

## コンピュータ制御教材における信号入出力技法の研究

森 正人\*・志田 寛\*・佐藤信安\*

(1996年12月9日受理)

Masato Mori, Yutaka Shida, Nobuyasu Sato

Development of an I/O interface for Instraction of Computer controlled systems

中学校「技術・家庭科」の情報基礎領域において使用されるパーソナルコンピュータによる制御の教材（以下、コンピュータ制御教材とする。）に利用できる多ポート入出力インターフェースの開発を行った。このインターフェースの特徴は、学校現場でのパーソナルコンピュータの設置状況を考慮し、多くの機種に共通して装備されているプリンタポートとRS-232Cインターフェースを組み合わせる事により、異なる機種間で互換性のある多ポート入出力のインターフェースを実現していることである。さらに、多変数制御や変化の速い制御量を扱う制御機構にも対応出来るように入出力ルーチンの高速化を図ると共にMS-WINDOWS環境下でも教材開発・利用ができるようにMS-WINDOWS用DLL (Dynamic Linking Library)の開発も行った。また、開発したインターフェースの教材への応用例として交流波形を制御する制御システムを試作した。

キーワード：コンピュータ制御教材，インターフェース，プリンタポート，RS-232Cインターフェース，セントロニクスインターフェース

### 1 はじめに

#### (1) 本研究の背景

我々の身の回りでは多数の自動制御機器が稼働しており、特に最近では、マイクロコンピュータの小型化、低価格化に伴い、それらを組み込むことによって高度な制御がなされている機器が産業技術の分野のみならず一般の家庭にも浸透してきている。このような現状を考えると現在の我々の社会生活や家庭生活において自動制御技術は大きな位置を占めているといえ、技術教育においても自動制御に関する教育内容の充実が必要と考えられる。

現在、コンピュータ教育の中学校現場への導入に伴い、アプリケーションソフトを用いた情報活用能力の育成を目指す教育が広くおこなわれているが、一方ではコンピュータ教育にハードウェア、特に制御の教育をも取り入れる必要性が指摘されている<sup>1) 2) 3)</sup>。中学校指導書、技術・家庭科編では「(3)イ. コンピュータの利用分野を知ること。」の項で制御分野、ファクトリーオートメーションなどにも触れることを求めている。

コンピュータ制御を情報基礎領域で扱うことに関して、亀山は、「技術は、道具→機械→自動制御機械体系と発展してきた。機械から自動制御機械体系への進歩にはコンピュータによる制御が大きな役割を果たしていることを考慮するならばコンピュータ制御が重要な情報基礎の教育内容だと位置づけられることになる。そしてコンピュータ制御を取り入れることによって、情報基礎教育を技術科教育として明確に位置づけることが出来る。」<sup>1) 2)</sup>としている。また、鈴木は「技術・家庭科らしさを鮮明に出すには、コンピュータを生徒の主体的な学習活動に欠くことのできないツールとして位置づけ、思考力を鍛え、創造性を発揮させるような題材を選定する必要に迫られる。例えば、技術・家庭科の中の、①情報検索機能を用いる学習活動、②シミュレーション機能を用いる学習活動、③計測・制御機能を用いる学習活動、④情報の表現・創造を要する学習活動などに着目した、典型的な題材開発が望まれる。」<sup>3)</sup>としている。

コンピュータのハードウェアやソフトウェアと制御対象機器の関係を学習することは、ソフトウェアのみの教育では困難なコンピュータと外界との繋がりを学習する事ができ、更にコンピュータの他の技術への応用を示唆する上で有効であると考えられる<sup>4)</sup>。教育現場においても生徒の「生きて働く力」<sup>5)</sup>を養う為の要点を明らかにすると共に、情報活用能力の育成に向けてのポイントの1つとしてコンピュータによる制御の学習を取り入れた研究と実践がなされている。しかし、コンピュータ制御技術を教育する教材（以下コンピュータ制御教材という）の教育現場への導入はいまだ初期段階にあるといえる<sup>3)</sup>。その原因として経済的な事情から教具を現場教員が自作する必要性が高いことや、特に入出力インターフェース機器はコンピュータ制御には必要不可欠のものでありながら市販のものは、授業で使うのに相応の台数を揃えるのには高価であり、安価に自作できるものの開発が必要とされていること、等があげられる<sup>5)</sup>。こうした状況を打開するために、制御教材開発のための多角的な研究が要請されている。

## (2) コンピュータ制御教材に求められる要件

「制御」はコンピュータの出現以前から機械機構や電気回路の形で実現されており、現在においても変わらずに利用されているものも多い。その中で敢えてコンピュータによる制御を教材化する場合は「多変数制御が比較的容易に行える。」「制御の手順や設定値などの変更がプログラムの変更により容易に出来る。」などのコンピュータによる制御の特徴を生かした題材を選択する事が望ましいと考える。また、このような教材を開発するためには、多数の入出力信号を高速に取り込むことが出来るインターフェースが必要となるが市販のものでこの条件と経済性を同時に満たす事は難しい。そこで、多数の入出力信号を高速に取り込むことが出来、かつ、現場教員が自作できるインターフェースの開発が必要であると考える。

## (3) 本研究の目的

本研究はコンピュータ制御教材開発において有用な複数のポートに高速にアクセスする

事が出来、かつ、教育現場での製作が可能なパーソナルコンピュータ入出力インターフェースのハードウェア及びソフトウェアの開発を目的とする。

## 2 制御教材用入出力インターフェース開発環境の検討

### (1) 制御教材用インターフェースに求められる要件

コンピュータ制御の特色を生かした制御教材を開発するためには、複数の入出力ポートに高速にアクセス可能なインターフェースが必要である。また、教育現場での利用を想定すると、インターフェースには教育現場特有の以下のような要件を満たす必要があると考える。

#### ①ポート数の複数化

多変数制御やアナログ量を扱う教材の場合、複数の入出力ポートを必要とする。

#### ②入出力の速度

変化の速い制御量を扱う場合は制御量の変化に追従できる速度が不可欠であり、また、多変数制御の場合には入出力数が増える分だけ入出力の速さが求められる。

#### ③異機種間の互換性

学校の設備更新や教員の移動などで使用する機種が変わった場合や他校の実践を取り入れる際に若干の変更、或いはそのままに対応できることが望ましい。

#### ④回路の簡易性と経済性

授業で利用する場合、相応の台数の教具を準備する必要がある。その為には出来るだけ簡易かつ安価なものが望ましい。

#### ⑤設置、撤去の容易さ

授業終了後に教具を撤去しなければならない可能性が高いので、設置、撤去の容易なものが望ましい。

以上の5点を踏まえた上でハードウェア及びソフトウェアの開発環境の検討を行った。

### (2) ハードウェア環境の検討

開発するインターフェースが利用するパーソナルコンピュータ側のインターフェースの選定を行うために、パーソナルコンピュータの入出力インターフェースの内、教育現場で利用されている拡張I/Oポート、プリンタポート、RS-232Cインターフェースについて制御教材としての利用の観点から検討した。まとめを第1表に示す。

#### (イ) 制御教材での利用の観点からの各種インターフェースの比較検討

##### ①拡張I/Oポート

コンピュータ入出力インターフェースの基本といえるものである。コンピュータのバスラインに直接接続されておりプログラミング言語の入出力コマンドで直接、バイトまたはワード単位のパラレル入出力を高速に行うことが出来る。

しかし、インターフェースポートの異機種間の互換性が全くなく、ボードをコンピュータ内部に挿入するため装着、取り外しが面倒である。市販品は高価な上、自作にはある程度の知識と技能が必要なため、制御教材用に台数を揃えるのは負担が大きい。

##### ②プリンタポート

市販パーソナルコンピュータの多くがセントロニクス規格に準拠しているため異機種間でも比較的互換性があるインターフェースで、プリンターコネクタへの接続なので、接続、撤去が容易である。8ビットデータ出力に関してはプリンタポートから直接、信号線を引き出して利用できるが入力端子の内、異機種間での互換性が高いのはBUSY端子のみであるため、そのままでは1ビットの取り込みしかできない。このため、シリアル伝送により8ビットの入力を行う必要があるが、シリアル伝送のために簡単な回路とプログラムが必要になる上、データの取り込みに時間が掛かるという欠点がある。

### ③RS-232C

市販パーソナルコンピュータの殆どが装備しており異機種間での互換性が高く、専用コネクタへの接続なので接続、撤去が容易である。しかし、シリアル伝送のため伝送速度が遅く、入出力共に制御対象側に通信回路が必要となるためにハードウェアの扱いが難しい。また、8ビットを越える入出力を実現するのは難しい。

第1表 各種インターフェースの比較

	拡張I/Oポート	プリンタポート	RS-232C
ポート数	多	少	少
速度	高	低	中
互換性	無	中	高
回路の簡易性	難	易	中
設置・撤去	難	易	易

(ロ) 先行研究に見られるプリンタポートとRS-232Cインターフェースの利用

プリンタポートやRS-232Cインターフェースは、元来プリンタやモデムなどの周辺機器とコンピュータの通信のために設けられたインターフェースであるが、これまでも、これらを利用して8ビットデータの入出力を試みる研究がなされてきた。以下にその代表的なものの特徴をまとめる。

#### ①「セントロニクスインターフェイスを利用した逐次比較型A/Dコンバータのデータ入力手法」<sup>7)</sup>

- ・プリンタポートのBUSY端子とSTRB端子を使用してシリアル伝送する事によってA/D変換8ビット信号を取り込む。
- ・シリアル伝送コードはBASICによって構築。
- ・10Hzの正弦波をディスプレイ画面にグラフィック表示した。

#### ②「RS-232Cを利用したA/Dコンバータの教具化」<sup>8)</sup>

- ・RS-232Cインターフェースを利用してA/D変換8ビット信号を取り込む。
- ・BASICによる伝送コードで9600bpsデータ伝送
- ・1kHzの正弦波(半波)をディスプレイ画面にグラフィック表示した。

両研究はプリンタポートとRS-232Cインターフェースの汎用入出力インターフェースとしての利用を示唆する画期的なものでハードウェア的に異機種間での互換性のあるイン

ターフェースを実現していたが、入出力信号数は8ビット1ポートのみであり、伝送速度に限界があった。

以上の考察から、これらのインターフェースをそのまま用いても制御教材用インターフェースに求められる要件を満たすことは難しいと考えられる。そこで、プリンタポートとRS-232C インターフェースの2者を組み合わせることによって既存インターフェースの短所を補い多ポート・高速化を実現することとした。

### (3) ソフトウェア開発環境

インターフェースのハードウェアを駆動するソフトウェアの開発に使用するプログラミング言語等についてについて検討を行った。

#### (イ) MS-DOS環境における制御ソフトウェア開発

コンピュータ制御教材のソフトウェア開発で使用される言語としては、BASICとC言語が適当と考えられる。現在、コンピュータ制御教材用ソフトウェアの殆どはBASICで記述されている。BASICは中学校「技術・家庭科」の教科書にも採用されているプログラミング言語であり、入出力関連のコマンドも一通り用意されているため、コンピュータ制御教材のプログラミングに最も利用される頻度が高い。また、C言語は生徒が自作するプログラムに使用される事は殆ど無いと思われるが、現在最も普及しているプログラミング言語の一つであり、開発性の高い言語であるので教員自身が開発する教材用ソフトウェアでの利用が考えられる。

#### (ロ) MS-WINDOWS環境における制御ソフトウェア開発

現状では学校現場へのMS-WINDOWSの導入は進んでいないが一般向けパーソナルコンピュータ市場の動向から見て、いずれは導入されていくであろう。MS-WINDOWS用プログラミング言語としてはVisual BASIC, C/C++, DELPHI等が普及しているが学校現場へ導入される言語としては取り扱いの容易さや一般への普及度などから考えてVisual BASICが最も有力と思われる。

しかし、Visual BASICのプログラムでは、I/Oポートやプリンタポートなどを直接制御する事が出来ない<sup>6)</sup>ため、単体では制御プログラムを作成することは出来ない。また、MS-WINDOWSそのものも、I/Oポートやプリンタポートなどを直接制御するAPI(Application Programming Interface)を提供していないためVisual BASICで制御プログラミングを行うためには独自のDLL(Dynamic Linking Library)を作成してVisual BASICアプリケーション上から呼び出せる環境を作る必要がある。

以上の検討から、インターフェース駆動用ソフトウェアの開発はBASIC及びC言語で行い、MS-WINDOWS用のDLLの開発も行うこととした。

## 3 制御教材用入出力インターフェースの開発

### (1) 開発の方針

前章での考察に基づき、開発方針を以下のように定めた。

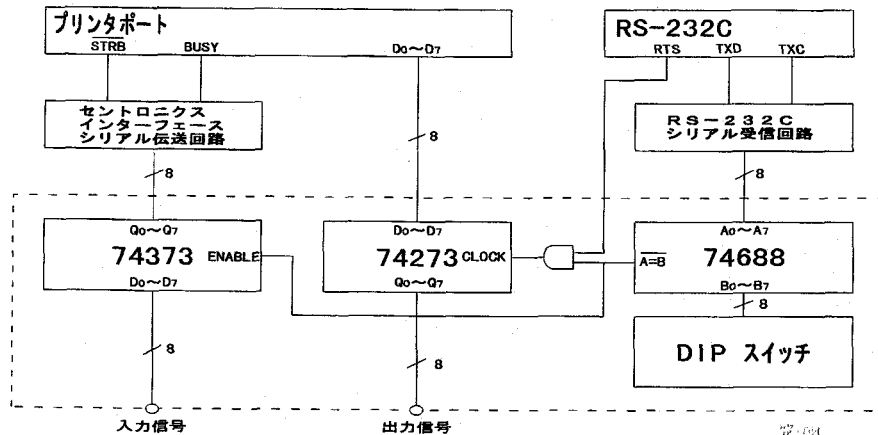
- ①プリンタポートとRS-232Cインターフェースを組み合わせることにより、複数の入出力ポートを確保できるようにする。

- ②異機種間での互換はソフトウェアのみの対応で可能なるようにする。
- ③入出力ポートの増設を容易に行えるようにする。
- ④ソフトウェアの改良により入出力の高速化を図る。
- ⑤インターフェースの開発においては学校現場での利用を想定して教員が調達、利用しうる部品や設備で製作が可能なるように留意する。
- ⑥MS-WINDOWS上で利用できるようにDLLの開発を行う。

## (2) ハードウェア構成と回路の動作

### (イ) インターフェース回路ハードウェア構成

第1図にインターフェース回路のブロック図を示す。本インターフェースはプリンタポートとRS-232Cインターフェースを組み合わせ、RS-232Cをポート番号の指定にプリンタポートをデータ入出力に割り当てることによって多ポート入出力を実現している。第1図において点線で囲まれた部分が1組の入出力ポートになっており、入出力するポートの数だけ、この回路を並列接続する。各ポートはDIPスイッチで任意のポート番号が設定できるようになっている。入出力を行う際には使用するポートの番号をRS-232Cから出力し、このポート選択信号とDIPスイッチの設定値が一致したポートのみがプリンタポートに接続される。選択されたポートは、出力の場合にはプリンタポートの8ビット出力信号端子の信号をラッチし、入力の場合は8ビット信号をプリンタポートのシリアル伝送回路に接続する。このラッチのタイミングはRS-232CのRTS(送信要求)信号を利用してパーソナルコンピュータから伝送される。



第1図 インターフェース回路ブロック図

### (ロ) プリンタポートシリアル伝送回路

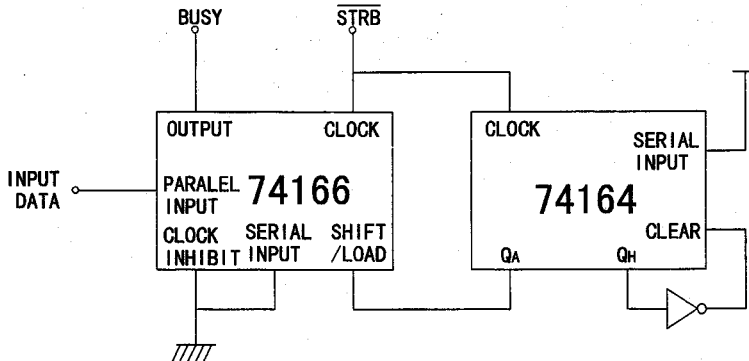
第2図にプリンタポートシリアル伝送回路図を示す。この回路はプリンタポートのSTRB(ストローブ)端子からパルスを受信し、これをクロックとしてプリンタポートのBUSY端子へ8ビットデータをシリアル伝送する。回路を構成する74164, 74166は共にシフトレジスタであるが74166は8ビットのパラレルデータをシリアル化するために用い、74164は受信クロックのパルス数をカウントするためにジョンソンカウンタとして使用している。

初期状態では74164の平行出力 $Q_A$ の $L_0$ の信号が74166のSHIFT/LOAD端子に入力され、74166は8ビットデータLOADの状態にある。ここでSTRB 端子からクロックパルスが入力されると74164の $Q_A$ が $L_0$ から $H_i$ に変化し、74166はLOADからSHIFTの状態に変化し、以後クロックパルスに従って1ビットずつBUSY端子に信号を出力していく。クロックパルスが8つ入力されると74164は自らをクリアし、74166もLOADの状態になって次にクロックパルスが印加されるまで待機する。

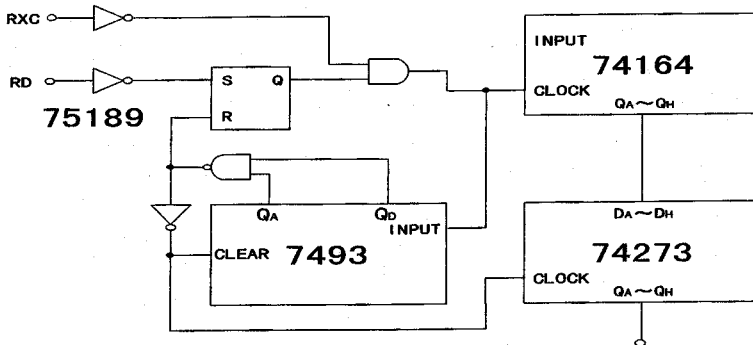
#### (ハ) RS-232C受信回路

第3図にRS-232C 信号受信回路図を示す。この回路はパーソナルコンピュータから送られる通信用クロックパルスによってシリアルデータをシフトレジスタで平行化する。回路を構成する7493は16進カウンタ、74164はシフトレジスタ、74273はDフリップフロップである。

RD(データ受信)端子からスタートビットを受信するとRSフリップフロップの出力 $Q$ が $H_i$ に変化しRXC(受信エレメントタイミング)から受信されるクロックパルスが74164に印加されるようになり以後8ビットシリアルデータが順次、平行化される。7493はスタートビットから数えて9個のパルスをカウントすると74273のクロック端子にパルスが印加され8ビット平行データが74273の出力端子にセットされる。これと同時に7493自身とRSフリップフロップがリセットされ、74164へのRXCの供給も停止され次にパーソナルコンピュータからスタートビットが送信されるまで待機する。



第2図 プリンタボードシリアル伝送回路図



第3図 RS-232Cシリアル伝送回路図

## (二) 使用部品

インターフェース回路はシリアル通信回路も含めて市販の汎用ICを使用するように留意して設計した。使用したICのリストを第2表に示す。

リスト中の75189のみはRS-232Cレベルコンバータ専用ICであるが、このICは広く普及しているICであるので教育現場での調達にも支障はない。

## (3) プログラムコード

第4図にBASICによるプログラムコードを第5図にC言語によるプログラムコードを示す。これらのプログラムコードはBASICにおいてはサブルーチンとして、C言語においては関数としてユーザプログラムに組み込んで利用できるようにした。

入力プログラムコードにおいてはBASIC、C言語共にシリアルデータ受信を行っており、プリンタポートのBUSY端子にパルスを送る毎にSTRB端子の値を読み込み、ビットの重み付けを行った後に変数に順次加えていく動作をしている。

使用に際してはBASICの入力の場合、変数ADDRESSに入力を行うポート番号を代入してからサブルーチンにジャンプすると変数INDATAに入力データの値が代入される。出力の場合は変数ADDRESSに出力を行うポート番号、変数OUTDATAに出力するデータを代入してからサブルーチンにジャンプすると指定したポートにデータが出力される。

C言語の場合も同様に入力の場合は入力関数にポート番号を引数として渡してやれば入力データが戻り値として返される。出力の場合も関数にポート番号と出力するデータを引数として渡すと指定したポートにデータが出力される。

第2表 インターフェース回路使用IC一覧

7400	NANDゲート
7404	NOTゲート
7493	16進数カウンタ
74164	シフトレジスタ
74166	シフトレジスタ
74273	D-フリップフロップ
74573	3ステータラッチ
74688	一致検出回路

```

240 *OUTP 'データ出力ルーチン-----
250 OUT &H30, ADDRESS 'アドレス番号出力
270 OUT &H40, OUTDATA 'データ出力
280 OUT &H32, &H1 'RTSにパルス出力
290 OUT &H32, &H21
295 RETURN

```

```

300 *INPT 'データ入力ルーチン-----
305 INDATA=0
310 OUT &H30, ADDRESS 'アドレス番号出力
330 FOR I=0 TO 7 'データ取り込み
340 OUT &H44, I28
350 OUT &H44, 0
360 INDATA=INDATA+(INP(&H42) AND 4)/4*2^I
370 NEXT I
375 INDATA=255-INDATA
380 RETURN

```

第4図 BASIC入出力プログラムコード

```

/* データ出力 */
void out(char address, char out_data)
{
    outp(0x30, address); /* RS-232Cからアドレスデータ送出 */
    while ((inp(0x32) & 0x4)) {} /* 送信終了を待つ */
    outp(0x40, out_data); /* プリンタポートから出力データを送り出す */
    outp(0x32, 0x1); /* RTSにパルス出力 */
    outp(0x32, 0x21);
}

/* データ入力 */
int inpt(char address)
{
    int data = 0;
    count = 0;
    outp(0x30, address); /* RS-232Cからアドレスデータ送出 */
    while ((inp(0x32) & 0x4)) {} /* 送信終了を待つ */
    /* プリンタポート シリアルデータ取り込み
    STRB端子からクロックパルスを送出し、BUSY端子から
    シリアルデータを取り込む */
    for (count = 0; count <= 7; count++) {
        outp(0x44, I28); /* STRB 端子からパルス出力 */
        outp(0x44, 0);
        data = data + ((inp(0x42) & 4) / 4 << count); /* データ取り込み */
    }
    return 0xff - data;
}

```

第5図 C言語入出力プログラムコード



## (4) 入出力の高速化

## (イ) コードの工夫

本インターフェースは8ビット信号のシリアル伝送を行っているために入出力に時間がかかるという欠点がある。そこでシリアル伝送コードのマシン語化とRS-232C インターフェース回路の直接制御によってシリアル伝送速度の高速化を図った。第7図にマシン語化した入力コードのリストを第6図にRS-232C インターフェースの設定コードを示す。第7図のマシン語コードは前掲のシリアル伝送コードをそのままマシン語化したものである。第6図のコードは、パーソナルコンピュータ内でRS-232C のクロックパルスを作り出しているタイマIC の設定とRS-232C通信ICのクロックカウントレートを直接設定し、高速伝送を実現できるようにしている。

```

100 *SETUP 'RS232C設定ルーチン-----
110 TIMER=16                                *タイマカウント値
120 TIMERH=(TIMER AND &HFF00)/&HFF        *カウント値は下位8ビット、上位8ビットの順に書き込む
130 TIMERL=TIMER AND &HFF
140 OUT &H77, &HB6                          *タイマ動作モードセット
150 OUT &H75, TIMERL                        *下位8ビット書き込み
160 OUT &H75, TIMERH                        *上位8ビット書き込み
170 FOR I=0 TO 2                            *シリアル通信デバイス8251セット
180 OUT &H32, 0:NEXT I
190 OUT &H32, &H40                          *8251リセット
200 OUT &H32, &H4D                          *8251モードセット
210 OUT &H32, &H21                          *8251コマンドセット
220 RETURN

```

第6図 RS-232C設定プログラムコード

```

/* データ入力 */
int read(char address)
{
    asm {
        mov al, address    /* アドレス送信 */
        out 0x30, al
    a:
        in  al, 0x32        /* 送信終了を待つ */
        and al, 0x4
        jz  a
        mov ax, 0x0000      /* プリンタポートデータ取り込み */
        mov bl, 0x04
        mov cl, 0x00
        mov dl, 0x00
    c:
        mov al, 0x00        /* STPBに1パルス出力後HSY端子の番号を読み込む */
        out 0x44, al        /* 動作を8回続ける */
        mov al, 0x00
        out 0x44, al
        in  al, 0x42
        and al, 0x04
        div bl
        shl al, cl
        add di, al
        inc cl
        cmp cl, 8
        jnz c
    };
    return 0xff - _DL;    /* ビットを反転して戻り値を返す。 */
}

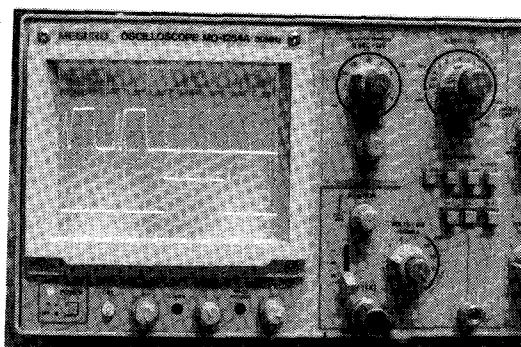
```

第7図 シリアル伝送マシン語プログラムコード

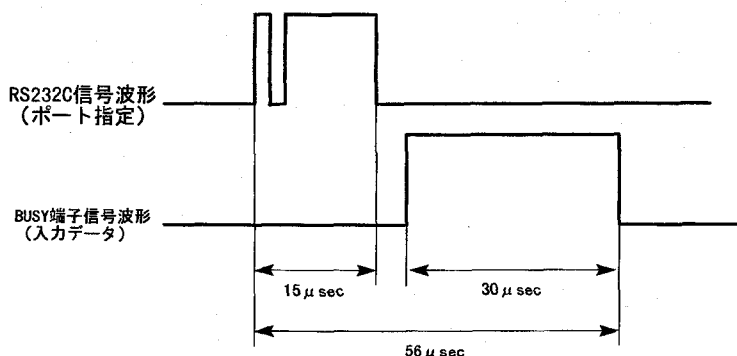
## (ロ) 動作速度試験

上記のコードを利用して8ビットデータのシリアル入力動作速度試験を行った。観測にはNEC PC-9821V12(Pentium 120MHz)を使用した。第8図にシリアル伝送信号波形観測オシロスコープ画面を、第9図に波形説明図を示す。なお、この観測ではポートへのデータ出力を行った後にその出力データをそのまま入力するプログラムを実行させているためにRS-232Cの信号波形にポート指定信号が2回出現している。

先にRS-232Cよりポート指定信号が614400bpsでシリアル伝送され、その伝送終了後にプリンタポートのBUSY端子に8ビットデータがシリアル伝送されている。これら一連の動作は56μsecで完了しており、サンプリング周波数にすると16kHz以上であることから音声信号にも対応可能であると考えられる。



第8図 シリアル伝送波形オシロスコープ画面



第9図 シリアル伝送波形解説図

#### (ハ) DLL (Dynamic Linking Library) の開発

インターフェース回路をMS-WINDOWS環境で利用するためWINDOWS3.1用のものとWINDOWS95用の2種類のDLLを開発した。開発にはVisual C/C++を用いているが、MFC(Microsoft Foundation Class)を用いず通常のC/C++構文を用いてプログラミングされている。プログラムコードは各関数がエクスポート関数パスカル型として宣言される以外は前掲のC プログラムコードと同一なので、ここではリストを省略する。

#### (ニ) インターフェースの仕様

本インターフェース回路とマシン語化された入出力コードの組み合わせにより、第3表に示す特性が実現が出来た。

第3表 インターフェース特性表

・ポート数	最大256ポート(8ビット)
・8ビット信号出力所要時間	20 $\mu$ sec (1ポート当たり)
・8ビット信号入力所要時間	56 $\mu$ sec (1ポート当たり)
・ハードウェア互換性	セントロニクスインターフェース(プリンタポート) 及びRS-232Cを装備しているコンピュータ

※動作測定環境 NEC PC-9821V12(Pentium120MHz)

#### 4 教材への応用例

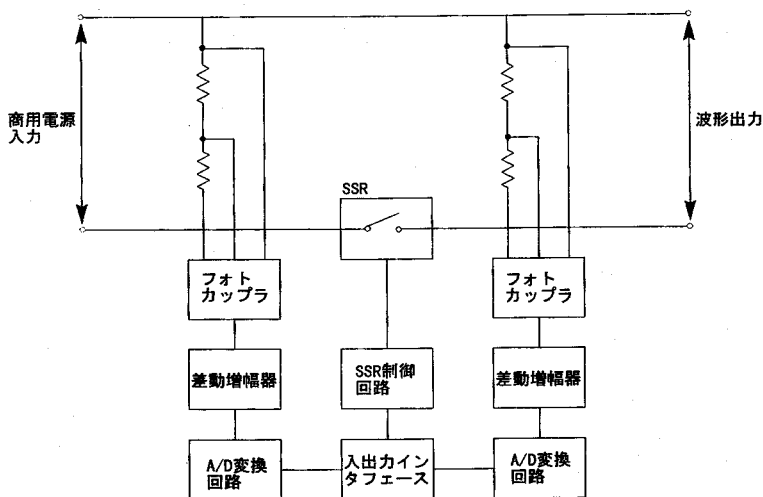
##### (1) 交流波形制御システムへの応用

本インターフェースの応用例として、商用電源の交流波形を制御するシステムを作製した。この制御システムは商用電源(交流 100V 50Hz)の回路をスイッチで開閉する事によって波形の制御を行うものである。このスイッチの動作はパーソナルコンピュータで制御されており、パーソナルコンピュータは商用電源の交流電圧を取り込みソフトウェアの設定にしたがってスイッチに動作信号を出力する。さらに、これと併せて商用電源の元の交流波形と制御された波形の両方をサンプリングし、パーソナルコンピュータのディスプレイ画面にグラフィック表示を行っている。

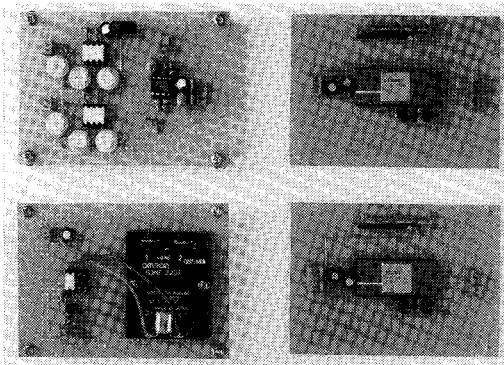
このシステムではデータサンプリングや A/D変換回路と半導体スイッチの制御の為に合計 18 ビットの入力と 3 ビットの出力をパーソナルコンピュータから入出力する必要がある。この為にパーソナルコンピュータに接続する入出力インターフェースには少なくとも 3 ポートの入力ポートと 1 ポートの出力ポートが必要となる。また、データサンプリングの速度は、サンプリングする信号の周期の少なくとも 10 倍程度は必要となる。従来のプリンタポートや RS-232C を利用したインターフェースではこの条件を満たすことは出来ないが、今回開発したインターフェース回路と入出力コードはこの条件を満たす事が出来る。

##### (2) 交流波形制御システムの構成と動作

第10図に交流波形制御システムのブロック図を第11図に回路基板を示す。ハードウェアのシステムは商用電源の波形と制御された出力波形のアナログ値をコンピュータに取り込む為の回路、商用電源を断続するための半導体スイッチ及び入出力インターフェース回路からなる。今回は商用電源の100Vを扱うので不慮の事故からパーソナルコンピュータやその他の回路を保護するためにフォトカップラによる絶縁を行っている。



第10図 交流波形制御システムブロック図



第11図 交流波形制御システム回路基板

第4表

フォトカップラ	TLP521-2 (東芝)
差動増幅器	NJM4558D (新日本無線)
A/D変換器	ADC5742C (DATEL)
分解能	12 / 8ビット
アナログ入力	ユニポーラ +10V / +20V
	バイポーラ ±5V / ±10V
変換時間	25 $\mu$ sec
Solid State Relay	G3NE-220 (OMRON)
絶縁方式	フォトトライアック絶縁
入力信号電圧	5V
出力適用負荷	100~240V 5A

## (イ) 交流波形取り込み部

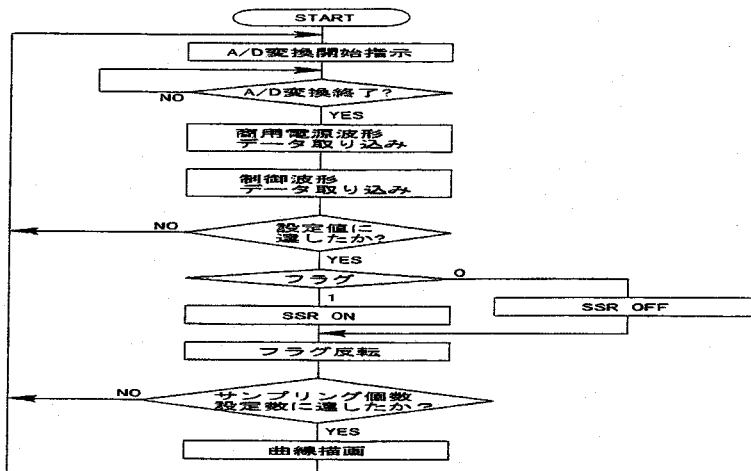
交流波形を取り込む回路は、抵抗器によって分圧された交流電圧の正負両方向の波形を2つのフォトカップラを用いて取り込んでいる。フォトカップラを介して取り込まれた波形は差動増幅器で0~5Vの直流アナログ信号に変換された後 A/D変換器でデジタル変換されインターフェース回路に入力される。

## (ロ) 半導体スイッチ部

スイッチにはSSR (Solid State Relay) を用いた。SSR は従来の電磁リレーと同様に入力信号が印可されている間はONとなり、入力信号が無くなればOFF になる素子である。また、入力-出力間のノイズ対策や絶縁が素子の内部で施されており、コンピュータやICで直接駆動が可能である。交流波形制御システムでは商用電源の回路をこのSSR で開閉することによって制御している。

## (3) ソフトウェアの概要

ソフトウェアはBorland Turbo C++ Ver4.0J を用いて作成されており、前述のRS-232C 設定ルーチンとマシン語化した入出力ルーチンを関数化したものを組み込んでいる。第12図にソフトウェアのフローチャートを示す。



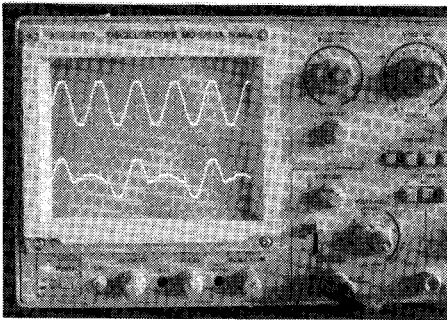
第12図 交流波形制御プログラム・フローチャート

#### (4) 動作結果

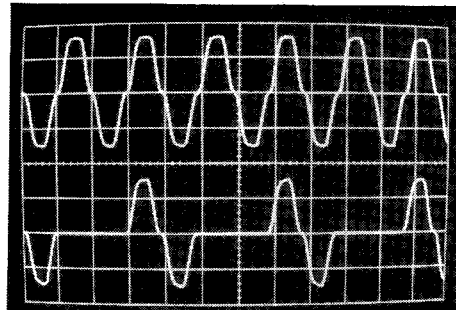
商用電源の元波形と制御された波形のオシロスコープ画面を第13図に、パーソナルコンピュータ画面に表示された波形のグラフィックを第14図に示す。

オシロスコープ画面において上の波形が商用電源の元波形で下の波形が制御された波形であり、制御された波形は1周期毎に断続されている事が分かる。なお、波形が切られている部位が直線になっていないのはSSRによって電源から切り離された出力端子にノイズが乗ってしまったためである。同じくパーソナルコンピュータディスプレイ画面においても上が商用電源の元波形で下の波形が制御された波形である。波形には信号絶縁に使用しているフォトカップラ特性により、0 V 付近ではつぶれる為に歪みが見られる。また、同じ理由からオシロスコープ画面で見られたノイズの波形もパーソナルコンピュータには取り込まれず電源から切り離されている区間は綺麗な直線で描画されている。

以上の結果により、今回開発したインターフェース回路と出力ルーチンの組み合わせることにより、複数の変化の早い信号を処理する制御に十分対応できるシステムが構築できる事が確認した。なお、実験にはNEC PC-9801BX (CPU 80486SX 20MHz)を使用した。



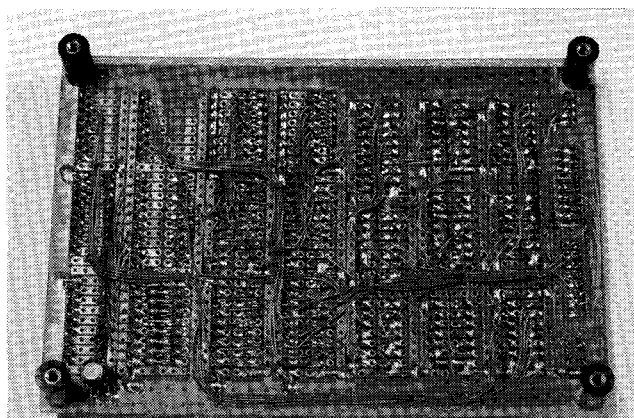
第13図 交流波形制御オシロスコープ画面



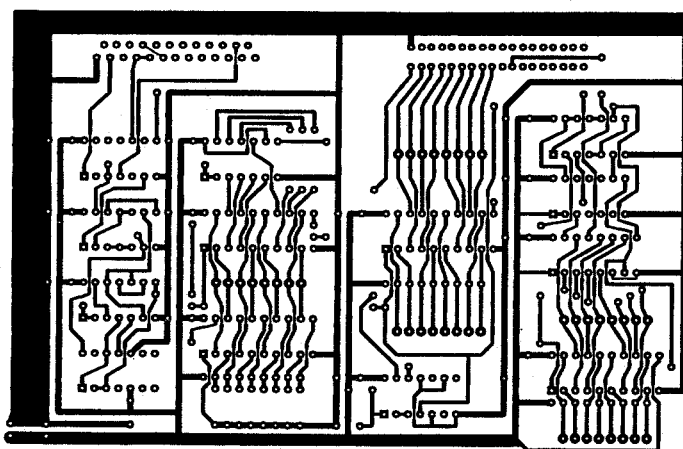
第14図 交流波形制御ディスプレイ画面

### 5 開発したインターフェースの教育現場への普及のための工夫

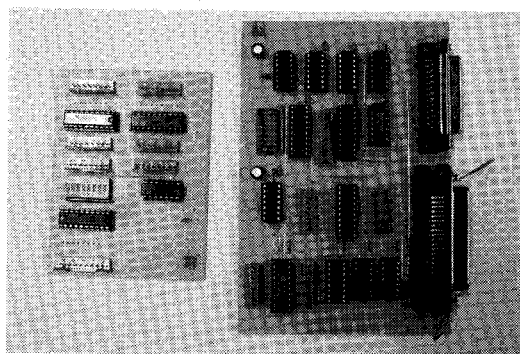
今回開発したインターフェースは回路構成が比較的煩雑であり、万能基板での手配線では現場教師が製作するのは難しいと思われる。(第15図)そこで、インターフェース回路とポート増設基板などのプリントパターンを設計し、現場に供給するための工夫をおこなっている。第16図にインターフェース回路のプリントパターンを、第17図に試作したインターフェース基板の写真を示す。このプリントパターンは、パーソナルコンピュータのビットマップファイル化している。ビットマップファイルはMS-WINDOWS付属の「ペイントブラシ」をはじめ多くの画像処理ソフトで処理できるのでパターンをプリンタで印刷し、そのまま感光基板作成の原版として利用することが出来る。このビットマップファイルをインターネットを利用して配布することも可能である。



第15図 インターフェース回路万能基板



第16図 インターフェース回路プリントパターン



第17図 インターフェース回路プリント基板  
(右：インターフェース基本回路 左：ポート増設基板)

## 6 ま と め

学校教育現場で利用しやすい異機種間での互換性があり、多変数制御に対応しうる最大256ポートの増設が容易にできるインターフェースを開発が出来た。また、入出力ルーチンの改良により、音声信号にも対応できる最大16kHzのサンプリング周波数を実現すると共にMS-WINDOWS環境下でも制御教材の開発・利用が容易に行えるようにDLLの開発も行った。さらに、多ポート、高速インターフェースの応用例として交流波形制御システムを作製して複数の変化の早い信号を処理する制御に十分対応できるシステムが実現できる事が確認された。

今後は、このインターフェースの応用例を幾つか設計して、そのハードウェアとソフトウェアを現場に供給するための作業を進め、これと同時にインターフェース技法の教育現場への供給の方法について、インターネット利用などの検討も行っていく必要がある。

### 参考文献等

- 1) 亀山 寛「情報基礎」の教育内容」(日本産業技術教育学会誌第30巻3号, 1988年)
- 2) 佐々木亨・近藤義美・田中喜美編 「技術科教育法」  
(学文社 亀山 寛: 第Ⅱ部第6章「情報基礎の学習指導」, 1990年)
- 3) 鈴木寿雄「情報基礎」の実施をめぐって」(K G K Jurnal VOL.28 No2)
- 4) 佐伯胖, 坂村健, 赤城昭夫「コンピュータと子供の未来」(岩波書店, 1988)
- 5) 第35回全日本中学校技術・家庭科研究大会  
第36回東北地区中学校技術・家庭科研究大会  
第39回岩手県中学校技術・家庭科研究大会  
岩手大会要録 '96
- 6) WWW:マイクロソフト社ホームページ「サポート技術情報」  
<http://www.microsoft.co.jp/pss/tdoc/vb/BVJ090.htm>
- 7) 杵渕 信, 菅野徳明「セントロニクスインターフェイスを利用した逐次比較型A/Dコンバータのデータ入力手法」(日本産業技術教育学会誌第32巻4号, 1990年)
- 8) 杵渕 信, 安孫子啓「RS-232Cを利用したA/Dコンバータの教具化」  
(日本産業技術教育学会誌第33巻1号, 1991年)