

リアルタイム OS におけるマルチタスク機能の実装方法と評価 (The Implementation of Real-time tasks and its evaluation)

有馬 明日香[†] 田中 和明[‡]

九州工業大学情報工学部[†] 九州工業大学情報工学研究院[‡]

1. 諸言

近年, IoT(Internet of Things)に注目が集まり, 組込みシステムの需要が拡大しているが, 組込み分野における技術者不足や開発コストの問題などが増えてきている. 組込みシステムでの重要な機能として, 複数の機能を実現するマルチタスク処理があげられる. 現在マルチタスク処理を行う際はリアルタイム OS(RTOS)を利用することが一般的である. RTOS を使用したマルチタスク処理は高いリアルタイム性能を持っているが, 使用するには敷居が高い. そのため, 本研究では可読性と生産性が高いプログラミング言語 Ruby を組込み向けに軽量化した mruby/c に注目した. mruby/c は VM のコンカレント実行により, 擬似的なマルチタスクを実現する機能が提供されている. そこで本研究では, モータ制御を具体例として, RTOS の一つである TOPPERS OS のマルチタスク機能と, mruby/c によるコンカレント機能のそれぞれを用いて実装し, 評価した.

2. システム概要

2.1 mruby

mruby は Ruby の高い開発効率, 安全性の良さを保持したまま, 組込みソフトウェア開発に適用できるように軽量・最適化したものである. Ruby プログラムをコンパイルしてバイトコードを生成し, RiteVM で実行するため, ハードウェアに依存しない.

2.2 mruby/c

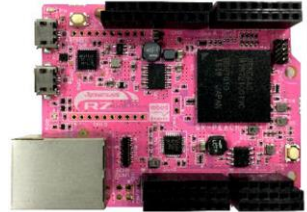
mruby/c は上記の mruby をさらに軽量化したものである. 消費メモリは 30KB 程度で消費電力も少ない. mruby/c は, 複数のバイトコードを切り替えながらコンカレントに実行することで擬似的なマルチタスクを実現する機能が提供されている. RiteVM が切り替えを行うため, OS がなくても動作する.

2.2 TOPPERS/ASP カーネル

TOPPERS/ASP カーネルはオープンソースの RTOS で, 高い信頼性・安全性・リアルタイム性を要求される組込みシステムに適用されている. 本研究では, ASP カーネルを用いたシステム開発を行った.

2.3 GR-PEACH

本研究ではルネサス社の GR-PEACH を使用した. 外観と仕様を図 1 に示す.



項目	仕様
CPU	ルネサスエレクトロニクス社製 RZ/A1H(Cortex-A9)
clock	400MHz
RAM	10MB
Flash ROM	8MB

図 1: GR-PEACH の外見と仕様

3. システム構成

TOPPERS OS では, メインプログラムの中に各タスクを loop 文で用意してコンパイルしたのちに GR-PEACH に搭載する. 言語は c++を使用した.

mruby/c では, まず Ruby プログラムを各タスク分作成し, コンパイルする. その際に「-E -Bary(配列名)」をオプションとして追加すると, ary という配列にバイトコードを格納した c 言語の中間コードが作成される. 次にタスク数だけ

The Implementation of Real-time tasks and its evaluation

Asuka Arima[†] Kazuaki Tanaka[‡]
School of Computer Science and Systems Engineering[†]
Graduate School of Computer Science and Systems Engineering[‡]

VMを用意し、作成したC言語の中間コードを割り当てる。コンパイルし、GR-PEACHに搭載する。

4. 研究内容

本研究では、モータの制御を4つの機能に分けて作成した。それぞれのタスクの役割を表に示す。また、モータ制御の機構とタスクがどの部分で使用されているかの構成を図2に示す。

表1：モータ制御におけるタスクの役割

タスク 1	PWM 制御でモータを回転させる。
タスク 2	フォトインタラプタから信号を受け取り、回転数を計算する。
タスク 3	モータの回転数を設定した回転数と比較し、異なっていた場合はモータの回転数を変化させる。
タスク 4	モータの回転数を7セグメントに表示する。（ダイナミック点灯）

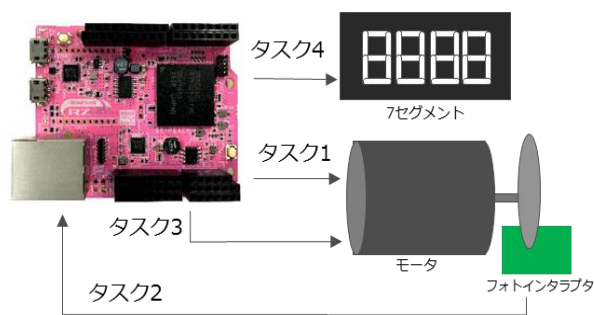


図2：モータ制御図

4.1 TOPPERS OS における動作検証

まず、GR-PEACH への TOPPERS OS の搭載だが、使用しているルネサス社の Web コンパイラに標準で TOPPERS/ASP カーネル用テンプレートが搭載されているため、今回はそれを使用した。動作確認は7セグでの回転数表示と PC 上のターミナルソフトを用いて行った。

4.2 mruby/c における動作検証

mruby/c 内の src フォルダにある C プログラムとヘッダファイルをプロジェクト内のメインプログラムと同じ場所にコピーする。Ruby で表1に示す4つのタスクを分けて作成したプログラムをコンパイルしたものを、C言語の中間コードに変換した。VMなどはC言語で記述されているため、Ruby で使用したメソッドとC言語の関数との紐づけを行った。メインプログラムではVMを4つ作成し、それぞれにC言語の中間コード

を割り当てた。このとき、C言語の中間コードはメインプログラムの中ですべて include しておく。動作確認は7セグでの回転数表示と PC 上のターミナルソフトを用いて行った。

5. 研究結果

フォトインタラプタで信号を検知するのにかかった時間を5000回ずつ計測し、各値が何回得られたかを図3,4に示す。図3,4を比較すると TOPPERS OS の方が、信号を受け取る時間が少なく、値のばらつきも少ないことが分かった。

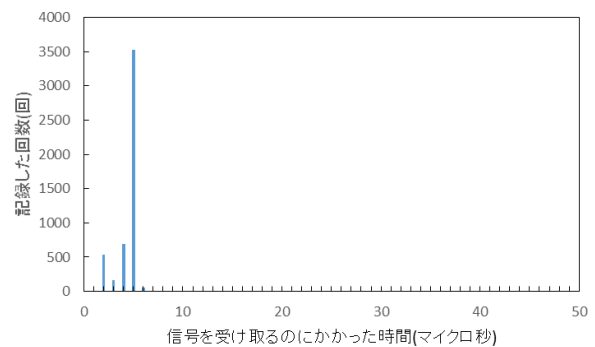


図3：TOPPERS OS の評価結果

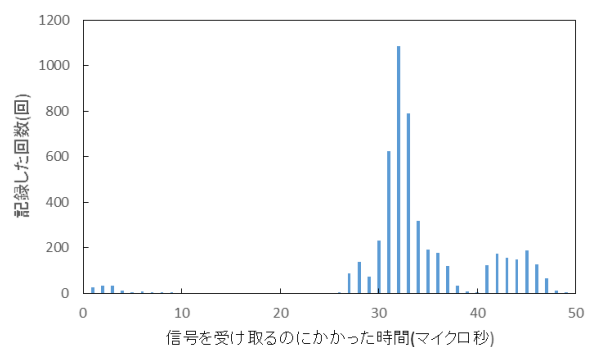


図4：mrub/c の評価結果

6. 結言

TOPPERS OS と mruby/c を使用し、マルチタスクでのモータ制御に成功した。それらを比較、評価し、信号検知時間および、計測の揺らぎを測定することができた。

評価の結果、マルチタスク処理をしたい場合、リアルタイム性が重要になってくる場面では TOPPERS OS が適している。リアルタイム性があまり重要でなく、開発効率や生産性の高さが求められる場面では mruby/c の利用が期待される。