

# 体験型プログラミング学習装置の設計・開発

伊 藤 敏

井 上 祥 史

## 概 要

本研究では、プログラミング入門者対象が学習意欲を維持しながら、本質的な点に集中できるシステム構築を目的とする。「見て、聞いて、触って」体で実感ができる学習装置をプログラムで動かすシステムである。本稿ではシステムに用いる学習装置の設計・開発について述べる。学習者は学習装置を意図通りに動かす課題で目的が明確になりアルゴリズムを考えやすくなる。また、具体から抽象、抽象から具体化へと思考の行き来が可能になる。これらを実現するために、ビット操作の学習装置を設計・開発し、インターフェイスを通じてコンピュータからプログラムを行うシステムを構築した。開発したシステムの特徴を下記にまとめる。

- ビット操作の学習装置としてライトレースロボットを設計した。4ビットの命令で平面（床）上を自由に移動可能で、床面の明暗情報を2ビット情報としてコンピュータへ送ることが可能である。
- コンピュータとのインターフェイスにUSBを用いた。これにより、専門的と思われるバスライン経由ではなく、安全にビット入出力情報をやり取りすることが可能である。
- プログラミングは通常の教育環境のコンピュータを用いることが可能である。

## 1. はじめに

情報化社会の進展とともに、社会活動に必要な情報の収集・処理・利用に情報システムがかかわるようになってきている。また社会（組織体）だけでなく、個人でも情報を判断・評価し、その結果に基づいて行動をすることが必要となってきている。情報技術の進歩は速い。新しい技術の導入によって、情報システムは大きく変化をする可能性がある。情報化社会は常に変化することを属性として持つ。

流動性をもつ情報化社会において、情報教育は定型作業が中心のコンピュータ操作教育から、情報化社会を生き抜くための、独創性・創造性を育成する教育観へ移行している<sup>[1],[2]</sup>。すなわち、「問題を発見し、解決をする」という能力が要求されている。このような能力の育成に、プログラムを作成する教育は有効な方法の一つである<sup>[3]</sup>。プログラムを作成する教育、すなわち、プログラミング教育では課題に対して、問題点を発見し、解決の方法を探して、それに基づき変数とアルゴリズムを設計し、それらをプログラミング

言語へコーディングする必要がある。アルゴリズムの設計過程で問題点を明らかにし、解決すべきことを抽出し、解決法を考える。さらに、コーディングの過程で、解決法の具体化を行う。これらの過程が大切である<sup>[4],[5],[6]</sup>。

プログラミングに対する態度はどうであろうか。普通高校の生徒72名にプログラミングに対する18項目の態度尺度項目、各5段階尺度（値が大きいほど肯定的評価）からなるアンケートを実施（参考：向後<sup>[7]</sup>）した。その結果から、つぎの三つの因子を抽出した<sup>[8],[9]</sup>。

- (1) 能力因子：分かる、得意だ、できない など
- (2) 好悪因子：好きだ、嫌いだ など
- (3) 効用因子：必要だ、役立つ など

能力因子は平均で1.73、好悪因子は2.19、効用因子は3.22あった。これらは、プログラミングは「できない」、「嫌いだ」と考えており、一方で「必要だ」「役立つ」と感じている結果である。また、調査件数は少ないが大学生においてもほぼ同様の結果が得られた。すなわち、学習者はプログラミングの必要性は感じつつも、「できない」、「嫌いだ」として、困難を感じている。これら学習者の苦手意識の状態を克服するには、入門段階では本質な事柄、アルゴリズムの基本である順次処理・条件分岐・繰り返し処理に集中できるよう配慮され、学習意欲を維持するシステムが必要である<sup>[10]</sup>。

プログラムした結果がロボットなどの学習装置の具体的なものの動きとして反映するシステムがあれば、学習意欲を維持するのに有効であろう。プログラミング入門者は、学習装置の具体的動きから、動きの手順（アルゴリズム）を経てプログラムを作成する。さらに、作成したプログラムの実行結果が学習装置の具体的動きとして反映することで、具体から抽象へ、抽象から具体へと繰り返し学習することになる。これにより学習意欲を維持しながら、能力を高めることが可能と考えられる<sup>[11]</sup>。

本稿の目的は、「プログラム⇔学習装置の動き」を実現するシステムを提案し、そのシステムの一部を担う学習装置の設計・開発を行うことにある。

以下に2でシステムと学習装置の設計の基本方針を、3で学習装置の開発を、4でまとめを述べる。

## 2. システムと学習装置設計の基本方針

### 2.1 アルゴリズムの学習

アルゴリズムの学習に、数学の解法を例にとると、論理的に明快<sup>[12]</sup>である。しかし、抽象的で難解と誤解される場合が発生する。では、課題を具体化すればよいかと言うと、必ずしもそうとはいえない。日常生活を想定した行動パターンを例にとる場合、自分の行動分析ゆえに具体的であり、困難が少ないように見える。しかし、日常生活では暗黙の了解

事項が多くあり、その暗黙の了解事項に気づかないとかえって困難な課題になる。また日常生活の行動の区切りにおける選択肢の多様性のために、区切り方によりかえって困難な課題になる場合がある。たとえば、「歯を洗う」行動を想定した場合、洗面所へ行って歯を洗う という行動を、洗面所まで行くプロセスも含めて行動分析を行った場合、「3歩前進したら、右折して4歩先で停止」などといった記述をする場合もありえる。

## 2.2 従来型学習装置

ロボットなどは自ら判断をする能力を持たないため、動作の順番を一つひとつ指示する必要がある。かつ、その指示結果である動作が視覚・聴覚・触覚で確認できる点で学習教材として大変優れている。一方で、学習装置として用いられる既存のロボットなどは自律型が多く、指令をするコンピュータ部分はマイクロコンピュータ（以下マイコンと略す）が利用されている<sup>[13],[14]</sup>。マイコンは機能が限定されていることから、開発言語は一般にアセンブラやC言語が用いられる。入門者が使うプログラミング言語としては敷居が高い。さらに機械工学・電子工学などの制御技術や知識が要求され、ハードウェアも高価になり、教育で幅広く実施することは難しくなる。

## 2.3 設計の基本方針

既存のロボットなどの指令部分であるマイコンをデスクトップやノート型コンピュータに置き換えことで、制御に用いるプログラミング言語の選択肢が広がる。さらに、学習装置ハードウェアを数ビットの情報に基づき操作可能なものに限定し、動作を単純化することで、コンピュータとの接続も含め、学習装置の取り扱いが容易（すなわち制御の単純化）になり、価格も低く抑えることが可能になる。かつ、音が出て動きが目で確認できる利点は残すことができる。

そこで、次に述べる条件を満たすシステムを設計する。

1. 命令を出す部分は通常の教育環境にあるコンピュータを用いる
2. 学習装置は数ビットの命令で動作する
3. 学習装置から数ビットの周辺情報をコンピュータに供給可能にする

プログラミング学習システムの概要を図1に示す。大別すると学習装置（ハードウェア）、インターフェイス、プログラミング作成環境のソフトウェアが載ったコンピュータの三部分で構成される。学習者はソフトウェアからインターフェイスへビット情報の入出力指示を行うことで学習装置を操作する。本稿では学習装置であるハードウェア、入出力インターフェイスについて述べる。ソフトウェア環境については別稿に譲る。

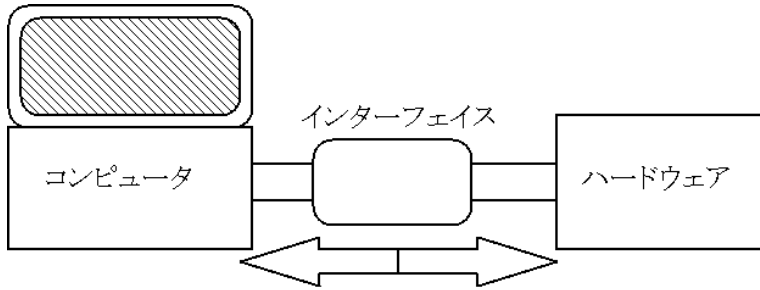


図1. プログラミング学習システムの概要

### 3. 学習装置の開発

#### 3.1 概要

開発の基本方針に基づき、学習装置としてライトレースロボット（以下ライトレーサと略す）を選んだ。図2に用いたライトレーサの概要を示す。独立で動くモータ2台に車輪を装着し、自由に回転できる車輪とあわせて3点で車体を支える。二つのモータ回転方向および ON-OFF 制御により平面（床）上を任意の動作で移動可能である。各車輪（ギヤを通じてモータに連結）は2ビットの情報により前進・停止・後進する。速度を変化させる機能は持たず、移動距離は車輪を回す時間で制御する。

車両先頭部左右（図2の上部）には光センサーと光源の組を2組持つ。各光センサーは光源と組み合わせ、走行路面の明度により明暗の1ビット情報を提供する。

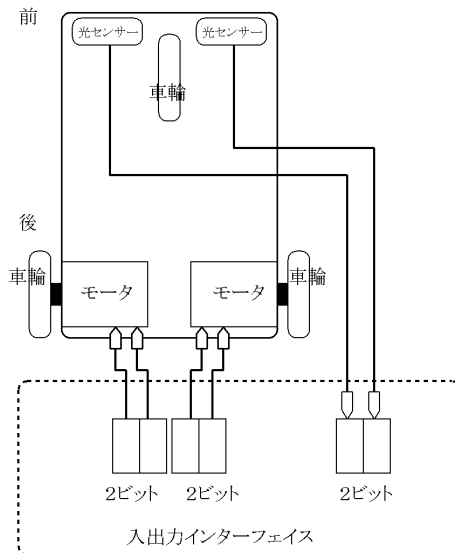


図2. ビット操作で動くライトレーサ

学習者へはライトレーサの完成品を与える。ライトレーサはサイズ10cm×18cm×5cmで、モータをビット操作する電子回路基板（電源はUSBより供給）、単3電池2本（モータ駆動用）、および入出力インターフェイスを搭載している。そのため、学習者はライトレーサをコンピュータとUSBケーブルで接続をするだけで利用可能である。なお、このライトレーサを動作させるには1m四方程度の面積が必要である。

ライトレーサのハードウェアは次の三部分から構成される。

#### 1. ビット情報を受けて動作する部分

ライトレーサ車体、モータ駆動部、駆動電子回路からなる

#### 2. 周辺情報をビット情報として提供する部分

発光部と受光部、およびビットに変換する電子回路からなる

#### 3. コンピュータとのビット情報のやり取りをするインターフェイス部分

1、2については既存のライトレースロボットの技術を用いた<sup>[13],[14]</sup>。ハードウェアの詳細は省略する。3のインターフェイス部はコンピュータからの指令を学習装置へ伝える役割を果たす。コンピュータのUSB経由でビット情報の入出力を担当する。詳しくは3.3節にて述べる。

### 3.2 ビットによる操作の設計・開発

ライトレーサはコンピュータから4ビットの情報を受け、振舞いを変化させ、2ビットの床情報をコンピュータへ供給するように設計された。コンピュータからの4ビット命令とコンピュータへの2ビット情報供給の順で述べる。

#### ・コンピュータからの4ビット情報を受け取り動作

学習装置の制御レベルは モータの制御⇔車輪の動作⇔ライトレーサ車両の振舞いと三段階に分かれている。コンピュータからのビット情報はモータの回転方向・停止の指示に使われる。しかし、学習者の立場で考察すると、車輪の動作をビットで指示の方が直感的である。そこで本システムでは見かけ上ビット情報で直接車輪の動作を指示する仕様にした。車輪への指示を組み合わせることで車両の振舞いを決定することができる。

本システムの車両で処理される情報の流れは

ビット情報→モータの回転方向→車輪の回転方向→車両の振舞い

である。モータの制御と車輪の動作と車両の振舞いは表1の関係になる。ここで、回転方向の定義を図3に示す。図3左にモータの回転方向を示す。モータの正回転方向はモータの軸が伸びた側の正面から見て軸が右回転する方向を正回転と定義する。図3右に車輪の回転方向を示す。車両につけられた車輪の正回転方向は車両が前進する方向をもって正回転とする。したがって、左車輪と右車輪ではモータの正回転方向と車輪の正回転方向が一致しない。この不一致は学習者に混乱を生じされる可能性がある。前述のように、車輪の

正回転方向の定義

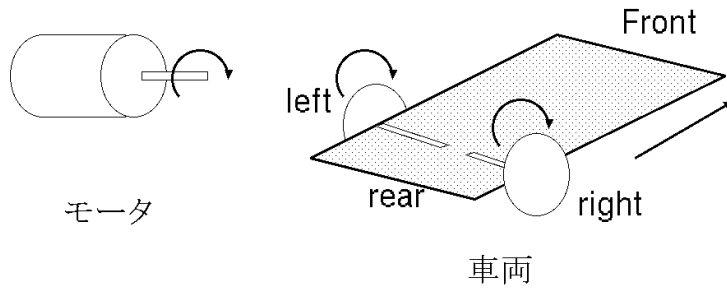


図3. モータおよび車両につけられた車輪の回転方向定義

動作を中心に。これらのハードウェアとビット操作の違いはすべて電子回路とモータの接続方法で吸収をした。

表1で示すように、直接の制御対象であるモータの動きに対して、4ビット中、上位2ビットを左モータに、下位2ビットを右モータに割り当て、制御する。その結果として車両につけられた車輪の動作は上位2ビットが左車輪、下位2ビットが右車輪への動作指示とし、それぞれ00が停止、01が前進、10が後進とした。ここで、左車輪と右車輪の前進方向回転はモータの回転方向では左と右で逆転しているが、前述のように、この違いを電子回路とモータ結線で吸収している。

表1. モータの制御と車輪の動作と車両の振り舞い

<b>制御対象</b>	左モータ 2ビット 停止・逆回転・正回転 00, 01, 10	右モータ 2ビット 停止・正・逆回転 00, 01, 10
<b>車輪の動作 (モータの動きの結果)</b>	左車輪 2ビット 停止 00 前進方向回転 01 後進方向回転 10	右車輪 2ビット 停止 00 前進方向回転 01 後進方向回転 10
<b>車輪の振り舞い (2つの車輪の組み合わせによる結果)</b>	<b>車輪 4ビット</b>	
	停止 (0000)、前身 (0101)、後進 (1010)、前進左折 (0001)、前進右折 (0100)、 後進左折 (0010)、後進右折 (1000)、 左旋回 (左右車輪逆回転、上から見て) (1001)、右旋回 (左右車輪逆回転、上から見て) (0110)	

車両につけた車輪の動作と車両の振舞いを図4にまとめた。図4では四角で車両を表し、前方中央部に自由に動く車輪を示し、後方両側面に独立して動く駆動車輪を表している。図4上の列右から2つ目の場合、左車輪の停止および右車輪の正回転が車両を前進左折させる。この場合、2つのモータへ指示されるビット情報は左モータへは停止の00が上位2ビットになり、右モータへは前進の01が下位2ビットである。あわせて4ビット情報0001により車両は前進左折する。

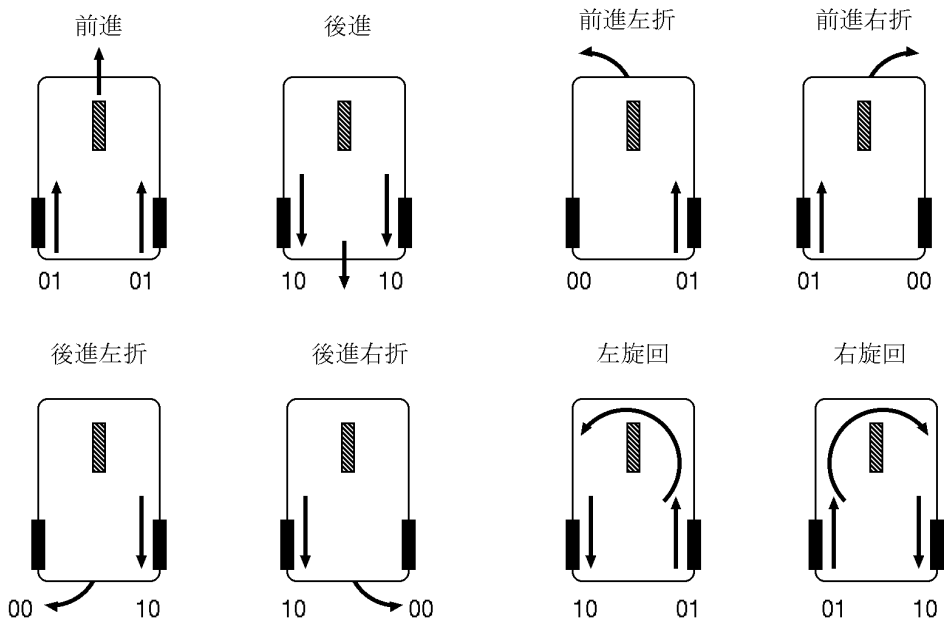


図4. 車輪の動作とライトレーサ車両の振舞い

・コンピュータへ2ビット情報を渡す

学習装置はコンピュータへ2ビットの床の明暗情報を提供する。上位1ビットが左側、下位1ビットが右側の床情報を提供し、0が暗、1が明とした。2ビット情報と床面の関係を図5に示す。図5と同様に四角で車両を表し、車両前方の左右にある丸で光センサーを表している。

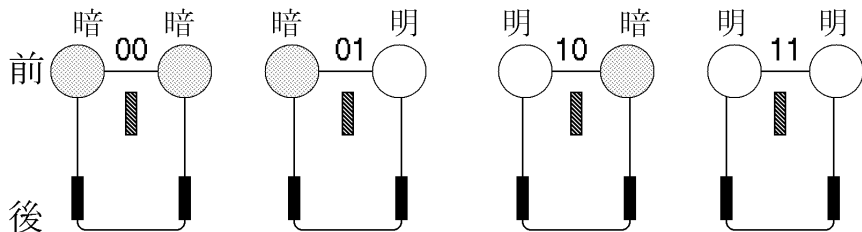


図5. 光センサーによる床の明暗情報

コンピュータへ提供する入力情報をまとめると、車両前面の左右にセンサーがつけられ、床の明暗情報を2ビット、4状態、左右とも明(11)、左右とも暗(00)、左のみ明(10)、右のみ明(01)となる。この床面明暗ビットは床面の変化とともに常時に提供されている。コンピュータ側では提供される値を常に監視するプログラムの処理を必要とする。

### 3.3 入出力端子インターフェイス

コンピュータからバスラインを通じてビット入出力情報を取り出すことは簡単ではない。デスクトップコンピュータではバスラインを通じて取り出す場合、条件によってはコンピュータに回復不可能な損傷を与える危険を伴う。またノート型コンピュータではバスラインの取り出しが困難な場合が多い。そのために、本システムのようなコンピュータから外へビット情報の入出力を目指す研究はコンピュータインターフェイスの専門的な知識を要する分野として見られてきた。

これらの困難を避けるために、以前は、入出力端子の外部への取り出しに、EIA232やプリンタポートが利用されていた<sup>[15]</sup>。これらの利用にもある程度専門的な知識が要求され、端子を取り出すインターフェイスも電子回路で自作する必要がある、簡単な事柄ではなかった。しかも、最近のコンピュータではこれらのインターフェイスを備えていない場合が増えてきた。この傾向は、今後さらに増すと思われる。その代わりに、USB端子を備えるコンピュータが普通の存在になってきた。本システムではコンピュータと外部の接続にはUSB端子を用いる。しかし、USBは取り扱いの簡便さとは逆に入出力情報を取り出すために複雑な処理が必要となる<sup>[16],[17]</sup>。

本システムでは、ビット情報入出力端子を持つインターフェイスはUSB-IOを利用する<sup>[18],[19]</sup>。USBの煩雑な信号処理を一つのIC(サイプレス社製CY7C63001)で行うusb1.1用のインターフェイスであり<sup>[20]</sup>、次の仕様を持つ。

USB-IOの基本仕様

- ・ 2ポートを持つ
- ・ ポート0として8ビット、ポート1として4ビットの入出力端子を持つ
- ・ ポート毎に入力・出力を設定可能(ビットごとの入出力設定は不可)
- ・ すべての端子はIC内部でプルアップされている(未接続状態では1を出力)
- ・ コンピュータのUSB端子より5Vの電源供給が可能

また、ハードウェアを操作する基本ソフトウェアがすでに書き込まれた状態でキットとして提供されており、次の特徴を持つ。

- ・ Windows環境でヒューマンインターフェイスデバイスとして自動認識される
- ・ 制御するためのDynamic Link Libraryが存在する

さらに、Linuxなどのunixオペレーティングシステムでもドライバが開発されており、



利用可能である。

これらの既存キットを利用することでコンピュータから安全にビット入出力端子を取り出すことが可能になる。

本システムでは USB-IO が持つ 2 つのポートを、ポート 0 の 8 ビットを出力専用、ポート 1 の 4 ビットを入力専用としてハードウェアとのビット情報交換に用いた。前述のビット操作ハードウェアの仕様より、コンピュータからハードウェアへのビット出力命令は 4 ビットである。図 6 に示すようにポート 0 の下位 4 ビットを出力部として用い、上位 4 ビットは未接続とした。コンピュータへの入力 2 ビットはポート 1 の下位 2 ビットを用い、上位 2 ビットは抵抗を通してグランドへプルダウン回路を作成した。これにより、ポート 1 の読み出しは常に上位 2 ビットがゼロとなり、下位 2 ビットの値に応じて 00 00、00 01、00 10、00 11 を出力し、10 進法で記すと 0 から 3 までの値を出す仕様にした。

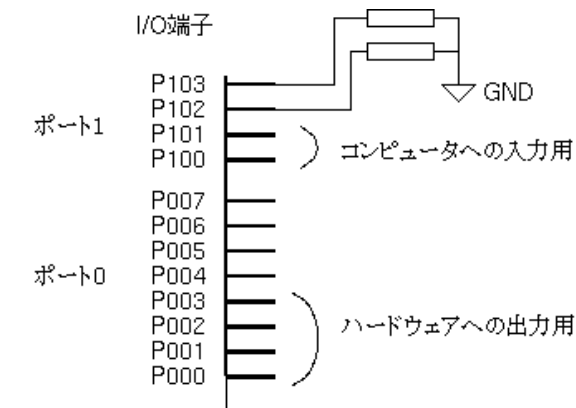


図 6. 用いた USB-IO のポート 0 と 1

ハードウェア（ライントレーサ）との接続はポート 0 の下位 4 ビットをビット信号線へ接続し、モータの制御に用いた。ハードウェアからの 2 ビット入力情報はポート 1 の下位 2 ビットに接続し、ハードウェアからの床情報を取得するのに用いた。また、USB-IO 経由でハードウェア制御基板へ 5 V の電源供給を行った。

### 3.4 ソフトウェア環境

現在、USB-IO を操作可能な Dynamic Link Library（以下 DLL と略す）は 3 種類出ている。1 つは川平氏の作になる HSP で利用できる hspusbio.dll 無料で利用可能である<sup>[21]</sup>。2 つ目は川平氏のソースコードからバケさんが作られた Visual Basic for Applications 用の vbausbio.dll、これは Visual Basic などからも利用可能であり、無料で利用可能である<sup>[22]</sup>。3 つ目は三石印房の製品 USBIODLL.dll で Visual Basic for Applications か

らも利用可能である。一部機能制限されたものは無料で利用可能である<sup>[23]</sup>。

これらの DLL を用いることで、入門者にとって抵抗の少ないプログラミング言語環境での教材が提供可能である。たとえば、Excel と Visual Basic for Applications を組み合わせたシステムを構築することが可能である<sup>[24],[25],[26]</sup>。

この学習装置を用いることで、多種多様なコンピュータ言語環境で、アルゴリズムの基本である、順次処理、繰り返し、条件判断を学ぶ教材を作成可能となる。たとえば、学習者は学習装置を用いて算用数字「2」を描く動きをする課題を実施することで順次処理を学び、四角形を描く課題で繰り返し処理を学ぶ。さらに、明暗情報で停止をする課題で条件判断を学ぶ。これらの動作を組み合わせて、ラインをトレースする課題で順次・繰り返し・条件判断を組み合わせたプログラム作成を学ぶことが可能である。

現在利用可能な DLL を表 2 に示す。

**表 2. USB-IO 操作 Dynamic Link Library**

Windows 環境

- 川平氏 (<http://www.chichibu.ne.jp/~kawahira/index.html>)  
HSP USB-IO 用プラグイン Ver 0.21 (05/06/22 製作)  
HSP (<http://www.onionsoft.ne/hsp/>) 上で利用可能 hspusbio.d11
- バケさん <http://bake-san.com/> vbausbio.d11  
Visual Basic および Visual Basic for Applications で利用可能
- 三石印房 <http://www.kyokusendo.com/> USBIODLL.d11  
VB、VBA、VCC で利用可能

Linux 環境

- 成松氏 <http://www.narimatsu.net/nari-page/musbio.html>

Java の <http://fl.aaa.livedoor.jp/~fexx/usbio/index-ja.html>  
を用いることで Windows、Linux の両環境利用可能。

#### 4. まとめ

既存の技術であるライントレースロボットとコンピュータからビット情報の入出力を取り出すインターフェイスを組み合わせて、ビット操作可能な学習装置を設計・開発した。この学習装置は 4 ビットの命令で振る舞いを変化させ、2 ビットの明暗情報を提供可能である。

この学習装置を作動させるソフトウェア環境はオペレーティングシステムでは、Windows 環境と Linux が利用可能である。コンピュータ言語は、Java、Visual Basic、Visual Basic for Applications、Visual C++、HPS など多岐にわたる。これにより、従来によ

うにアセンブラやC言語に限定された環境ではなく、学習者の学習進度に応じた多彩な環境での教材作成が可能となる。

## 謝 辞

本研究は、文部科学省科学研究費、基盤研究C研究代表者 伊藤敏 課題番号17500657、および基盤研究C研究代表者 井上祥史 課題番号18500696より補助を受けた。

## 参 考 文 献

- (1) 教育システム情報学会情報教育特別委員会、「情報教育推進のための提言書」、2003
- (2) 大学における一般情報処理教育の在り方に関する調査研究委員会、「大学における一般情報処理教育の在り方に関する調査研究（文部科学省委託調査研究）平成13年度報告書」、2002
- (3) 原田悦子、「文科系大学・学部における情報教育—その目的と問題—」、情報処理、第41巻 No.3、2000、pp227-233
- (4) 西野和典、「高等学校における情報教育実践」、教育システム情報学会誌、Vol. 17 No.2、2000、pp139-143
- (5) 大岩元、「高等教育機関における情報教育」、教育システム情報学会誌、Vol. 17 No.2、2000、pp110-113
- (6) 家本修、「プログラミング教育」、教育工学事典、2000、pp459
- (7) 向後千春、「Webベース個別教授システム(PSI)によるプログラミング授業の設計、実施とその評価」、教育システム情報学会論文誌、Vol. 20 No.3、Jul、2003、pp.293-302
- (8) 伊藤敏、井上祥史、「体で実感できるプログラミング教材の開発」、電子情報通信学会 信学技報、ET2006-13、2006、pp7-12
- (9) 伊藤敏、井上祥史、「見て・聞いて・触れるプログラミング教材の開発」、教育システム情報学会論文誌投稿中
- (10) 香川考司、「プログラミングの楽しさを教えるには?」、教育システム情報学会誌、Vol. 22 No.3、2005、pp225
- (11) 伊藤敏、井上祥史、「見て、聞いて、触って操作する教材開発」、教育システム情報学会 研究報告、Vol. 19 No.2、Jul、2004、pp.56-60

- (12) 土居範久、 笈捷彦、『コンピュータ入門 1』、岩波書店、1987.
- (13) 水川真、安藤吉伸、春日智恵、『ライントレースロボット入門』、オーム社、東京、2003.
- (14) 梵天丸、<http://www.inrof.org/TORO/>
- (15) 渡辺 明禎、『VBと製作で学ぶ初めてのパソコン応用工作』、CQ出版社、2002.
- (16) CQ 出版社編、『USB ハード&ソフト開発のすべて』、CQ 出版社、2001.
- (17) 鈴木一海、五十嵐顕寿、『入門 USB』、技術評論社、2001.
- (18) テクノキット、<http://www.technokit.biz/>、Km2Net 商店、<http://www.km2net.com/>
- (19) 永島智二、『手作り USB 機器』、RBB PRESS、2005.
- (20) サイプレス社、<http://www.cypress-japan.co.jp/>
- (21) 川平航介、<http://www.chichibu.ne.jp/~kawahira/soft/forsp.html>
- (22) バケさん、「VBA 用 dll (vbausbio.dll)」、<http://bake-san.com/>
- (23) 三石印房、[http://www.geocities.jp/m\\_m\\_m\\_ishi/](http://www.geocities.jp/m_m_m_ishi/)
- (24) 石澤祐治、帷子誠、高橋光広、伊藤敏、井上祥史、「USB-IO を用いた制御教材」、日本産業技術教育学会第23回東北支部大会講演論文集 A-7 Nov、2005、pp13-14
- (25) 伊藤敏、井上祥史、「USB を用いた身体体験型プログラミング教材の開発」教育システム情報学会 研究報告、Vol. 20 No.6、Mar、2006、pp.162-165
- (26) 伊藤敏、井上祥史、「見て、聞いて、触れるプログラミング教材 —Excel を通した bit 操作—」、教育工学論文集 (計測自動制御学会 中部支部 教育工学研究委員会)、Vol. 28 2006、pp21-23