組み込み向け軽量 Ruby の研究・開発 ～TJ ボードを使った CPU とハードウェア支援について～

佐藤雄亮†

組み込みソフトウェアは私達の生活に深く密着しており、安全性も含め、高品質が要求され、その開発は容易ではない。一方、組み込みソフトウェア開発の多くは C、C＋＋言語が用いられているが、習得が困難であったり、コードが長く生産性が低いという問題がある。
そこで、Ruby という日本製の開発言語は、手軽で記述しやすく C 言語のソースを簡単に利用できるという特徴を持つ広く普及しているスクリプト言語であり、その結果、高生産性、高品質なプログラム開発が可能である。
本研究ではこの Ruby を組み込みソフトウェア開発に使用できるように簡略化、軽量化すると同時に、チップ化して使用できるかどうかなどの研究を行なっている。

Research and development of lightweight Ruby for embedded system ~ About CPU using TJ board, and hardware assists~

Yusuke Sato

Embedded software is deeply close to our life, high quality also including safety is required, and the development is not easy. On the other hand, many of the embedded software development has been used C and C + + language. but there is a problem that difficult to learn and low productivity more code Ruby development language that is made in Japan, which is widely used scripting language. It has features that you can easily use the C language source and write code. As a result, program development is capable of high quality high productivity We study it which developers can use Ruby for embedded software development.
Specifically, we performed a simplified, lightweight, and chipping of Ruby.

1. 背景・研究目的

現在、組み込み機器の適用場所は様々な分野は様々な分野に広がっており、その全体の傾向としてプログラムの複雑化、開発の短期間化、低コスト化が求められている。そこで、現在開発言語として多く使われている C、C++言語に比べ、コードが短い、記述が容易、生産性が高い、高品質、という特徴を持つ Ruby 言語を用いることによって複雑化、短期間化、低コスト化に対し C、C++言語で開発するよりも開発効率を上げることができる。しかし、Ruby 言語は、VM が 50MB もありメモリ消費も大きく、組み込み機器には適していない。

そこで本研究では、組み込み機器でも Ruby 言語を使用できるように RubyVM の軽量化（以下 RubyVM）、中間表現を使用し組み込み機器に適用させるように研究開発を行う。

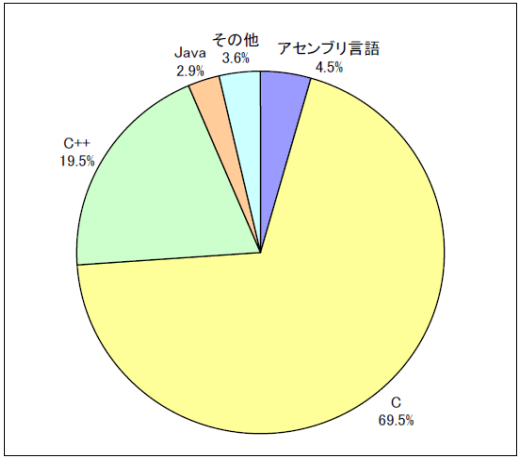

図 1　2010 年度組み込み開発で使用されるプログラミング言語（記述行数比率）
Figure 1　Programming languages which are used to embedded development in FY 2010 (ratio of the number of lines written)

† 九州工業大学大学院
Kyushu Institute of Technology Graduate School of Information Systems Department

2. 研究内容

本研究の最終目標は、軽量 RubyVM のチップ化が最終目的である。その前段階として FPGA に実装することによって最終的にチップ化できるかの検証を行う。

具体的には、実際に FPGA が動作させるための環境開発や軽量 Ruby が、より快適に利用できるように RTOS を実装・動作を行う。

3. システム構成

3.1 Ruby

Ruby とは、まつもとゆきひろ氏が開発した日本産のマルチプラットフォームな、オブジェクト指向スクリプト言語である。Web アプリケーションフレームワークである Ruby on Rails によって大ブレイクした。

Ruby の主な特徴としては、以下のことが挙げられる。

・インタプリタ言語のため、コンパイルの必要がない。
・変数に型がなく、変数宣言が必要ないので、シンプルに文法を書くことが可能。
・オブジェクト指向なので、クラス、継承、メソッド、Mix-in 等の機能、概念が存在。
・正規表現や例外処理の機能が存在。
・C プログラムやライブラリを呼び出す拡張モジュールを組み込むことが可能。
・フリーソフトウェアである。

3.2 軽量 Ruby システム機能

軽量 Ruby は、既存の Ruby の長所を残し組み込みシステム開発向けに軽量化、最適化を行ったプログラム言語である。機能としては、既存の Ruby との違い ISO/JIS 規格で定められた機能、組み込みライブラリ部分になっている。よって、一般的な Ruby に比べ、組み込みに必要な機能を絞り込んでいる。

また、ライブラリ内の機能の縮小を行う必要に応じてミニマルライブラリ、スタンダードライブラリ、フルライブラリを使い分ける。

軽量 Ruby の機能範囲を図解したものを以下に示す

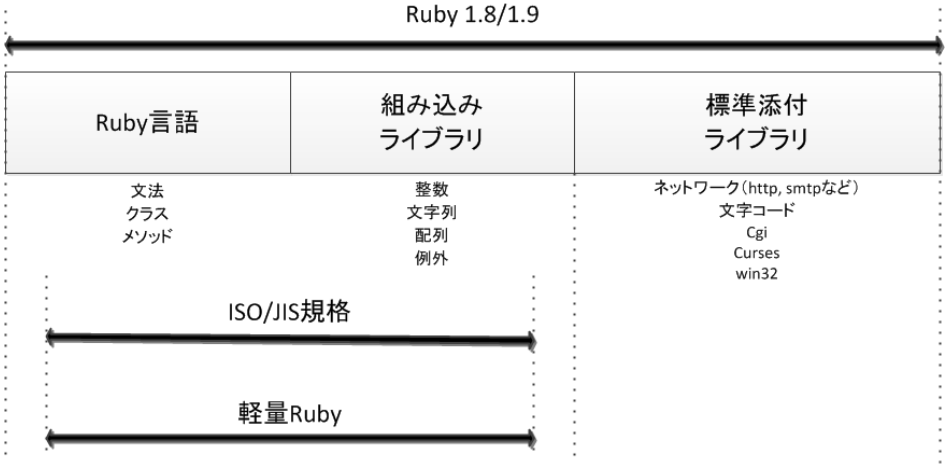

図 2　軽量 Ruby の機能範囲
Figure 2　The function of light Ruby

3.3 軽量 Ruby の動作方法

軽量 Ruby の動作は以下のようになっている。

1、 開発環境（Windows、Linux など）上で Ruby ソースファイルを専用のコンパイラを用いて、Rite バイナリファイルに変換する。
2、 実行環境上に Rite バイナリファイルとライブラリ、RiteVM など動作に必要なファイルを追加する。
3、 実行環境上で、RiteVM を用いて Rite 中間表現を実行する。

また、軽量 Ruby コンパイラの機能として、Ruby ソースコードを C 言語に変換することもできる。その場合は、変換した後のコードを一般的な C 言語での開発方法とほぼ同じ方法で開発することができる。

基本的な流れとして Ruby ソースファイルのみを使用した軽量 Ruby の開発方法を以下に示す。

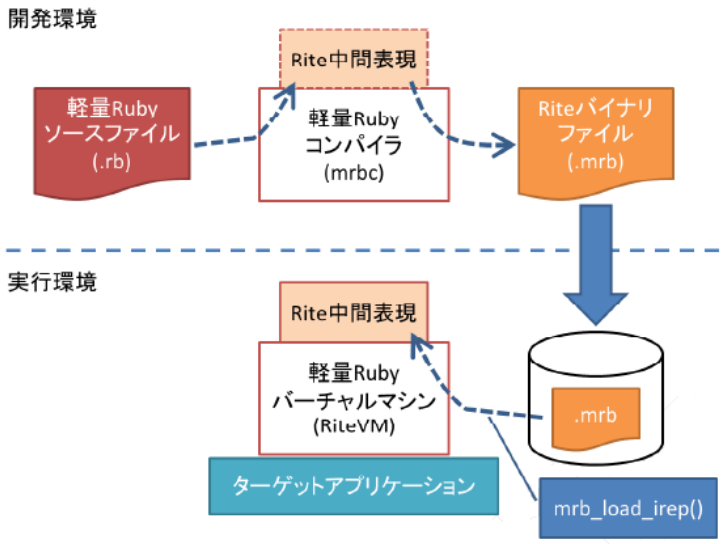

図３　軽量 Ruby の動作方法
Figure 3　How to use light Ruby

4.　実装

軽量 Ruby をチップ化させる前段階として軽量 Ruby が FPGA ボードで動作することが出来るか、2 種類の FPGA ボードで検証を行い動作させた。
また FPGA の開発に使用したソフトウェアは以下の３つである

Altera SOPC Builder 10.1 Build 153	ＣＰＵの設定
Quartus Ⅱ　10.1　Web　Edition	ＰＩＮ配置、ＣＰＵ書き込み
Nios II 10.1 Software Build Tools	Ｃ言語のＩＤＥ、プログラムの書き込み

4.1　FPGA

FPGA とは、**F**ield **P**rogrammable **G**ate **A**rray の頭文字をとったもので、現場（Field）で、書き換え可能（programmable、プログラム可能な）、LSI（論理ゲート（Gate）が格子（Array）状に並んでいるセミカスタム LSI）のことであり、後からでも回路の書き換えが可能なロジックデバイスである。

FPGA の特徴としては、以下のような特徴がある。

・高速、小型、軽量、低消費電力。
・小さなロジックモジュールを組み合わせて設計ができる為、設計の自由度が高い。
・PC 上で設計し、PC から FPGA に実装できるため開発期間が短い。
・設計の自由度の高さから、任意の論理回路の実現。
・多入力、多出力の並列処理が可能。
・プログラムを何度でも書き直すことができるため、柔軟に回路の変更が可能。
・実装した回路の速度やゲートの消費効率などの回路の速度が設計者に依存。

4.2　評価用 FPGA ボード

ALTERA 社の cyclone Ⅱ EP2C20F484C7N という一般的市販されている開発学習 FPGA ボードを用い実装を行った。

メインプロセッサ	AlteraCycloneII
CPU	NIOSII　: 50MHz
メモリ	FlashROM : 4MBytes

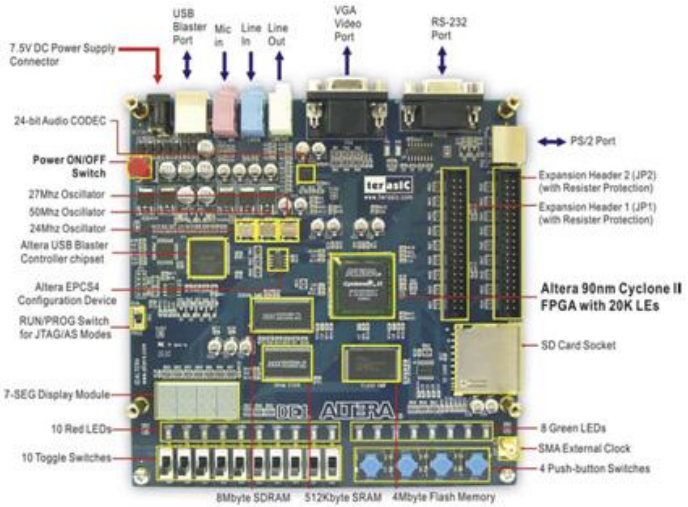

図４　　一般的な評価用 FPGA ボード EP2C20F484C7N
Figure 4　FPGA evaluation board for the general

4.3 軽量 Ruby 評価用 FPGA ボード

軽量 Ruby の評価をするために制作された軽量 Ruby 評価用 FPGA ボードで軽量 Ruby の実装を行った。図 4.1 の一般的な評価用ボードと基本的に変化はないが、メインプロセッサ、メモリ等がより商用に変更されている。

メインプロセッサ	AlteraCycloneIII
CPU	NIOSII : 50MHz
メモリ	SDRAM :512Kbytes
	SDRAM : 32MBytes
	FlashROM : 64MBytes

図 5　軽量 Ruby 評価用 FPGA ボード
Figure 5　FPGA Evaluation board of light Ruby

5. OS の実装と結果

この軽量 Ruby システムは、様々なタスク処理を行う、大規模な組み込み機器を対象としている。よって、タスク処理を行う RTOS の実装が不可欠であるため実装を行った。本研究では、軽量 Ruby 評価用 FPGA ボードに toppers と uClinux 両方の実装を試みた。

5.1 RTOS について

RTOS（リアルタイム OS）とは、一般的な OS に比べ時間資源の保護及び実行時間の予測に特化したものである。様々なタスクの処理をするスケジューリング等を行う。

5.2 uClinux について

uClinux は、小規模の組み込み機器に適した Linux であり、組み込み用 RTOS の中でも人気の高い RTOS である。

今回、軽量 Ruby 評価用 FPGA ボードを用い実装を行ったが、カーネルブート中に止まるというエラーが発生した。

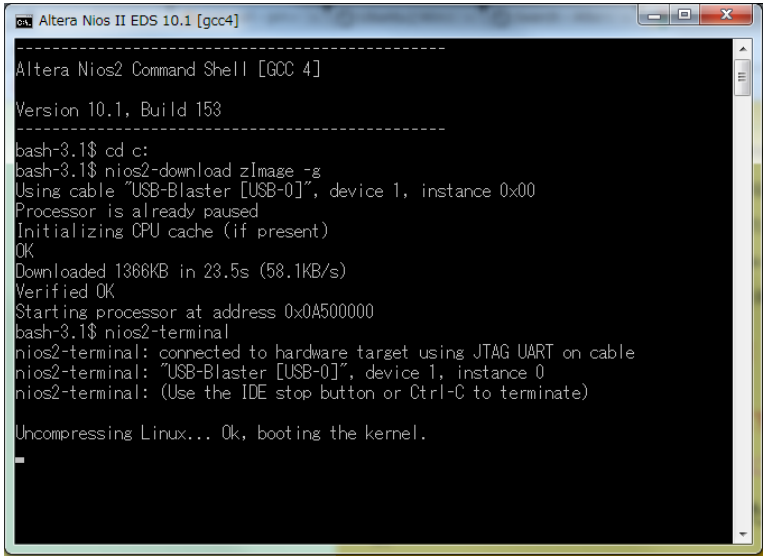

図 6　軽量 Ruby 評価用 FPGA ボードにおける uClinux のカーネル読み込みエラー画面
Figure 6　uClinux kernel read error screen with FPGA Evaluation board of light Ruby

5.2.1 評価用ボードを用いた場合について

図 6 のエラー発生の原因を調べるために図 4 の一般的な評価用 FPGA ボードを用い uClinux を実装し動作させた。

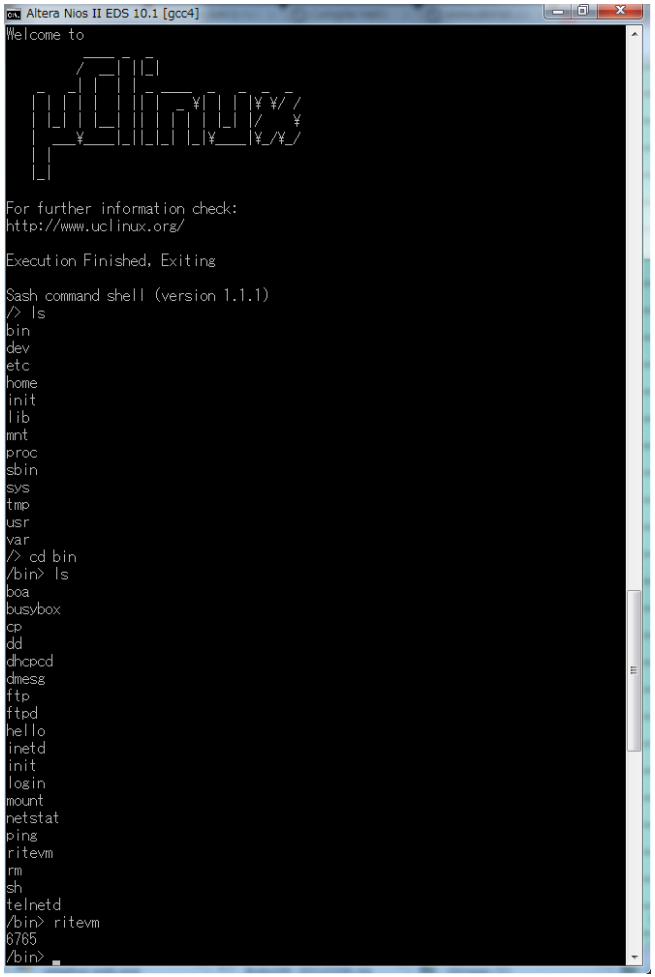

図 7　評価用 FPGA ボードにおける uClinux の動作画面
Figure 7　Operation of the toppers screen with FPGA Evaluation board

5.3　toppers について

toppers は、uClinux よりも更に小規模の組み込み機器に適した TRON 系の RTOS である。国産ということもあり、国内では人気の高い RTOS である。

軽量 Ruby 評価用 FPGA ボードを用い実装を行い動作したものを以下に示す。

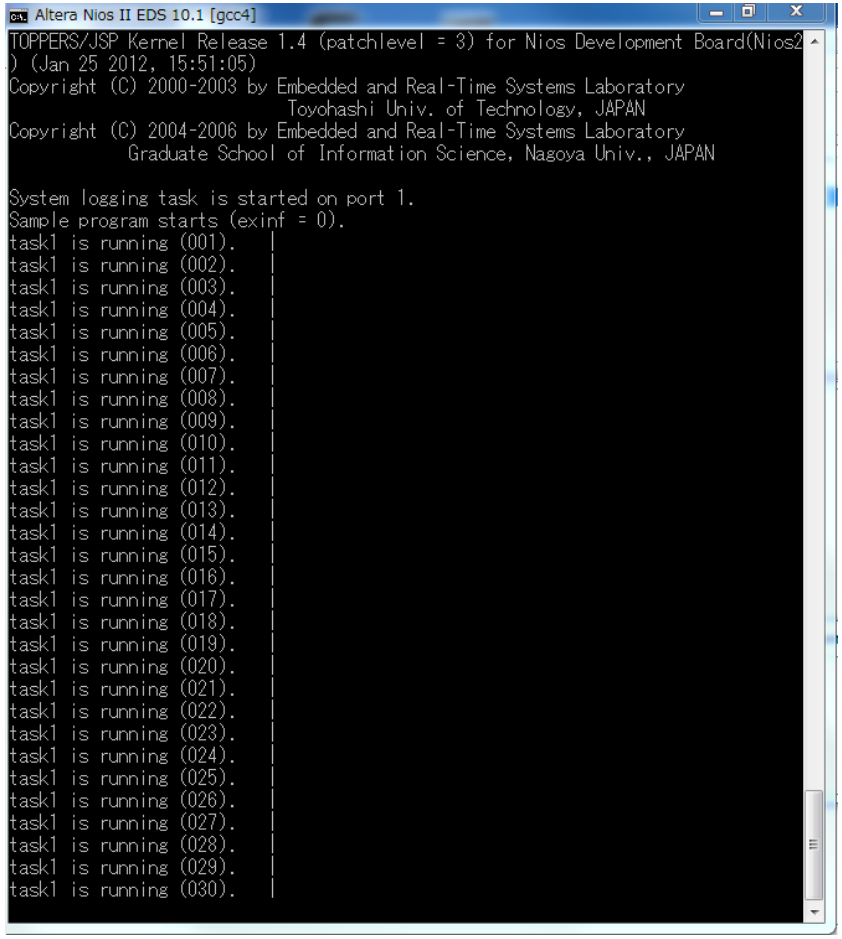

図 8　軽量 Ruby 評価用 FPGA ボードにおける toppers の動作画面
Figure 8　Operation of the toppers screen with FPGA Evaluation board of light Ruby

6. 考察と今後の展望

今回、２種類の RTOS の実装を行い toppers は動作させることができたものの uClinux を動作させることが出来なかった。

原因は現在調査中であるが、可能性として考えられることは、toppers が FPGA の CPU 設定に依存しないことに対して、uClinux は FPGA の CPU 設定に大きく依存することが考えられる。なので、その辺りをもっと調べ改善することによって、軽量 Ruby を使用する範囲を広げ、更に使いやすいプラットホームを開発することができるので、解決すべき問題点であると考えている。また、軽量 Ruby 評価用 FPGA ボードに toppers を実装した際にまだ OS しか実装しておらず軽量 RubyVM 及びプログラムを実装していないので、実装することも、今後の課題である。

参考文献

１）経済産業省　2010 年版組み込みソフトウェア産業実態調査
http://www.meti.go.jp/policy/mono_info_service/joho/downloadfiles/2010software_research/index.htm