

自走ロボットを利用したリアルタイムシステムの学習支援環境の開発

横山 太一郎 西野 洋介 早川 栄一 高橋 延匡
拓殖大学大学院 工学研究科 電子情報工学専攻

〒193-0985 東京都八王子市館町 815-1

E-mail : taichiro@os.cs.takushoku-u.ac.jp

あらまし

本研究では、自走ロボットを利用したリアルタイムシステムの学習支援環境の開発について述べる。リアルタイムシステムを学習するうえで問題となるのは、システム内部の動作は目に見えにくいといった点と、知識や経験を得るための実験演習できる学習環境がないといった点がある。このような問題に対して、我々はハードウェアに Mindstorms を利用し、GUI 環境を使用して学習を支援することを提案する。本システムの特徴は、(1) いくつかのタスクスケジューラと OS に与えるパラメータを GUI を使用して学習の支援をする (2) Mindstorms 上で動作する legOS をベースに RTOS を提供する、さらに(3)OS のスケジューラ、特にリアルタイム OS におけるタスクスケジューラの振舞いを可視化することである。

キーワード リアルタイムシステム、学習支援

Development of a Automotive Robot based Learning Support Environment for Realtime System

Taichirou Yokoyama, Yousuke Nishino, Eiichi Hayakawa and Nobumasa Takahashi
Takushoku University

815-1 Tatemachi, Hachioji-shi, Tokyo 193-0985 Japan

E-mail : taichiro@os.cs.takushoku-u.ac.jp

Abstract

This paper describes the development of a real time learning support environment based on automotive robot. It is difficult to acquire knowledge or experience of real time system programming owing to the invisibility of kernel behavior and the insufficiency of attractive experiment system. To solve these problems, we developed the learning support environment that is unified between LEGO Mindstorms based robot controlled real time operating system and graphical user interface (GUI) oriented development environment on PC. Features of this system are (1) to provide GUI in order to easily attempt several kernel and application parameters, (2) to support real time operating system enhanced the LegOS and (3) to visualize task scheduling behavior when the Mindstorms is moving.

key words Realtime System, Learning Support

1. はじめに

近年 CPU の低価格化によってマイクロコントローラが使用された家電製品が次々と開発され、組み込みシステムの需要は拡大してきている。ネットワークに対応した機能や、機器制御という本来の仕事もさらに高度化、複雑化する事によって、組み込みシステムで使用されるソフトウェアも高度な処理が必要とされている。また、開発期間の短縮や低コストの要求から組み込みシステムでもリアルタイムオペレーティングシステム(RTOS)の需要が拡大しているが、OS を含むシステムプログラミングの知識や経験を持った開発者が不足しているといった現状がある。そのため、システムプログラミングの知識や実践的な経験を持った開発者を教育する環境が必要である。

しかし、現状ではこのような教育環境はあまり存在していない。この理由として、設計者、又は開発担当者が現実の問題に追われて、このような支援系を開発する余裕を持っていないこと、また、大学等では実際に開発する場から遠い所にあることが考えられる。

例えば、汎用の OS や応用に即したオンラインリアルタイムシステムの中身を理解することは、開発担当者以外には、情報が公開されないことが多く、公開されたとしても学習の初期段階の学習者が容易に理解するのは難しい。この問題を解決するには学習者のレベルにあった学習環境を提供し、OS の動作が現実簡単に目でたちどころに判断できることが重要だと考えた。

2. 本研究の目的

このような問題を踏まえて、本研究では次の目的を設定した。

- (1) 実ハードウェアを用いた学習支援環境の提供
- (2) RTOS の提供
- (3) 可視化による学習支援

具体的には、次の三点の特色を持たせる。第一の特徴は、ブロックで組み立てることを可能にした自走式ロボットである Mindstorms[1]を使用することによりハードウェア環境の単純化を図ることである。すなわち、プログラムの設計不良はたちどころに動作不良となって誰でもが見分けることが可能となる。

第二の特徴は、OS として legOS[2]をベースとして RTOS を提供する。具体的にはタスクスケジューラをいくつかの方式で設計し、リアルタイム性とスケジューラとの関連を明らかにすることが容易となる。

第三の特徴は、OS のスケジューラ、特にリアルタイム OS におけるタスクスケジューラの振舞いを可視化することである。

以上、本研究は三つの特徴を具体化することを目指す。

3. 問題分析

本章では、リアルタイムシステムの学習を支援する環境を提供するうえで問題となる点について述べる。

3.1. 演習できる学習環境の要求

現在、一般的な科目での教育環境では、講義による説明とそれと並行してテキストを用いて学習させている。計算機科学における教育環境でも同じ方法を用いている。プログラミング言語の講義やソフトウェア開発などの演習では、計算機上でアプリケーションを作成し実行する学習環境が与えられる。これは、講義による説明とテキストによる学習だけでは習得するのが難しく、自らものを作り体験することにより学習内容を習得し、学習者自身の力になるからである。

リアルタイムシステムを学習する際にも講義による説明とテキストを用いた学習の後に実験演習できる環境があれば、構造の理解や動作のイメージをするのに役立つ学習した内容をより確実に習

得できると考えられる。

3.2. 演習する学習環境の問題点

リアルタイムシステムを実験演習できる環境を構築するには、プログラムを作成する環境と作成したプログラムを実行する環境が必要になる。

リアルタイムシステムで用いられる AP や RTOS を実体験して学習できる環境として二つ取り上げ、問題点を示す。

(1) 実ハードウェアを用いる方法

最近の半導体技術の進歩で、CPU の機能の拡大は目覚ましい。また、CPU キットもかなり高機能化されている。学習環境として CPU キットを用いて実験する環境が考えられる。市販されている CPU キットを用いた評価ボードにはリモートデバッガやクロスコンパイラなどが付属し、プログラミングできる環境は整っている。しかし、これらを使いこなすには、情報工学科の学生にとってみればコンピュータアーキテクチャ、プログラミング言語、OS、データ構造等の多くの知識と、それに関するプロトタイプの開発経験が必要であり、学習者がいきなり OS を開発することや、デバイスを作成しドライバを書くのは困難が予想される。

(2) 仮想的なハードウェアで実現する方法

ターゲットとなる CPU のシミュレータを作成し、PC などの計算機資源上でアプリケーションを動作させることを可能にする方法である。CPU だけでなくデバイスも仮想的に提供することで、学習環境で使用することができる。しかし、リアルタイムシステムでは実機上で動作することを保証することが重要であり、仮想的な空間では現実性に乏しいといった問題がある。例えば、CPU シミュレータ上では仮想的な時間での同期的な割込みはシミュレートとできるが、実時間での非同期的な割込みをシミュレートさせることは難しい。

4. 学習する項目

RTOS ではひとつひとつの仕事に時間的属性をつけ、ポリシーに依存する優先度の高いプログラムを先に処理し、優先度の低いプログラムは後まわしにする。早く処理してほしいプログラムほど高い優先度をつけている。

割込みなどの外部要因に対しては、予め決められた時間内にシステムは定まった様式で処理を行い、その結果に基づいて外部に応答を返すのが基本である。それもある時間のデッドライン内に行わなければならない。このような時間的制約を守るための重要なパラメータとして、タスクに CPU 時間を与えるタイミングと割込み応答時間がある。CPU 時間をタスクに与えるタイミングを決定するのは RTOS で使用されるリアルタイムスケジューラであり、割込み応答時間とは一般的には割込みが発生してから割込みハンドラにコントローラが移るまでの時間を指す。

そこで、リアルタイムシステムで重要なタスクスケジューリングと割込みに関して着目し本学習支援環境での学習項目とする。

5. 設計方針

そこで本学習支援環境では、次に示した三つの設計方針を定める。

(1) 実ハードウェアを使用した学習環境を容易に提供できること

学習環境として実ハードウェアを用いる方法での問題点から、マイクロプロセッサとデバイスが一括に提供されている必要がある。また、そのハードウェア上に AP を作成するプログラミング環境も提供されている必要がある。そこで LEGO 社の提供する Mindstorms を使用して本学習支援環境を構築する。この Mindstorms を用いることで、実践的な経験を積むことも可能となる。

8bitCPU である H8 を搭載して、モーター出力 3

ポート、センサー入力 3 ポートを備えた RCX モジュールにLEGO社のブロックで組み立てることで自由にロボットを設計できる。

(2)一つのシステム上で複数のスケジューラを変更できること

リアルタイムシステムではタスクごとに実行時間を見積もり、ある決まった時間内で処理を実行しなくてはならない。そのため実行するタスクを決めるためには、リアルタイムスケジューラと呼ばれるポリシーに依存する。

そこで、本学習支援環境では、代表的なリアルタイムスケジューラを提供し RTOS の学習を可能にする。また、一般的なタスク管理方法も学ぶことを可能とする。そこで次のスケジューラを提供する。

- (i)ラウンドロビン
- (ii)優先度方式
- (iii)多重レベルラウンドロビン
- (iv)RM(Rate Monotonic)
- (v)EDF(Earliest Deadline First)

(3)学習の初期段階に対する支援をすること

学習の初期段階にある学習者がスムーズな学習ができるよう支援する。具体的には、OS のソースを変更する手間を省いて OS に与えるパラメータ

を操作できる環境を提供する。また、OS の拡張された機能をイメージできるような仕組みや OS や API の解説、サンプルになる AP の作成なども提供する。

6. 全体構成

本システムは、RCX 上で動作するリアルタイム拡張 legOS と、OS を生成する PC 上のデスクトップから構築されている。拡張 legOS は、カーネルライブラリ、複数のスケジューラ、およびドライバ群と、これらを用いて実行するタスク群がある。

これらのうち、スケジューラやタスクに関するパラメータは、デスクトップ上で GUI を用いて変更することができる。

学習者は legOS が提供するライブラリ、ドライバ、またスケジューラに関連する API を利用し AP を作成する。作成した AP を赤外線通信を利用し RCX にダウンロード後実行する。そしてタスクスケジューリングをログとして保存し、実行後 PC に転送する。PC 上ではログからタスクの状態遷移を表示する。全体構成図を図 1 に示す。

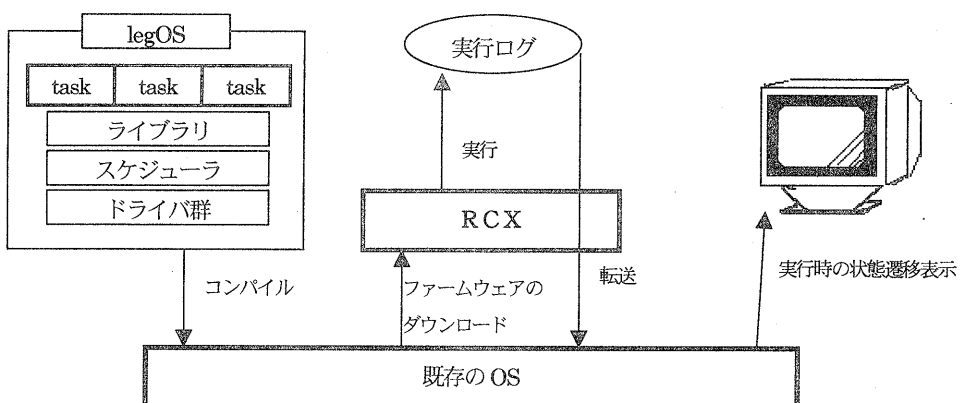


図 1. 全体構成図

表1. 拡張したAPI

リアルタイムタスクに関連するAPI	
set_rt_task()	指定されたタスクを周期タスクに設定する。起動時間, 起動周期, 最悪実行時間, デッドラインを設定し、引数はスケジューラに依存する。
rt_kill 0 / rt_exit 0	リアルタイムタスクを削除する。
rt_yield0	次の周期まで実行権を放棄する。
スケジューラの実装に必要な内部インタフェース	
rt_get_time0	現在のシステム時間 (sys_time) を得る。
rt_set_timer0	システム時間が指定時間になったらスケジューラが起動するよう設定する。

7. 機能設計

7.1. タスクスケジューラの拡張設計

(1) legOS のタスク管理

legOS は、タイムスライス方式のプリエンプレイブルマルチタスクを提供している。legOS デフォルトのスケジューラは優先度方式の多重レベルラウンドロビンを採用し、タスク管理構造の実装には双方向リストを利用している。タイマハンドラが1ms周期で起動し、実行中のタスクがタイムスライス値を使い切るとスケジューラが起動してpreemptされる。

(2) タスク管理スケジューラの拡張方法

タスク管理は、プロセッサ依存のディスパッチャと、プロセッサ非依存のスケジューラから構成される。タスク管理アルゴリズムの変更にはスケジューラ部分の変更をする。RTOSにはタスクごとに起動時間、起動周期、最悪実行時間、デッドラインといったパラメータが必要となる。legOSにはリアルタイム性を実現するためのメカニズムを持っていないためスケジューラ以外に次に示すような変更を行いリアルタイムスケジューラの実行を可能とする。

- (1) TCBに必要なパラメータを追加
- (2) リアルタイムタスクの生成、削除
- (3) システム時間に関するAP

リアルタイムスケジューラ実装のための拡張

APIを表1に示す。拡張後のタスクの状態遷移を図2.に示す。

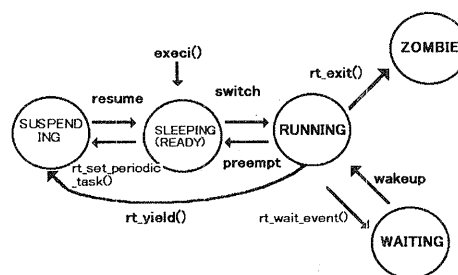


図2. タスクの状態遷移

リアルタイムタスクの生成には、通常のタスク同様 execi() でエントリを生成した後、set_rt_task() でリアルタイムタスクに必要なパラメータを設定する。リアルタイムタスクは無限ループとして記述され、タスク間でのループが1周期に相当する。各周期の最後にrt_yieldを実行し、次の周期が来るまで実行権を放棄する。

7.2. GUIを用いた学習支援機能

legOSの拡張項目を表現するためと、学習の初期段階を支援するために、タスク管理アルゴリズムの選択とOSに与えるパラメータをGUIを用いて変更できるようにする。図3にそのスナップショットを示す。このGUI画面では、画面上部でタスクスケジューラを選択、下部に各パラメータを表示している。下部の左側には各スケジューラ

ごとの変更できるパラメータを示し、右側にはデバイスのパラメータを変更するスライドを示す。

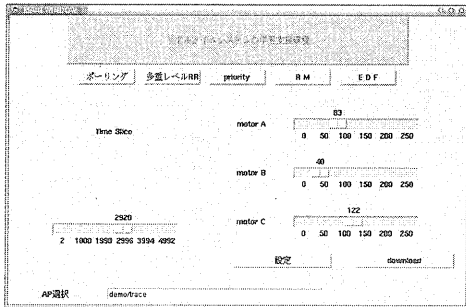


図3. パラメータの変更を可能にする GUI

本 GUI 環境で変更できるパラメータの一覧を次に示す。

- (i)ラウンドロビンでのタイムスライス
- (ii)タスクスイッチ時の時間間隔
- (iii)モータのスピード

(i)ラウンドロビンでのタイムスライスはまさにタスクに与える実行時間を設定する値であり、この値がシステムに与える影響をみる事ができる。(ii)はタスクが切り替わる時間間隔を変更して、タスクスケジューラを起動するまでの時間をコントロールするものである。システムが実行する時間がリアルタイムタスクの設計にどのように影響するかを見るものである。

変更可能なパラメータは、ソースコード中にタグを入れて、変更できるようにする。変更後コンパイル、ファームウェアのダウンロードを行って、各パラメータに対応した OS を作成できるようにする。また、変更したパラメータを示すソースコードを同時に見る事が可能である。

7.3. タスク可視化機能

本学習支援環境では、プログラムロジックをテストのためと、実行状態を学習者に示すためにタスクの状態遷移表を示す。プログラムロジックをテストすることは、スケジューリングが不可能な

タスクとして設計された AP の動作を検証するために必要となる。ロジックテストでスケジューリングが可能になっても、実際に動作を保証できない場合 RCX ではその原因を知ることはできない。

これに対して、スケジューリング時にタイムスタンプとタスクのパラメータを保存し、その情報を赤外線を通して PC に送り、PC 上で実行時の状態遷移を復元する。これにより学習者に実行状態を示すことが可能となる。

7.4. OS 基本機能の可視化環境と統合

本学習支援環境では、自走ロボットを利用してタスクレベルでシステム内部の可視化を可能としているが、可視化による OS 基本機能の学習支援システムの開発[3]では OS を可視化して学習を支援する環境を提案している。この環境では、従来の計算機環境の上に CPU シミュレータを作成し、その上で学習用 OS と AP を動かすことで、OS の動作を制御する。またシミュレータを通して可視化機構を提供し、OS 自体の動作や、AP と OS の協調動作を可視化し OS の概念的な学習を行えるものである。

本学習支援環境と可視化による学習支援環境を用いて実ハードウェアを用いた実践的な学習とシミュレータを用いた OS の概念的学習を一つの学習支援環境とすることを考えている。

8. 実現

本学習支援環境を設計し実現するにあたっての実現環境と使用言語を次の表 2 に示す。

表 2. 実現環境と使用言語

実現環境	
実行環境	RCX1.0
legOS バージョン	legOS-0.2.3
開発環境	Linux Vine1.1
使用言語	C 言語(Linux / glibc)

次に本学習支援環境での実行例について述べる。走行中の写真を図4. に示す。

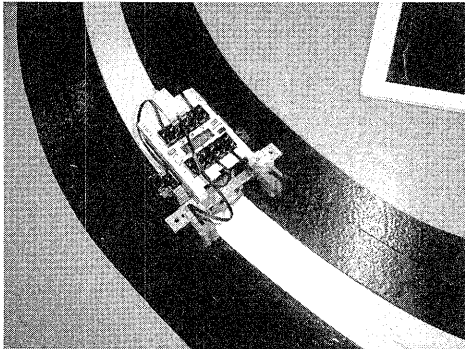


図4. ライントレースの様子

この例では、ライントレースによる実行例について示している。RCX モジュールにモータと2つの光センサを接続し、ライン上を自走するロボットとして構成している。光センサが白いラインを認識すると黒いラインを認識するまでモータにトルクを与えるタスクを光センサごとに一つ与え、ラウンドロビンでタイムスライス時間だけ与える。

アプリケーションに与えるパラメータとしてモータのスピードと、タイムスライス値を変化させることでシステムの全体への影響を見る。また、モータやセンサの取り付け位置を変化させ、そのロボットの構造上の変化がシステムにどう影響するかを見る。

9. おわりに

本報告では、自走ロボットを利用してリアルタイムシステムの学習を支援する環境を提案について述べた。この学習支援環境によって、リアルタイムシステムを実験演習できる環境を容易に提供でき、ロボットの動作とシステム内部の動作を関

連づけて学習することが可能となる。

今後は、ビデオを用いてロボットの動作を撮影し、タスクスケジューリングとロボットの動作、さらには OS 基本機能の可視化環境とで統合して動作を再現していく。

参考文献

- [1] Hitachi Single-Chip Microcomputer H/3297 Series Hardware Manual 3rd Edition
- [2] Markus L. Noga "legOS Home Page" <http://www.noga.de/legOS/>
- [3] 西野洋介、早川栄一、高橋延匡：可視化による OS 基本機能の学習支援システムの開発、情報処理学会研究報告、2000-OS-84、pp.173-180、2000.