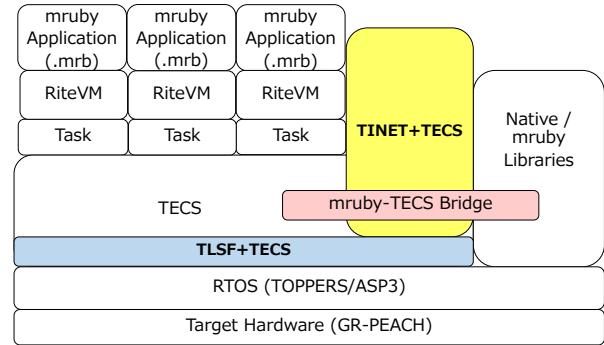


図 1 システムモデル

```
1 begin
2   io = AnalogIO.new(A0, INPUT)
3   cep = TCP.new()
4   cep.accept
5   loop do
6     val = io.read
7     cep.snd val.to_s + "\n"
8     RTOS.delay(1000)
9   end
10 rescue => e
11   puts "[ERROR]" + e
12 end
```

図 2 提案フレームワークでのアプリケーション例

る. TLSF はメモリ利用効率が良く常に $O(1)$ で動作する高速なアロケータであるが, 複数のスレッドが並行動作すると, メモリ競合が起きる場合がある. TLSF+TECS では, TLSF コンポーネントが独自のヒープ領域を保持するため, 排他制御なしで複数のスレッドを動作させることができる. さらに, コンポーネントの特性により, ヒープメモリのサイズ変更が柔軟になる.

図 2 は, 提案フレームワーク上で動作させるアプリケーションの例 (1 秒周期でセンサデータを取得し, サーバへ送信する) を示している. mruby プログラムから TINET の機能を利用でき, IoT デバイスに適用できるネットワークソフトウェアを開発できる.

4 結論

mruby on TECS フレームワークの拡張として, IoT デバイス向けのソフトウェアを開発するフレームワークを提案した. 提案フレームワークでは, TCP/IP プロトコルスタックである TINET の機能を mruby プログラムから呼び出せる. さらに, ソフトウェアコンポーネントとして TINET+TECS と TLSF+TECS を実装し, コンポーネント化によるソフトウェア開発の生産性向上を示した.

参考文献

- [1] T. Azumi, Y. Nagahara, H. Oyama, and N. Nishio, "mruby on TECS: Component-Based Framework for Running Script Program," in Proc. of IEEE ISORC, pp.252-259, 2015.