情報工学実験１
実験４
～A/D 変換～
学籍番号：035764C 若津 大悟

グループ H
実験実施日：平成１６年６月 15 日
提出〆切り日：平成１６年６月 22 日
共同実験者名
035762G：吉永　安磨
035763E：饒平名隆一

1　実験目的

A/D 変換の仕組みを学と共に、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

実験で作成したプログラム (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) のソースおよびサンプルプログラムからの変更点の説明。

```
#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 (その機材ごとに異なる)*/
#define CHLS 8
   /* チャンネルの数を指定 */

void main(void)
{
  union _tag ClockData = { 999999 };
     /*サンプリングクロックの長さの設定*/
  union _tag CountData = { 99 };
     /*サンプリングカウント数の設定*/

  int ModeData = 5;
     /* mode data を決める。入力があると
        サンプリングを停止する Comand Mode に設定*/
  int AiData, i, Count = 0;
     /* サンプリングカウント */
  float AiVolt;
```

```
outp(ADR+6, 0);
   /* アナログ信号の初期化*/
outp(ADR+6, 1);
   /* サンプリングモードの設定*/
outp(ADR+7, ModeData);
   /*チャンネル、クロック、ストップの設定が可能*/
   /*Channel=multi,Clock=intenal,Stop=command*/
outp(ADR+6, 2);
   /* サンプリングクロックの設定*/
outp(ADR+7, ClockData.c[0]);
outp(ADR+7, ClockData.c[1]);
outp(ADR+7, ClockData.c[2]);
outp(ADR+6, 3);
   /*サンプリングカウントの設定*/
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);
   /* サンプリング開始 */
do{
  if(inp(ADR+2) &2){
     /* DRE(Data Read Enable) ステータスを確かめる */
    Count++;
    if(Count % 10 == 0){
      printf("%7d ", Count/10);}
    for(i=0; i < CHLS; i++){
      AiData = inpw(ADR);
         /* データの入力*/
      AiVolt = (float)AiData * 20 / 4096 - 10;
      if(Count % 10 == 0){
        printf("%+7.3fv ", AiVolt);
           /* Count が 10 の倍数なら電圧を表示。*/
           /* つまりは、１秒おきに表示させるようにする*/
        }
```

```
      }
    }
  }
while(!kbhit());
    /*キーボードからの入力があったらループを終了。*/
  outp(ADR+6, 4);
    /*サンプリング停止コマンド*/
}
```

- サンプリングクロックの長さ

$$ClockData = \frac{100,000,000}{100} - 1 = 999,999$$

- プログラムの実行結果

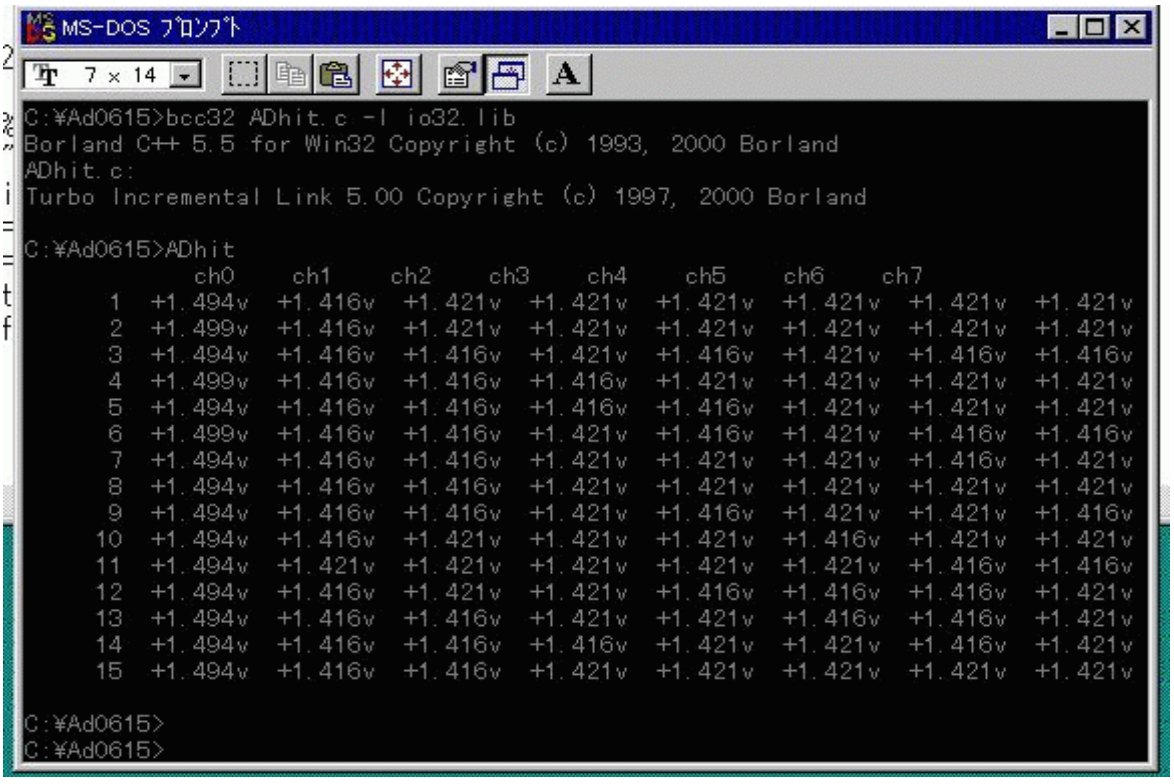

図 1: プログラムの実行結果

3　逐次比較型以外の A/D 変換器について

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

- 積分型
  積分器に一定の直流電圧を入力すると、出力電圧は積分時間に比例する。その積分器の出力電圧とアナログ入力電圧の比較をして、出力電圧が入力電圧と等しくなるまでの時間をディジタル表示すれば A/D 変換回路が実現できる。
  - 利点
    - 高性能で、比較的安価
    - ノイズに強い
  - 欠点
    - 変換速度が遅い
- 二重積分方式
  増幅器と抵抗、コンデンサの組み合わせによって構成される回路である。入力される直流の時間積分の上昇が、入力された直流の大きさに比例することから、パルス計数でカウントする。入力の直流値をディジタル値に変換する積分型 A/D 変換器の回路に、直流オフセットなどの誤差要因があると、それも積分されて、誤差が累積する問題点を、正方向と負方向から積分して、誤差を打ち消すようにしたもの。
  - 利点
    - 比較的簡単な回路構成で高精度な変換が可能
  - 欠点
    - アナログ的要素をもつ回路部分が多く、ディジタル LSI 中に搭載することが難しく、コストダウンが難しい
- カウンタ制御式
  これは積分型の逆の方法でホールドされた値になるように、クロックで内部カウンターの数字を大きくしていき、その値を DA 変換してコンパレーターで比較する。コンパレーターの値が引っくり返ると、そこで終了する。 かかる時間はクロックの $2^n - 1$ 倍。
  - 利点&欠点
    - 処理時間がクロックの $2^n - 1$ 倍であるということ

- 電荷平衡式
  二重積分方式に似ているが、入力電圧に比例したパルス列を出力する変換器によって構成され、発生パルス数をカウンターで数えることで変換する。二重積分方式の利点に加え、変換時間が入力によらず一定であり、積分器の精度に対する要求は二重積分方式よりも小さく、積分容量の非線形性の影響が少ない点が優れている。変換時間が $10\mu s$ から $10ms$ のものが市販されている。

  - 利点
    - 比較的安価で高精度
    - 積分容量の非線形性の影響が少ない
  - 欠点
    - 変換速度が遅い

- 省ビット式
  逐次比較型と原理的には同じであるが、最も下位のビットのカウントアップとカウントダウンを 0、+1、もしくは −1、0、+1 の信号として出力する方式。出力を受信する側にカウンターを持てばよいので、わずか 1 ビットで $12bit$ 精度の出力が得られる。データ量を $1bit$ に圧縮することができるので、オーディオや影像のように DC 成分の少ない信号や通信のビットレートが限られている場合には威力を発揮する。

  - 利点
    - わずか 1 ビットで $12bit$ の精度の出力が得られる
    - データを $1bit$ に圧縮することができる
  - 欠点
    - 電圧変化量の大きい急峻な変化には追従できない

- フラッシュ(並列) 型
  フラッシュA/D 変換は、ビデオ信号や超音波信号など高速な A/D 変換が必要な信号に用いられる。原理は単純で、比較器を量子化していき、値の個数だけ並べる。出力が 1 の比較器群と 0 の比較器群の境界の位置を、プライオリティーエンコーダなどの論理回路により 2 進数化して出力する。

  - 利点
    - 原理が最も単純で、高速である
  - 欠点
    - bit 数が増えると、回路素子が非常に増えるため値段が高くなる

4 参考文献・URL

- 電子回路〜基礎からシステムまで〜 著：安藤 繁 出版：培風館
- AD 変換
  http://http://www.comb.kokushikan.ac.jp/lecture/envmeasure/node58.html

5 実験に使用した器具

- A/D カード
- ノート PC