

情報工学実験 1

A/D変換

035743A : 比嘉雅樹

実験日 : 2004/06/18

提出日 : 2004/06/25

共同実験者 : Fグループ

035739B : 任家林

035741D : 浜川ありさ

1 実験の目的

A/D 変換の仕組みを学ぶと共に、市販の A/D ボードの使用法を習得することを目的とする。

2 実験結果

2.1 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 秒おきに表示するプログラム

```
#include<stdio.h>
#include<conio.h>
#include<io32.h>

union _tag{
long l;
char c[4];
};

#define ADR 0x240 // I/O アドレス
#define CHLS 8

void main(void)
{
int ModeData = 5; // モードデータをコマンドストップモードとマルチチャンネルモードにする
union _tag ClockData = {999999}; //0.1 秒おきにするよう設定
union _tag CountData = {9}; //サンプリングカウント数の設定
int AiData, i, Count=0;
float AiVolt;

outp(ADR+6,0); // アナログ信号の初期化
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);    //サンプリングを開始
do{
    kbhit();           //何かキーを押したら1を返す
    if(kbhit()==1){   //kbhit()が1、すなわち何かキーが
        押されたらサンプリングを停止する
        outp(ADR+6,4);
    }
    else if(inp(ADR+2) &2){
        if(Count%10==0){ //1秒ごとに表示させるようにする
            printf(" %5d: ",Count/10);
        }
        ++Count;
        for(i=0; i<CHLS; i++){
            AiData=inpw(ADR);
            AiVolt=(float)AiData * 20 / 4096 - 10;
            if(Count%10==0){
                printf("%+7.3fv ",AiVolt);
            }
        }
    }
}while(inp(ADR+2) & 3);
}

```

2.2 プログラムの実行結果

2.1 で書いたプログラムを実行すると、図1のようになる。

```

C:\#AD-0618>a1
  ch0      ch1      ch2      ch3      ch4      ch5      ch6      ch7
0: +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v
1: +1.416v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v
2: +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v
3: +1.421v +1.421v +1.416v +1.421v +1.421v +1.421v +1.421v +1.421v
4: +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v
5: +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v
6: +1.421v +1.421v +1.416v +1.421v +1.421v +1.421v +1.421v +1.421v
7: +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.416v
8: +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v
9: +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v
10: +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v
11: +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v
12: +1.421v +1.421v +1.421v +1.416v +1.421v +1.421v +1.421v +1.416v
13: +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v
14: +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v
15: +1.416v +1.421v +1.421v +1.421v +1.416v +1.426v +1.421v +1.421v
16: +1.421v +1.421v +1.416v +1.421v +1.416v +1.421v +1.416v +1.421v
17: +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v +1.421v

```

図 1: プログラムの実行結果

3 報告事項

3.1 A/D 変換探は逐次比較型以外にもいくつか存在する。代表的な物について調べ、それぞれの利点、欠点を述べよ。

- 追従型 A/D コンバータ

図 2 のような回路でカウンタに 0 から順にカウントさせ、その値（デジタル）を D/A コンバータに供給すると、D/A コンバータからそのデジタルデータに相当する電圧が発生されます。

その電圧と A/D 変換したい電圧 (V_{in}) をコンパレータで比べ、両者が一致したときのカウンタの値が、A/D 変換した結果ということになります。この方法の欠点としては、カウンタが常に 0 からカウントしていくため変換に時間がかかる上に、アナログ入力電圧の値によって変換時間がまちまちになってしまうことです。

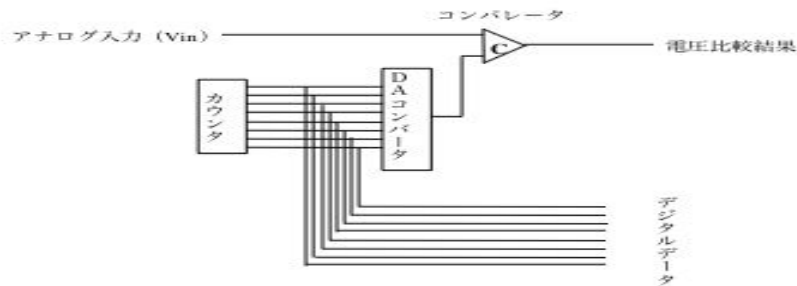


図 2: 追従型 A/D 変換コンバータ

● フラッシュ（並列）型 A/D コンバータ

逐次比較型方式は追従型に比べると高速の変換ができ、アナログ入力電圧の値に関係なく一定の変換時間が得られるので広く利用されています。しかしそれでもカウンタの値を 1 bit ずつ操作するので、それなりの変換時間がかかります。そこで「一発変換」しようとするフラッシュ（並列）型方式が考えられました。

この方式は、たとえば 8bit の A/D 変換をするなら、あらかじめ 256 通りの基準電圧と、256 個のコンパレータ（比較器）を用意しておいて、変換したい電圧をすべての基準電圧と一度に比べようというもので、下図のような回路を組むことで実現できます。

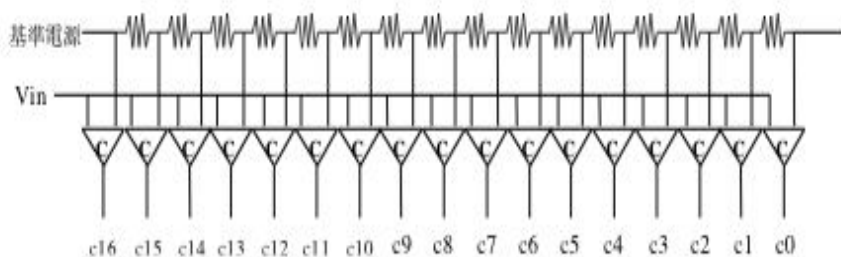


図 3: フラッシュ型 A/D コンバータ

この回路に A/D 変換する電圧 (V_{in}) を与えると、 $c_0 \sim c_{16}$ のどこかを境に大小が分かります。そこで $c_0 \sim c_{16}$ をエンコーダに供給し、デジタル値に変換します。

この方式では非常に高速な A/D 変換ができますが、変換の bit 数を 1 bit 上げると回路は 2 倍必要になるという欠点を持っています。

● 2 ステップ・フラッシュ型 A/D コンバータ

フラッシュ型では多くの回路を使用するため、変換速度をあまり落とさずに回路を少なくするための工夫として考えられた方式で、A/D 変換する電圧が大体どの範囲にあるかをまず調べ、次に細かく調べるという 2 段階の比較をする方式です。

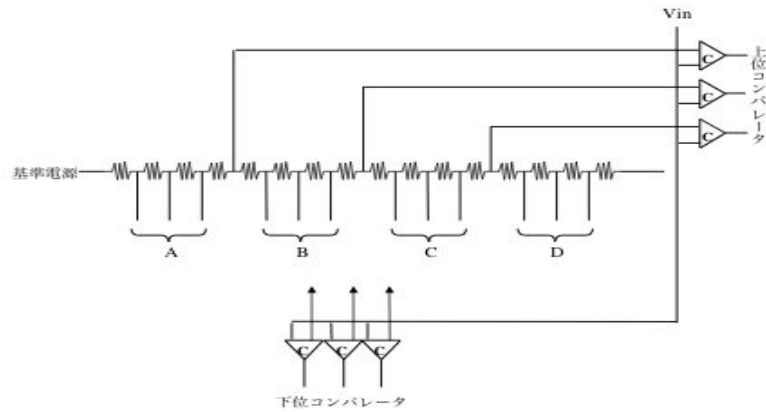


図 4: 2 ステップ・フラッシュ型 A/D コンバータ

上図のように、とびとびの基準電圧と V_{in} を上位コンパレータで比較し、 V_{in} がどの範囲にあるかを調べ、範囲が分かったらその範囲の基準電圧（図の A～D のいずれか）を下位コンパレータにスイッチしてやり、大小の分かれ目がどこにあるかを調べます。そして上位・下位コンパレータの信号をエンコーダに送りデジタル値を求めるものです。

4 使用器具

- ノート PC (Win95)、罫 01
- AD12-8(PM)

参考文献

- [1] 計測・制御ボードの仕様が分かる … 読む用語辞典 pp.3～5：アドテックシステムサイエンス
- [2] AD12-8(PM) 解説書