

システムソフトウェア教育支援環境「港」における 教育用組込み OS の開発

川上亮太郎[†], 田中裕樹[†], 大角圭吾[†], 西野洋介[†], 川口貴弘[‡], 早川栄一[‡]

[†] 拓殖大学大学院工学研究科 〒193-0985 八王子市館町 815-1

[‡] 拓殖大学工学部 〒193-0985 八王子市館町 815-1

E-Mail : [†]kawa@os.cs.takushoku-u.ac.jp, [‡]hayakawa@cs.takushoku-u.ac.jp

あらまし

本稿では、システムソフトウェア教育支援環境「港」における組込みシステム教育支援環境の概要と、その一部である教育用組込みオペレーティングシステム(以下 OS)について述べる。

本研究は教育を指向した組込み OS を開発するものである。この OS は特徴として、コードリーディングを容易にするために小規模で、コメントを詳細に付けたコードを持つこと、「港」で開発された可視化やデバッグツールと連携することにより、動作を視覚的に見るせることのできる機能を持つことがある。また、ターゲットハードウェアにロボットを使用することで、学習者のモチベーションを高めることができる。これらを用いることで、組込み OS の学習を概念から実装までを、動作イメージを伴う形で学習することを可能にした。

キーワード： 教育 組込み OS

Development of an Instructional Embedded Operating System in System Software Educational Support Environment “MINATO”

Ryotaro KAWAKAMI[†], Yuki TANAKA[†], Keigo OOSUMI[†], Yosuke NISHINO[†],
Takahiro KAWAGUCHI[‡], Eiichi HAYAKAWA[‡]

[†] Graduate School of Engineering, Takushoku university 815-1 Tatemachi, Hachioji City, Tokyo, 193-0985 Japan

[‡] Faculty of Engineering, Takushoku university 815-1 Tatemachi, Hachioji City, Tokyo, 193-0985 Japan

E-Mail : [†]kawa@os.cs.takushoku-u.ac.jp, [‡]hayakawa@cs.takushoku-u.ac.jp

Abstract

In this paper, we describe the outline of an Embedded System Educational Support Environment in System Software Educational Support environment “MINATO” and the Instructional Embedded Operating System (OS).

We develop a small-scale OS that user is able to read codes easily, apply to codes in detail and it can use educational support environment functions. Educational support environment functions are the Visualization and the Debugging Tools in “MINATO”. This Instructional Embedded OS uses the robot to make it study enthusiastically.

Students are able to learn the concept and the programming using an Instructional Embedded OS.

Keyword : Education Embedded Operating System

1. はじめに

近年、組込みシステムが様々なところで用いられるようになっており、その需要が高まっている。

その一方で組込みシステム開発に従事している技術者のうち 80%がスキル不足を認識していると

いう報告¹⁾があり、組込み技術者の育成が重要視されている。そのような現状から、学校教育で行うべきことの一つとして組込みシステムが挙げられており、講義や演習の中に取り入れられてきている。

現在の組込みシステムは、高機能化や、開発期間

の短縮への要求もあり、OS が用いられることが多くなってきている。したがって、今後は組込みシステムについて学ぶときにはアプリケーションだけでなく組込み OS についても学ぶ必要がある。

しかし、従来の組込み OS 学習では問題として：

(1) 不適切な規模・構成の学習教材、(2) OS 内部の見えにくさという 2 点が存在した。

そこで、本研究ではこの 2 点の問題を解決した教育用組込み OS の開発を目的とする。教育用組込み OS の特徴は、学習に必要な機能だけで構成された教育用 OS と、OS の動作の中から学習に必要なデータを取得して出力する、教育支援機能と分離したところにある。それにより、学習者に学習教材として提供するコードを小さく読みやすいものにする。教育支援機能によって OS 内部のデータを出力することができる。この内部データを、早川研究室で研究が進められているシステムソフトウェア教育支援環境「港」²⁾の可視化ツール、デバッグツールと連携をとることで、学習の支援を行うことができる。

2. システムソフトウェア教育支援環境「港」

まず、システムソフトウェア教育支援環境「港」についての説明を行う。その後で組込み OS 学習環境について述べる。

2.1 「港」の概要

システムソフトウェアを学ぶことは計算機科学を学ぶ者にとって重要である。「港」プロジェクトシステムソフトウェアの教育を支援することを目的として研究が進められている。しかし、システムソフトウェア学習には次の問題がある。

- (1) ソフトウェアとハードウェア両方の知識が必要
- (2) 動作のわかりにくい OS の構造知識と記述能力が必要
- (3) ハードウェアとソフトウェアの協調動作への理解が必要

「港」は次の設計方針を挙げて、これらの問題を解決する。

- (1) 概念から実装まで幅広い教育支援を行える環境
- (2) 可視化を用いた、視覚的な教育支援を行える環境
- (3) ハードウェアとソフトウェアや異なる学習項目間で協調した動作も行うことができる環境

これらの設計方針に基づいて「港」は設計されており、個々に、独立した動作はもちろん、それぞれが協調した動作も行うことができる。Fig.1 に全体構成を示す。

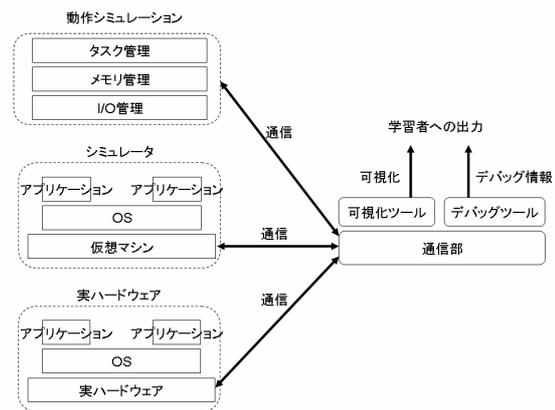


Fig.1 港の全体構成

2.2 「港」における組込みシステム学習環境

組込みシステム学習の問題は、システムソフトウェア学習の問題に加えて、実ハードウェアベースでの学習が必要なことが挙げられる。

そのため、組込みシステム教育では、次の教育支援ツールを学習者へ提供することで教育支援環境の構築を行う。

- (1) OS 動作の可視化ツールの提供
- (2) デバッグツールの提供

また、それを可能にするために対象ハードウェアとして、次のものを提供する。

- (3) 学習用ロボット
- (4) 計算機資源に余裕のある組込みボード
- (5) 無線 LAN の使用

学習用ロボットをターゲットにすることで、学習者のモチベーションを高めること、ロボットの挙動から、学習者の予定通りに動作しているかを確認することができる。このときロボットは制御の学習に向くよう、単純なデバイスを使用する。また、計算機資源に余裕を持たせることで、最適化をせずともリアルタイム処理に支障がないようなものにする。ロボットは動作を伴うため、利用のしやすさから無線 LAN によるデータ通信をサポートする。

この環境上で実行する、学習用に作成した組込み OS を合わせて提供し、組込みシステム学習を行わせる。

Fig.2 に可視化ツールの画面を、Fig.3 にデバッグツールの画面を、Fig.4 に学習用ロボットを示す。また、ロボットに搭載されているデバイスを Table.1 に示す。

本研究では無線 LAN 機能については試作中であり、ezTCP³⁾を用いて作成している。

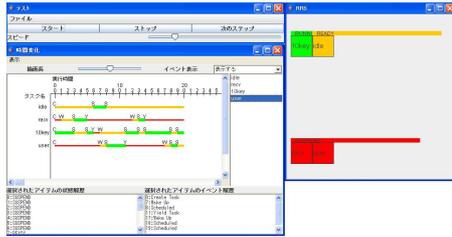


Fig.2 可視化ツールの画面



Fig.3 デバッグツールの画面

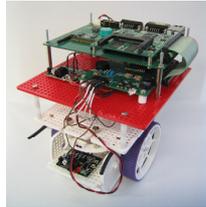


Fig.4 学習用ロボット

Table.1 ロボットデバイス

デバイス名	個数
可視光センサ	1
赤外線センサ	1
タッチセンサ	2
タクトスイッチ	2
モータ	2
LED	4

3. 問題分析

3.1 組込み OS 学習の問題分析

既存の組込み OS 学習で用いられている OS には次の問題がある。

(1) 学習優先度の低い機能

OS のような規模の大きくなりがちなシステムでは、オープンソースの流行に伴ってコードリーディング⁴⁾という、コードを読んで機能を理解する学習方法をとることで、各機能への理解が深めることができる。一方で、従来の学習に使用されている組込み OS では、組込み OS の教育用としては優先度が低く、不要な機能が多く含まれており、次のような問題があったため、コードリーディングが困難であった。

・ コードサイズが大きい

大きなコードは理解するのに時間が必要であるが、時間に制限のある演習講義などでは、学習者の負担となる。

・ コードが複雑

様々な組込みボードに対応していることから、ハードウェア依存のコードが多く、全体を理解しにくい。

(2) コメント・ドキュメントが不足

コメント・ドキュメントが不足していると学習者が全体の把握やコードの動作を理解するのが困難である。

(3) 単体での学習

多くの環境では、ハードウェア単体を学習環境と位置づけて提供されており、あまりデバッグなどの支援機能に注意が払われていない。そのため動作の確認がしにくく、教育支援が不十分である。

3.2 OS 演習の分析

2004 年度、拓殖大学情報工学科 3 年生を対象とした、5 回の講義時間を持つ OS 演習の講義で、学習用教材として利用した OS は 2000 行程度であった。そのとき、コードリーディングによって OS 機能の学習を行ったが、細かなコメントがない場合には学習者の理解が遅く、演習中に OS に詳細なコメントを付け加えるという経緯があった。その経験から学習用として使用するためにはコメントを詳細につける必要があることがわかった。

4. 設計方針

4.1 教育用組込み OS 機能の分離

本システムを、教育用組込み OS と、教育支援機能群を持ったファームウェアに分離することで、次の利点を得る。

(1) 教育用 OS

学習者へ提供する教育用 OS を、学習に必要な機能だけで構成することができる。それによりコードリーディングによる学習が容易になる。

(2) ファームウェア

ファームウェアの教育支援機能で、可視化ツールやデバッグツールとの連携をとることができ、それによって、動作を視覚的に捕らえ、イメージを伴った学習ができる。

4.2 教育用 OS の設計方針

4.1 で述べたとおり、問題点を解決するため、ここでは教育用組込み OS の方針を次に述べる。

コードリーディングを短期間でもできるように配慮し、コードサイズの上限を 2000 行として OS を作成していく。

(1) 構成

学習に必要な機能だけで構成した OS を提供する。その機能は次のとおりである。

- ・ タスク管理
- ・ タスクスケジューラ
- ・ 同期・排他
- ・ 割り込み
- ・ デバイス制御

ハードウェアの制御の学習を行う場合には、デバイス制御、割り込みについて学ぶ必要がある。

リアルタイム性についての学習を行う場合には、リアルタイムスケジューリングアルゴリズムやタスク管理、同期・排他について学ぶ必要がある。

以上の機能は、いずれも OS 学習では欠くことのできない機能である。

(2) コメント・ドキュメントの整備

学習する上では学習対象の仕様を記述したドキュメントは必要である。

ドキュメントでは、複雑になってしまう OS の構成を、判りやすく見せることが必要である。

4.3 ファームウェアの設計方針

ファームウェアは、OS の学習を支援するための機能の集まりである。ここでは、ファームウェア機能の中でも、教育支援に使う機能について述べる。

教育用組込み OS は学習者が書き換えることを前提としている。

(1) 教育用組込み OS の状態の取得

教育用組込み OS が管理している情報の中で、学習に必要なものをログとして残し、教育用組込み OS の動作を見ることを可能にする。そのためにログデータを作成する。教育用組込み OS の持つログデータの取得は、学習者からは見えないように行われるようにし、教育用組込み OS を書き換えても問題なく取得できるようにする。

(2) データの送信

動き回る学習用ロボットを対象としているため、学習者の使い勝手の面から、データ通信には無線を使用する。

4.4 コースウェア

コースウェアを Fig.5 に示す。組込みシステムの流れとして、制御プログラミングの学習をまず行い、外界の状況の変化に対応する制御アプリケーションについて学ぶ。次に組込み OS の概念について学び、組込み OS の実装について学ぶ。OS 学習では、コードリーディングで OS 機能の把握を行う。また、同期・排他やリアルタイムスケジューリングといったソフトウェアの学習と、デバイス制御というハードウェア制御の学習を、実際に実装を行うことで学ぶ。最後に応用制御プログラミングでロボサッカーを行わせることで、モチベーションを高めたまま、今までの学習の成果を利用し、かつ、ネットワークによる他のロボットとの協調動作についての学習もできる。

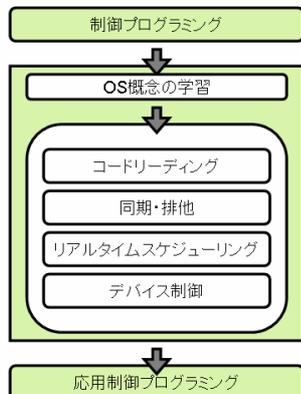


Fig.5 組込み OS 学習のコースウェア

5. 設計

5.1 全体構成

教育用組込み OS の全体構成を Fig.6 に示す。

本研究ではこの OS を、教育用 OS とファームウェアに分離する。

教育用 OS では、組込み OS の学習に必要な機能だけをもち、スケジューラについては複数用意した。スケジューラはリアルタイム機能の学習でスケジューラを交換して実行することができ、各スケジューラの特徴について比較しながら学ぶことができる。

ファームウェアには教育支援機能として、ログ生成、送信機能がある。ログ機能は学習者の PC へ、教育用 OS の動作記録を送信し、PC にある可視化ツール、デバッグツールとの連携をとる。

5.2 教育用組込み OS の設計

OS はコードリーディングによる学習の妨げにならないよう、小さい構成をとり、コードサイズも小さく抑える。OS 機能を次に挙げる。

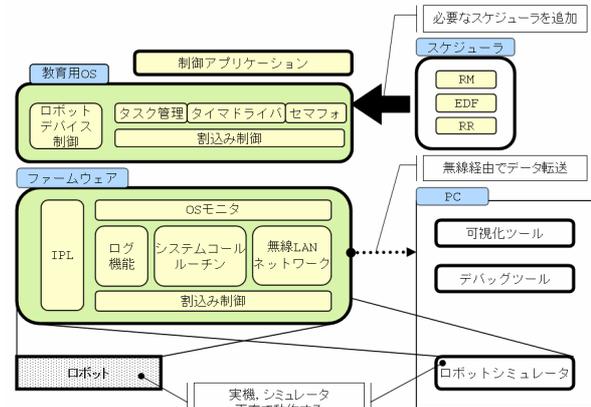


Fig.6 システムの全体構成

(1) スケジューラ

リアルタイムスケジューリングアルゴリズムの中で、最も基本的なものは RM と EDF である。また、非リアルタイムスケジューリングアルゴリズムであるラウンドロビンスケジューリング(以下 RR)も併せて提供し、それぞれの共通点と相違点を分離することで変更を容易にする。

(2) タスク管理

タスク管理機能を最小限にして提供するために、提供する主な機能としては task_create(タスク生成)・task_exit(削除)・task_yield(CPU 使用権の放棄)・task_suspend(実行保留)だけを持つ。

(3) 割り込み

割り込み機能はハードウェアに強く依存する。しかし、そのような依存した処理よりも割り込みの動作が重要である、提供する主な機能は、int_regist(割り込みハンドラの登録)・int_clean(割り込みハンドラの削除)である。

(4) 同期・排他

同期・排他はマルチプログラミングを行う上で重要な機能である。この機能を提供するため、セマフォがよく用いられている。セマフォには幾つかの種類があるが、本研究では、セマフォの中でも単純な機能であるバイナリセマフォを提供する。提供する主な機能は `sem_create`(セマフォ生成)・`sem_clean`(セマフォ削除)・`sem_p`(P 命令)・`sem_v`(V 命令)である。

(5) デバイス制御

OS が対象とするハードウェアを一つに絞ることで、デバイスの種類を限定することができる。ロボットに搭載されているデバイスを制御する。これらは操作関数だけを提供し、コードを小さくする。

5.3 コメント・ドキュメント

コメントは、学習者によってわかりにくい場所が違うため、C 言語のコード、アセンブリ言語を問わず、詳細につける。

ドキュメントは `doxygen`⁵⁾を利用して自動生成するものを利用する。Fig.7 にコメント付けされた OS コードの一部を示す。Fig.8 に `doxygen` によって生成された画面を示す。この中には関数呼出しの順序やコメントによる関数の説明が入る。さらにリンクを使って他のコードとのつながりも見やすくなっている。

```

/**
 * センサ値を特定のアドレスから取得する関数.
 * static 変数へ値を保存する.
 * A/D 変換で得られるデータは 10 ビットであることに注意すること.
 */
ushort get_light( void )
{
    ushort dt;                               /* 16 ビットデータ保存用 */

    irq_lock();                               /* 割込み禁止 */
    dt = read_b( ADDRRAH ) << 8;             /* 上位 8 ビットを取得 */
    dt += read_b( ADDRDAL );                 /* 下位 2 ビットを取得 */
    dt >>= 6;                                /* 10 ビットデータに直す */
    light_data = (ushort)dt;                 /* 取得した値を保存 */
    irq_unlock();                             /* 割込み許可 */

    return dt;                               /* 取得したデータを返す */
}

/**
 * A/D 変換を行い、データを ADDRRAH, ADDRDAH へ値を書き込ませるため、
 * A/D 変換割込みを開始する.
 * A/D 変換割込みは、現在の設定では一度割込みを行うと、
 * 次に許可するまで禁止状態になることに注意すること.
 */
void start_light_interrupt( void )
{

```

Fig7 コメント付けされた教育用組込み OS のコード

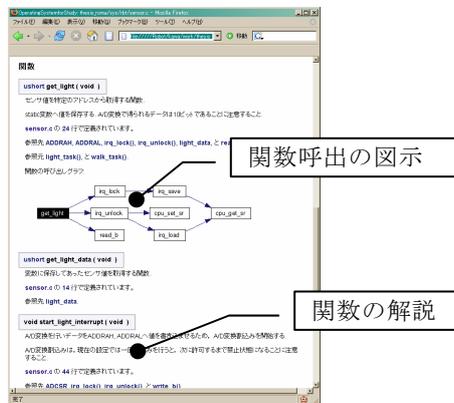


Fig.8 doxygen によるドキュメント

5.4 ファームウェア機能

ファームウェアは学習を支援するための機能の集まりである。学習を支援する機能とは次の機能である。

(1) ログ機能

ログは、教育用組込み OS の動作を記録して、その動作をデバッグツールや可視化ツールなどの教育支援機能に伝えるためにある。そうすることで、学習者は教育用組込み OS の持つ機能がどのように動作したかを、視覚的に捉えることができ、イメージとともに実装の学習を行うことができる。

送信するには有線、無線があるが、ロボットの動くという特徴から無線で送信する方が、作業効率がいい。さらには、学習者は OS を書き換えて、何度もロボットへダウンロードするので、その効率もよくするため、無線 LAN を用いてデータ通信を行う。

送信するログデータとしては次のものがある。

① タスクデータ

これらのデータは教育用組込み OS 内にデータとして保持している。タスクは TCB 内に情報を持っているが、これの変化をスケジューラ内で、ロボットの状態変化はログデータを作成するタイミングで自動的に取得して、データを作成する。Fig.9 に動作を示す。

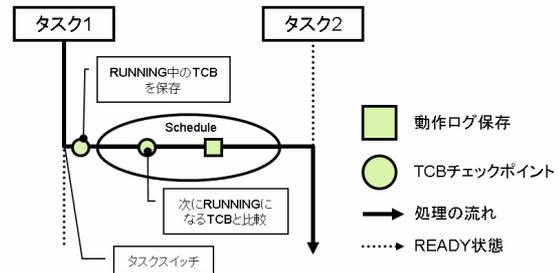


Fig.9 タスクスイッチ時の処理

Fig.9 の TCB チェックポイントで TCB データを一度保存する。その後、schedule 後の情報との差分から TCB の変化した値をすることが出来る。TCB を静的に、配列として確保することで、使わなくなった無効なデータかどうかタスクの状態から判断できるようにする。

学習者が TCB のデータ構造を変更することも考えられるため、ログに必要なデータは、TCB 構造体の中の決まった位置に配置する。そうすることで、ファームウェアがデータの意味を理解し、ログデータを作成する。

② 入力デバイス

学習用ロボットには Table.1 の可視光センサ、赤外線センサ、タッチセンサ、タクトスイッチという入力デバイスがある。

入力デバイスは、割込みで取得するときには学習者が記述した割込みハンドラの直前にファームウェアが記述したハンドラを呼び出す。そこで、デバ

イスからの入力値をログとして保存してから、OSのハンドラを呼び出す。ポーリングの場合は、明示的にログ機能呼び出す必要がある。

③ 出力デバイス

Table.1のモータ、LEDが、学習者が制御する出力デバイスである。これもデバイスに値を出力する際に、明示的にログ機能呼び出す必要がある。

これらにより、ログを取得することができる。

(2) 無線LANネットワーク機能

教育用組込みOSは学習者に何度も書き換えられるが、フラッシュROM領域に置くのではなくRAMに置くことで、書換え制限を気にしなくてよくなる。しかし、そのためにRAMへ教育用組込みOSを転送する必要が出てくる。本研究では利便性を考え、無線LANを用いたOSの起動やログの送信などを提供し、学習を支援するための機能として利用する。

6. 実現

実現した本研究のシステムにおいて、実現規模をTable.2に示す。また、ログを使用することで動作の履歴を取得し、PCへ送信することが可能になった。

(1) 教育用組込みOSの開発

システムの機能を分離し、小さいコードサイズと必要な機能だけの構成で作成した。これにより、学習者がコードリーディングすることで、機能の理解が容易になる。

(2) 支援機能の開発

ログの生成、送信機能を作成することで、「港」の可視化ツールやデバッグツールをはじめとした、教育支援ツール群との連携をとることが可能になった。

Table.2 システムの実現規模

機能	コードサイズ (行)
教育用組込みOS	1849
ファームウェア	3516
合計	5365

7. 関連研究

ここではまず、既存の学習に用いられている組込みOSが学習に適切かについての検討を行う。その後で、組込みに限らず学習用のOSを持つ特徴と組込みOS学習のため、利用可能かを検討する。

また、学習に用いられる組込み学習OSのコードサイズ比較表をTable.3に示す。

(1) TOPPERS

組込みOS学習でよく用いられているものとして、TOPPERS⁶⁾がある。

TOPPERSは実用性の重視や、移植性が高さから

コードサイズが大きくなっている。また、実用指向なので複雑な構成になっている。そのため、コードリーディングによる学習のように、学習対象となるコードを読む学習方法では、OSが持つ機能の理解が困難である。

(2) brickOS

brickOS⁷⁾はターゲットが固定されている分ハードウェア依存のコードは少ないといえる。しかし、教育用として作成されていないため、コードが複雑で読みにくい。また、コメントやドキュメントも整っておらず、教育用には向いていない。

Table.3 学習用OSの比較

機能	コードサイズ (行)
教育用組込みOS	1849
TOPPERS	125548
brickOS	17618

8. おわりに

本研究では、教育用組込みOSと教育支援機能を持ったファームウェアを開発した。

本研究により、学習者のコードリーディングが容易になり、OS機能を読むことで実装を理解しやすくなった。

また、学習支援機能との連携をとることによって、OSの動作を視覚的に捉えることができ、イメージを伴った学習ができるようになった。

これらの成果から、概念から実装までを連続して行える幅広い学習が可能になった。

今後の課題として、教育的な評価を行う必要がある。また、本研究ではezTCPを用いているが、これは通信速度が遅いため、直接無線LANカードを制御する必要がある。

[参考文献]

- 1) 経済産業省商務情報制作局, 2005年度版組込みソフトウェア産業実態調査報告書
- 2) 西野洋介, 早川栄一, Development of an OS Visualization System for Learning Systems Programming, HCI International 2003, pp.1116-1120 (2003)
- 3) ezTCP, アルファプロジェクト, <http://www.apnet.co.jp/index.html>
- 4) Diomidis Spinellis, Code Reading: The Open Source Perspective (Effective Software Development), Addison-Wesley Pub (Sd), ISBN: 0201799405
- 5) Dimitri van Heesch, doxygen, <http://www.stack.nl/~dimitri/doxygen/>
- 6) TOPPERSプロジェクト, TOPPERS/JSPカーネル, <http://www.toppers.jp/>
- 7) Stephan M Moraco, brickOS, <http://brickos.sourceforge.net>