

ロボットを用いた組み込みシステム学習環境の開発

川上 亮太郎[†] 田中 裕樹[‡] 杉田 圭祐[‡] 早川 栄一[‡]

概要

本研究は、サッカーロボットをベースとして、OS や制御アプリケーションなどのプログラム群を学習教材として提供することで、単一の環境下で組み込みシステムの学習を可能にするものである。特に、教材となるプログラムのサイズを小さく抑え、単純な仕様にする事で全体の動作を把握しやすくする。かつ、プログラムを読み理解できるようにすることで、学習者の学習に対する負担を軽減する。教材としてロボットハードウェア、OS、JavaVM を一つのシステムとして提供する。また、ロボットの動作とシステムの各層との対応を容易に取ることを可能にしデバッグの支援を行う環境を提供する。

キーワード ロボット 組み込みシステム 学習教材

Development of Educational Environment for Embedded System with Robot

Ryotaro KAWAKAMI[†] Yuuki TANAKA[‡] Keisuke SUGITA[‡] Eiichi HAYAKAWA[‡]

Abstract

This research enables the study of the embedded system to use a soccer robot. Especially, it make a small program size and simple specification to easy understand, to understand a program of the robot system, a learner can read source code of whole system. It is offered as a system that include soccer robot, OS and JavaVM. Moreover, It offered an environment to support debugging to easy understand the relation of movement of the robot and each layer of system.

Keywords Robot Embedded system Learning material

1 はじめに

計算機科学においてオペレーティングシステム(以下 OS)やネットワーク等は概念や実装を理解すべき重要なものである。特に実装の技術は本を読むだけでは得られない経験が必要な部分であり、大学などでは演習という形で教育を行っている。これまで、OS 等、規模の大きなシステムの実装学習には、教育用に開発されたものが用いられてきた。しかし、組み込みシステムが注目され、その教育が重要

となっている現在、既存の教育用システムでは組み込みシステムに必要な機能が存在しない等の理由から、

組み込みシステムの学習用としては不十分であったため新たに組み込みシステム学習用の環境が必要となった。

一般に組み込みシステムは特定の目的を達成するための専用システムとして開発される。そのシステムには OS が無い場合もある。学習者が技術者としてシステムを開発するようになったとき、ブートストラップやタスクのディスパッチ、バケット送受信といった低レベルなものを理解することで、単体のソフトウェアとして動作する高機能なソフトウェアの開発が可能になる。また、システムの機能がどのように連携しているのかを理解することで、必要な機能のみを選択、構

[†] 拓殖大学大学院工学研究科

Graduate School of Engineering, Takushoku University

[‡] 拓殖大学工学部

Faculty of Engineering, Takushoku University

築することが可能になる。そのため組込みシステム学習は重要なものである。

組込みシステム学習環境では、これまでの教育用システムでは動作に必須だった機能の中で、組込みシステムの視点からは不要になるものを排除する。それによりプログラムのサイズを小さく抑え、学習者がシステム全体を把握しやすくなることが期待できる。

また、多くの組込みシステムは、システム外部の状況や時間、タイミングに処理が大きく左右されるような厳しい要求のある、再現性の低い状態に置かれる。これはエミュレーションなどでは再現しにくい部分であり、実ハードウェアを用いることでそれらを学習する必要がある。しかし、組込みボード単体では作業の確認しにくくなるため目的を見失いやすい、デバッグがしにくいという問題がある。

そこで、我々はロボットを実ハードウェアとして用いることにした。ロボットを用いて

学習を行わせることで、組込みボード単体で学習を行うよりも目的がはっきりする、意欲的に学習に取り組むことが期待できる、動作を目で確かめることができるため、デバッグがしやすくなるという利点が存在する。

そこで、本研究ではロボットを用いて、複数の実装学習を行うための教材と、学習環境の開発を併せて行う。

2 問題分析

既存のシステム学習環境に存在する問題について述べる。

(1) プログラムサイズが大きい

ソフトウェアは実用的になるにつれ、エラー処理などが多くなり、プログラムサイズが大きくなる。その結果、学習者が全体を把握するのに時間がかかるので、短期間で学習を行う必要のある学校教育には向いていない。そのため、大規模なシステムの学習には、実用的なプログラムではなく、教育用に実装されたプログラムが教材として用いられる。MINIX^[1]をはじめとした教育用ソフトウェアはターゲットアーキテクチャに x86 を選択していることが多い。これは普及率が最も高いためである。しかし、ハードウェアの構成が複雑でプログラムサイズが大きいという問題がある。

(2) ブラックボックスの存在

教材に、学習者にまったく見せない部分があると仮定した場合、学習者はその部分を理解しないままに学習を終わらせてしまい、全体を理解する機会を逃すことになる。学習者が将来、技術者として組込みシステムの開発をする場合、必要な機能の選別や実装を行うことが考えられる。その際、PC 上でのアプリケーション開発とは異なり、自力で起動、動作を行い、かつ、低水準なインタフェースしかもたないハードウェアの制御を行うプログラムを開発する必要に迫られる可能性がある。そのため、組込みシステム全体の把握からハードウェア操作の仕方まで、広範囲に渡っての理解が重要になる。全体の理解が重要な組込みシステムの場合それでは不十分である。

(3) デバッグが困難

組込みシステムにはモニタなど文字を出力するデバイスがないことが多い。そのような状況でも動作中のシステムの変数値などを知るために LED を点滅させたりする。しかし、それでは複雑なことはできず外部に情報を示すことが困難である。動作が正常に行われているかを知る手がかりが少なく、デバッグの効率も悪いと言える。実装の学習の多くの時間がデバッグ作業に費やされることが考えられる。組込みでは再現性の低いバグも発生しやすいため、デバッグのし易さや効率の向上は実装学習を行う上で重要な要素である。

(4) オブジェクト指向言語が未サポート

今後、組込みの分野でもシステムの規模が大きくなるにつれ、開発で用いられることが多くなることが予想される。そのため組込みシステムにおけるオブジェクト指向言語の学習は重要である。しかし、教材として、学習環境がサポートしていないことが多い。これは学習者に組込みシステムを対象としたオブジェクト指向言語の適用に関する考察や実装学習する機会が与えられていないということの意味する。

3 設計方針

上述した諸問題を解決し、組込みシステムの実装学習を円滑に行えるようにするため、設計方針に次の五つを挙げる。

(1) 必須な機能だけ実装した教材の提供

ここでいう必須な機能とは実際に動作する最小限の機能のことである。

各教材の仕様を単純にし、全体のコードサイズを小

さく抑えることで学習者が教材を理解することを容易にする。

(2) すべてのプログラムの提供

システムすべてのプログラムを提供する。それにより学習者はすべてのプログラムを読むことができるので、システムの各機能の実装と、それらの連携の取り方を理解することができる。また、コードを順に追っていくことで処理の中心部分に必ず行き着くため、実装を理解することができる。

これは(1)で述べた設計方針とともに、学習者がシステム全体の把握と理解することに役立つ。

(3) デバッグ環境の提供

学習中はデバッグが中心作業の一つになる。しかし組込みシステムはデバッグが難しい。PC用のアプリケーションであれば変数などの値を画面に表示すれば内部状態がわかり、デバッグすることができる。しかし、組込みシステムでは表示デバイスがないことも多く、また、リアルタイムで動作することが多いため表示が追いつかない、という問題が存在する。そこで、内部状態をログとして取り、PC側でそれを解析することでデバッグを支援する環境を提供する。これでデバッグ時にデータや動作の確認や、デバッグ自体が困難な再現性の低いバグを何度でも再現し確認できるようにする。

(4) オブジェクト指向言語でアプリケーション作成

ソフトウェアの規模が大きくなると、開発言語にオブジェクト指向言語が採用されることが考えられる。そこで学習環境もオブジェクト指向言語のサポートを行う。

(5) 複数の学習テーマに用いることのできる単一の環境の提供

学習者が複数の学習テーマの教材として単一システムを用いる場合、一度学習したことが他の学習の際にも利用できる。システム学習ではアーキテクチャの情報が重要であるが、実装時に必要となるだけで、学習の中心になるわけではない。実装にあてる時間が増えるため学習効率が上がることが期待できる。

4 学習の教材

本システムでは学部生を対象として、複数の講義用教材となるシステムの開発を目的としている。

本システムにより、一つのプラットフォームで複数の実装学習を行うことが可能になる。組込みプログラ

ミングではアーキテクチャを理解することも重要であるが、それは学習の準備である。プラットフォームが一つであればアーキテクチャの学習が一度だけで済み、残りの時間をソフトウェアの学習にあてることが可能になる。さらに、一つの演習の中で得た知識が他の演習のときにも役立ち、最終的に全体像が掴めるようになることが期待できる。

学習対象となる教材を次に挙げる。

(1) システム全体

組込みOSのサポートを受けてアプリケーションを作成することは、現在では携帯電話、ロボットなど、様々なところで行われている。本システム上に、ユーザプログラムとして、ロボットの制御用アプリケーションをロードし動作させる。学習者に制御アプリケーションを作成させることで組込みシステムの学習を行わせる。これは組込みシステムを学ぶ最初のステップとして位置付けることができる。時間や動作の不安定さ、それを踏まえた上での柔軟なプログラムの記述方法などを学習させる。

(2) OS

OSは組込みシステムではアプリケーションと近い関係にある。単体のアプリケーションをハードウェア上で動作させる場合、OSのブートストラップ、メモリ管理など、OSの機能をアプリケーションが自分で持っている必要がある。組込みシステムではOSの機能の一部をアプリケーションが含みやすいといえる。したがってOSの学習は組込みシステムを学ぶものにとっては必須である。

提供された組込みOSのプログラムを読む、拡張するなどの演習を通して組込みOSの実装を学習する。

学習者へ提供されるOSは最小限の機能を持つコード量の少ないものである。これを理解することで組込みOSの実装を学ぶことができる。また、拡張作業や新機能の追加といった課題を通して理解を深めることが期待できる。

(3) ネットワーク

組込みシステムの多くは少資源である。それを補うため、ネットワーク機能を持つことが多い。ネットワークの実装には、HTTPをはじめとした、プロトコルの使用をとおしてのネットワークプログラミング学習と、システムとしてネットワークはどのように実装されているのかを理解するためのプロトコルスタックの実装についての学習がある。どちらも組込みシステム

では多く用いられる重要な技術であり、これを学習する必要がある。

本システムではプロトコルスタックの実装を提供することで、それを用いてのネットワークプログラミングや、プロトコルの追加、拡張を行うことが可能になる。

(4) オブジェクト指向言語

本システムにある仮想マシンを用いて演習を行う。

C言語のような他の言語と、オブジェクト指向言語での開発効率の差や、この言語が組み込みシステムに向いているかどうかなどといった考察、実際にプログラムを作成する経験を学習者は得ることができる。

5 教材を用いた学習

(1) 段階的な学習

本システムを用いて段階的にシステム全体を学習することが可能になる。一例として、学部二年でシステムプログラミング学習を行い制御プログラミングを学ぶ。次に学部三年でOSやネットワーク学習を行うことで、システムの下層はどのように動作しているのか、システム全体としてどのように動作しているのかを学習するという流れを持った学習作ることができる。

(2) 課題の提案

我々は本システムを用いた課題の一つとしてサッカーを提案する。サッカーでは周囲の状況を把握する必要があり、また、プレイヤー間の意思の疎通が重要である。これらをロボットで実現するためにはセンサやモータデバイスの制御、ネットワーク機能の使用、といった組み込みシステムを学習する上で重要な要素を含んでいるためである。

この課題を学習者に与えることで、目的を強く持ちながら学習することが期待できる。

6 全体構成

本システムでは組み込みボードにCAT709^[2]を、ロボットにサッカーロボ915^[3]を用い開発を行う。また、仮想マシンのベースとしてWabaVM^[4]を用いる。

組み込みボードは、教材として複数の実装学習に利用されることを考慮し、組み込みシステムとしては処理能力が高く、メモリ容量の大きいもの、かつ、OSの講義で使用されることを考慮し、仮想記憶が使用できるものを選択した。

組み込みハードウェアにはサッカーロボットを用いる。

サッカーでは外部状況を判断するためセンサが用いられる。これは組み込みシステムの最も基本的なものであり、これを用いることで組み込みシステムの基本を学ぶことができる。

WabaVMはJavaVMのサブセットであるため、完全にはバイトコードを理解しない。しかし、実装は小さく単純な構成になっているため、仮想マシンとして採用した。これは主にオブジェクト指向言語の習得という目的で採用しているが、学習者がコード全体を把握できる規模で実装されているため、このコードも提供する。しかし、WabaVMにはスレッドや例外といった機能がない。それらはプログラムの記述性を高めるために必要なので拡張を行う。

全体構成を次の図1に示す。我々が試作したロボットを図2に示す。

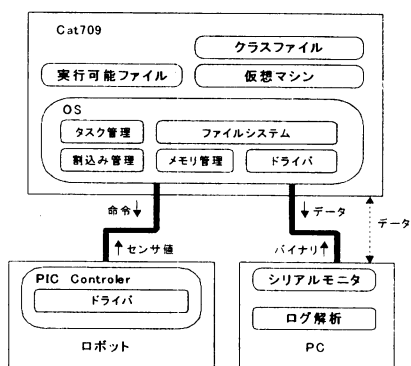


図1 全体構成

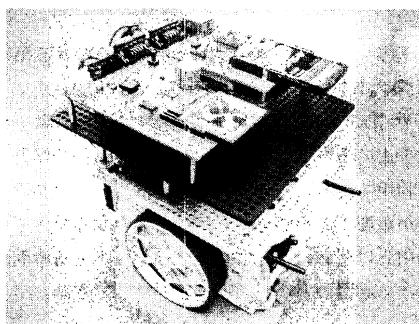


図2 試作ロボット

次に各部の設計を説明する。

6.1 OS

教育用組込み OS の設計について述べる。

(1) タスク管理

OS にはマルチタスク機能を持たせる。

タスクスイッチの実装は理解しにくい部分である。実装例を学習者に提供することでコードを読みながら学習することが可能になる。

また、本研究ではロボットを扱うが、このようなハードウェアは PC 等のシステムよりも時間の正確性が求められることが多いことを考慮し、リアルタイムスケジューラを用いる。

(2) メモリ管理

メモリ管理には Buddy 等、幾つかのアルゴリズムが存在する。しかし、組込み用である、コード量を抑えたい、動的にメモリを取得することがあまりない、という理由から複雑なものは不要であると考え、最低限必要である機能としてメモリの確保、解放の機能を作成、提供する。

(3) 割込み管理

本システムにおける割込みの機能はタイマ、シリアルである。

タイマはスケジューリング時のトリガや時間の計測など、重要な要素であり、学ぶべき機能である。通常割込み処理として記述されるこの機能は、割込みの処理の例ともなる。

シリアルは通信インターフェースとして単純なインターフェースを持つ。タイマよりは多少複雑であるが、基本的な割込みの一例として学習者へ実装を提供する。

(4) ファイルシステム

ファイルシステムには FAT16 を用いる。このファイルシステムは構造が単純であり、コード量が少なくすることが可能になる。単純なので理解することも容易である。

6.2 ネットワーク

UDP プロトコルスタックと無線 LAN を用いたネットワーク環境を提供する。ネットワークプログラミングやそれを提供するプロトコルスタックへの理解は、組込みシステムにネットワーク機能を持たせるために必要となる。そのため、モジュールとして追加可能にする。UDP プロトコルスタックは仕様単純でコードサイズも小さく実装できるため教育用に向いている。

6.3 デバッグ環境

本システムではログ解析アプリケーション^[5]を用いてログ解析を支援する手段を学習者へ提供する。

組込みシステムでは動作中の内部状態を知る手段が少ない。ロボットの動作を見ることで多くの問題を発見することは可能となるが、詳細な状況までは把握できない。そこで動作中の内部状態をログとして記録し、デバッグ時の参考データとして学習者へ提供する。ログから特定の時間におけるプログラム内で実行中の関数名、ロボットに搭載されたセンサの値が取得できる。しかし、そのままでは数字や文字の羅列にすぎず理解しにくい。そこで、それを見て理解できる形にするための解析ツールを提供する。これにより時間を追ってセンサ値の変動やそのとき実行されていた関数名が容易に判別できる。

また、組込みシステムでは再現性が低く、特定の状態が組み合わさったときのみバグが発生する、といったことも考えられ、その場合のデバッグは困難であることが考えられる。それを解決するため実行時の動画をログと同期して学習者へ見せることで再現性の低いバグを何度でも見ることが可能になる。

実行画面を図 3 に示す。

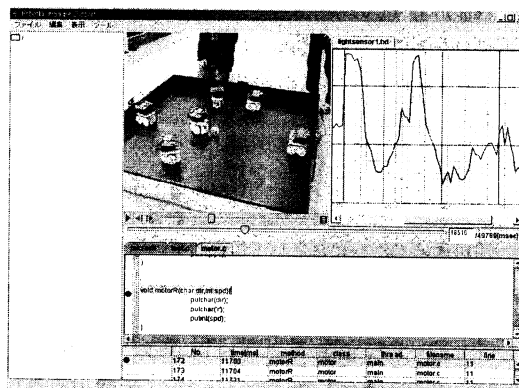


図 3 ログ解析アプリケーションの画面

7 考察

試作した OS のプログラムステップ数を表 1 に示す。また、MINIX、brickOS^[6]のカーネルのステップ数との比較を表 2 に示す。

MINIX は OS の教材として用いられることが多い。しかし、組込みシステムとして備えているべき機能がないため、組込みシステムの学習環境としては不向きである。また、組込みという視点からは余分なプログ

ラムも多く、学習者への負担を考えると全体を把握することは難しいといえる。試作 OS のステップ数は MINIX2.0 の約 18%である。全体としてプログラムサイズは小さく抑えてあるといえる。

Nachos^[7]はPC上でアプリケーションとして動作する教育用 OS である。これは小さい不完全なプログラムを学習者へ提供し、機能を作成することで実装学習を行わせる。しかし、学習者が機能の実装例を事前に学習していない場合、通常のプログラミングとは異なる部分があり、学習が困難である。また、プログラム群の構成が演習ごとに分かれるため、全体を把握しにくい。

LEGO MINDSTORM^[8]上で、C 言語で記述されたアプリケーションを実行するための OS 開発がオープンソースで行われている。現在、組込みシステム学習に多く用いられている brickOS や legOS^[9]はオープンソースで開発されている LEGO MINDSTORM 上で動作する OS である。これは小さく実装された組込み OS だが、試作 OS とは異なり教材としては作成されていない。そのためデバッグがしにくいという欠点がある。そのため学習環境としてシミュレータが開発されている^[10]。また、製品の構成から、ネットワーク機能が IrDA のみである。これは一般のネットワークプログラミングには利用されないため、ネットワークプログラミングの実装学習には用いることができない。また、

これらと比較し、我々が試作した組込みシステムはプログラムサイズを小さく抑えてあるため、全体を把握し易い、制御アプリケーションの実行環境と、教育用組込み OS を教材として提供している、かつ、デバッグ環境を整えたシステムであるといえる。

表 1 試作 OS のステップ数

機能	ステップ数
ブートストラップ	68
割り込み処理	834
メモリ管理	94
タスク管理	702
ドライバ	438
ロボット制御用ライブラリ	173
ファイルシステム	627
メイン処理その他	233
プロトコルスタック	1336
合計	3854

表 2 ステップ数の比較

OS	カーネル ステップ数
試作 OS	3854
MINIX 2.0	20798
brickOS 0.2.6	5891

8 おわりに

本報告では、ロボットを用いた組込みシステム学習環境の開発について述べた。本研究をとおしてロボットに実装する OS、WabaVM、ログの取得、解析、解析結果の表示機能と、動画との協調機能の開発を行った。今後は教育的評価を行う。

参考文献

- [1] Andrew S. Tanenbaum, "Operating System: Design and Implementation"
- [2] シリコンリナックス, <http://www.si-linux.com/index.html>
- [3] エレキット株式会社, <http://www.elekit.co.jp/index.php>
- [4] wabasoft, <http://www.wabasoft.com>
- [5] 吉田恵美 山本茂樹 西野洋介 早川栄一, ロボットプログラミング学習支援環境の開発, FIT2003 N-026
- [6] brickOS, <http://brickos.sourceforge.net/index.html>
- [7] Wayne A. Christopher, Steven J. Procter, Thomas E. Anderson "The Nachos Instructional Operating System" USENIX
- [8] LEGO MINDSTORM, <http://mindstorms.lego.com/japan/>
- [9] legOS, <http://www.noga.de/legOS/>
- [10] THOMAS ROBLITZ, OLIVER BUHN, FRANK MUELLER, "LegoSim: Simulation of Embedded Kernels over Pthread" 2002, Journal on Educational Resources in Computing
- [11] Henrik Hautop Lund, "Robot Soccer in Education" 1999, <http://legolab.daimi.au.dk/Publications/pdf/SocEdu.pdf>