

高等学校産業科における組込みシステム学習支援環境の開発と実践

産業科における組込みシステム学習支援環境研究グループ

西野 洋介、小幡 章、土谷 哲司、大嶋 亮平、影山 伸

要約

本研究は、新たに発足した高等学校産業科における組込みシステム教育を支援するものである。組込みシステム学習における導入段階において、生徒がシステムの動作概念イメージがつかめず、理解促進を妨げている点に着目し、自立型の自走ロボットの挙動を利用した内部動作の可視化によって動作概念を具体化した教材の開発、実践、評価を行うことが目的である。これにより、組込みシステムの概念を視覚的、直観的に理解することができることを明らかにする。

産業高校では組込み機器や制御系の教育を行っているが、従来のテキストや実習装置では組込みシステムの動作、構造のイメージがつかめず、動作概念の理解が難しい現状があった。これに対し、本環境は組込みシステムの概念からロボットなどの制御、組込み OS まで連続的に教育支援を行うことができる。また、従来の組込みシステム学習支援環境は学習者の理解度に応じた教材の設計が困難であり、教授者のコストが大きくなっている点に着目し、様々な学習教材を可視化できる設計を行っている。さらに、環境を BOOT 可能な 1CD、1USB に集約することで、あらゆる PC 上で、実習環境を手軽に構築することが可能となる。本研究では、自走ロボットの開発、ロボットの内部動作のモニタ開発、ロボット内の OS の可視化、3D によるロボットシミュレータの開発を行う。

本研究は従来の制御系教材と比べ、内部動作を可視化によって制御とプログラミングの関係を直観的に理解することができる。特に、制御にかかわる授業では、ロボットを作成し、プログラミングによって制御する展開が多いが、外乱などによって動作の再現性を確保することが困難である。そのため、学習者はデバッグの際、ハードウェアの問題なのか、ソフトウェアの問題なのか、その他の要素かを判断することが難しい。また、リソース不足から、ロボット内部のプログラムの挙動を観察できず、ロボットの挙動とプログラミングの関係をイメージしにくい。

そこで、本環境ではロボットの動作をビデオキャプチャしつつ、同時にロボット内部のセンサ値、出力タイミング、実行中の命令をロギングし、動作後に PC 環境からこれらに関連付けて出力することで、上述の問題を解決する。また、これらのシステムは MVC モデルに基づいて再生しているため、あらゆるロボット環境を同じユーザインタフェースで扱いことが可能となり、教材作成のコストダウンにも大きく寄与することが可能である。

本報告では、本システムの概要、組込みシステム学習のコースウェアを示すとともに、本システムを産業高等学校の実習において実践し、本システムが制御プログラミングを含めた組込みシステム学習において一定の効果が得られたことについて述べる。

1. 研究の背景

現在、日本の産業においてコンピュータが組み込まれている機器は、日々増加の一途をたどっている。その結果、組み込みシステム産業に従事する技術者は多く、これらの技術者にソフトウェアを提供する組み込みシステム関連産業は日本の産業界にとっても大きな割合を占めており、このような制御プログラムの開発を行う組み込みシステム技術者の需要は高い。

このような背景もあり、初等中等教育における情報技術の理解は重要な課題であり、技術立国を支える重要な要素であると認識されている。特に、高等学校における教科「情報」の実施や、中学校技術家庭科における「プログラムによる計測・制御」の必修化に代表されるように学校現場での情報教育が推進されている。

一方、産業高等学校(以下産業高校)や工業高等学校では教科「情報」の単位を「情報技術基礎」で代替している。産業高校は普通高校に比べ、実験・実習も多く行い、専門的な知識を持った生徒を育成している。特に組み込みシステムに関しては、簡単な制御に始まり、組み込みOSを用いたプログラミングまで幅広い教育を行っており、ロボット教材の普及、低価格化もあり、生徒の組み込みシステムへの興味、意欲は高い。しかし、このような取組みにも関わらず、現状では、組み込みシステムの理解は難しく、挫折してしまう生徒も多い。その原因として次の点が考えられる。

- ・システムの中の様子、処理の流れが見えにくいので、動作のイメージができない
- ・外乱が大きく、何が問題点なのか特定できない

これらの問題点に着目し、組み込みシステムの統合的な学習支援環境の開発および産業高校における実践と学習効果の評価を行った。

2. 組み込みシステム教育の現状と要件

2.1 産業高校における現状

産業高校では情報技術に関する教育を行っている。特に近年では組み込み機器の発達、普及や、ロボコン等の盛り上がりによって代表されるように、制御プログラミングや組み込みシステムといった分野のニーズが高まり、さらに、中学校技術家庭科における「プログラムによる計測・制御」の必修化に対応し、カリキュラムにおいてもこれらを重視した編成、教育を行っている。特に、情報系学科に所属する生徒はロボット等への興味関心が高く、学ぶ意欲は大きい。しかし、多くの生徒が実際にロボットを利用した組み込みシステムの授業において、システム全体の総合的な学習を行うにつれその意欲が小さくなり、挫折してしまう場合も多い。その理由として、次の点が考えられる。なお、これらの問題点は、生徒へのアンケート調査や現場での経験から得られた生徒の様子から推察したものである。

生徒側の問題点

- ・中身の様子がわからない、イメージできない
機器の内部動作が見えにくいので、どのような状態になっているのかイメージできない
- ・外乱、再現性

組み込みシステム特有の外乱の影響や、動作の再現ができないため、うまく動作しない原因が何なのか特定できない

- OS やネットワークの知識

機器の高性能化に伴った組込みやネットワークの充実によって、これらの理解も必要となっている

- 統合的な環境

実装的な制御プログラミングと概念レベルの理論的な座学を関連付けて理解することができず、単に学習機材の使い方に終わってしまっている

教員側の問題点

- 教員の準備労力

限られた時間のなかで多くの機材準備や教材開発労力が大きく、古い機材や現実に即しておらず実用的ではない教材を使っている

2. 2産業高校における組込みシステム学習環境の要件

上述の問題点を解決するシステムとして、産業高校の組込みシステム教育段階における学習環境の要件について述べる。

(1) 可視化を中心とした直観的な学習教材の提供

組込みシステム学習では、内部動作が見えにくいいため、生徒にとってその挙動が非常に分かりにくくなる。このことが生徒の理解の妨げとなってきている。そこで、可視化を中心とした環境を提供することで、内部動作のイメージ付けや複雑な動作を抽象化して、分かりやすい形で生徒に提供する。また、内部動作にロボットの挙動を関連付けて可視化することで、生徒の学習意欲を保ちつつ、ロボットの挙動を通じて制御プログラミングの概念を理解する環境を提供する。

高校生の学習段階においては、ロボット動作の不具合が想定外の外乱、ソフトウェアのバグ、いずれの原因によるのかを直観的に判断することは難しく、繰り返し実行して判断する生徒が多い。本来、組込みシステム開発においては想定される外乱の影響を考慮したプログラミングを行わなければならないが、制御プログラミングの初学者にとっては、その前段階であるプログラムと機器の動作の対応を理解することさえ困難な場合がある。そこで、挙動の再現性を確保でき、内部動作の様子をモニタできる環境があれば外乱の影響なのかソフトウェアの問題なのかを容易に判断することができる。

(2) 概念学習から実装学習までを単一環境でサポート

教育機関においては、学習時間の確保は重要な問題である。情報技術の発展に伴い、組込みシステムだけに多くの時間を割くことは難しい。そこで、システムの操作などの習熟にかかる時間を減らすために、単一の環境で講義での概念学習の支援から、実験演習などでの実装学習の支援までをサポートする。

(3) 教員が教材を作成しやすい環境の提供

組込み分野の進歩は著しいが、その一方で教員の数や質は必ずしも一定ではない。そこで教員にとって使いやすく、教材を作成しやすい環境を提供することが必要である。特に公立高校等においては、教員の異動が頻繁に行われるため、その教員でなければ動かすことのできない教材は利用しにくい。そこで、教員のスキルになるべく依存せずに利用することができる教材が望ましい。

この問題に対して、実応用を想定した CPU プラットホームや部品群、サポート OS、ロボットを用意して解決

する。

3. 組込みシステム学習支援環境の概要

組込みシステム学習支援環境の概要、全体構成について述べる。本システムの全体構成を図 1 に示す。

産業高校における組込みシステム学習環境の要件を満たす必要から、本システムは可視化を中心とした学習支援を提供する。また、学習の各フェーズでの利用を可能にするために、システム全体を、学習を行うための教材群、生徒の理解を支援する可視化ツール群、この二つの群のあいだの通信制御をおよびログ情報の管理を行う制御ツールから構成する。教材群は、OSシミュレータ、SH3 仮想マシン、SH3 ロボット、LEGO NXT 仮想マシン、LEGO NXT ロボットから構成される。また、可視化ツール群としては、システムモニタ、OS モニタ、3D ロボットシミュレータから構成される。

これらの教材群と可視化ツール群は、独立性の高いモジュールとなるように設計されている。学校現場では学習効率の観点から、生徒の理解、興味に応じた可視化教材を用いたが、時間的、専門的理由から、教員が教材をカスタマイズすることは難しい現状がある。本システムはこれらの問題を解決し、生徒の理解、学習進度に応じた柔軟な構築を行うことができる。

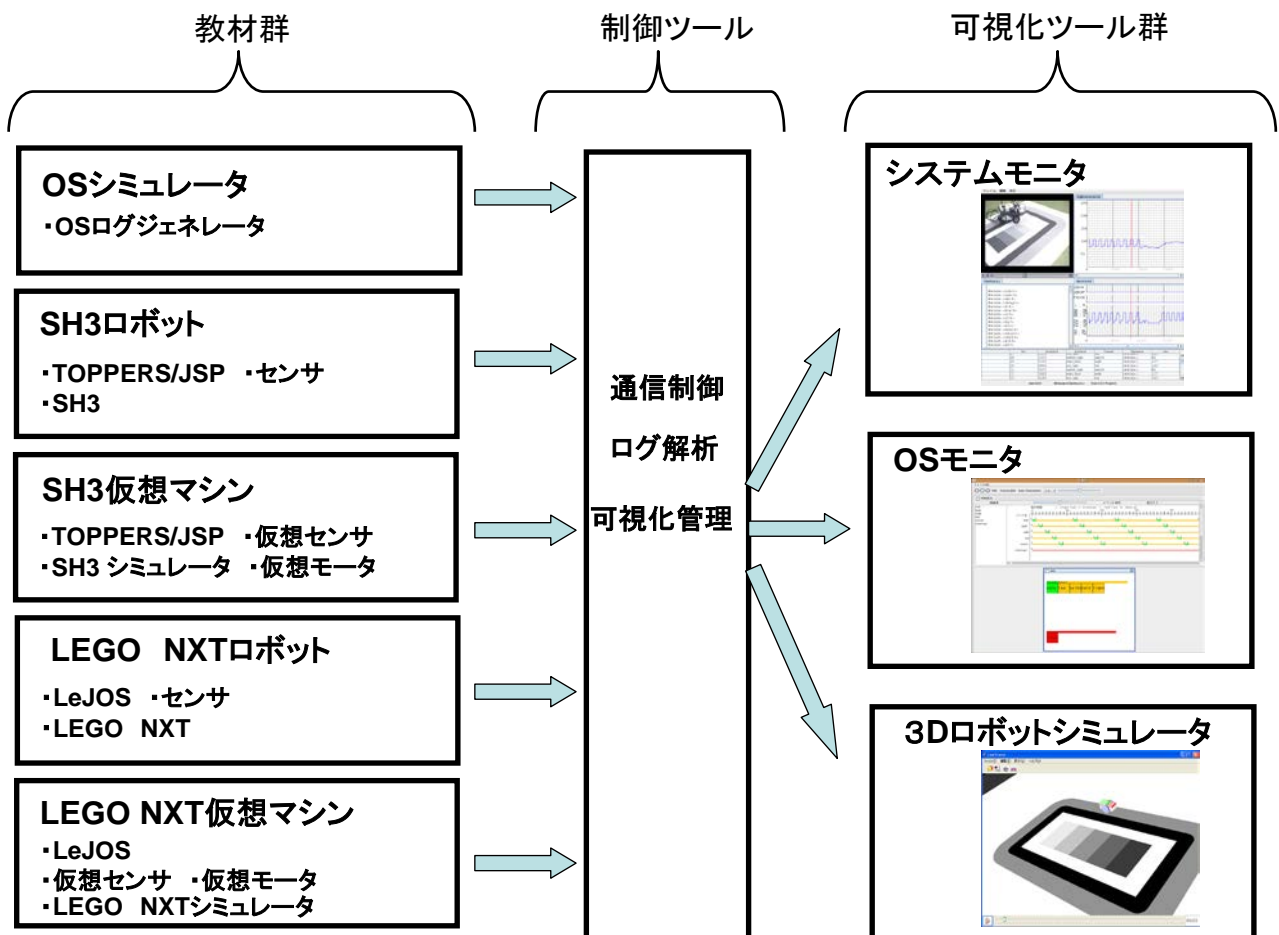


図 1 システムの全体構成

Fig. 1 Overall structure of the environment

3. 1 可視化ツール群

可視化ツール群は本システムの中核であり、生徒の組込みシステムの理解を支援するものである。本システムでは、一つの教材に対して、複数の可視化表現を提供する。従来の学習支援システムが、システム内の一つの現象の可視化に留まっていたのに対して、本システムでは、同じ現象を複数の動作や異なる可視化粒度によって生徒に見せていく。これによって、生徒は、組込みシステムの複数の側面をより具体的に理解することが可能になる。本システムでは3つの可視化を提供する。

(1) システムモニタ

システムモニタは、組込み装置が備えているセンサ類の入力を生徒に分かりやすい形で可視化して提供する。組込みシステムでは、センサからの入力イベントによって各タスクを起動する形態を採用することから、センサ値の時系列での変化を把握することは重要である。また、システムモニタでは動画による組込みシステムの動作の可視化を合わせて提供する。生徒は、ロボット自体の挙動を監視しているので、その時点では現象を把握している。しかし、挙動を精査し、デバッグやテストを行う段階では、挙動とセンサ値の変化やプログラムのフローを、関連付けてチェックすることが難しくなる。その理由として、ロボットの挙動を完全に再現させることができないことが挙げられる。実世界のシステムでは、ロボットの初期形態、バッテリー残量によるモータのスピード、センサ値の変化、外乱などといった物理現象が装置に影響を与えるために、同じ動作をさせることは難しい。このことから、従来では生徒に興味をもたせやすいロボットを、組込みシステム教育の演習教材として採用することが困難であった。これに対して、本システムでは、動画、センサ値、プログラムフローをそれぞれ関連づけて表示することでこの問題を解決する。

図2に、システムモニタのスクリーンショットを示す。システムモニタは、左上にWebカメラから取得したロボットの動作を動画によって表示する。この動画の挙動に連動して右上の入力デバイスのフォトセンサの値、右中の出力デバイスであるモータのデューティ値、LEDの変化をダイアグラムとして表示している。左中では、そのときの実行中のソースコードを表示し、ロボットの挙動、センサ、出力デバイス、プログラムの対応を1画面で把握することが可能である。また、スライドバーによって、動画を任意のポジションから再生することが可能である。これによって、生徒は、納得するまで何度でも動作を見直すことができる。これによって、センサ値の変化がシステムにどのような影響を与えるかを知ることができる。このダイアグラム表示は、組込み装置から取得したログデータを動画データと同期させることによって実現している。

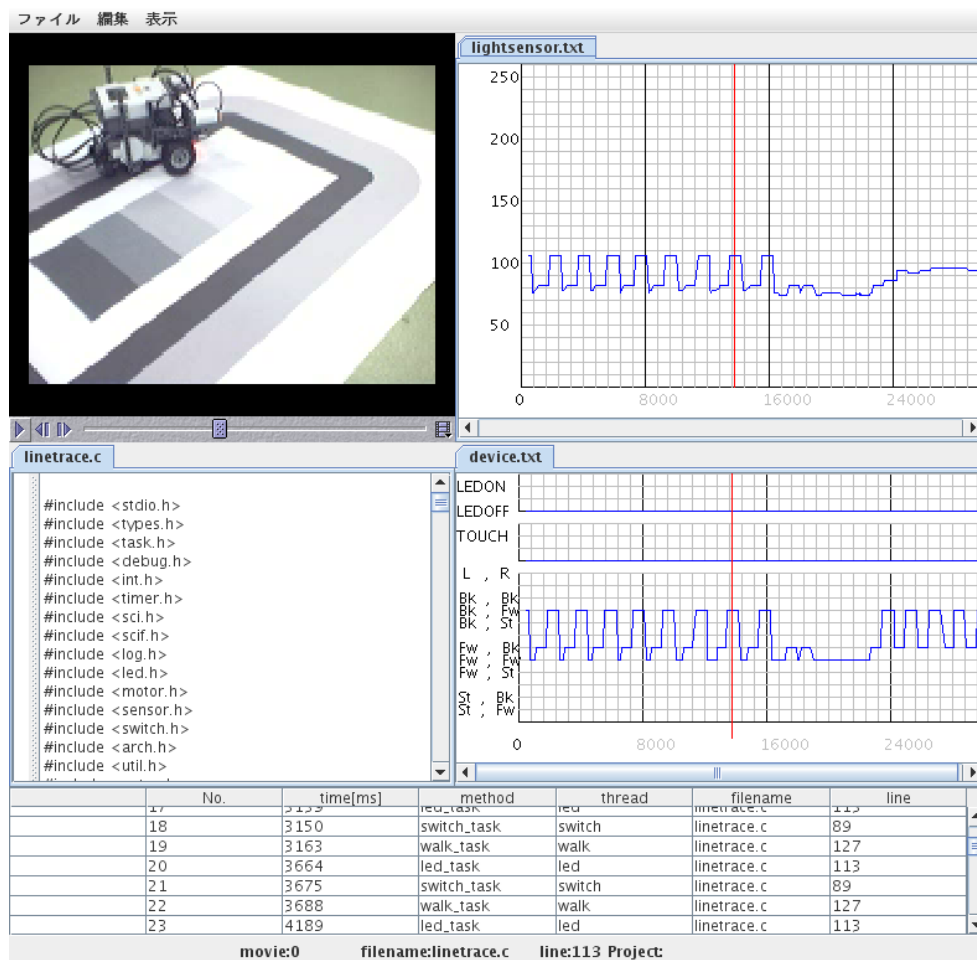


図 2 システムモニタ

Fig. 2 Screenshot of system monitor

(2) OS モニタ

OS モニタは、組込み OS の内部を図的に表現する。組込み OS の利用を考えた場合、組込みシステムの動作は多層的になることから、複数の階層での動作の監視が必須となるが、従来のテキストベースの表示ではその把握が難しい。さらに、センサからのイベントのように、同時発生する複数の現象を把握するには、情報を適宜取捨選択できる必要がある。このための可視化表現として図的な表現が必要になる。

また、これらの情報は、従来隠蔽されており、外部からその動作をモニタリングすることが難しかった。単純なロギング情報による表示では、階層ごとの同時動作を理解することは難しい。これに対して、本 OS モニタは適度に抽象化しつつ、同時にその変化を表示することで、導入学習から応用学習への連続的な支援を可能にする。

本 OS モニタはタスクの状態変化を、スケジューリングと合わせて表示することで、OS 内部状態の変化を可視化している。これによって、OS の動作を把握するとともに、OS 上で生徒が作成したアプリケーションの動作を把握することが可能になる。図 3 に OS モニタのスクリーンショットを示す。



図 3 OS 可視化ツール

Fig. 3 Screenshot of OS visualization tool

上側のウィンドウは、タスクスケジューリングの時間的な変化を示している。図 3 の例では、ラウンドロビンスケジューリングのタスクスイッチタイミングについて示している。このようなグラフ表示によって、実際の頻度と実行時間が適切に対応しているかどうかを直観的に理解することができる。

また、可視化はより直観的な理解を得られるよう、内部動作の状態遷移を配色によって表現する。特に、実行中は緑、実行待ちは黄、停止は赤と、信号色を用いることで、生徒は直観的に理解することができる。

(3) 3D ロボットシミュレータ

3D ロボットシミュレータの動作画面を図 4 に示す。本システムでは、システムモニタに用いるロボットの動作動画の代わりに 3D ロボットシミュレータをシステムモニタの画面に表示させることが可能である。この機能により、生徒は新しく操作を覚えることなく、従来のシステムモニタとシームレスに操作することができ、生徒のシステム操作学習にかかる時間を削減することができる。

3D ロボットシミュレータは、動画表示と同等の環境を提供する。さらに、3D ロボットシミュレータの特徴として、次のものが挙げられる。

- 3D シミュレータを用いた学習のサポート
- ロボット等ハードウェアからのログの再生機能
- ログ再生の巻戻し、早送りによるデバッグのサポート
- 従来のシステムとの統合

これによって、各ロボットシミュレータ動作による結果の表示だけでなく、実際の装置やロボットからのログデータによる動作を表示することができ、ロボットの挙動を、二つの視点から比較検討できることになる。

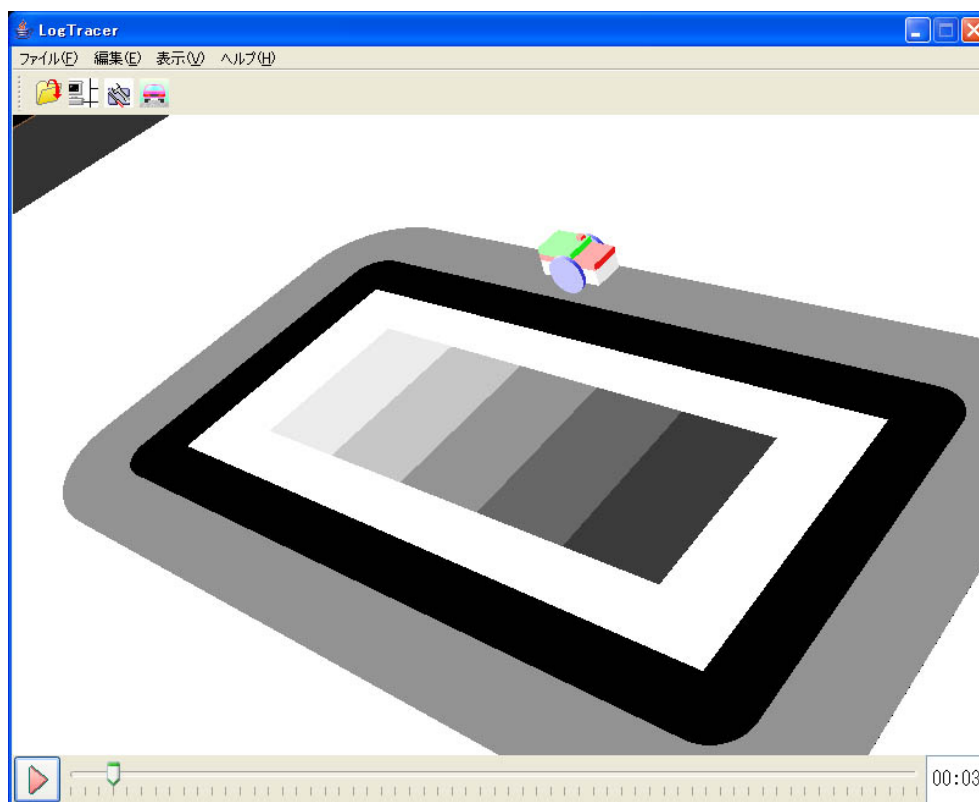


図 4 3D ロボットシミュレータ

Fig. 4 Screenshot of 3D robot simulator

(4) 制御ツール

制御ツールは通信制御、ログ解析、可視化管理を行っており、本システムのフレームワークとして動作している。

通信制御は各教材群と可視化ツール群のデータのやり取りを中継し、それぞれの通信におけるデータの差異を吸収している。これにより、教材群と可視化ツール群の任意な組み合わせを実現すると共に、新たな教材、可視化ツールの作成時においても本フレームワークの通信プロトコルに合わせるだけで本システムの一環として動作することができ、教材作成の手間を減らすことが可能となった。

また、ログ解析および可視化管理では各教材群より送られてきたログデータを解析し、可視化データとして変換した後、可視化ツール群に提供するものである。可視化データには可視化に必要な OS の動作ログ、ロボットの内部状態、各センサ値、モータ出力、実行プログラムの箇所、挙動のビデオデータなどを含んでいる。

3. 2教材群

教材群は組込みシステムの内部動作を、実際の動作として表現する教材である。組込みシステムの学習であっても、従来のワンボードコンピュータのような単純な入出力装置を備えた教材では、生徒がその興味を持続させることが難しくなっている。また、シミュレータ環境では、実際の物理現象から来る問題を体験できず、組込みシステムの学習としては不十分である。

これに対して、ロボットはソフトウェアの動作をロボットの挙動として見せることができ、生徒の意欲を高めることができる。その一方、外乱など物理的な現象によって動作が再現しにくく、教材として扱いにくい側面

がある。

本システムでは、生徒が動作させて面白いと感じること、学習しやすいように、単純な構成であることの両方を満たす学習教材を提供する。それぞれの詳細について述べる。

(1) OS シミュレータ

OS シミュレータは組込みシステム学習における初期段階で組込み OS の動作、機能を擬似的に再現するものである。組込み OS の動作を具体化し、初学者の混乱要素である複雑な動作を排除した動作やログを作成し、概念学習に必要なイメージの理解を促進することができる。

さらに、任意のタイミングや、実際に実行した組込みシステムからのログデータを入力データとして与えることができる。これによって、生徒はいろいろなタイミングでの動作を検証することができ、プログラムのテストに役立てることが可能である。

(2) SH3 ロボット

SH3 ロボットは、組込み OS を動作させて、組込みシステムによる装置制御の結果を、生徒に目に見える形で表示させるための実装的な教材であり、TOPPERS/JSP を搭載している。SH3 ロボットは自走型ロボットで構成し、CAT709(SH-3、117MHz)CPU ボードをベースとして、32MB メモリ、および無線 LAN の通信環境を提供している。また、タッチセンサ、可視光センサなどを備えている。また、仮想マシンモニタ上で OS を動作させることで、生徒がハードウェアを単純化して扱うことやドライバ類を安全に作成することが可能になる。

生徒は SH3 ロボットの内部動作をロギングし、内部状態を可視化することで挙動、ソースコード、内部動作を関連付けて理解することができる。ロボットからは、システムイベントログ、センサ値、割り込み、メモリアクセス、タスクスケジューリングのデータが PC 側に送られる。これらの情報が、可視化の基礎データとなる。SH3 ロボットを図 5 に示す。

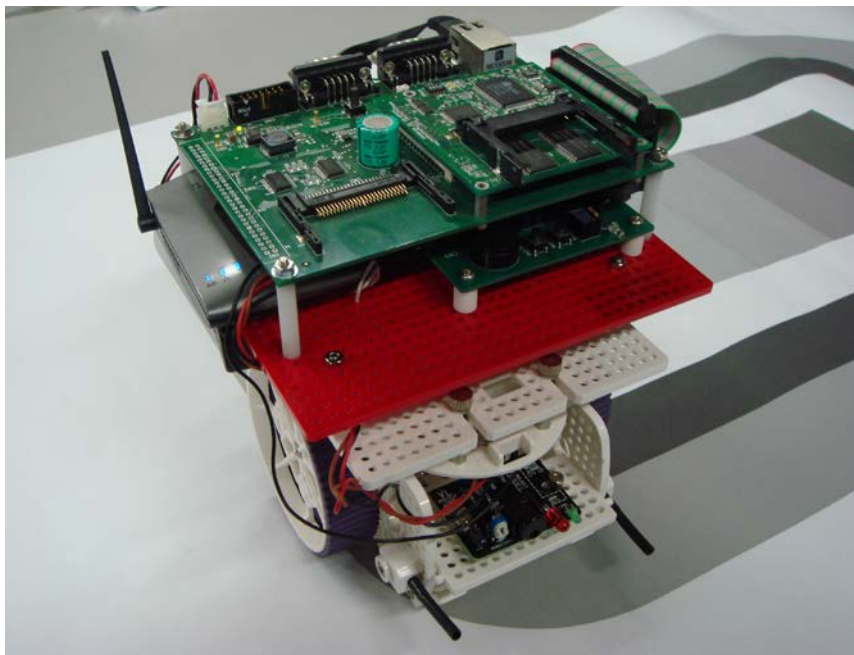


図 5 SH3 ロボット

Fig. 5 SH3 robot

(3) SH3 仮想マシン

SH3 仮想マシンは、前述の SH3 ロボットを仮想的に実現するものである。SH3 仮想マシンは、SH-3 CPU シミュレータおよび、仮想センサ、仮想モータを備え、SH3 上で動作する TOPPERS/JSP を制御 OS として搭載していて、SH-3 ロボットで用いるユーザプログラムをそのまま PC 上で動作することができる。各可視化ツール群は、ロボットが理想的なコースを走行した場合の動作をテストすること、およびロボットが存在しない在宅でのユーザプログラムの作成、走行テストを可能にしている。これによって、教育機関における授業時間の制約による学習時間の不足を解消する。

(4) LEGO NXT ロボット

実装学習段階の教材として Lego Mindstorms NXT を提供する。NXT を採用した理由はロボットをそろえるコストや、より手軽に本システムを試すためのプラットフォームを考慮した結果である。NXT ロボットを図 6 に示す。



図 6 NXT ロボット

Fig. 6 NXT robot

NXT は CPU に 32bit の ARM7 と 8bit の AVR を搭載し、USB や Bluetooth で外部と通信することが可能である。また、各種センサもそろっており、手軽にロボット環境を試すことができる。さらに、NXT 上で動作する LeJOS を組み入れることで、Java による制御プログラミング環境に加え、学習を支援する可視化ツールを利用することで、より視覚的、直観的に理解できることが可能となる。

NXT からは、システムイベントログ、センサ値、割込み、メモリアクセス、タスクスケジューリングのデータが PC 側に送られる。これらの情報が、可視化の基礎データとなる。

(5)LEGO NXT 仮想マシン

LEGO NXT 仮想マシンは、前述の LEGO NXT ロボットを仮想的に実現するものである。LEGO NXT 仮想マシンはNXT シミュレータおよび、仮想センサ、仮想モータを備え、NXT 上で動作する LeJOS を制御 OS として搭載している。実際の NXT ロボットで用いるユーザプログラムをそのまま PC 上で動作させ、その挙動を可視化ツール群で確認することが可能となる。これにより作成したユーザプログラムを仮想ロボットで動作した際の基本的な挙動の確認や、ロボットが理想的なコースを走行した場合の動作のテストが可能となる。

3. 3学習教材の作成、カスタマイズ

本システムは学習教材作成やカスタマイズにおいて、教員の負担を軽減できることを方針の一つとしている。本システムにおいて、教員がカスタマイズできる部分は、図 1 における教材群である。教材群は可視化を行うためのログデータを採取するモジュール以外は自由にカスタマイズすることが可能であり、これらを利用した実習教材を作成することができる。例えば、各ロボットや仮想マシンにおいて、組込み OS のスケジューラを変更し、より効率的なスケジューリングを生徒に考えさせ、その結果を可視化するなどの利用を想定している。

また、図1の可視化ツール群や制御ツールを含めて Java で記述されていて、改良は容易である。各部分のモジュール化、およびログデータや通信プロトコルを扱うライブラリの提供により、新しい教材作成も容易となっている。

4. 学習環境の 1CD/1USB 化

4. 1要求分析

本システムは、弾力的、統合的な教育支援を実現するため、さまざまな教材、ツール群から構成されている。従来では、本システムの環境構築や多くの PC への導入に際し、大きな手間がかかった。そこで、本システムを統合し、導入コストや運用コストを減らすことができる環境を作成する。

本システムの教材群および可視化ツール群の組み合わせの際および学校導入時の教員の要求を次に述べる。

(1)簡単に本システムを設定できること

本システムを導入する際に、手軽に、誰でも試すことができることが条件となる。本システムは高等教育機関のみならず、中学校の技術家庭におけるロボット教材の利用や、高等学校普通科における教科「情報」、産業高等学校など中等教育機関での利用も想定している。そのため、PC に関する知識が多くない教員や生徒でも、簡単に導入、設定できる学習支援システムが望ましい。

(2)スペックが低い PC でも動作すること

本システムは、ロボットを用いて組込みシステム学習をサポートするが、ロボットが用意できない環境や、自宅での復習などの利用も想定し、各ロボットシミュレータや 3D ロボットシミュレータを搭載している。そのため、どの PC でも手間をかけずに同じ環境を構築できることが望ましい。

(3) 本システムを簡単に初期状態に戻せること

多くの学校での学習用 PC 環境は、1 台の PC を不特定の生徒が利用する。故意、偶然を問わず生徒がシステムを改変し、不具合が生じた場合、教員はその修復に多くの時間を割き、授業展開に支障が生じることが非常に多い。そのため、常に同じ状態で教材が起動できることが望ましい。さらに、生徒にシステムの改変を公開することが可能となる。

4.2 方針

前述の要求を満たすために、本学習環境を 1CD/1USB にまとめた。これにより、学習環境の導入コストの削減や運用時の管理コスト削減が期待できる。OS には Ubuntu Linux を用いた。これは、LiveCD/LiveUSB により、PC の場所を選ばないこと、OS と環境導入が CD 一枚や USB メモリ単体で済むこと、ウィンドマネージャや日本語環境が整備されていることが理由である。

その他の可能性として、仮想マシンを用いた環境が考えられるが、学校、自宅を問わず、気軽に、多くの知識を必要とせずに学習環境を構築できるのは 1CD/1USB 環境が優れていると判断した。

4.3 導入事例と結果

1CD/1USB 化した本学習環境の導入事例およびユーザビリティの検証を行った。導入実験では、実習の一環として、システムソフトウェアの概念教育および、ロボット制御を想定した組込みシステム教育を行う環境を対象とした。すでに少数の PC に Windows 環境による本システムの導入がなされているが、管理、導入のコストの問題点が明らかとなっていた。これらの問題を解決しつつ、従来のユーザビリティを確保できるかどうか導入実験の目的である。なお、校内事情により、LiveUSB ではなく LiveCD を用いて実験を行った。

実験の詳細を表 1 に示す。また、対照群として、従来の Windows 環境への導入事例を合わせて示す。

実験環境：

台数 40台(デスクトップ型20台、ノート型20台)

表 1 導入実験結果

Table 1. Result of Initial Cost

| | LiveCD 環境 | 従来環境 |
|--------------|--------------|------------|
| OS | Ubuntu Linux | Windows XP |
| 導入時間(1 台あたり) | なし | 約 1 時間 |
| 起動時間 | 約 90 秒 | 約 30 秒 |

導入実験から得られた知見

導入実験から得られた知見を、導入コスト、運用コストそれぞれの利点、欠点をから述べる。

(1)導入コスト

導入に際する利点は次のとおりである。

・総導入時間の削減

従来の環境ではインストールの時間が、1 台あたり約 1 時間かかり、120 台に準備するには多くの時間が

かかった。しかし、LiveCD 環境では、インストールの手間が省けるため、大きなコストダウンになる。

- 1 台あたりの導入の手間

従来の Windows 環境では、本システムの導入に、システム本体、実行環境(JAVA)、各種ドライバ(Web カメラ等)、開発環境など様々なインストール作業に時間がかかった。加えて、すべての実習用 PC にインターネット環境が整っているわけではなく、スタンドアロンの PC が多く存在する。これらの面からも、各種ドライバや環境をメディア等で配布しなければならないという手間が削減できた。

- 手軽さ

本システムは、中等教育でも利用できる教材として設計しているため、利用時は PC に関する多くの知識を必要としない。しかし、本システムの導入にはある程度の PC の知識が必要となる(JAVA やドライバなどのインストール作業)。学校現場の教員すべてがこれらの知識を持っているわけではないため、LiveCD で手軽に利用できるのは大きなメリットである。

一方、導入に際する欠点は次のとおりである。

- 台数分の CD 作成の手間

本システムを同時に利用する場合、その台数分だけの CD を作成しなければならない。一度作成すればよいが、大きな手間である。

- 起動時間が遅い

LiveCD 環境のデメリットとして、起動が遅い点がある。生徒によっては、この起動時間が待てなく、集中力が切れてしまうことがある。

(2)運用コスト

運用面での利点は次のとおりである。

- 常に同じ状態で起動できること

LiveCD のメリットとして、ローカルディスクへの書き込みがないことがある。中等教育の場合、興味本位や知識不足、あるいは悪意から、ローカルディスクの改変を行い、次の生徒が使えないと言う事例が非常に多い。LiveCD によってこれらの問題は排除できる。また、作成したユーザプログラムは、LAN 内のファイルサーバや USB ストレージ等に保存するため、連続的な実習が可能である。

- GUI、日本語環境の充実

以前の Linux 環境では、GUI や日本語環境が整っていないく、“Linux は難しい”というイメージから、敬遠していた教員も少なからず存在した。しかし、現状ではこれらの問題は過去の問題となっており、初めて Linux 環境を使う生徒でもまったく問題なく利用することができる。

一方、運用面での欠点は次のとおりである。

- 動作が遅い

LiveCD 環境の本質から動作が遅いことがある。生徒によっては起動時の問題と同様に、集中力や意欲が続かない場合もある。

4.4 考察

導入実験から得られた知見から、LiveCD 環境で明らかとなった点を述べる。

- ・導入コストの大幅な削減ができる

LiveCD によって、環境のインストールの手間を省くことが可能となった。特に高等学校など、多くの実習環境を準備しなければならない事例では、教員の手間を大きく除くことができ、その分、教科指導へ時間を割くことが可能となった。

また、CD を入れて起動するだけで学習環境を利用することができるので、PC に関する知識が少ない教員でも気軽に試すことが可能となる。

- ・常に同じ環境を提供できる

LiveCD の特徴として、起動の度に同じ初期状態の環境を呼び出すことができる。これは、学校のような、1 台の PC を不特定の学習者で共有する場合には非常に有効である。故意、偶然を問わず、システムを改変し、次の利用者が学習できない事例が多いが、LiveCD により起動毎にリセットされるので、PC の保守、管理コストの削減、生徒が自由にシステムを触ることができることが利点としてある。

5. 産業高校における評価実験の実践と考察

5.1 産業高校における評価実験の実践

本システムの有効性を検証するために、産業高校の実習において評価実験を行った。今回の評価実験では、産業高校生を対象とし、ロボットを用いた制御プログラミング学習において、従来の組込みシステム教材に比べ、本システムのロボット教材および可視化ツールの有効性を検証することが目的である。実験授業の対象となる学生および講義は次のとおりである。

科目名:システム実習Ⅱ(3学年)3単位

対象校:東京都立八王子桑志高等学校

対象人数:産業科システム情報分野37名

回数:計6週18コマ(1コマ45分)

生徒の前提知識:プログラミングや制御に関して、入学より2年間一通り学んでいる。ロボット環境での制御プログラミング実習は別のキットロボットを用いて経験している。

本授業のねらいは、これまで行ってきた制御に関する実習と比較し、自律ロボットを用いて制御とプログラミングの関係、少ない資源下でのプログラム開発、リアルタイム OS を含めた組込みシステムの理解を目的としている。特に今回の実験授業では、制御とプログラミング両者を総合的に学び、組込みシステムの概念理解へつなげていくことを目的としている。

機器制御に関する実習はこれまでにシーケンス制御、組み立て型ロボットを用いた制御などを経験している。これらの制御教材は、制御とプログラミングとの関係性が薄く、プログラミング、内部動作、挙動が関連づけて理解しにくい。

実験授業は3グループに分け(1グループ15~16人)、課題解決に重点を置いた授業を行った。ロボットはLEGO Mindstorms NXTを用いて、それぞれROBOLAB Ver2.9のみ、LeJOS環境のみ、および本システム(教材群であるLEGO NTX 仮想マシンと、NTXロボット、および3種の可視化ツール群で構成、以下本システムの実験構成)を用いて行った。

ROBOLABはタフツ大学で開発されたGUIベースのプログラミングソフトウェアであり、タイルプログラミングによるコーディングができ、学習者が直観的に制御プログラミングを体験することができるため、多くの教育機関において利用されている。しかしROBOLABはNXTの内部状態を把握することができないため、プ

プログラムとロボットの挙動を関連付けて理解することは難しい。

一方、LeJOS は NXT 上で Java によるプログラミングを行うことができるため、Java の初学者などの導入教育としても利用されている。LeJOS は Java による NXT 制御を行うことができるため、ユーザプログラムの自由度が高い。しかし、ROBOLAB と同様に NXT の内部状態を視覚的に把握することができないため、本授業で目指している制御プログラミングの直観的な理解を得ることは難しいと予測できる。

講義は、NXT の使い方、制御プログラミングの概要、各種センサの利用方法および入力センサ値に対応したプログラミングを利用し、ライントレースのプログラミングに加えてマルチスレッドプログラミングについて行った。なお、LEGO という自由度の高いハードウェアを用いるに当たり、ハードウェアの変更による問題可決を避けるため、ロボットの組み方は両者同じものを使用した。筆者らの 10 年以上のロボットを用いた教育の経験から、ロボットを用いた制御プログラミングの導入学習では、本来ソフトウェアレベルによって解決すべき問題を、ロボットのブロックを組み替えることで解決しようとする傾向が強い。今回の実践では、制御プログラミングや組込みシステムの理解を目的としているため、ハードウェアの変更による問題解決は極力させないものとした。

図 7 に LEGO を用いた実習風景を示す。

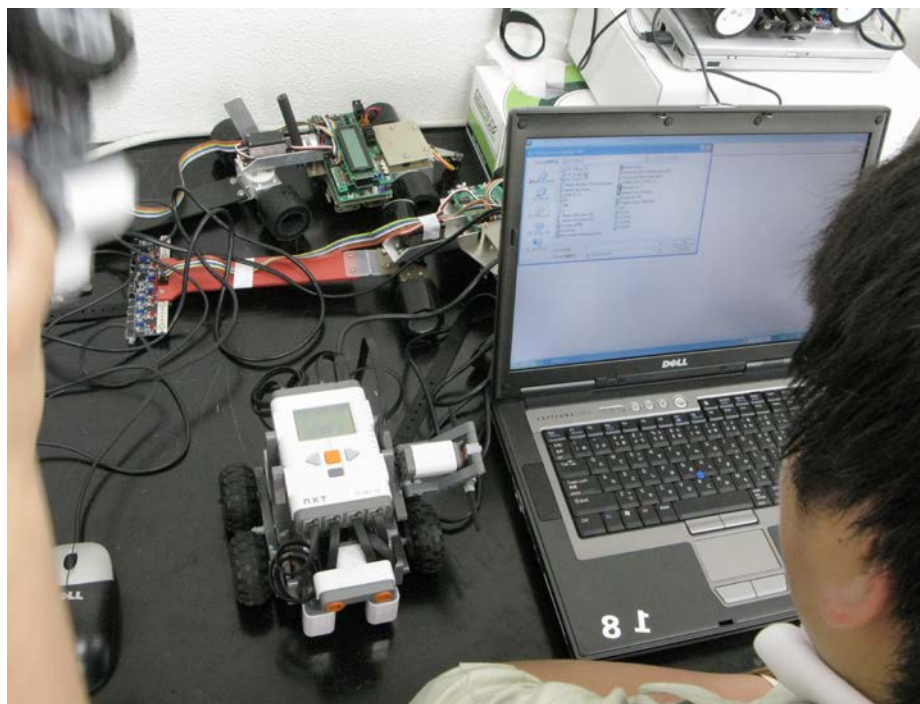


図 7 LEGO による実習風景

Fig. 7 Screenshot of lesson using LEGO

5.2 授業の実践から得られた生徒の様子

産業高校における授業の実践を通して得られた様子を、ROBOLAB、LeJOS および本システムの実験構成それぞれの観点から述べる。なお、これらの生徒の様子は、教員の教授活動の過程において、机間巡視、相談、レポートや雑談から知り得たものである。

(1) ROBOLAB

ROBOLAB を用いた実践では、直観性の高いタイルプログラミング環境によって、特にプログラミングの知

識が低い生徒でも短時間でプログラミングインタフェースを理解していた。

しかし、頭の中で思い描いていた挙動と、実際に動かしてみた挙動が大きくかけ離れており、何度もパラメータやプログラムを変更し、試してみるといったプログラミングを行っていた。また、そもそものロジックを理解していないため、まったく違う挙動が現れても、なぜそうなるのか理解できないといった例も見られた。これは、制御プログラミングの本質を理解していない、または内部動作がイメージできないといった原因が考えられる。

さらに、動作に再現性がないために、バッテリーの状態や照明、スタート位置などの外因に挙動が左右され、そのつどプログラムを直す試行錯誤な作業が続き、効率の悪い実習となった。

(2) LeJOS

LeJOSを用いた実践では主に Java 言語を利用するため、ROBOLAB のタイルプログラミングと比較してプログラミング自体に苦勞している様子が見られた。さらに、ロボット内部の様子や、センサの値、挙動が予想できないため、何度も試行錯誤する様子が見られた。限られたリソースでのプログラミング作業やデバッグ作業は高校生にとって非常に難しく、組込みシステムの理解意欲が失せてしまい、組込みシステム嫌い、プログラミング嫌いを生んでしまう結果にもなりかねない。

(3) 本システムの実験構成

本システムの実験構成を用いた実践では、主に Java 言語によるプログラミングを行ったため、ROBOLAB のように手軽にロボットを動かしてみるという様子は見られなかった。

しかし、本システムの実験構成では、内部動作やセンサ値をシステムモニタリングツールによって可視化、挙動を再現できるため、ROBOLAB や LeJOS を用いたときとは対照的に、試行錯誤が少なく、数回のデバッグで思ったとおりの挙動を得ることができていた。今まで学習したプログラムや制御などの知識が今までは断片的な理解であったが、本システムの実験構成を用いたことで関連付けて理解できたという生徒も現れたことは大きな成果である。

5.3 評価実験

本システムの実験構成の有効性を検証するための評価実験を行った。実験は、本システムの実験構成を実験群、ROBOLAB 環境、LeJOS 環境を対照群とし、1 グループ 15 名で学力がほぼ対等となるように分け、課題解決による対照実験によって有用性を判定した。具体的には一人 1 台の NXT を使い、15 名がそれぞれ課題を解決する方法を取り。ライントレースの課題をクリアするまでの時間、試行錯誤の工程(ロボット実際に走らせ、プログラムのデバッグを行った回数)を比較した。

また、プログラム規模は ROBOLAB においてはプログラムタイル約 10 パーツ、LeJOS では 100 から 150 行、本システムの実験構成でも同様に 100 から 150 行である。課題解決時間やデバッグ回数は教員による机間巡視および自己申告での記録である。なお、表内の数値は 15 名の実験結果の平均値である。評価実験結果を表2に示す。

表2 評価実験結果

Tab. 2 Result of evaluation test

| 使用環境 | クリア時間 | パラメータ調整 | ロジック変更 | 大幅修正 |
|------------|--------|---------|--------|-------|
| ROBOLAB | 1時間45分 | 32.5回 | 7.2回 | 7.7回 |
| LeJOS | 3時間15分 | 78.9回 | 13.5回 | 11.2回 |
| 本システムの実験構成 | 1時間11分 | 12.3回 | 5.4回 | 4.5回 |

評価実験結果から、本システムの実験構成を用いた実験群において、課題クリア時間およびデバッグ回数は、ともに少ない時間や少ない工程で実現できていることが明らかである。

次にアンケート結果を述べる。アンケートは各 15 名に対し無記名で行ない、利用環境のよかった点、悪かった点、感想を記述式で取った。アンケートはそれぞれの環境の印象について意見の多かったものを挙げている。

(1) ROBOLAB

- ・タイルの組み合わせのできるのでわかりやすい
- ・数値も見えるのでパラメータの調整がしやすい
- ・同じ動作ができないので、何が原因なのかわかりにくかった
- ・簡単すぎてつまらない、もっと本格的なロボット制御をやりたかった

これらの結果から見えるように、ROBOLAB 等タイルスタイルによるプログラミングは、生徒がプログラムを直観的に理解でき、興味を引いていることがわかる。しかし、ロボットの内部状態を把握することが難しく、動作を再現することが難しいため、外乱の影響を考慮したプログラミングを行うこと自体が難しい様子が見られた。

(2) LeJOS

- ・プログラムを考えるのが難しかった
- ・ロボットの様子や動作がプログラムと一致しなかった
- ・思っていた動作ができなくて苦労した
- ・最後までクリアできなかった

LeJOS を用いた感想からは、制御プログラミングにおいて、プログラム作成で想定した動作と実際の動作が一致せず、何が原因なのか検討がつかないことがわかる。外乱による不具合なのか、プログラムミスなのかそもそも区別できず、途中でモチベーションを切らしてしまう生徒も多いた。高校生を対象とした制御プログラム実習では、生徒の理解を促進させてやる支援環境の必要性がある。

(3) 本システムの実験構成

- ・ロボットの動作がビデオでもう一度見られるのがよかった
- ・プログラムは難しかったが、ロボットの動きとリンクしているのがよかった

- もっと早い学年でやってみたかった
- CDでもできるので家でもやってみたい
- プログラムを作る画面が使いにくい

本システムの実験構成を用いた感想からは、ロボット制御への興味を示し、プログラムとロボットの挙動を結びつけて理解していることがわかる。これは本システムの実験構成によるロボット内部動作の可視化が生徒に有利に働いた結果であると言える。また、本システムの実験構成の導入の手間が低いため、自宅において自学自習してみたいという意欲が現れたことは大きな成果である。プログラムを作成する段階では、本システムの実験構成のプログラム作成画面の改良の必要性や、Java 以外の、より導入教育をしやすい言語の利用も検討する必要性が生じた。

6. 考察

評価実験から得られた様子やアンケート結果から、本システムの実験構成の学習効果について明らかとなった点を述べる。前述したように、評価実験結果から本システムの実験構成を用いた場合、ROBOLAB や LeJOS 等既存の教材に比べ、課題クリア時間、デバッグ回数ともに少ない時間、工程で実現できていることが明らかである。工程時間が少なく、デバッグ回数も少ないということはロボットの内部動作を理解し、課題解決を促進していると解釈できる。

(1) 内部動作の可視化によって生徒の理解が促進されることを確認した

表2のデータより、本システムを用いた場合のクリア時間が最も短く、パラメータ調整の回数が最も少ない。これは、可視化によってプログラム内のパラメータの調整時間の減少が要因である。可視化環境では、生徒が設定したパラメータがどのような状況で動作に関連しているのかを示すことができることから、原因の特定が容易になっている。このことは、ロジックの変更が少ないことから分かる。これに対して、ROBOLAB および LeJOS では、パラメータ値と動作の調整とか関連づけにくく、ロジックの変更による原因の切分けが必要であり、試行回数が増加していると考えられる。

これらのことから、本システムの方針が、2章で述べた産業高校での組込みシステム教育の生徒側の問題を解決できたと考えられる。

(2) 統合的な教育環境が導入教育では必要であることを確認した

産業高校のように、専門性の高い学習を前提とする環境では、導入教育での理解の徹底が重要であり、導入段階で躓いてしまった生徒は、発展的な学習や実習について行けず、学習意欲を低下させる場合が多い。本システムは多くの機能を持つ教育環境であるが、生徒のアンケートからは肯定的な評価が多く、生徒の学習意欲を維持向上させるのに貢献しているといえる。また、LeJOS と同様の Java による制御プログラミングであるが、ロボットや、可視化および統合環境を提供し、制御やシステムの動作を分かりやすく示すことで、生徒が初期段階で躓くことを防いでいる。このことから本システムのように、統合的かつ直観的に学習できる環境によって、導入教育段階において生徒の理解を促進することができることを確認した。また、ロボットのように興味を引く教材を用いることで、生徒の学習意欲を得ることができることを確認した。

(3) プログラミング環境の見直しの必要性を確認した

今回の環境では、プログラミング環境では Java 言語を用いた。現状ではプログラミング環境は統合的に

まとまっていないため、生徒にとっては直観性に欠け、学習効率に影響がある。より直観的に扱える言語環境の導入が今後の課題として明らかとなった。

7. 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において、そのモデルおよび開発したツール群、さらに、学校現場での評価実験について述べた。

本システムが提供する特徴としては、学習対象の組込みシステムの挙動を、ロボットや動画と連動したシステムデータ、OS といった複数の視点から可視化できる環境を提供した点、およびそれらを単一のシステムでサポートしている点にある。これにより、産業高校生の組込みシステム学習において、可視化による理解の促進やそれに伴う効率的な課題解決を促進することが明らかとなった。

また、本システムを1CD、1USBでの動作を可能とした。さらに実際に高等学校において評価実験を行い、制御プログラミングの導入学習において、ロボットを教材とした本システムが有効あることを確認した。

今後の課題は、システムの公開による社会へのフィードバックである。また、これによって将来に向けて組込み技術者の増大、スキルアップに繋がるよう初等中等教育現場での普及を目指す。

実施場所

東京都立八王子桑志高等学校 産業科 システム情報分野

193-0835 東京都八王子市千人町4-8-1

042-663-5970

参考文献

- 1) LEGO Mindstorms <http://www.legoeducation.jp/mindstorms/>
- 2) LejOS, <http://lejos.sourceforge.net/>
- 3) 文部科学省:新高等学校学習指導要領 工業
http://www.mext.go.jp/b_menu/shuppan/sonota/990301d/990301n.htm
- 4) 志子田有光, 御園生高太郎, 桃野健, 宗形太, 松澤茂, 井口巖, 佐藤徳男, 河田拓朗, 川田徳明, 高橋一夫, 山下哲央, 五十嵐充志, 佐藤明子, 渡辺典子, 及川幸恵:高等学校におけるオープンソースソフトウェア活用教材の研究 Open School Platform Project の可能性, 電子情報通信学会技術研究報告, 106(437) pp.61-64 (2006)
- 5) 中西恒夫, 野田厚志, 北須賀輝明, 福田晃:模型を使った組込みソフトウェア設計教育試行とその評価, 情報処理学会研究報告, Vol.2005 No.118, pp.31-36, (2005)
- 6) 菊池達也, 大村光徳, 渡辺忠史, 安原雅彦: 教育訓練用組込みマイコン教材の試作, 職業能力開発総合大学校研究紀要第 23 号, pp108-114, (2008)
- 7) 藤枝直輝, 渡邊伸平, 吉瀬謙二: 教育・研究に有用な MIPS システムシミュレータ SimMips, 情報処理学会論文誌, Vol.50 No.11, pp2665-2676(2009)
- 8) ROBOLAB, <http://www.ceeo.tufts.edu/robolabatceeo/>
- 9) 経済産業省, 2005 年度組込みソフトウェア産業実態調査報告書, 2005 年 6 月
- 10) 名古屋大学組込みソフトウェア技術者人材養成プログラム,
<http://www.nces.is.nagoya-u.ac.jp/NEXCESS/>
- 11) i-GarnET, <http://www.micronet.co.jp/i-GarnET/>
- 12) Luiz Paulo Maia, Francis Berenger Machado, Ageu C. Pacheco: A constructivist framework for operating systems education: a pedagogic proposal using the Sosim, ACM SIGCSE Bulletin, Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education ITICSE '05, Volume 37 Issue 3. (2005)
- 13) James H. Hill, Aniruddha S. Gokhale : Education: Visual OS: design and implementation of a visual framework for learning operating system concepts, Proceedings of the 43rd annual southeast regional conference - Volume 1 ACM-SE 43. (2005)
- 14) Jones.D, Newman.A :RCOS.java: a simulated operating system with animations. Proceeding of the Computer-Based Learning in Science Conference. Rep. Tchecha (2001)
- 15) brickOS, <http://brickos.sourceforge.net/index.html>