Ｗｉｎｄｏｗｓ上での計測と制御

―簡易ＡＤコンバータによる計測―

大久保　政俊

本センターで開発したプリンタポート使用の簡易ＡＤコンバータをオペアンプを使って増幅と同時に正負の電圧が取り込めるように改良した。また，このコンバータをWindows上で制御できるように，開発言語Delphiを用いて完全実行形式のファイルを作製した。これによりWindows上でもフロッピーディスクからソフトを起動させて計測することが可能になり，活用範囲が飛躍的に拡大した。

［キーワード］高等学校理科　コンピュータ　ＡＤコンバータ　Windows　計測・制御　Delphi

Ⅰ　はじめに

Windowsになってハードウエアを直接アクセスすることが制限されるようになって，自作の周辺装置を接続して動かすことが難しくなっている。例えばVisualBasicでI/Oポートを通してアクセスしようとしたら，他の言語（例えばVisual C++）で作ったDLLファイルをWindowsのシステムにコピーしておく必要が出てくる。つまり，VisualBasicでは完全実行形式の実行プログラムはできないので，DLLファイルも一緒に配布するわずらわしさがある。

ここでは，従来MS-DOS上のBasicで制御していたＡＤコンバータを，Windows上でDelphiのアセンブラを使ってI/Oポートを制御し，測定できるようにした（フロッピーディスク上の完全実行形式のEXEファイルから計測・制御できる）ことについて説明する。また，正負の電圧も取り込めるようにオペアンプを使って改良したことにより，活用範囲が広がったので，これに関する実験実習例や授業実践例もあわせて報告する。

Ⅱ　ＡＤコンバータ

このＡ／Ｄコンバータは北海道立理科教育センターが平成４年度長期研修員の田中佳典先生，北海道高等学校理科研究会の菅原陽先生，萬木貢先生らの協力の下に製作した逐次比較型の８ビットＡ／Ｄコンバータで電源電圧３～９Ｖで作動し，回路電流も小さいため，パソコンから電源を得て使用することができる。またこの回路は以前報告した２チャネルＡＤコンバータ(p25の図4b)の姉妹品で仮グランドを持ち，＋－の電位を測定することができる。しかも変換速度が２９μｓと速いため音声の取り込みも可能である。このボードを作動させるソフトは従来はＭＳ-ＤＯＳ上で制御していたが，開発言語 Delphiで実行型のファイルを作成したため　Windows上で制御することができるようになり,ＡＤコンバータが低価格で作製できることとあいまって活用範囲が飛躍的に拡大した。

1.　ＡＤ変換のための機器構成

使用機種　　ＮＥＣ9801,9821シリーズ,
　　ＰＣ／ＡＴ互換機
接続端子　　プリンタ端子
オペアンプ　ＴＬＣ27Ｌ2
プログラム MS-DOS上でN88BASIC，Windows上では Delphi(Pascal,Assembler)を使用

このボードは，汎用性を高めるため，ＬＴＣ1098に高入力インピーダンスのオペアンプＴＬＣ27Ｌ2を接続し，パソコンにはプリンタ端子を通して信号の交換を行う。

２．ＬＴＣ1098の主な特徴

ＬＴＣ1098には次のような特徴がある。

(1) 作動させるためには３～９Ｖの間の電源電圧を供給しなければならない。このとき，フルスケールは電源電圧と同じである。
(2) 消費電力が小さく，パソコンから電源を供

給することができる。
(3) あまり大きな電流をセンサーに供給することができない。
(4) 入力チャンネル数は，シングル２チャンネルと差動１チャンネルがソフト上で選択きる。
(5) 入力インピーダンスが大きくセンサーの電圧を直接測定することができる。
(6) サンプルアンドホールドが内蔵されている。
(7) 価格はDIP品で１個800円くらい
(8) 測定データは８ビットのシリアルデータとして送出する。

３．ＡＤコンバータ：ＬＴＣ1098

ＬＴＣ1098のトップビューは図１のとおりである。

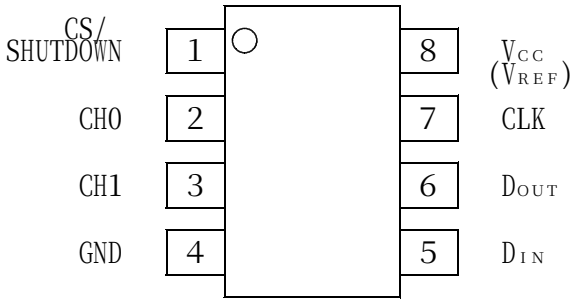


図１　ＬＴＣ1098　TOP VIEW

４．プリンタ端子との接続

通常，パソコンにはプリンタを接続するためのプリンタ端子がついていて，ほとんどがセントロニクス社が定めた規格に準拠している。プリンタ端子の各ピンのうち，データ有効フラッグ(PSTB)に１点(端子1)，データ転送用(DATA)に８点(端子2,3,4,5,6,7,8,9)，データ転送許可フラッグ(BUSY)に１点(端子11)，グランド(GND)に１点(端子19)が割り当てられている。すべての信号は，0～5VのＴＴＬレベルで転送されており，その速度は毎秒数十万バイトと高速である。

パソコンへの接続は図２のように行い，計測したデータはMS-DOS上でN88BASIC，Windows上ではDelphi(Pascal,Assembler)を使用して，パソコンのプリンタ端子から読み取る。

プリンタ端子			ＬＴＣ１０９８
ＰＳＴＢ	1○	――	7（ＣＬＫ）
D0	2○	――	5（D_{IN}）
D1	3○	――	1（ＣＳ）
D2	4○		
D3	5○		
D4	6○	――	8（V_{CC}）
D5	7○	――	8（V_{CC}）
D6	8○	――	8（V_{CC}）
D7	9○	――	8（V_{CC}）
ＢＵＳＹ	11○	――	6（D_{OUT}）
ＧＮＤ	19○	――	4（ＧＮＤ）

図２　ﾌﾟﾘﾝﾀ端子とADｺﾝﾊﾞｰﾀ端子の接続

５．プリントパターン

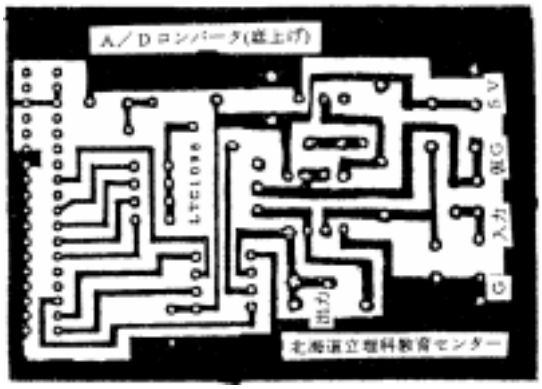


図３－a　ＡＤコンバータプリントパターン（改良版，１チャネル増幅，底上げ用）

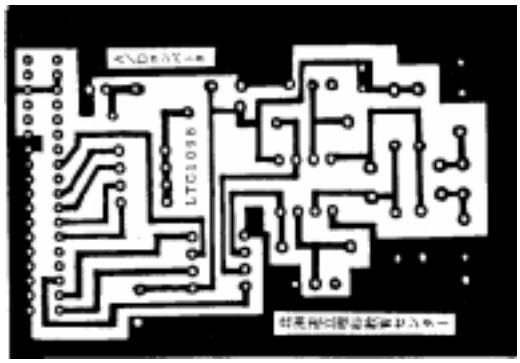


図３－b　ＡＤコンバータプリントパターン（２チャネル増幅，差動用）

６．回路図

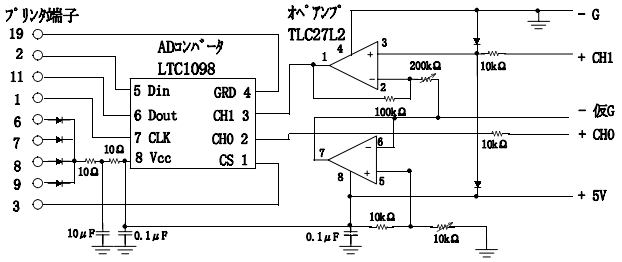


図４－ａ　ＡＤコンバータ（LTC1098）及びオペアンプ部回路図（改良版，１チャネル増幅，底上げ用）

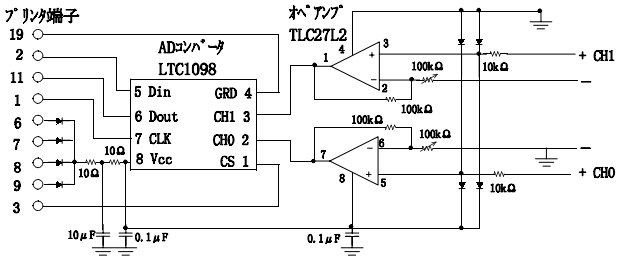


図４－ｂ　ＡＤコンバータ（LTC1098）及びオペアンプ部回路図（２チャネル増幅，差動用）

７．オペアンプによる増幅

オペアンプには消費電力が小さく，入力インピーダンスが大きいＴＬＣ27Ｌ2（ﾃｷｻｽｲﾝｽﾂﾙﾒﾝﾂ）を使用した。このオペアンプには２つの増幅器が内蔵されており，１チャンネルの入力を1.5倍以上に増幅できるように非反転増幅回路で増幅回路を作成した。

増幅率　Ａ＝１＋100／Ｒ
Ｒは200ＫΩの可変抵抗

オペアンプの電源は3～9Ｖの間で正常に作動するが，この回路の場合，Ａ／Ｄコンバータ同様パソコンのプリンタ端子からとっているため，供給されている電圧は 5Ｖ弱程度である。ノートパソコンの場合は３Ｖ強程度である。一般にオペアンプで増幅した後の電圧の上限は供給電源の70%くらいであるので，電圧５Ｖの場合，増幅後の電圧の上限は3.6Ｖくらいである。

８．オペアンプによる上げ底（仮グランド）

１個のオペアンプで図５のように底上げ（０～2.5Ｖ）をして仮グランドを作っているので，＋と－の電圧を出力するセンサーを仮グランドと入力（１チャネル）に接続することで電圧の測定が可能である。また増幅は仮グランドを基準にしてなされる。ただし，センサーから大きな電流が流れこむと仮グランドが不安定になることがあるので，使用するセンサーの特性に注意を払う必要がある。

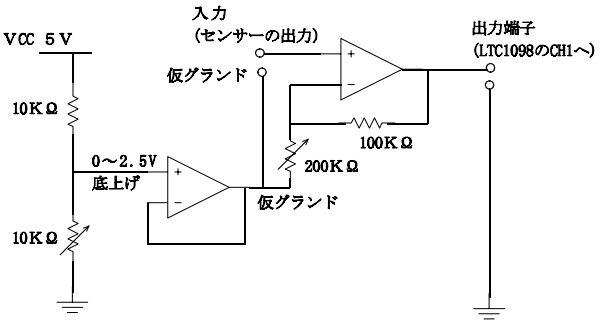


図５　オペアンプによる底上げと増幅

９．ＬＴＣ1098の優位点

リニアテクノロジーのＬＴＣ1098は消費電力が小さく，パソコン本体から作動電源を供給することが可能である。このＡ／Ｄコンバータは逐次変換方式で，サンプルアンドホールド内蔵の８ビットＡ／Ｄコンバータで，２チャンネルをソフトウエアで切り替えて使用することができる。また，DIFFモードでは，CH0とCH1の間の電位差をソフト上で極性を選択の上測定することができる。作動時の電流は80μＡ，非作動時にはＣＳをＨにしておくと3μＡに低下する。

電源は 3～9Ｖの単一電源で作動し，最大クロックは500KHzで，このときの変換時間は16μＳである。

入力インピーダンスが高く，各種センサを直接接続することが可能である。

Ⅲ　簡易ＡＤコンバータの製作

１．材料

プリント基板（感光基板，現像液，エッチング液，プリントパターン，アセトン）

２．道具

ハンダごて，ハンダ，こてスタンド，ニッパー，ラジオペンチ

３．部品

プリント基板			１枚
抵抗	茶黒黒	10Ω	２個
	茶黒橙	10ＫΩ	３個
	茶黒黄	100ＫΩ	１個
可変抵抗	W103	10ＫΩ	１個
	W204	200ＫΩ	１個
電解コンデンサー		10μＦ	１個
セラミックコンデンサー		0.1μＦ	２個
スイッチングダイオード		IS1588	６個
ＩＣ用ソケット（８ピン）平ピン足			２個
ＡＤコンバータ LTC1098 ﾘﾆｱﾃｸﾉﾛｼﾞｰ			１個
オペアンプ TLC27L2 ﾃｷｻｽｲﾝｽﾄﾙﾒﾝﾂ			１個
プリンタ端子（セントロニクス）			１個

４．製作順序

(1) プリントパターンをケント紙に黒インクで描き，原画を作成する。
(2) 原画をコピー機で原寸大になるように縮小して複写し，ネガを作成する。黒色が鮮明でないときは，レジストペンで修正する。
(3) ポジ感光基板に焼き付ける（15Wの蛍光灯で約15分間照射する）
(4) ポジ感光基板を現像液（液温約35℃）に浸し現像する。
(5) 水洗い後，修正が必要ならレジストペンで修正する。
(6) エッチング液（塩化第二鉄溶液）（液温約40℃）につけ，不要な銅を溶かし流す。
(7) アセトンで残っている感光皮膜を取り除く。
(8) プリント基板にドリルで径0.8㎜の穴を空ける。基板にフラックスを塗り，錆を防ぐ。
(9) 部品をハンダ付けする。

＊注意

① プリンタコネクタを最初につける。コネクタのピンを穴に差し込むときは無理に押し込まずに片側からひとつずつ順番に差し込む。
② ダイオード，電解コンデンサー，可変抵抗の向きに注意する。
③ ジャンパー線は，抵抗を付け終わった段階で，切断した線を利用する。
④ ＩＣ（ＬＴＣ1098，ＴＬＣ27Ｌ2）は一番最後にソケットに向きを間違えずに差し込む.

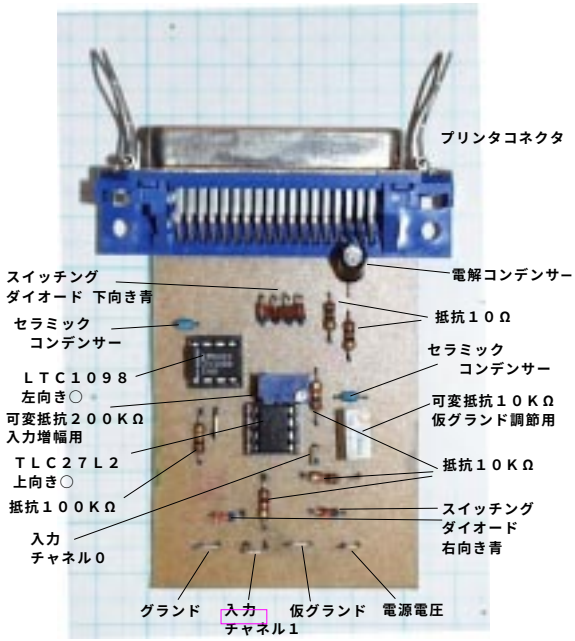


図６　簡易ＡＤコンバータ

５．動作確認（ソフトを起動させ確認。詳細はp44のⅤ(4)測定プログラムの動かし方参照）

(1) 入力チャネルと仮グランドをリード線でつないで可変抵抗10ＫΩのネジを右に回すと仮グランドの値（上げ底の値）が増えることを確認する。
(2) 可変抵抗200ＫΩを右に回して入力値が増える（増幅されている）ことを確認する。

＊部品入手先

・ＡＤコンバータ LTC1098　ﾘﾆｱﾃｸﾉﾛｼﾞｰ
株式会社アドバンセル　TEL 03-3445-5122
・オペアンプ TLC27L2　ﾃｷｻｽｲﾝｽﾄﾙﾒﾝﾂ
たくみ商事株式会社　TEL 03-3343-9630

Ⅳ　Ｉ／Ｏ制御

1．情報の内部処理

コンピュータの本体はＣＰＵを中心としてメモリー，Ｉ／Ｏポート等から構成されている。

(1) ＣＰＵ

コンピュータ全体をコントロールし，メモリやＩ／Ｏポートから受け取ったデータを処理する。

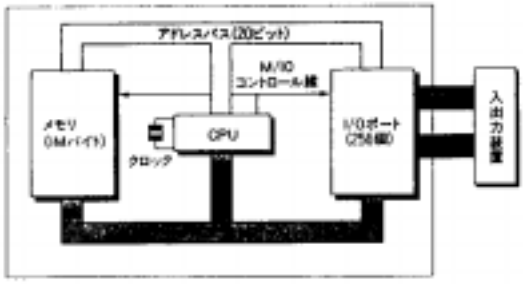


図７ コンピュータの構成

(2) Ｉ／Ｏポート

外部からコンピュータにデータを入力（Input）したり，反対にコンピュータから外部にデータを出力（Output）するための場所，実際はＬＳＩ。ポートは港の意味である。ＡＤコンバータなどから取り込まれたデータもここを通ってコンピュータの内部に入る。

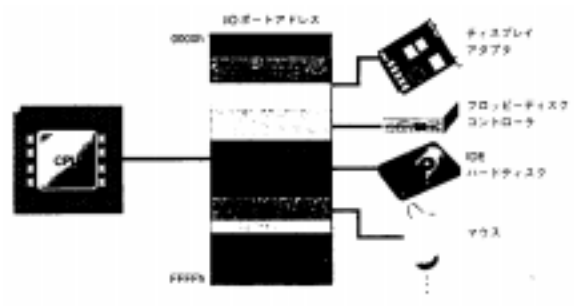


図11　Ｉ／Ｏポートアドレスとメモリ空間

(3) Ｉ／Ｏポートアドレス

ＣＰＵが周辺機器とデータをやりとりするときに使うためにに用意された６４Ｋバイトのメモリ空間である。データを通すこともあるが，主に制御のために使われる。

通常は0000ｈ～FFFFｈまでの640Ｋバイトの空間である。

図９ プリンタポートのＩ／Ｏアドレス

(4) メモリ

コンピュータ内部での情報の記憶場所である。通常はＲＡＭと呼ばれるＬＳＩで構成される。最低６４０Ｋバイトの記憶容量を持っている。Ｉ／Ｏポートとメモリにはそれぞれ独立に番地（アドレス）が割り当てられ，番地を指定することにより，メモリ内の個々の情報やＩ／Ｏポートの入出力場所を選ぶ。

(5) バス

ＣＰＵとメモリやＩ／Ｏポートの間をつなぐ信号線。バスには，データが通るデータバスと，メモリやＩ／Ｏポートの番地を示すためのアドレスバスがある。

２．プリンタインターフェイスの制御

パソコンのプリンタインターフェースの殆どは，セントロニクスインターフェースを使用している。PC98ではインターフェースには8255AというＬＳＩ（ＩＣ）が使用されている。

(1) 8255A

８ビットＣＰＵ（8085A）のコンピュータシステム用に開発されたプログラマブル汎用デバイスで，３組（１組：８ビット）の入出力ポート（ポートＡ，Ｂ，Ｃ）があり，各ポートはプログラムで入出力を制御できる。

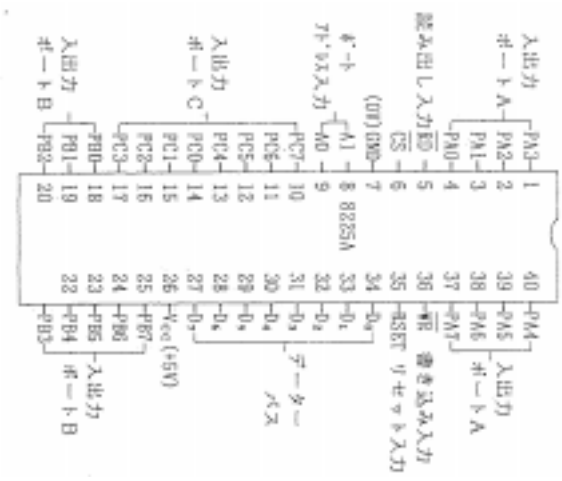


図１０　8255A　ＬＳＩ

(2) プリンタインターフェースのＩ/Ｏアドレス

ＰＣ９８ではプリンタインターフェース(8255A）のＩ／Ｏアドレスは次のように割り当てられている

ポートＡ	：４０Ｈ
ポートＢ	：４２Ｈ
ポートＣ	：４４Ｈ
コントロールレジスタ	：４６Ｈ

(3)　8255Ａの制御

①　ＣＰＵはコンピュータを起動したとき ＯＵＴ　４６Ｈ，８２Ｈを出力して8255Aを，モード０（ポートＡ：出力，ポートＢ：入力，ポートＣ：出力）に設定する。

表１　8255Ａのコントロールコマンド

8255 Aのコントロールコマンド

ポートA	ポートB	ポートC		コマンド
		0～3	4～7	
出力	出力	出力	出力	80
出力	出力	入力	出力	81
出力	入力	出力	出力	82
出力	入力	入力	出力	83
出力	出力	出力	入力	88
出力	出力	入力	入力	89
出力	入力	出力	入力	8A
出力	入力	入力	入力	8B
入力	出力	出力	出力	90
入力	出力	入力	出力	91
入力	入力	出力	出力	92

②　プリンタのデータ（ＤＡＴＡＢＵＳ）の出力は，ポートＡ（８ビット）を用いる。

③　ストローブ（ＰＳＴＢ）はポートＣの７番（Ｃ７），ビジー（ＢＵＳＹ）はポートＢの２番（Ｂ２）に割り当てられ，データのやりとりのタイミングをとるために使用されている。

④　Ｉ／Ｏアドレスの４０Ｈにデータを書き込むとＰＡ０～ＰＡ７に出力される。

⑤　４２Ｈを読み込み，２ビット目を調べることにより，ＢＵＳＹの状態がわかる。ＢＵＳＹ端子と8255Aの間にインバータは入っているため，信号が反転するので注意する。

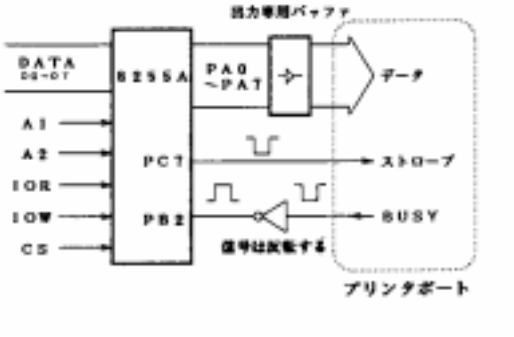


図14　プリンタインターフェース

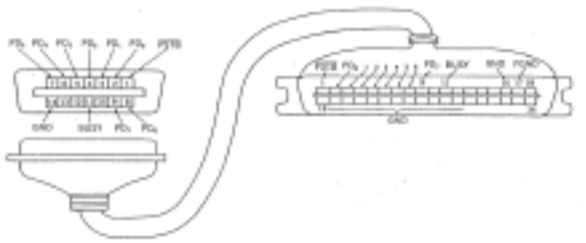


図15　プリンタコネクタのピン配置
（ＰＣ9801シリーズ用）

(4) プログラムによる8255Ａ制御

①　ＢＵＳＹの確認：
Ｉ／Ｏアドレス（＄Ｈ４２）を読み込み

```
ＩＮ　ＡＬ　＄４２
```

２ビット目を調べる
ＢＵＳＹがＨ（５Ｖ）であれば"０"となりＬ（０Ｖ）であれば"１"となる。

② ＰＡ０～ＰＡ７(ＤＡ０～ＤＡ７)の確認
Ｉ／Ｏアドレス（&Ｈ４０）にデータを書き込む。
プログラム
例　１ビット目と２ビット目がＨ（５Ｖ）

```
MOV AL, $3
OUT $40, AL
```

例　モニタＬＥＤの点灯
ＯＵＴ命令＄４０への書き込みデータとモニタＬＥＤの関係から表２のように10進数，2進数，16進数の関係がわかる。

表２　モニタＬＥＤの点灯とデータの書き込み

10進数	モニタLED	2進数	16進数
0	○○○○○○○○	00000000	0
1	○○○○○○○●	00000001	1
2	○○○○○○●○	00000010	2
3	○○○○○○●●	00000011	3
4	○○○○○●○○	00000100	4
5	○○○○○●○●	00000101	5
6	○○○○○●●○	00000110	6
7	○○○○○●●●	00000111	7
8	○○○○●○○○	00001000	8
9	○○○○●○○●	00001001	9
10	○○○○●○●○	00001010	A
11	○○○○●○●●	00001011	B
12	○○○○●●○○	00001100	C
13	○○○○●●○●	00001101	D
14	○○○○●●●○	00001110	E
15	○○○○●●●●	00001111	F
16	○○○●○○○○	00010000	10
17	○○○●○○○●	00010001	11
18	○○○●○○●○	00010010	12
19	○○○●○○●●	00010011	13
20	○○○●○●○○	00010100	14
32	○○●○○○○○	00100000	20
64	○●○○○○○○	01000000	40
128	●○○○○○○○	10000000	80
255	●●●●●●●●	11111111	FF

③ ボタンを押すとモニタＬＥＤがつく(消える)プログラム（Delphi使用）

```
unit Unit1;
interface
uses
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;
type
  TForm1 = class(TForm)
  Button1: TButton;
  procedure Button1Click(Sender: TObject);
  private
    { Private 宣言 }
  public
    { Public 宣言 }
  end;
  var
  Form1: TForm1;
implementation
{$R *.DFM}
procedure TForm1.Button1Click(Sender: TObject);
begin
  asm                    {アセンブラの始まり}
  mov AL,$FF     {$FFの代わりに$00とすると全部消える}
  out $40,AL     {ポート$40にALつまり$FFを出力する}
  end;           {アセンブラの終わり}
  end;
end.
```

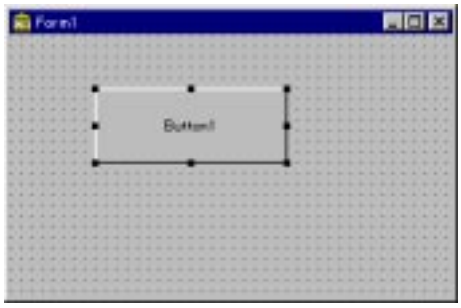


図１３プログラムの起動画面

(5) Windows上でのＩ／Ｏポートの制御

Windowsになってハードウエアを直接アクセスすることが制限されるようになって，自作の周辺装置を接続して動かすことが難しくなっている。

例えばVisualBasicでI/Oポートを通してアクセスしようとしたら，他の言語（例えばVisual C++）で作ったDLLファイル（I/Oポートを通してアクセスするように作成）をDeclairの宣言文で読み出さなければならない。つまり，VisualBasicでできない点を他の言語で作成したDLLファイルで機能追加することになる。そのためには事前にDLLファイルをWindowsのシステムにコピーしておく必要が出てくる。また完全実行形式の実行プログラムはできないので，プログラムを配布のときにDLLファイルも一緒に配布するわずらわしさがある。

では，Windows上でI/Oポートを直接制御できるC++を使えばという話になるが，C言語に特有の何でも設定しなければならず，グラフィックの部分の表示はなかなか難しく手ごわい。それよりはグラフィックを表示するコンポーネントを貼り付けて時間をかけずに簡単にプログラミングできる言語の方が楽である。

ところで，Borland社(最近Inprise社に改名)から出ているDelphiという言語は，Pascal言語であるが，実際に使ってみるとコンパイルや実行速度が速く，またAPI関数もDeclairの宣言文なしに関数のように直接呼び出すことが可能である。また容易にアセンブラが使えるので直接I/Oポートを制御することもできる。また，100近くのコンポーネントがついていて簡単に機能が追加でき，かつ完全実行形式の実行ファイルができ，大変優れものである。

VisualBasicもコンポーネントがついているが，これを使って他のコンピュータで実行形式のプログラム(EXEファイル）を動かそうとしたらコンポーネントに対応するOCXファイルをシステムに次から次へとコピーしていかなければならない。Delphiと違ってVisualBasicは完全実行形式のファイルを作成できない。

学校現場でこども達や先生方に自作のプログラムソフトを自由に使ってもらうためにはどうしても完全実行形式のファイルが必要である。

ここでは，従来MS-DOS上のBasicで動かしていたADコンバータをWindows上のDelphiのアセンブラを使ってI/Oポートを制御し，動かせるようにしたこと（当然ながら完全実行形式のEXEファイルができる）について説明する。

＊ Pascal言語の中で直接Assemblerを呼び出してポートを制御する。

① ポートへの出力：

```
MOV  AL, $(出力したい値を16進数の数字く)
OUT $(ポートの番号を16進数の数字) , AL
```

または

```
mov Ax ,$
mov dx,$ (DX指定でﾎﾟｰﾄ番号を格納すると
          16ﾋﾞｯﾄのﾃﾞｰﾀ (0～65535,HFFFF)
           の番号指定)
out dx,AX   { AXﾚｼﾞｽﾀと出力ﾎﾟｰﾄの間で
              ﾃﾞｰﾀ出力,
```

② ポートからの入力 ：

```
IN AL,$(８ビットで表示されるポートの番号
        を16進数の数字で書く)
```

または

```
MOV dx,$ (DX指定でﾎﾟｰﾄ番号を格納すると
          16ﾋﾞｯﾄのﾃﾞｰﾀ (0～65535,HFFFF)
         の番号指定)
IN AX,DX    { AXﾚｼﾞｽﾀと入力ﾎﾟｰﾄの間で
              ﾃﾞｰﾀ入力,
```

＊ Delphiでは組み込みアセンブラをasmとendを使ってObjectPascal文中に直接記述できる。

```
        ・・・・・・・ObjectPascal文
    asm         {アセンブラの始まり}
          ・・・ 組み込みアセンブラ文
          ・・・・・・・・・
    end;        {アセンブラの終わり}
        ・・・・・・ObjectPascal文
```

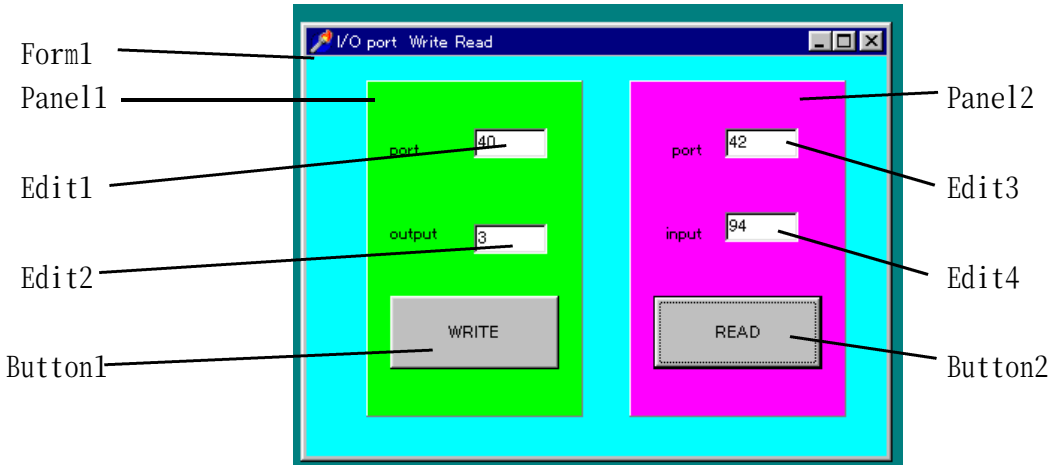


図１４　I／Oポートへの読み書きソフト起動画面

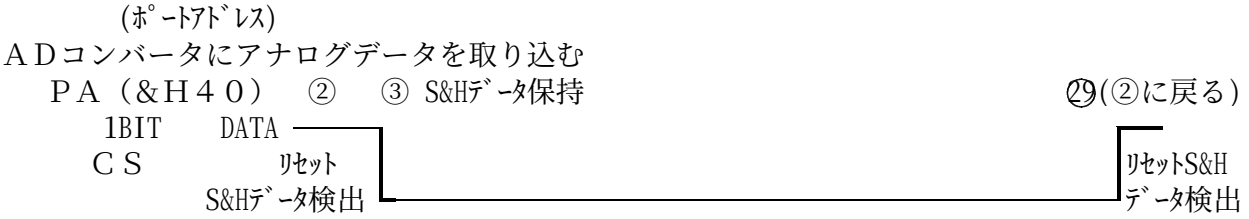
```
unit IOPORT;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls;
type
  TForm1 = class(TForm)
    Button1: TButton;
    Edit1: TEdit;
    Button2: TButton;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Panel1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Panel2: TPanel;
    Label3: TLabel;
    Label4: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private 宣言 }
  public
    { Public 宣言 }
  end;
var
  Form1: TForm1;
```

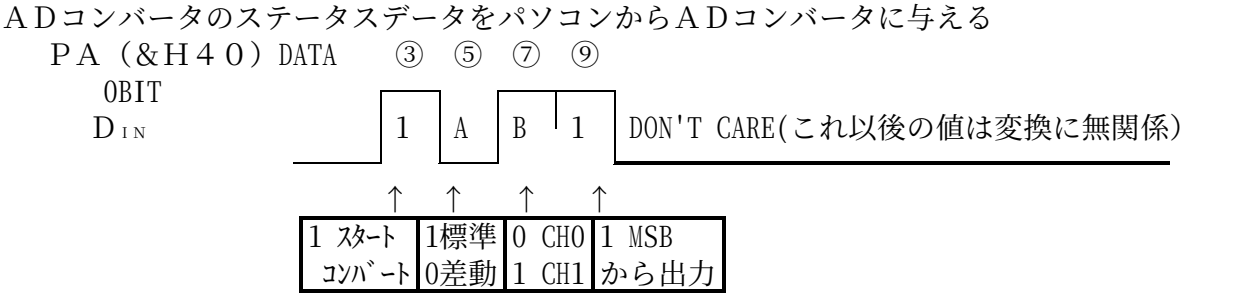
```
implementation

{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);     {ポートへの書き込みをする手続き}
var
 w:word;
 port:word;
begin
  w:=strtoint('$'+Edit2.text);        {書き込むデータを読み込み，文字からWordへ変換}
  port:=strToint('$'+Edit1.text);{書き込むポートアドレスを読み込み，文字からWordへ変換}
   asm                               {アセンブラの始まり}
     mov Ax ,w
     mov dx,port
     out dx,AX
   end;                              {アセンブラの終わり}
end;

procedure TForm1.Button2Click(Sender: TObject);     {ポートからの読みこみをする手続き}
var
 ri:integer;
 r:word;
 port:word;
begin
 port:=strToint('$'+Edit3.text); {読み込むポートアドレスを読み込み，文字からWordへ変換}
 asm                               {アセンブラの始まり}
   mov dx,port
   in ax,dx
   mov ah,0
   mov r,ax
 end;                               {アセンブラの終わり}
     ri:=r;                              {読み込んだデータをワードから整数に変換}
     Edit4.text:=inttoHEX(ri,2);         {読み込んだデータを１６進に変換して表示}
end;

end.
```

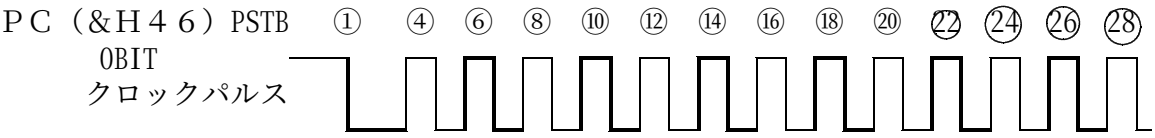
Ⅴ　ＡＤコンバータＬＴＣ１０９８用測定プログラム
(MS-DOS上はN88Basic，Windows上はDelphiを使用)

１．ＡＤコンバータＬＴＣ１０９８の制御のタイミング

ＬＴＣ１０９８はチップセレクト（CS）をアクチブ（LOW）にした後，最初のWAKEUP（SGL，CH0，CH1，DIFF，＋，－の選択を含む。）に続いてクロックに従い８ビットのデータが順次送出される。







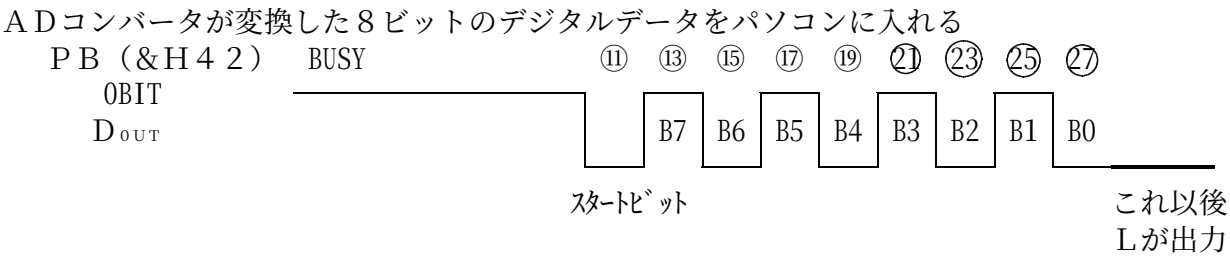


図１５　LTC1098 AD変換タイミングチャート

図１５の変換タイミングチャートにもとづいて，Ｉ／Ｏポート（＄４０，＄４６）にBIT情報を書き込み，ポート＄４２からBUSY信号を読み込むプログラムをMS-DOS上はN88Basic，Windows上はDelphiを使用してプログラムを作成した。

ＡＤコンバータの制御操作手順は次の表３の通りである。

表３　制御操作手順（ポートアドレスはＰＣ９８２１用）

操作順番	ﾎﾟｰﾄｱﾄﾞﾚｽ １６進	出力(入力) 整数	16進	ﾌﾟﾘﾝﾀ １番ﾋﾟﾝ CLK	ﾌﾟﾘﾝﾀ ２番ﾋﾟﾝ Din	ﾌﾟﾘﾝﾀ ３番 ﾋﾟﾝ CS	ﾌﾟﾘﾝﾀ ６番～９番 ﾋﾟﾝ Vcc	操作内容
①	46	14	E	L	L	L	L	ｸﾛｯｸ L
②	40	242	F2	L	L	H	H	power on 240で6～9がH ﾁｯﾌﾟｾﾚｸﾄ ﾘｾｯﾄ
③	40	241	F1	L	H	L	H	ﾁｯﾌﾟｾﾚｸﾄ ｱｸﾃｨﾌﾞ，ｽﾀｰﾄｺﾝﾊﾞｰﾄ１
④	46	15 14	F E	H L	H H	L L	H H	ｸﾛｯｸ H ｸﾛｯｸ L
⑤	40	241	F1	L	H	L	H	標準１
⑥	46	15 14	F E	H L	H H	L L	H H	ｸﾛｯｸ H ｸﾛｯｸ L
⑦	40	241	F1	L	H	L	H	1ﾁｬﾈﾙ選択
⑧	46	15 14	F E	H L	H H	L L	H H	ｸﾛｯｸ H ｸﾛｯｸ L
⑨	40	241	F1	L	H	L	H	ｽﾃｰﾀｽﾃﾞｰﾀ 1
⑩	46	15 14	F E	H L	H H	L L	H H	ｸﾛｯｸ H ｸﾛｯｸ L
⑪				L	H	L	H	ｽﾀｰﾄﾋﾞｯﾄ１が出力され操作なし
⑫	46	15 14	F E	H L	H H	L L	H H	ｸﾛｯｸ H ｸﾛｯｸ L
⑬	42			L	H	L	H	BIT7を入力
⑭	46	15 14	F E	H L	H H	L L	H H	ｸﾛｯｸ H ｸﾛｯｸ L
⑮	42			L	H	L	H	BIT6を入力
⑯	46	15 14	F E	H L	H H	L L	H H	ｸﾛｯｸ H ｸﾛｯｸ L
⑰	42			L	H	L	H	BIT5を入力
⑱	46	15 14	F E	H L	H H	L L	H H	ｸﾛｯｸ H ｸﾛｯｸ L
⑲	42			L	H	L	H	BIT4を入力
⑳	46	15 14	F E	H L	H H	L L	H H	ｸﾛｯｸ H ｸﾛｯｸ L
㉑	42			L	H	L	H	BIT3を入力
㉒	46	15 14	F E	H L	H H	L L	H H	ｸﾛｯｸ H ｸﾛｯｸ L
㉓	42			L	H	L	H	BIT2を入力
㉔	46	15 14	F E	H L	H H	L L	H H	ｸﾛｯｸ H ｸﾛｯｸ L
㉕	42			L	H	L	H	BIT1を入力
㉖	46	15 14	F E	H L	H H	L L	H H	ｸﾛｯｸ H ｸﾛｯｸ L
㉗	42			L	H	L	H	BIT0を入力
㉘	46	15 14	F E	H L	H H	L L	H H	ｸﾛｯｸ H ｸﾛｯｸ L
㉙	42							②に戻る

これ以後は②～㉘を繰り返す。

２．ＭＳ-ＤＯＳ上でのＮ８８Basicによる作成

ＮＥＣ９８０１の場合，プログラム命令と出入力ピンの関係は次のとおりである。

```
ＯＵＴ　＆Ｈ４６，＆Ｈ０Ｅ １ピンがＬ:CLK
ＯＵＴ　＆Ｈ４６，＆Ｈ０Ｆ １ピンがＨ:CLK
ＯＵＴ　＆Ｈ４０，＆Ｈ０１ ２ピンがＨ:DIN
ＯＵＴ　＆Ｈ４０，＆Ｈ０２ ３ピンがＨ:ADCS
ＯＵＴ　＆Ｈ４０，＆Ｈ００　２～９ピンがＬ
ＩＮＰ（＆Ｈ４２）11ピンからDATAを読み込む
```

この関係はＰＣ9821シリーズのWindows上でも同じである。

なお，プリンタ端子の１ピンは通常Ｈになっており，プログラムの終了時には，次のプリンタ使用のため，OUT &H46,&H0Fにして，１ピンをＨにしておかなければならない。

＜シングル計測プログラム＞

シングル計測のためのプログラムは表４のとおりである。

表４　Single計測（CH0,CH1）のプログラム

手順

ア　ＣＳ（チップセレクト）をＬにする。

イ　最初の４ビットのＤｉｎ（データイン）で，入力チャネル（ＣＨ０，ＣＨ１）を選択する。

ウ　ＣＬＫ（クロック）に従い８ビットのデータが順次送出される。

```
100  'SAVE "LTC1098
110   STOP  ON: ON STOP GOSUB *END
120   OUT &46,14       '1PIN LOW
130   OUT &40,242 '3,6,7,8,9 PIN HIGH
140   GOSUB *CH0  'CH0 MEASURE
150   LOCATE 20,10 :PRINT "CH0= ";DX
160   GOSUB *CH1  'CH1 MEASURE
170   LOCATE 40,10 :PRINT "CH1= ";DX
180   GOTO 140
190  *CH0
200    AL=11      '1BIT  1,1,0,1
210    GOSUB *ADIN
220    RETURN
230  *CH1
240    AL=15      '1BIT  1,1,1,1
250    GOSUB *ADIN
260    RETURN
```

```
270  *ADIN
280    AH=AL
290    FOR I=0 TO 3    '4BIT DATA IN
300    GOSUB *BTWT
310    NEXT I
320    DX=0             'DATA RESET
330    FOR I=0 TO 7     '8 BIT COUNT
340    GOSUB *BTRD
350    NEXT I
360    OUT &40,242      'CS HIGH
370    RETURN
380  *BTWT              '1BIT WRITE
390    ADCS=240          'CIP SELECT   CS LOW
400    AL=(AH AND 1) + ADCS
410    AH=INT(AH/2)
420    OUT &40,AL
430    GOSUB *CLK
440    RETURN
450  *BTRD               '1BIT READ
460    GOSUB *CLK
470    AL=INP(&H42)      'DATA READ
480    AL=((NOT AL) AND 4)/4     ’2BIT  2²=4  CHECK
490    DX=DX*2+AL
500    RETURN
510  *CLK
520    OUT &H46,15       'CLK HIGH
530    OUT &H46,14       'CLK LOW
540    RETURN
550  *END
560    OUT &H40,0        'AD CONBERT OFF 2-9 PIN LO
570    OUT &H46,15       '1PIN HI
580    CLS 3
590    END
```

３． Windows上でのDelphi(Pascal,Assembler)による作成

＊ Pascal言語の中でAssemblerを呼び出してポートを制御する。

・ポート番号が８ビット（＄００～＄ＦＦ）の場合
　ポートへの出力：　MOV　AL, $(出力したい値を16進数の数字で書く）
　　OUT　$(ポートの番号を16進数の数字で書く）, AL
　ポートからの入力　IN AL,$(ポートの番号を16進数の数字で書く）
・ポート番号が１６ビット（＄０１００～＄ＦＦＦＦ)の場合
　DXレジスタを使ってポートを指定しAXとDXの間で入出力する。
　具体的にはp30およびこれ以後のPC/AT互換機用のプログラムを参照のこと

＊　ＰＣ９８２１とＰＣ／ＡＴ互換機用のプログラム比較

(1) パラレルポートのアドレス

		ＰＣ９８２１	ＰＣ／ＡＴ互換機	
Ｉ/Ｏ ｱﾄﾞﾚｽ	データ	４０ｈ	２７８ｈ	３７８ｈ
	制御入力	４２ｈ	２７９ｈ	３７９ｈ
	制御出力	４４(４６)ｈ	２７Ａｈ	３７Ａｈ
機種		NEC 9821V12 9821NE2 9821Nr166	EPSON VN512ET ｹﾞｰﾄｳｪｲ Solo 5100 富士通 FMV-BIBLO NC13D	ｼｬｰﾌﾟﾒﾋﾞｳｽ MN-530-X1 NEC NX LW23/4

(2)　プリンタポートの使用方法(NEC)

端子名	ｱﾄﾚｽ	BIT	DAT	使用法	PI
DATA	H40	0	D0	ﾃﾞｰﾀ出力	2
		1	D1	ADCのCS	3
		2	D2		4
		3	D3		5
		4	D4	電源供給	6
		5	D5	電源供給	7
		6	D6	電源供給	8
		7	D7	電源供給	9
PSTB	H46			ADCｸﾛｯｸ	1
BUSY	H42	2	BUSY	ﾃﾞｰﾀ入力	11

プリンタポートの使用方法(PC/AT)

端子名	ｱﾄﾚｽ	BIT	DAT	使用法	PI
DATA	H378 H278	0	D0	ﾃﾞｰﾀ出力	2
		1	D1	ADCのCS	3
		2	D2		4
		3	D3		5
		4	D4	電源供給	6
		5	D5	電源供給	7
		6	D6	電源供給	8
		7	D7	電源供給	9
PSTB	H37A H27A			ADCｸﾛｯｸ	1
BUSY	H379 H279	7	BUSY	ﾃﾞｰﾀ入力	11

(3)　　ＰＣ９８２１用プログラム

ＯＵＴ　Ｈ４６，　Ｈ０Ｅ　1ピンがL:CLKL ＯＵＴ　Ｈ４６，　Ｈ０Ｆ　1ピンがH:CLKH
ＯＵＴ　Ｈ４０，　Ｈ０１　2ピンがH:DIN ＯＵＴ　Ｈ４０，　Ｈ０２　3ピンがH:ADCS ＯＵＴ　Ｈ４０，　Ｈ００　２～９ピンがＬ
ＩＮＰ（Ｈ４２）11ピンからDATAを読み込む busy信号は反転している。 Low　FF018594 High　FF018590 LowとHighの差が４＝2^2 ２ビット目が変化

ＰＣ／ＡＴ互換機用プログラム

ＯＵＴ　H3(2)７A，＆Ｈ０Ｅ　1ピンがH:CLKH ＯＵＴ　H3(2)７A，＆Ｈ０Ｆ　1ピンがL:CLKL
ＯＵＴ　H3(2)７8，＆Ｈ０１　2ピンがH:DIN ＯＵＴ　H3(2)７8，＆Ｈ０２　3ピンがH:ADCS ＯＵＴ　H3(2)７8，＆Ｈ００　２～９ピンがＬ
ＩＮＰ（H3(2)79）11ピンからDATAを読み込む busy信号は反転している。 ＊ EPSON VN512ET　Low　FF000FF8 High　FF000F78 ＊ ｼｬｰﾌﾟﾒﾋﾞｳｽ MN-530-X1　Low　FFFFEFEF High　FFFFEF6F ＊ ｹﾞｰﾄｳｴｲ　Solo　Low　FFFFEFCF High　FFFFEF4F ＊ NEC NX LW23/4　Low　FFCFCFFF High　FFCFCF7F LowとHighの差が８０ｈ＝１６×８＝2^7 ７ビット目が変化

ボタンをクリック

```
procedure TForm1.Button1Click(Sender: TOb
ect);
 var
roop:integer;
xmaxint:integer;
DXASM: word;
begin
    asm             {Assemblerの開始}

     mov al,$E      {ｸﾛｯｸL}
     out $46,al

   end;   {Assemblerの終わり

  xmaxint:=StrToint(Form1.Edit22.Text);
 for roop:=1 to xmaxint
 do begin
     CH1BWR;
    Data[roop]:=DXASM;
  end;
end;
```

すると測定を開始する手続き

```
procedure TForm1.Button1Click(Sender: TObjec
t);
 var
roop:integer;
xmaxint:integer;
DXASM: word;
begin
    asm          {Assemblerの開始}
①←――p34の表３の操作順番の番号
      mov ax,$F          {ｸﾛｯｸL}
      mov dx,$37A
      out dx,ax
 end;         {Assemblerの終わり}

  xmaxint:=StrToint(Form1.Edit22.Text);
 for roop:=1 to xmaxint
 do begin
    CH1BWR;
    Data[roop]:=DXASM;
  end;
end;
```

```
  {PC9821 １回データを読み込む手続き}
procedure CH1BWR;   {ＣＨ1から入力}
begin
    asm           {Assemblerの開始}

     mov AL,$F2 {240+2;poweron6～9とcsをH}
     out $40,AL




    mov Al,$F1   {241=240(poweron pin6～9)
    out $40,AL       +1(ﾁｯﾌﾟｾﾚｸﾄﾘｾｯﾄCSをL)
                       ｽﾀｰﾄｺﾝﾊﾞｰﾄ１}

     mov al,$F  {clock}
     out $46,AL
     mov al,$E
     out $46,AL

{ ALﾚｼﾞｽﾀと入出力ﾎﾟｰﾄの間でﾃﾞｰﾀの入出力,
 ﾎﾟｰﾄの指定ｲﾐﾃﾞｨｴｰﾄ値は8ﾋﾞｯﾄ(0～255,HFF)
 ﾃﾞｰﾀで示すﾎﾟｰﾄ番号}  *入出力で使用できる

     mov Al,$F1  {241=240(poweron pin6～9)
     out $40,AL          +1(標準１)}



     mov al,$F   {clock}
     out $46,AL
     mov al,$E
     out $46,AL




     mov Al,$F１ {$F1 240+1 1 ﾁｬﾈﾙ選択}
     out $40,AL  {$F0 240+0 0 ﾁｬﾈﾙ選択}
                 {240(poweron pin6～9)}

     mov al,$F   {clock}
     out $46,AL
```

```
  {PC/AT互換機１回データを読み込む手続き}
procedure CH1BWR;    {ＣＨ１から入力}
begin
    asm           {Assemblerの開始}
②
     mov Ax ,$F2   {240+2;poweron6～9とcsをH}
     mov dx,$378
     out dx,AX


③
    mov Ax ,$F1   {241= 240(poweron pin6～9)
    mov dx,$378      1(ﾁｯﾌﾟｾﾚｸﾄﾘｾｯﾄCSをL)
    out dx,AX              ｽﾀｰﾄｺﾝﾊﾞｰﾄ１}
④
     mov Ax ,$E    {clock}
     mov dx,$37A
     out dx,AX     { AXﾚｼﾞｽﾀと入出力ﾎﾟｰﾄの間で
     mov Ax,$F     ﾃﾞｰﾀ入出力, DX指定でﾎﾟｰﾄ番
     mov dx,$37A   号を格納すると16ﾋﾞｯﾄのﾃﾞｰﾀ
     out dx,AX     (0～65535,HFFFF)でﾎﾟｰﾄ指定
                    できる}
レジスタはAX,ALﾚｼﾞｽﾀだけである。
⑤
     mov Ax ,$F1   {241=240(poweron pin6～9)
     mov dx,$378          +1(標準１)}
     out dx,AX
⑥
     mov Ax ,$E    {clock}
     mov dx,$37A
     out dx,AX
     mov Ax ,$F
     mov dx,$37A
     out dx,AX
⑦
      mov Ax ,$F1   {$F1 240+1  1 ﾁｬﾈﾙ選択}
      mov dx,$378   {$F0 240+0  0 ﾁｬﾈﾙ選択}
      out dx,AX     {240(poweron pin6～9)}
⑧
     mov Ax ,$E      {clock}
     mov dx,$37A
```

```
    mov al,$E
    out $46,AL



    mov Al,$F1 {241=240(poweron pin6～9)
    out $40,AL         +1ｽﾃｰﾀｽﾃﾞｰﾀ１}


    mov al,$F   {clock}
    out $46,AL
    mov al,$E
    out $46,AL



    XOR DX,DX   {DX=0}

    mov al,$F   {clock}
    out $46,AL
    mov al,$E
    out $46,AL



IN AL,$42 {B7  read  11pin busy信号}
AND AL,$4 {2bit,High;0 AL=0,Low;2² AL=1}
SUB AL,$1{High0-1=0 CF=1,Low1-1=0 CF=0}
ADC DX,DX {DX=DX+DX+CF Highの時ｷｬﾘｰﾌﾗｸﾞ
                                CF=1}


    mov al,$F   {clock}
    out $46,AL
    mov al,$E
    out $46,AL



     IN AL,$42  {B6  read}
```

```
    out dx,AX
    mov Ax ,$F
    mov dx,$37A
    out dx,AX
⑨
     mov Ax ,$F1   {241=240(poweron pin6～9)
     mov dx,$378        +1(ｽﾃｰﾀｽﾃﾞｰﾀ１)}
     out dx,AX
⑩
    mov Ax ,$E    {clock}
    mov dx,$37A
    out dx,AX
    mov Ax ,$F
    mov dx,$37A
    out dx,AX

     XOR CX,CX     {CX=0}
⑪
    mov Ax ,$E     {clock}
    mov dx,$37A
    out dx,AX
    mov Ax ,$F
    mov dx,$37A
    out dx,AX
⑬
MOV dx,$379    {B7  read  11pin busy信号}
IN AX,DX
AND AX,$80 {7bit,High;0 AL=0,Low;2⁷ $80 AL=1}
SUB AX,$1{High0-1=0ﾎﾞﾛｰ=1 CF=1,Low1-1=0 CF=0}
ADC CX,CX   {CX=CX+CX+CF Highの時ｷｬﾘｰﾌﾗｸﾞ
                                    CF=1}
⑭
    mov Ax ,$E      {clock}
    mov dx,$37A
    out dx,AX
    mov Ax ,$F
    mov dx,$37A
    out dx,AX
⑮
     MOV dx,$379     {B6  read}
```

```
AND AL,$4
SUB AL,$1
ADC DX,DX
```

```
mov al,$F   {clock}
out $46,AL
mov al,$E
out $46,AL
```

```
 IN AL,$42   {B5  read}
AND AL,$4
SUB AL,$1
ADC DX,DX
```

```
mov al,$F   {clock}
out $46,AL
mov al,$E
out $46,AL
```

```
 IN AL,$42   {B4  read}
AND AL,$4
SUB AL,$1
ADC DX,DX
```

```
mov al,$F   {clock}
out $46,AL
mov al,$E
out $46,AL
```

```
 IN AL,$42   {B3  read}
AND AL,$4
```

```
 IN AX,DX
 AND AX,$80
 SUB AX,$1
 ADC CX,CX
```

⑯

```
 mov Ax ,$E      {clock}
mov dx,$37A
out dx,AX
mov Ax ,$F
mov dx,$37A
out dx,AX
```

⑰

```
 MOV dx,$379     {B5  read}
 IN AX,DX
 AND AX,$80
 SUB AX,$1
 ADC CX,CX
```

⑱

```
mov Ax ,$E       {clock}
mov dx,$37A
out dx,AX
mov Ax ,$F
mov dx,$37A
out dx,AX
```

⑲

```
  MOV dx,$379   {B4  read}
  IN AX,DX
  AND AX,$80
  SUB AX,$1
  ADC CX,CX
```

⑳

```
mov Ax ,$E     {clock}
mov dx,$37A
out dx,AX
mov Ax ,$F
mov dx,$37A
out dx,AX
```

㉑

```
  MOV dx,$379   {B3  read}
  IN AX,DX
```

```
SUB AL,$1
ADC DX,DX
```

```
mov al,$F   {clock}
out $46,AL
mov al,$E
out $46,AL
```

```
IN AL,$42   {B2  read}
AND AL,$4
SUB AL,$1
ADC DX,DX
```

```
mov al,$F   {clock}
out $46,AL
mov al,$E
out $46,AL
```

```
 IN AL,$42   {B1  read}
AND AL,$4
SUB AL,$1
ADC DX,DX
```

```
mov al,$F   {clock}
out $46,AL
mov al,$E
out $46,AL
```

```
 IN AL,$42   {B0  read}
AND AL,$4
```

```
AND AX,$80
SUB AX,$1
ADC CX,CX
```

㉒

```
mov Ax ,$E      {clock}
mov dx,$37A
out dx,AX
mov Ax ,$F
mov dx,$37A
out dx,AX
```

㉓

```
MOV dx,$379     {B2  read}
IN AX,DX
AND AX,$80
SUB AX,$1
ADC CX,CX
```

㉔

```
mov Ax ,$E      {clock}
mov dx,$37A
out dx,AX
mov Ax ,$F
mov dx,$37A
out dx,AX
```

㉕

```
MOV dx,$379     {B1  read}
IN AX,DX
AND AX,$80
SUB AX,$1
ADC CX,CX
```

㉖

```
mov Ax ,$E       {clock}
mov dx,$37A
out dx,AX
mov Ax ,$F
mov dx,$37A
out dx,AX
```

㉗

```
MOV dx,$379     {B0  read}
IN AX,DX
```

```
    SUB AL,$1
    ADC DX,DX



    mov al,$F   {clock}
    out $46,AL
    mov al,$E
    out $46,AL



    MOV DXASM,DX  {DXASMにdata  格納}

    XOR AL,AL      {AL=0}

  end;             {Assemblerの終わり}
end;             { procedureの終わり}
```

```
      AND AX,$80
      SUB AX,$1
      ADC CX,CX
㉘
      mov Ax ,$E       {clock}
      mov dx,$37A
      out dx,AX
      mov Ax ,$F
      mov dx,$37A
      out dx,AX

    MOV DXASM,CX       {DXASMにdata  格納}

    XOR AX,AX          {AX=0}

    end;             {Assemblerの終わり}
end;               { procedureの終わり}
```

(4) プログラムの流れ

a　ボタンをクリックすることで表３の操作手順①を実行する。コンピュータの起動時はプリンタポートの１番ピンは通常はＨｉｇｈになっているのでまずＬｏｗにする必要がある。

b　チャネル１からデータを読み込む手続きＣＨ１ＢＷＲをサンプリング数の最大値ＸＭＡＸＩＮＴに等しい回数，ループで呼び出す。

c　手続きＣＨ１ＢＷＲの中で表３の操作手順②～㉘を実行する。

d　ビットを読み込む操作手順毎に，BUSY信号がHighかLowか論理和（AND)で判定している。
BUSY信号は反転してコンピュータに入るのでHighの時０，Lowの時１がALまたはAXに格納される。

e　BUSY信号は反転してコンピュータに入るので，論理和の結果（ALまたはAXに格納されている値）と＄１の減算（SUB)を行って，キャリーフラグＣＦの値がBUSY信号がHighの時１，Lowの時０になる（反転を直す）ようにする。

f　キャリー付き加算(ADC)を行ってDX（PC9821の時）またはCXレジスタ（DOS/Vの時）に格納する。

g　手続きＣＨ１ＢＷＲの中でd～fを８回繰り返すことで８ビットの信号に2^{N-1}（Ｎビット目毎に）の重みがかかって加算され，アナログデータに対応するデジタルデータの値が得られる。

h　手続きＣＨ１ＢＷＲの最後でデジタルデータを変数DXASMに格納する。

i　手続きＣＨ１ＢＷＲを呼ぶループの中で，変数DXASMを配列変数DATA[roop]に格納する。
roopの最大値がサンプリング数になる。

４．Windows上での測定プログラムの動かし方

(1) 特徴

Delphiで作成し完全実行形式でFD上から実行できる。DLLやOCXをSYSTEMにコピーする必要がないので使いやすい。

(2) 対応機種

NECのPC9821ｼﾘｰｽﾞ用とDOS/V

(3) 使用方法

① 起動のさせ方

フロッピーから，スタート→ファイル名を指定して実行から，参照でEXEファイルを探す。これは起動するソフト以外のWindowを開かないためである。他のWindowが開いているとSYSTEMとこのWindowの間でメッセージのやりとりなどをして測定のソフトの実行が遅くなる。 ＊＊＊.EXEをダブルクリックすると図１６のような画面が立ち上がる。

図１６　測定プログラムの起動画面

② 測定間隔

ms単位で任意の数字を入て連続測定の時の値を決める。５５ｍｓ以下では不正確になる。

③ グラフ設定

グラフの幅，高さ，ｙの最大値，最小値，目盛，ｘの最大値，目盛を設定できる。ただし，ｘはサンプリング数で，これに測定間隔をかけると時間になる。

④ 数値の入力方法

カーソルを枠の中に移動させてクリックすると，文字が反転する。またはＴＡＢキーを押していくと順番に枠の中が反転し移動していく。反転後に数値を代入する。

⑤ 測定開始

連続測定表示ありのボタンを左クリックすると測定しながら左上の窓に数値を表示する。連続測定表示なしのボタンを左クリックすると測定するが表示はしない。グラフ表示するときは，こちらを使う。

⑥ グラフの表示

グラフ設定・表示ボタンを左クリックするたびに新規にグラフが作られる。あるいは，表示されたグラフ上でダブルクリックすると新規にグラフが作られる。

⑦ 測定しながらデータをグラフ上にプロット（折れ線グラフ）表示するには

連続測定表示なしのボタンを左クリック，グラフ設定・表示ボタンを左クリックした後，データプロット開始のボタンを押すか，またはグラフ上で右クリックするとデータをプロット表示（折れ線グラフ)する。Xの最大値を超えると，またグラフの 左からプロットする。データプロット停止のボタンを押すか，グラフ上で右クリックするとデータプロットが停止する。グラフ上で右クリックを押すたび毎にプロットを開始，停止が交互に行われる。開始は停止したところから再プロットされる。

⑧ 高速測定しながらグラフ上にデータをプロット（折れ線グラフ）表示するには

音声などを取り込む時には高速測定のボタンを押す。ただし，あらかじめグラフは表示しておく。グラフを表示しないで高速測定をするとグラフの設定・表示のメッセージが表示される。これは高速に取り込むためにデータを先にメモリに取り込んだ後，グラフにまとめてプロットするためである。Xの最大値で測定サンプリングは止まる。 ただしXの最大値は５０００。測定に要した時間は高速測定のボタンの下にｍｓ単位で表示される。この値はコンピュータのＣＰＵ及び状態に依存する。

Ⅵ　物理領域におけるＡＤコンバータを用いた計測

物理領域では瞬間的で非周期的な現象がよく現れるが，これを市販の計測器で測定しようとするとトリガ信号が検出できるシンクロスコープが必要となる。また現象を静止画像として再生するにはさらにメモリがついたシンクロスコープが必要である。このようなシンクロスコープは非常に高価であり，生徒用の実験台数を揃えるのはとうてい不可能である。しかし，これを数千円の簡易ＡＤコンバータを使ってコンピュータに取り込むことで容易に実現させることができる。

１．実験 音声の計測

(1) 準備

コンピュータ，A/D変換ボード（LTC1098)，マイク，シールドケーブル付き　変換コネクター，計測用プログラム，ドライバー

(2) 方法

① マイクの端子を変換コネクタに接続し，コネクタのケーブルの＋端子と－端子をそれぞれ，Ａ／Ｄ　変換ボード（ＬＴＣ１０９８）の＋入力と仮グランドに接続する。

② マイクのスイッチをオンにして，計測用プログラムを起動させ，連続表示ありボタンを押すと画面に値が表示されることを確認する。

③ 可変抵抗器（200KΩ）のボリュームをドライバーで右に回して増幅し，値が100～150程度になるように調節する。過度に増幅しすぎると波形がゆがむので注意する。

④ グラフ画面を表示した後，マイクに向かって声を出しながら，高速測定のボタンを押すと音声を取り込んだ後，グラフに音声の波形が表示される。

⑤ 下の波形が画面の下で切れる場合は上げ底の電圧を可変抵抗器（10KΩ）で大きくなるようボリュームをドライバーで右に回す。

(3) 考察

音声をいろいろと変えてみて（あいうえお，高い低い，大きい小さい），音声の波形の特徴を調べてみる中で，音の性質について定性的に調べる。

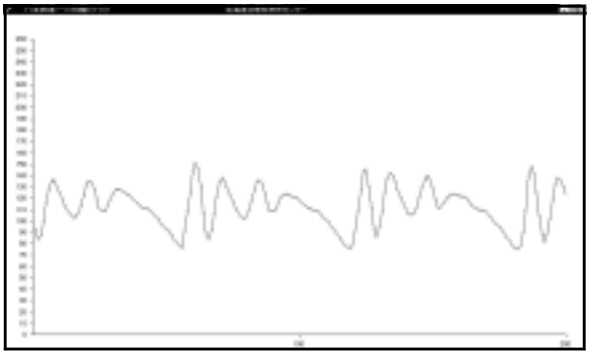


図１７　音声(あ)の波形

２．実験　電磁誘導の観察

(1) 準備

コンピュータ，A/D変換ボード（LTC1098），コイル，棒磁石，シールド線２本，計　測用プログラム，ドライバー

(2) 方法

① コイルの２つの端子とＡ／Ｄ変換ボード（ＬＴＣ１０９８）の＋入力と仮グランドをシールド線でつなぐ。

② 計測用プログラムを起動させ，連続表示ありボタンを押すと画面に値が表示されることを確認する。

③ コイルに棒磁石を出し入れをして値が100～150程度になるように可変抵抗器（200KΩ)のボリュームをドライバーで回して増幅，調節する。

④ グラフ画面を表示した後，測定時間間隔を50ｍｓ前後にして連続表示なしボタンを押し，グラフ画面上で右クリックすると測定値のグラフ表示が開始される。

⑤ コイルに棒磁石を出し入れすると，リアルタイムに変化の様子がグラフ表示される。

⑥ 下の波形が画面の下で切れる場合は上げ底の電圧を可変抵抗器（10KΩ）で大きくなるようボリュームをドライバーで右に回す。

⑦ グラフ画面上でさらに右クリックするとグラフ表示が停止する。

(3) 考察

コイルへの棒磁石の出し入れによって生じる電圧がどのように変化するか，検討する。

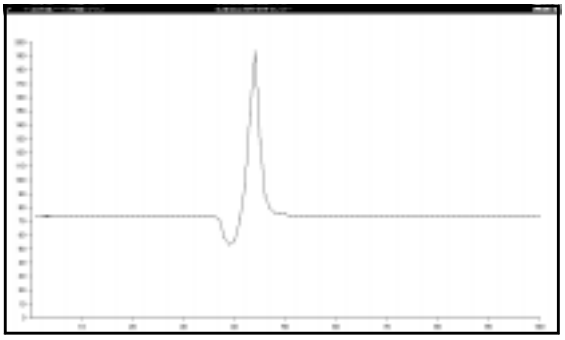


図１８　コイルから棒磁石を抜いたときの電圧変化

Ⅶ　化学領域におけるADｺﾝﾊﾞｰﾀを用いた 計測

１．実験　ゴムの断熱変化

(1) 準備

コンピュータ、計測ソフト、ＡＤコンバータ、K型熱電対、ＩＣクリップ、リード線、生ゴム（未加硫ゴム）、大型輪ゴム（加硫ゴム）、ナイフ、千枚通し、小型マイナスドライバー、凧糸

(2) 方法

① プリンターケーブルでＡＤコンバータとコンピュータを接続する。

② ＩＣクリップを用いて、熱電対の出力をＡＤコンバータの入力に接続する。

③ 生ゴムにナイフで長さ５㎜ほどの切れ目を入れ、次に切れ目に千枚通しを刺して切れ目を大きくして、そこに熱電対の先端を入れた後，千枚通しを抜く。

④ 切れ目に入れた熱電対が抜けないように、凧糸でしばる。

⑤ コンピュータの電源を入れ、WINDOWSを立ち上げる。

⑥ 計測ソフトをフロッピィデスクに入れる。

⑦ スタート → ファイル名を指定して実行 を左クリックする。

⑧ ファイル名を指定して実行の画面から、EXEファイルを確認して、OKを左クリックする。

⑨ ＡＤコンバータＬＴＣ１０９８の画面がでたら、連続測定・表示ありを左クリックする。

⑩ ＡＤコンバータの底上げ用の可変抵抗器をマイナスドライバーで右に回して増幅して、画面の数値を50程度に調整する。測定時間間隔を３００ｍｓにする。

⑪ 連続測定表示なしを左クリックし、グラフ設定・表示を左クリックする。

⑫ グラフの画面がでたら、右クリックして測定を開始する。

⑬ グラフを見ながらゴムをゆっくり引き、しばらくゴムを一定に伸ばした状態にした後、力を除いてゴムをもとに戻す。これを数回繰り返し、データを取る。

⑭ よいデータが得られないときは、ＡＤコンバータの可変抵抗器（200KΩ）をマイナスドライバーで右に回して増幅し調節する。

⑮ ゴムの代わりにに大型輪ゴムを用いて同様な実験をする。

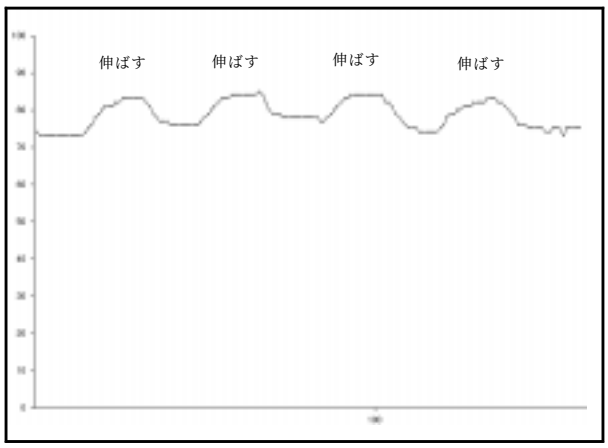


図１９　ゴムの断熱変化（温度変化）

(3) 考察

① 空気、ゴムの断熱変化について比較検討する。

② 未加硫ゴム、加硫ゴムの断熱変化について比較検討する。

Ⅷ　授業実践例（中学校地学分野）

１．実施日時，場所

日　時　平成１０年１１月１０日　５校時
場　所　北海道北広島市立東部中学校
　　３年Ｃ組　３４名　岩渕　健持 教諭

２．授業内容

本時　Ⅰ章　活動する大地
　　１　ゆれ動く大地　観察
　　地震によるゆれを調べよう(２校時目)

３．到達目標

地震によるゆれには２種類（初期微動，主要動）あることを見いだす。

(1) 地震シュミレーション装置の震源，震央を正しく指摘することが出来る。
(2) 地震による波の伝わり方をつかむことができる。
(3) 地震による観測点でのゆれ方をつかむことができる。
(4) 地震シュミレーション装置による地震と実際の地震はほぼ同じであることが理解できる。

４．授業の流れ

(1) 前の時間に押さえた重要事項の確認（震源，震央）のためコンピュータでアニメ中学校理科３年 第２分野（アドウィン）を見る。
(2) 地震発生時に撮影されたビデオを見せる。
　阪神淡路大震災記録　テレビ朝日作成ビデオ
(3) 地震シュミレーション装置によるゆれの伝わり方，観測点での揺れを観察する。
　地震シュミレーション装置（地震波モデル実験装置，図２０）

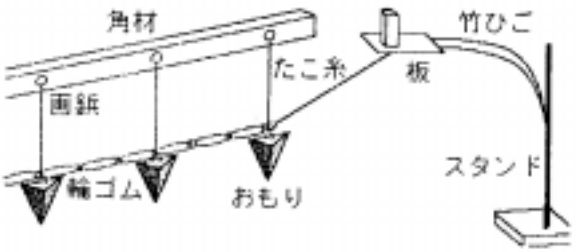


図２０　地震シュミレーション装置
（理科教育センター平成８年度長期研修生
岡久　保幸　教諭　作成
平成９年度東レ理科教育本賞受賞作品）

(4) 地震のゆれによる波形には２種類あることをつかむ。
　地震シュミレーションのゆれを地震計（ＡＤコンバータを使用）（図２１）で計測し，波形が２つの部分（初期微動と主要動）に分かれることを確認させる。

＊地震計
電磁誘導を利用している。地面に対応する板のゆれに応じて磁石が上下，左右に揺れることによってコイルに発生する起電力を計測する。

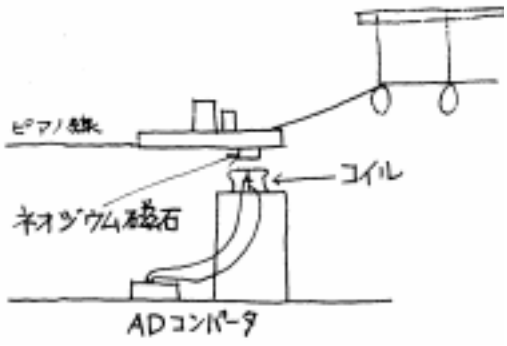


図２１　地震計

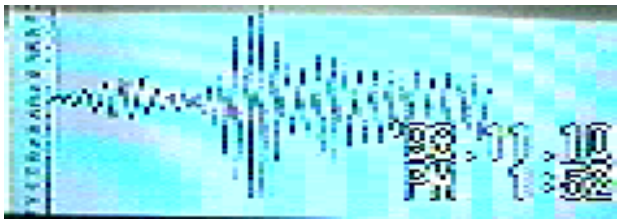


図２２　地震シュミレーション装置でのゆれの波形（ＡＤコンバータで計測）

(5) 実際の地震のゆれも地震シュミレーション装置によるゆれと同じ特徴（初期微動，主要動）をもっていることをつかませる。
地震のホームページより実際の地震によるゆれの波形をみて，ＡＤコンバータで計測したゆれの波形と比較する。（図２２と図２３の比較）

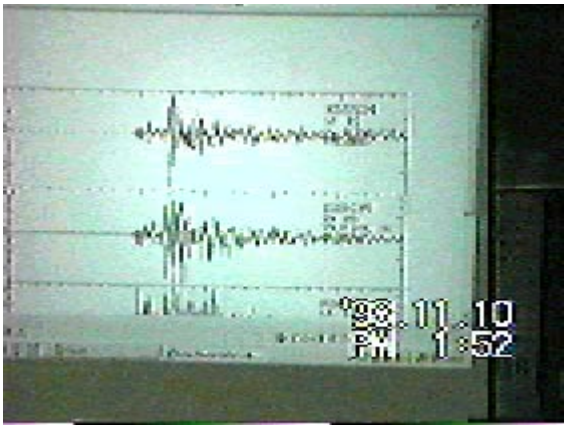


図２３　実際の地震のゆれの波形
（ホームページより）

(6) 地震によるゆれには初期微動と主要動の２種類があることをつかませ，まとめる。

Ⅸ　おわりに

今回開発したＡＤコンバータのハードおよびソフトについて大きく４点の特徴が挙げられる。

① ＡＤコンバータを正負の電圧が取り込めるように改良したことで今まで扱えなかった瞬間的非周期的な物理の現象にも対象範囲が広がった。

② DelphiのAssemblerでI/Oポートを制御することによりWindows上でＡＤコンバータを制御し高速計測することが可能になった。

③ PC9821およびDOS/Vの両方の機種で使える。

④ 開発言語のDelphiを使うと完全実行形式のファイルが作製でき，フロッピーディスク上から容易にＡＤコンバータを制御することができるようになった。

以上の①～④より簡単に高速な計測ができ，音声の変化を扱えるなど活用範囲が広がった。また，計測に要する時間が非常に短いので(１サンプリング約100μｓ弱)，電磁誘導の現象の観察などのように１サンプリングずつ計測する毎にリアルタイムにグラフ表示する（グラフ表示はCPUの時間がかかる）ことが可能になった。

今後，このＡＤコンバータが多くの学校で使ってもらえるよう，計画的に配布や貸し出しを検討していく必要がある。

最後に，今回のＡＤコンバータの改良製作にあたっては札幌南陵高校の菅原陽教諭，当センターの中村隆信事業課長に実際の技術開発を行って頂いた。また，ソフトの開発，特にAssemblerのプログラムについては当センターの平成４年度長期研修員の田中佳典教諭の研修集録によるところが大きいことを強調しておく。

Ⅹ　参考文献

中村隆信,田中佳典　LTC1098を用いた科学計測　北海道立理科教育センター研究紀要第５号　1993

中村隆信,萬木貢　LTC1098を用いた簡易型ＡＤ変換ボードの製作と化学実験教材　日本化学会第67春季年会講演予稿集Ⅰ　1994

大久保政俊　物理におけるコンピュータの活用　全国理科教育センター研究発表会物理　部会研究集録　1998

平成１０年度文部省情報教育指導者講座「高校理科」および「理科指導主事」テキスト　北海道立理科教育センター　1998

田中佳典　パソコンを計測機器として使用する方法の検討　北海道立理科教育センター　平成４年度後期長期研修集録　1993

コンピュータの活用編　理科教育指導資料第26集　北海道立理科教育センター　1994

互野恭治　万能ＡＤコンバータの製作　岩手県立総合教育センター　1996

高等学校の理科実験におけるコンピュータの活用（事例集）三重県総合教育センター　1996

コンピュータの計測・制御的な活用の研究　研究報告第９号　千葉県情報教育センター　1995

ボーランド Delphi3 ユーザーズガイド　インプライズ（旧ボーランド）株式会社　1997

（おおくぼ　まさとし　物理研究室研究員）