

情報工実験 1

A/D変換

035760A:横田敏明

1 実験状況

1.1 共同実験者

村山正嗣
宮城大輔

2 実験概要

2.1 A/D 変換とは

2.1.1 アナログとデジタル

アナログからデジタルへのデータ変換のことをいう。

アナログ信号は、定義された信号が時間に対して連続的で、なおかつ信号の値も連続的である。アナログとは、区間を無限倍まで拡大しても断続的な信号にはならない、連続的なデータのことをいう。

このアナログ信号を、ある一定の周期でサンプリングし、デジタルデータで近似することを A/D 変換という。サンプリングに関しては、その値の解析度とサンプリング周波数がデータの正確さを決定づける。

2.1.2 変換器の原理

変換は、入力と出力を比較し素早く近似することで実現している。例を示す。

出力を 3 ビットで表現すると、0~7 までの値を表現することができる。今、5 ボルトの入力電圧が発生したとすると、出力の最上位ビットが 1 になる。そのときの入力と出力を比較し、出力が 4、入力が 5 であるので、最上位ビットはそのままにする。次に、出力の上位から一つ下位のビットを 1 にする。このように繰り返し比較し、入力 5、出力 6 なので、今変更したビットを 0 にし、その一つ下位のビットを 1 にする。そして、デジタル出力の近似が完了する。

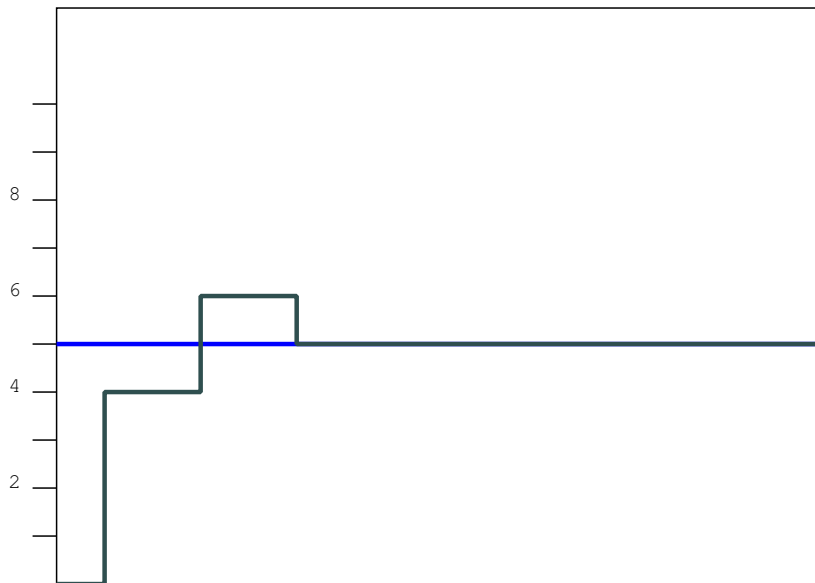


図 1: 入力と出力のチャート

2.2 目的

今回の実験では、市販の A/D 変換器を使用し、変換のしくみと変換器の使用法を学ぶ。使用した A/D ボードは AD12-8(PM)

2.3 実験内容

- サンプルプログラムが手元のノートパソコン上で作動することを確認せよ。何かキーを押すまで、全チャンネルで 0.1 秒おきに A/D 変換を行い、1 秒おきに表示。itemize

3 報告事項

3.1 サンプルプログラムが手元のノートパソコン上で作動することを確認せよ。

確認した。

3.2 何かキーを押すまで，全チャンネルで 0.1 秒おきに A/D 変換を行い，1 秒おきに表示

以下にソースコードを記す．

```
#include <stdio.h>
#include <conio.h>
#include <io32.h>
union _tag{
long l;
char c[4];
};
#define ADR 0x240
#define CHLS 8

int count;
float buf[80];

void main(void){
int ModeData =5;
union _tag ClockData ={ 999999 };
union _tag CountData={ 9 };
Int AiData,i,Count =0;
int AiVolt;
outp(ADR+6,0);
outp(ADR+6,1);
outp(ADR+7,ModeData);
outp(ADR+6,2);
outp(ADR+7,ClockData.c[0]);
outp(ADR+7,ClockData.c[1]);
outp(ADR+7,ClockData.c[2]);
outp(ADR+6,3);
outp(ADR+7,CountData.c[0]);
outp(ADR+7,CountData.c[1]);
//printf("          ");

for(i=0;i<CHLS;i++)
printf("  ch%d  ",i);
```

```

printf("\n");

outp(ADR+2,CHLS-1);
count=0;
do{
if(inp(ADR+2) &2){
count++;
for(i=0;i<CHLS;i++){
AiData=inpw(ADR);
AiVolt=(float)AiData *20/4096-10;
if(count%10==9){
printf("%d.3fv",AiVolt);
}
//Buf[count++]=AiVolt;
}
if(count%10==9){
printf("\n");
count=0;
}
if(kbhit()){
outp(ADR+7,1);
break;
}
}

}while( inp( ADR+2 ) & 3);
}

```

変更点を説明する .

```

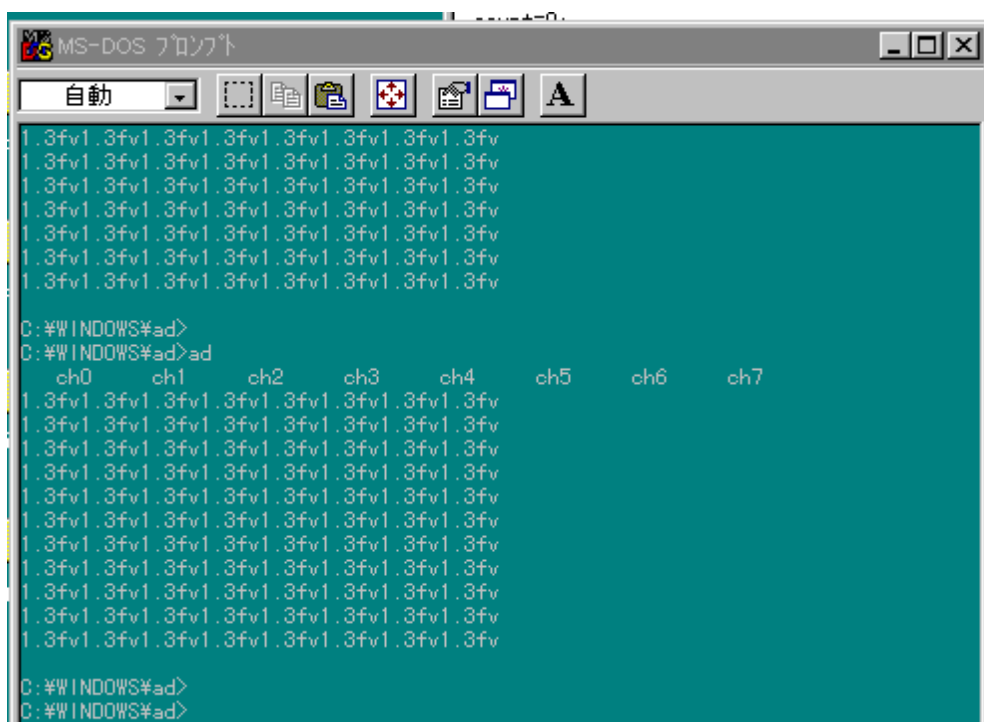
if(count%10==9){
printf("%d.3fv",AiVolt);
}

```

のように、カウント変数を用いて、10 回に 1 回出力するしくみに変更した。また、タイミングを 0.1 秒に合わせるため、

```
int ModeData =5;
union _tag ClockData ={ 999999 };
union _tag CountData={ 9 };
```

と変更した。



```
MS-DOS フロント
自動
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
C:\WINDOWS#ad>
C:\WINDOWS#ad>ad
  ch0    ch1    ch2    ch3    ch4    ch5    ch6    ch7
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1.3fv1
C:\WINDOWS#ad>
C:\WINDOWS#ad>
```

3.3 逐次変換以外の A/D 変換

3.3.1 フラッシュ型

コンパレータの数を贅沢に使い，処理速度を稼ぐ方式である．通常の 2 進数表示と異なり，1 であるビットの数そのものが値になるという．現在のように技術の発展で CPU の集積率が高くなってから実用的になった．

3.4 積分型

主に，アナログ電圧値を周波数変換するために用いる．

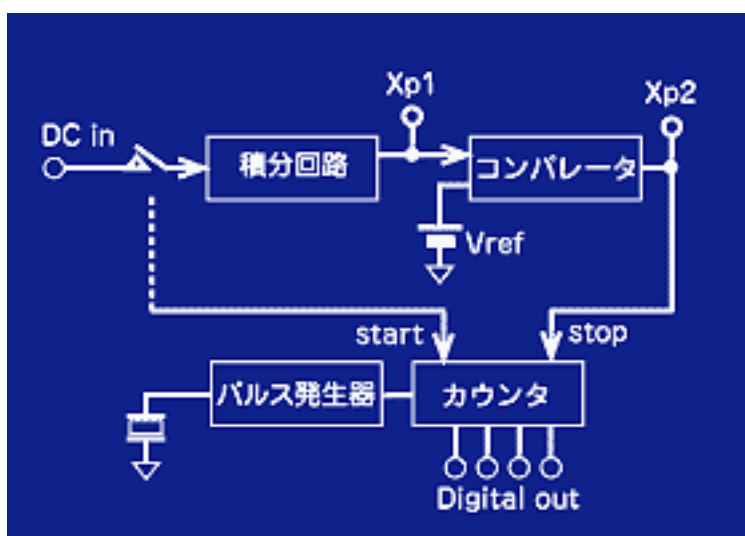


図 2: 積分型

直流 (定数) の時間積分が電圧入力なので，出力 (X_{p1} の信号) はゼロから直線的に上昇 (または下降) する．このとき，上昇率は入力された直流の大きさに比例する．積分器出力が上昇を続け，基準電圧を超えると，コンパレータの出力が反転する．コンパレータ出力の反転と同時に，下段のパルス計数を停止させる．停止した時点で出力する．

4 考察

今回の実験は，A/D 変換の概要を学んだ．

現実のあらゆる量は全てアナログ量であるが、コンピュータでの処理を行う場合はデジタルに変換しなければならない。

アナログからデジタルへ、変換することは近似することと同義であることがわかった。

変換プロセスとしては、今回の実験で逐次変換型を用いたが、調査によってフラッシュ型や積分型などがあることがわかった。