B-012

開発効率向上のためのスクリプト言語 mruby/c を用いたロボット制御
Robot Control by Scripting Language mruby/c for Rapid Development

福田 尚毅† 田中 和明†
Naoki Fukuda Kazuaki Tanaka

1. 序文

多くの産業用ロボットが世界中に普及し、それらはより高速かつ正確な動作を行うことが求められている。しかしながら、それらの開発に使用されるロボット言語に関する研究は未発達であり、ソフトウェアの技術もまた同様である。その理由はロボット制御のプログラムに C 言語が使用されていることにある。この言語の使用により以下のようなデメリットがあげられる。

- コード行数の増加と複雑なコード構造による開発工数の増加
- ポインタ操作やメモリ管理など言語特有の構造理解の困難さ
- ロボット制御を行うために必要な RTOS に関する知識がなければコード記述が不可能
- 新型ロボットの開発に伴う開発環境の変更による時間と労力の増加

以上の問題点はロボット開発だけでなく、あらゆる組込みソフトウェア開発においても同様である。そこで、このような問題点を解決するために mruby/c が開発された。mruby/c は組込みソフトウェアの一種であるため、産業用ロボットソフトウェアの開発で生じるあらゆる問題を解決することができると考えられる。

2. ロボット制御のためのスクリプト言語

2.1 MotoPlus

MotoPlus(Motoman Professional Programming Language for superior Use)は自身の PC で C 言語を使ってアプリケーションを作成し、ロボットコントローラのタスクとして実行するための専門プログラマー用 IDE である。本研究においては C 言語の代わりにスクリプト言語 mruby/c を使用可能にするために開発を行った。

2.2 MotoMINI

本研究で使用したマニピュレータを右図に示す(図 1)。MotoMINI は安川電機によって生産開発されている業界最小、最軽量(2017 年 6 月時点)の 6 軸垂直多関節ロボットである。重量は約 7kg で、人間でも容易に本体の運搬が可能である[1]。小型であるため主に電気・電子部品などの小物物品の搬送, 組立や加工機など装置内での小型物品の組立で使用されている。MotoPlus で作成したアプリケーションプログラムを専用コントローラである YRC1000micro にアプリケーションモジュールとしてインストールすればコントローラが MotoMINI に命令を出し、動作させることができる。

図 3 MotoMINI

2.3 mruby/c

mruby/c はワンチップマイコン等の小さいメモリ容量で使用できるように実装した言語である。使用するには、まず mruby という Ruby を組み込み向けにメモリ量を削減した言語を理解する必要がある[2]。mruby は以下のように実装されている(図 2)。

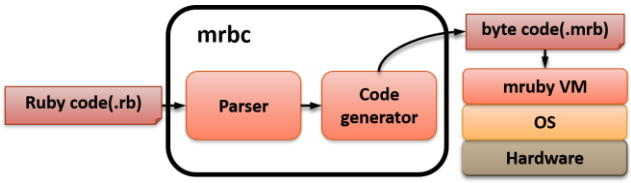

図 1 mruby

Ruby コードを自身の PC で作成し、そのスクリプトをパーサに受け渡し、そこで生成した構文木からバイトコード生成部分に受け渡し、mruby VM がバイトコードを実行するという流れだ。mruby/c はこの特徴を生かし、以下のように開発されている(図 3)。

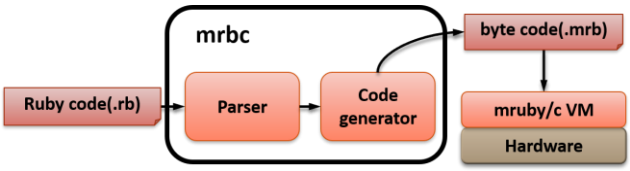

図 2 mruby/c

図の通り、バイトコードの生成までは mruby の特徴と同じで、バイトコードの実行部分である VM のみ置き換えて実装されている[3]。この mruby/c VM は OS のないような小型のハードウェアにも対応できるように mruby VM よりも少ないメモリ量で実装されている。

本研究ではメモリ不足にも困らず、実装がミニマムであるためこのスクリプト言語 mruby/c を採用した。ロボット制御を行うエンジニアは mruby/c の記述だけでロボット制御が可能となる。

3. MotoMINI 専用 mruby/c VM の開発

3.1 RTOS 内部の mruby/c VM

IDE である MotoPlus においてコンパイルができるように、C 言語特有の記述方法に変更を加え、ビルドを行った。その後生成したアプリケーションをコントローラにロードする際に RTOS に対応させられるように使用不可能な関数を

†九州工業大学 Kyushu Institute of Technology

制限した。 以下にコントローラ内部の開発に関する図を示す(図 4)。

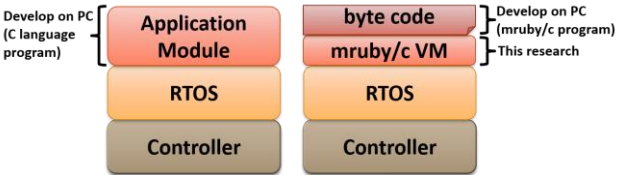

図 4　コントローラ内部

この図から分かるように従来の開発方法においては MotoPlus で開発した C 言語プログラムをアプリケーションモジュールとしてコントローラにインストールしていた。さらに、開発時に RTOS の知識がないとプログラムの記述が困難な場合もある。そのような場合、専門のプログラマーによるサポートが必要となり、時間や費用がかかってしまうという問題がある。そこで、本研究では mruby/c VM をあらかじめインストールしておくことで mruby/c（図における byte code の部分）の記述のみで開発可能となるため、技術者にとっては開発コストの大幅な削減が可能になる。また、C 言語に比べ、可読性、生産性が高いので現場で働く技術者でも簡単にロボット制御をすることができると考えている。

3.2 MotoMINI の特性

MotoPlus は MotoMINI を任意の場所に移動させる関数が用意されているが、mruby/c は汎用的組込みソフトウェアであるため、MotoMINI のようなロボットを制御できるような専用機能は搭載されていない。そこで C 言語で記述された MotoPlus の関数と mruby/c の紐づけを行い、mruby/c で記述したプログラムにより MotoMINI の動作を可能にする。

4. 実装と結果

mruby/c VM の中身は C 言語プログラムである。そのため C 言語コンパイラの機能を持つ Motoplus で対応することができる。ただし、コンパイル時にエラーが発生した。そこで以下の通り対応を行った。

- mruby/c VM は C99 に準拠した C 言語プログラムであるが、MotoPlus は ANSI に準拠した C 言語プログラムしか実行できない。そこで、コンパイル時に”-fms-extensions -std=c99”というコンパイルオプションを追加することで MotoPlus は C99 準拠の C 言語プログラムに対応することを可能にした。
- C99 準拠のプログラムで使用可能なメソッドとして「指示付き初期化子」が mruby/c VM のプログラム内で使用されている。MotoPlus は ANSI 準拠の C 言語プログラムしか読み取ることができないので、mruby/c VM のプログラムを読み取ることができるように初期化の書き方を改良した。
- MotoPlus では"stdio.h"や"stdlib.h"といった C 言語標準ライブラリと MotoPlus 専用のライブラリが提供されている。また、これらのライブラリで構成された"motoPlus.h"という専用ライブラリを宣言しなければ、C 言語プログラムをコンパイルすることができない。

作成したプログラムは MotoPlus のコンパイラでビルドが可能となり、コントローラにロードするためのアプリケーションの生成に成功した。

ロボットコントローラにもともとインストールされている RTOS が mruby/c を実行することができるように mruby/c VM のプログラム内記述を以下のように書き換えた。

- RTOS 専用の HAL とよばれるプログラムを作成した。mruby/c ではもともと、ハードウェアごとの HAL が用意されており、ハードウェア依存部を書き換えた。
- Motoplus では int や uint8_t などを motoplus.h 内で独自の名称で定義している。そのため、メモリアクセスのためのポインタのやり取りや配列ポインタの受け渡しの際に定義する型を変更した。

この改良によりコントローラに mruby/c VM をロードすることができ、RTOS 内にインストールされた。よって、Ruby による記述のみで MotoMINI が動作可能となった。

mruby/c にはもともとロボットを動作させる関数は用意されていないので、MotoPlus で用意されている関数を組み合わせることによって、新しい関数を作成した。以下にその例を示す。

表 1　ロボット制御関数

関数名	引数	用途
movj_x	v[mm/sec], d[mm]	X 軸方向に動作
movl_y	Type, v[mm/sec], dist[mm]	Y 軸方向に動作

そして、これらの関数を用いて Ruby の記述方法で以下のようなサンプルプログラムを作成した。

```
class MotoMINI
    #各メソッドを記述
end

robot1 = MotoMINI.new(0)

power 1
while true
    robot1.init()
    robot1.movj_x(6.25,70.0)
    robot1.movj_yz(12.50,-120.0,-125.0)
    robot1.movl_y(1,187.0,240.0)
    robot1.movj_yz(25.00,-120.0,125.0)
    robot1.start(2,3)
end
```

ソースコード 1　Ruby サンプルプログラム

以上の通り、Ruby コードを記述し、mruby/c のバイトコードを生成するだけで、プログラマーにとって容易なコーディングを可能にした。

5. 結論

mruby/c の記述だけで産業用ロボット MotoMINI の動作を可能にした。ロボット言語として C 言語ではなく、生産性、可読性の高いスクリプト言語 mruby/c を使用することでロボット開発の効率向上の足掛かりになることを期待している。

産業用ロボットを制御するためのプログラム作成として C 言語でプログラミングが行われることが多く、RTOS に

関する専門的な知識も必要となり、技術の未熟なエンジニアにとっては読み書きが困難である。今後 C 言語で可能なことはすべて mruby/c を使用し容易なコーディングを可能にする。

謝辞

本研究を完了するにあたり、産業用ロボット MotoMINI は、安川電機株式会社から学術支援プログラムの下で提供していただき、公式の安全トレーニングドキュメントとソフトウェア開発環境もサポートしていただいた。

参考文献

[1] 仕様 – MotoMINI – ハンドリング – ロボット - 製品情報 - トップページ
http://www.e-mechatronics.com/product/robot/assembly/lineup/motomini/spec.html
[2] http://www.mruby.org/
[3] http://github.com/mrubyc/mrubyc