

# USB 制御とパラレル制御に対応するための 自動化簡易言語とインターフェイスの改良 — 中学校技術科用教材「オートマ君」の事例にそくして —

村松 浩幸\*・砂岡 憲史\*\*・川瀬 忠\*\*\*・桜井 正広\*\*\*\*  
坂口 謙一\*\*\*\*\*・平舘 善明\*\*\*\*\*・木下 龍\*\*\*\*\*

## The Improvement of Automation Simple Language and Interface to Support USB Control and Parallel Control. — The Case of the “Automakun” Teaching Material for Junior High School Departments —

Hiroyuki MURAMATSU・Norifumi SUNAOKA・Tadashi KAWASE・Masahiro SAKURAI  
Kenichi SAKAGUCHI・Yoshiaki HIRADATE・Ryu KINOSHITA

### 要 旨

中学校技術科「情報とコンピュータ」において自動化学習用に開発した自動化簡易言語「オートマ君」の制御用 DLL および USB インターフェイスを、USB 制御とパラレル制御の両方に対応できるように改良した結果、次の成果が得られた。(1) USB 制御に対応させることで、OS およびハードウェアの制約を少なくし、より多くの環境で自動化学習が可能になった。(2) USB 制御とパラレル制御の両方に対応させることで、今までのハードウェア資産を生かしつつ、新しい環境での自動化学習に移行できるようになった。(3) 改良した制御用 DLL は使用法を公開し、他プログラム言語でも利用可能になった。

### 1. はじめに

中学校技術・家庭科の技術分野（以下技術科）の「情報とコンピュータ」では、選択であるが(6)プログラムと計測・制御という項目が取り上げられている。制御については1998年版学習指導要領改訂前の「情報基礎」の頃から様々な教材開発や実践がなされてきている<sup>1)</sup>。

情報技術を技術科で扱う場合、現実の技術をとらえさせるという観点からも、制御は学習内容の大きな柱の1つであると考えられる<sup>2)</sup>。アプリケーションの使い

方を中心とした基本操作重視の実践が数多い現状だが、今後の「情報とコンピュータ」を考えると、基本操作重視の方向から脱却し、「制御」と「情報通信ネットワーク」に重点をおくべきという現場の意見も多い<sup>3)</sup>。

筆者らは制御の学習を、プログラムが機械を自動化させたという観点から自動化学習と名付け<sup>4)</sup>、制御を手軽に行うための自動化簡易言語「オートマ君」（以下「オートマ君」）を教具やテキストと共に開発・公開してきた<sup>5)</sup>。「オートマ君」は技術科の中でも実践事例が多く、普及していると考えられる制御教材の1つである<sup>6)</sup>。しかし制御の学習は重要な事項でありながら、制御教具を使用したり、ハードウェアに密着していることから、OS やハードウェアの仕様変更の影響を大きく受ける。

例えば、「オートマ君」ではパラレルポートを用いたパラレル制御をしている。パラレル制御は利便性、簡便性、経済性に優れており、広く制御教材として用いられているが、OS やハードウェアの仕様変更のた

\* 三重大大学教育学部

\*\* ルクセンブルグ補習授業校

\*\*\* (有)テクノキット

\*\*\*\* 桜井技術教材社

\*\*\*\*\* 東京学芸大学教育学部

\*\*\*\*\* 東京学芸大学大学院連合学校教育学研究科

めに制御方法が難しくなったり、パラレルポートそのものを持たないコンピュータも増え、今後の使用に大きな不安を抱えている<sup>7)</sup>。

一方、制御教材にはパラレル制御のようにコンピュータからインターフェイスを通して直接制御教具を制御する方式の他に、シリアルポート経由や画面からの白黒発光の光信号あるいは音声データでプログラム転送する方式もある<sup>8) 9)</sup>。これらはパラレル制御ができなくとも実践可能であるが、プログラムを転送する作業時間が必要になり、パラレル制御のように直接動かし、すぐに修正できる試行錯誤の即時性に劣る点がある。

そこでパラレル制御の長所を生かした解決方法としてUSB制御が考えられる。USBはWindows98以降で標準サポートされ、広く普及している規格である。USB接続の機器はマウス、プリンタを始め多種多様な機器が販売され、その扱いも手軽である。

USBは技術的な壁が高く、技術科教員が自作する、あるいは技術科の教材関連会社において独自開発することが困難であるとされたが、亀山や飯島らにより、USBインターフェイスの制御教材が開発されたり<sup>10) 11)</sup>、技術科向けの教材としてUSB制御を可能にするUSBインターフェイス（以下USB-IO）が商品化され、USB制御が実践可能な環境が整ってきた<sup>12)</sup>。

USB-IOに対応すると同時に、パラレル制御での教材資産も生かす形で「オートマ君」およびインターフェイスを改良することができれば、従来から「オートマ君」の実践を行ってきた学校が機器更新でも実践を継続できるだけでなく、OSやハードウェアの変更で実践をあきらめていた学校でも実践可能になり、技術科における制御関係の学習に教材レベルで大きく寄与できると考えられる。

## 2. 研究目的

中学校技術科「情報とコンピュータ」において自動化学習用に開発した自動化簡易言語「オートマ君」の制御用DLLおよびUSBインターフェイスを、USB制御とパラレル制御の両方に対応できるように改良をする。

## 3. 研究方法

以下のように「オートマ君」とUSBインターフェイスに改良を加えることとする

- (1) USB-IOと同時にパラレル制御についてソフトウェア側で自動判別させ、ユーザーが意識することなくUSB制御とパラレル制御を切り替えて使用できるように「オートマ君」の制御用DLLを改良する。

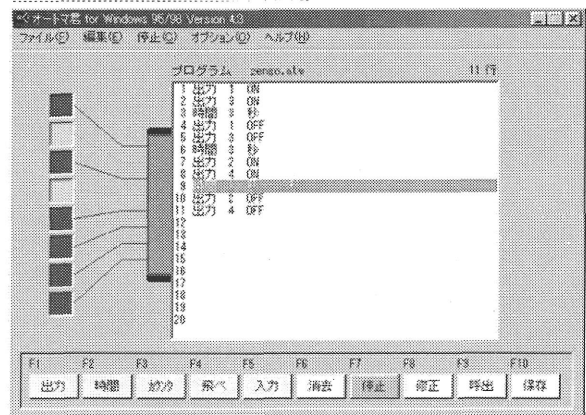


図1 自動化簡易言語「オートマ君」

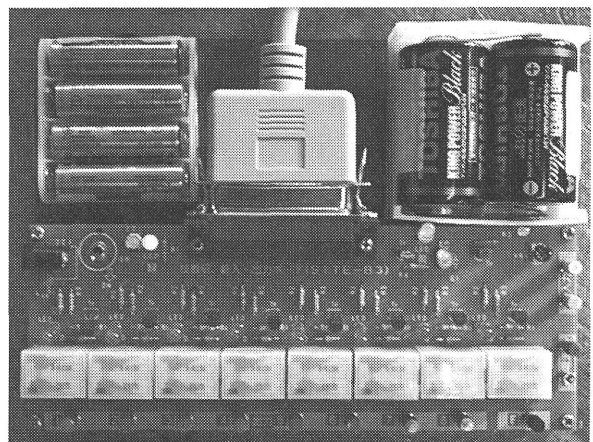


図2 出力ボード

- (2) 従来のパラレル制御インターフェイスもUSBから制御できるようにUSB-IOを改良する。
- (3) 改良されたDLLとインターフェイスを用いて動作テストを行い、複数OSや機種で動作確認をする。

### 3.1 自動化簡易言語「オートマ君」について

筆者らが開発した「オートマ君」は5つの命令をマウス操作で入力することで出力8ポート、入力1ポートの制御を可能にしている（図1）。

制御対象にはパラレルポートに接続したリレー8個を動作させる出力ボード（図2）を用いて、「オート三輪」と呼ばれる模型車を制御した各種課題のプログラムを試行錯誤したり、光センサを活用した課題を解決する学習を展開する。ネットワーク機能も持ち、プログラムをコメントと共にネット上でやり取りし、プログラムを複数人で共同開発する学習も展開している<sup>13)</sup>。

### 3.2 USB-IOについて

USB-IOはモルフィー企画が開発し、それを(有)テク

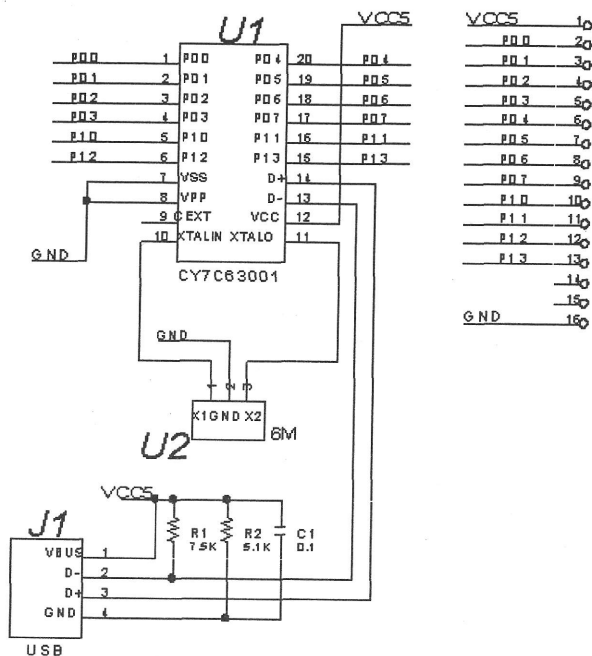


図3 USB-I/O の回路図

ノキット社が引き継ぎ、改良を加え商品化した USB 制御用のインターフェイスである。

米国 Cypress 社が開発した USB 用マイクロコントローラ CY7C63001 を用い、USB コネクタに挿すだけで、認識し、8 ビットと 4 ビットの 2 つの入出力信号を並列処理できる<sup>14)</sup>。回路図は図 3 に示す。本研究ではこの USB-I/O を用いて USB 制御をおこなう。

3.3 WinodowsNT, 2000, XP でのパラレル制御

中学校現場で広く使われる WindowsOS であるが WinodowsNT, 2000, XP ではユーザープログラムからの直接 IO アクセスが認められなくなり、パラレルポートを用いてのパラレル制御ができなくなった。これを回避するためには、リニアアドレスをマッピングさせ、IO ポートにアクセスを可能にするために、フリーウェアのデバイスドライバ giveio. sys を組み込む方法が知られている<sup>15)</sup>。

「オートマ君」でもこの giveio. sys を OS に組み込むことで従来通り、パラレル制御ができるようになる。しかしデバイスドライバの組み込みには専用のツールも開発され、公開されているが<sup>16)</sup>、デバイスドライバを扱うのは、誰もができるとは言い難い弱点を持つ。

3.4 制御用 DLL の改良

「オートマ君」本体と IO ポートの間で制御情報をやり取りをしているのが autoio. dll である。そこで USB-I/O についても同様に autoio. dll 経由で制御情報をやり取りできるようにした。

ベンダーID、プロダクト ID は、(有)テクノキット社

表 1 ベンダー ID、プロダクト ID

```
struct {
    USHORT VendorID;
    USHORT ProductID;
} DeviceAttributes[] = {
    0x12ED, 0x1003, // ベンダーID, プロダクト ID
    0, 0
};
```

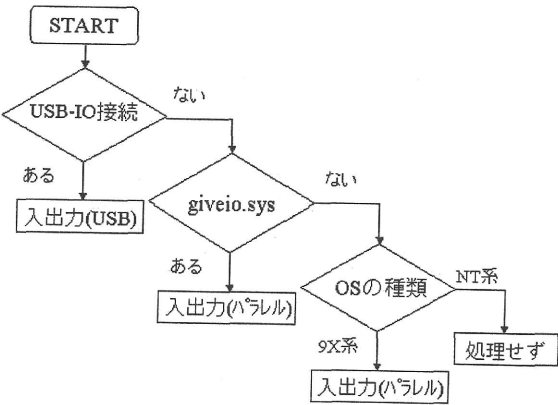


図4 USB-I/O および giveio. sys の判別処理

表 2 出力先を判別して出力する処理

```
__declspec (dlllexport) unsigned short __stdcall OutputWord
(unsigned short port, unsigned short dataword) {
    /* USB-I/O が繋がっているかどうか */
    if (!FindHid ()) {
        /* Windows 9x 系、若しくは NT 系で giveio. sys が組み込まれているか */
        if (OpenGiveIO ()) {
            return _outpw (port, dataword);
        } else {
            return 0;
        }
    } else {
        /* USB-I/O が繋がっている */
        return OutputDevice (OUT_PORT, (char) dataword);
    }
}
```

のサイトより参照した (表 1)<sup>17)</sup>。開発言語は MS Visual C++ 6.0 (SP6) を用いた。

次に自動化簡易言語を使用するユーザーの負荷をできるだけ減らせるように、autoio. dll に USB-I/O および giveio. sys が組み込まれているかどうかの判別処理を組み込んだ (図 4・表 2)。これにより、USB-I/O が接続されていれば自動的に USB-I/O に対応して入出力が行われ、接続されていなければ giveio. sys の存在をチェックし、対応。giveio. sys も存在しなければ OS の種類を調べ、9X 系はそのまま。NT 系であれば制御できないので処理を停止するようにした。

判別処理で USB-I/O が接続確認できれば USB-I/O

表3 USB-IO に出力をする処理

```
unsigned short OutputDevice (char port, char dataword) {
DWORD      BytesWritten = 0;
char OutputBuffer [8] = {0, port, dataword, 0, 0, 0, 0, 0};
BOOLEAN     Result;
/* 書き込み */
Result      = WriteFile (hDevice, OutputBuffer,
Capabilities.OutputReportByteLength, &BytesWritten, NULL)
;
/* 書き込み失敗 */
if (!Result) {
MessageBox (NULL, “書き込めません”, “エラー”,
MB_OK);
}

CloseHandle (hDevice);
return (unsigned short) Result;
}
```

表4 USB-IOとパラレルポートの入力値

	USB-IO からの返値	実際に返す値
入力有	0xFFFFFFFFE	0xCCFE
入力無	0xFFFFFFFFF	0xCC7E

への入出力を行う。USB-IO に出力する処理を表3に示した。

USB-IO で入力を取る場合とパラレルポートから入力を取る場合ではそれぞれ返る値が表4のように異なっている。そこで USB-IO からの入力の際は、パラレルポートからの戻り値に合わせ、返ってきた値が 0xFFFFFFFFE ならば、0xCCFE (入力有:「オートマ君」の命令で「入力 ON」) を返し、返ってきた値が 0xFFFFFFFFF ならば、0xCC7E (入力無:「オートマ君」の命令で「入力 OFF」) を返すように処理をした。

3.5 制御用 autoio.dll の他言語での活用

改良した制御用 DLL の autoio.dll を自動化簡易言語「オートマ君」のみならず、汎用性を持たせ、他言語で活用できるように使用法を公開した。表5に Visual Basic で使用する場合の使用例を示した。

3.6 インターフェイスの改良

USB-IO と共に、(有)テクノキット社より、モーター制御用インターフェイスも開発・販売されているが、今まで「オートマ君」で標準的に使用している出力ボードや、他にもパラレル制御を利用した制御用インターフェイスは数多い。それらのハードウェア資産も可能な限り継承できるよう、USB-IO からパラレル制御用のインターフェイスに接続できる変換コネクタを設計した。

「オートマ君」用に設計した出力ボードでは、信号レベルが H の時、アースに流れる電流で 2SC1815 を

表5 VBからのautoio.dllの呼び出し方

```
’ 入力命令
Public Declare Function InputWord Lib “autoio.dll” (ByVal
port As Integer) As Integer
’ 出力命令
Public Declare Function OutputWord Lib “autoio.dll” (
ByVal port As Integer, _
ByVal dataword As Integer) As Integer
※InputWord (入力ポートのアドレス)
※OutputWord (出力ポートのアドレス, データ)
```

通し、リレーを動作させている。しかし USB-IO の H レベルでは、2SC1815 のコレクタ電流は最大 30mA 程度と、リレーを動作させるには不足している。そこで、USB-IO の L レベルの時の吸い込み電流の方が大きいことを利用し、USB 電源を経由し、抵抗を接続することでコレクタ電流を最大 90mA 程度確保させ、リレーを動作可能にした。これで USB-IO と出力ボードを接続し、制御をすることができる (図5)。USB-IO→パラレル変換器→出力ボードという流れである。製作した USB-パラレル変換部の回路図は図6に、実際に製作した変換器は図8に示した。さらに USB-IO とパラル用の USB-IO を開発した (図7)。

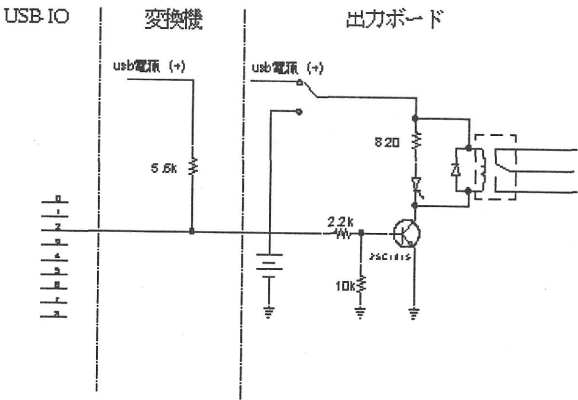


図5 USB-IO と出力ボードの接続回路

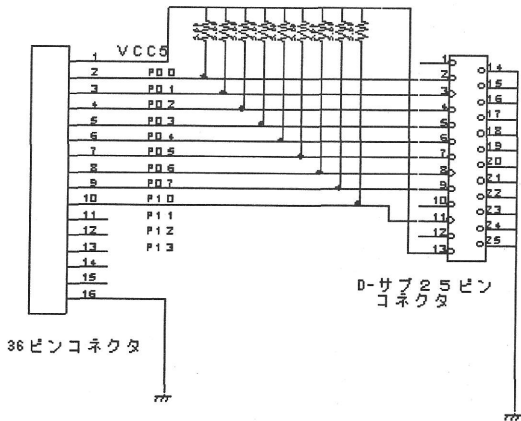


図6 USB-パラレル変換部回路図

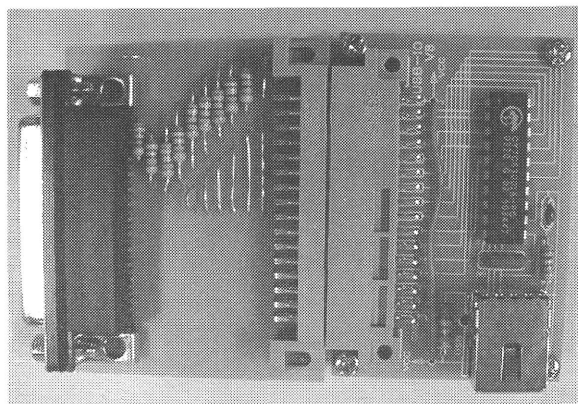


図7 USB-I/O とパラレル変換器

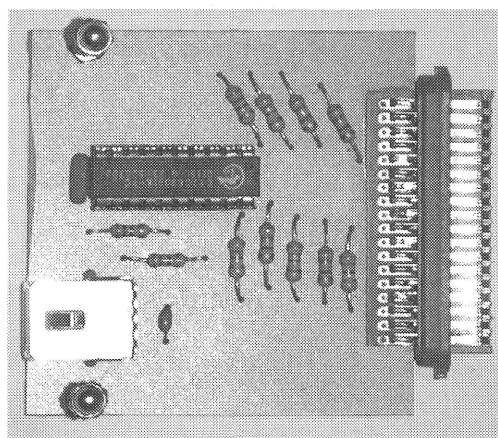


図8 一体型の USB インターフェイス

価格的にも数千円以内に押さえられた。

インターフェイスと共に出力ボードの電源を USB 電源から取することで、出力ボードの 6V 電源である単 3 電池 4 本または AC アダプタを不要にすることができた (図 5)。ただし全てのリレーを ON にしたり、他の USB 機器と併用する場合は外部電源をとる必要がある。

#### 4. 動作テスト

改良した autoio.dll と USB インターフェイスを用いて、以下の 4 つの動作テストプログラムでテストを行った。(1) リレーを全て ON に持続する動作を繰り返す USB 電源負荷テスト、(2) リレーの ON と OFF を繰り返すリレーの連続動作テスト、(3) 光センサの入力命令取得テスト、(4) 模型車と光センサを利用し、空き缶を発見するまで前進させる模型車の動作テスト (表 6)。

動作テストには「オートマ君」(Version4. 40)、出力ボードは USB 電源をとれるように改造したボードを用いた。

表 6 テストプログラム

(1) USB電源負荷テスト	(2) リレー連続動作テスト
出力 1 ON	出力 1 ON
出力 2 ON	出力 2 ON
出力 3 ON	出力 3 ON
出力 4 ON	出力 4 ON
出力 5 ON	出力 5 ON
出力 6 ON	出力 6 ON
出力 7 ON	出力 7 ON
出力 8 ON	出力 8 ON
時間 2 秒	出力 1 OFF
出力 1 OFF	出力 2 OFF
出力 2 OFF	出力 3 OFF
出力 3 OFF	出力 4 OFF
出力 4 OFF	出力 5 OFF
出力 5 OFF	出力 6 OFF
出力 6 OFF	出力 7 OFF
出力 7 OFF	出力 8 OFF
出力 8 OFF	カウンタ 99 回 飛べ 1 行へ
カウンタ 5 回 飛べ 1 行へ	
(3) 入力取得テスト	(4) 模型車動作テスト
入力 ON 飛べ 1 行へ	出力 1 ON
飛べ 1 行へ	出力 3 ON
	入力 ON 飛べ 3 行へ
	出力 1 OFF
	出力 3 OFF
	出力 2 ON
	出力 4 ON

表 7 動作テスト結果

Windows2000 Professional		
SONY	PCG-XR1SA	ノートブック型
自作	Pentium4 512MB	デスクトップ
SOTEC	PC Station M350V	デスクトップ
WindowsXP Home		
SHARP	PC-MV1-D5W	サブノート型
DELL	dimension 8300	デスクトップ
WindowsXP Professional		
DOS/V	Paradise Prime Solution	デスクトップ
Fujitsu	FMV-BibloMG75h	ノートブック型

上記 4 つの動作テストプログラムを異なる OS、および USB の電源を考慮し、デスクトップ型、ノートブック型、サブノート型の各複数メーカーのコンピュータで動作を確認した。テスト結果を表 7 に示す。

動作テストの結果、USB-I/O およびパラレルポートを経由しての制御は、テストに用いた機種で正常に動作することが確認された。

#### 5. おわりに

中学校技術科「情報とコンピュータ」において制御学習用に開発した「オートマ君」の制御用 DLL およ

び USB インターフェイスを、USB とパラレルの両方で制御できるように改良した結果、以下の成果が得られた。

- (1) USB 制御に対応させることで、OS やハードウェアの制約を少なくし、より多くの環境で自動化学習が可能になった。
- (2) USB 制御とパラレル制御の両方に対応させることで、今までのハードウェア資産を生かしつつ、新しい環境での自動化学習に移行できるようになった。
- (3) 改良した制御用 DLL は使用法を公開し、他プログラム言語でも利用可能になった。

なお、開発した DLL は以下のサイトに公開をし、広く制御の学習に利用できるようにした。

<http://www.gijyutu.com/g-soft/automa/index.htm>

<http://www.ddj.com/ftp/1996/1996.05/directio.zip>

<ftp://ftp.ddj.com/1996/1996.05/directio.zip>

- 16) p araffin 作 GiveIo. sys 簡単インストールツール  
INSTDRV Ver. 0. 16

<http://homepage1.nifty.com/paraffin/software/index.html>  
#INSTDRV

- 17) 旬テクノキット

<http://www.technokit.biz/usb-io/techno-vb23.zip>

## 参考文献

- 1) 技術科養育実践講座第 7 巻情報基礎 (1989) ニチブン
- 2) 河野義顕・大谷良光・田中喜美編著 (1999) 技術科の授業を創る、学文社、pp14-15
- 3) 村松浩幸・堀田龍也・竹野英敏 (2004) 中学校技術科「情報とコンピュータ」の実態調査と実践上の課題、情報処理学会情報教育シンポジウム (SSS2004)、pp. 1-6
- 4) 田中喜美、大谷良光、川俣純[他] (1992) 技術教育としてのコンピュータ教育の教材論-中学校技術科「情報基礎」の教材・教具研究 (1)、東京学芸大学紀要第 6 部門巻号 44、pp51-61
- 5) 川俣純 (1996) 自動化からはじめるコンピュータ学習指導書」、技術教育研究会
- 6) 大谷良光、工藤崇弘 (2004) 1998 年版学習指導要領全面実施下における青森県中学校技術科のカリキュラム調査研究、弘前大学教育学部紀要、19: pp51-58
- 7) 亀山寛・戸塚雅彦 (2002) USB インターフェイスを備えた制御教材の開発、日本産業技術教育学会第 45 号誌第 3 号 p135-141
- 8) 紅林秀治 (2004) ドリトルを使ったロボット制御、本多満正編集代表「実践情報科教育法」、東京電機大出版会、pp110-117
- 9) 例えば、山崎教育システム株式会社、音通信制御システム「自律制御ロボ」など  
<http://www.yamazaki-kk.com/>
- 10) 亀山寛・戸塚雅彦 前掲書
- 11) 飯島眞理 (2001) USB を利用した制御教材のための学習ソフトウェアの開発、東京学芸大学紀要、53 pp 7-12
- 12) 旬テクノキット USB-IOV8  
<http://www.technokit.biz/>
- 13) 村松浩幸編集代表 (2000) IT の授業革命「情報とコンピュータ」、東京書籍、pp92-124
- 14) 大石克己 (1997) USB 用ワンチップマイコン、「トランジスタ技術」1997 年 7 月号、pp250-257
- 15) Dale Roberts 作。