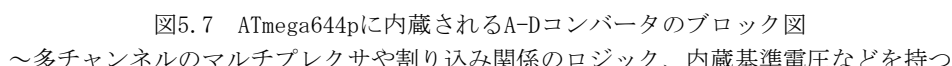


ATmega8 や ATmega644 など多くのAVRには基本的な性能は共通の10ビット逐次比較型のA-Dコンバータが内蔵されています。ATmega644pに内蔵されるA-Dコンバータのブロック図を図5.7に示します。多チャンネルのマルチプレクサや割り込み関係のロジック、内蔵基準電圧などを持ち、多様な要求に応えられる用になっています。



マイクロコントローラ内蔵型A-Dコンバータは性能的には単独のA-Dコンバータチップと比べてもそれほど見劣る物ではありません。しかしながら内蔵型特有の弱点もあります。1つは回路的にデジタル部分と独立していないため、クロックをはじめとして、様々なデジタル回路が発生源となるノイズに弱い点です。もう一つはA-D変換出力のバッファメモリとなる内部RAMがPCなどと比べて決定的に少ないことです。例えば1000点程度サンプルしてからFFTなどの信号処理をして、それから分析処理を行うといった、PCでは当たり前に行うことができることをマイクロコントローラチップ単体で行うのは困難です。

### 5.3 サンプルプログラム

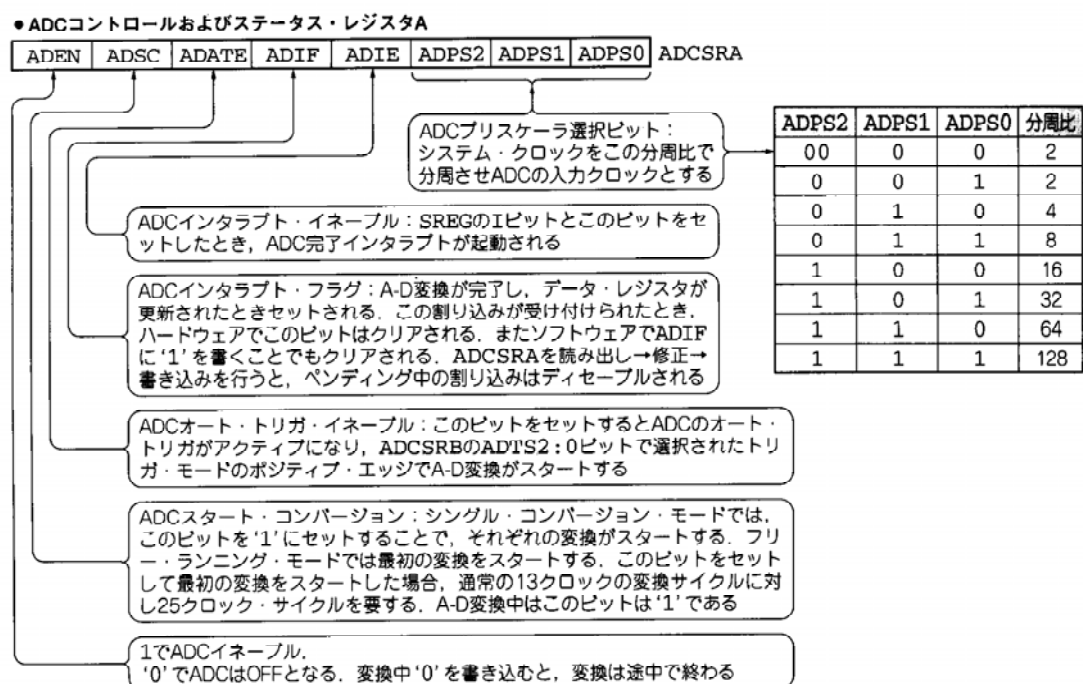


図5.9 ATmega644pのA-Dコンバータの制御レジスタ

#### ○温度計

ATmega644pボードには具体的な計測対象として温度センサICを設置しています。LM35は電圧出力の温度センサICで温度係数はリニアで+10.0mV/℃で、室温25℃の場合  $25 \times 10 = 250\text{mV}$  となります。このままではAVRのA-Dコンバータ入力には電圧値が低いのでオペアンプを使って少し増幅します。A-D変換の基準電圧の1/2程度に持っていくともっとも精度が得られますので、5.7倍の増幅として  $0.25\text{V} \times 5.7 \approx 1.4\text{V}$  を得ます。ATmega644pは1.1Vと2.56Vの2つの内部基準電圧を持ちます。2.56Vの基準電圧を使用すると、25℃での出力電圧1.4Vは基準電圧の1/2程度となります。

温度計プログラムをリスト5.2に示します。リスト5.2では計測のループの中で、温度センサからの電圧を変換しています。さらにオペアンプの増幅率を逆算するとセンサ本体からの電圧が求まります。

リスト5.2 温度計プログラム (m644\_502.c)

```
//
// m644_502ad.c
// 元 WinAVR A-Dテストプログラム prg0502.c
//
#include <avr/io.h>

void wait( unsigned char );
void segout2( int );
unsigned char segdat[10]={0b00111111, 0b00000110, 0b01011011, 0b01001111, 0b01100110,
                          0b01101101, 0b01111101, 0b00100111, 0b01111111, 0b01100111 };

int main( void )
{
    int n,v,i;

    DDRB=0b11111111;          // ポートBをすべて出力に設定
    DDRD=0b11111110;          // ポートDの1~7ビットを出力に設定

    ADCSRA = 0b10000100;      // ADイネーブル
    // |      +-----1/16クロック, 1/16=62.5kHz
    // +-----ADイネーブル
    ADMUX = 0b11000111;        // 基準, 入力選択
    // |||++++-----7ch選択
    // ||+-----右そろえ
    // ++-----内部2.56V基準電圧
    wait(2);                   // マルチプレクサが安定するまで待つ

    while ( 1 ) {              // 無限ループ, 周期約 1 秒
        ADCSRA = 0b11000100;
        // +-----変換開始
        while( ADCSRA & 0b01000000 ) // 変換終了待ち
            ;
        n = ADC ;
        v = (float)n / 1024.0 * 2.56 / 5.7 * 100.0 * 10.0;
        // 温度に変換

        for( i =0 ; i < 200; i++){ // 3桁表示
            segout2( v );
            wait(2);               // 約5ミリ秒
        }
    }

    void segout2( int i )
    {
        unsigned char j,x,p;
        static unsigned char cir=0;

        PORTB = p = 0;
        if( cir == 0){
            PORTD = 0b00100000;    // 1の桁点灯
            j = i % 10;
        }
        if( cir == 1){
            PORTD = 0b01000000;    // 10の桁点灯
            j = (i / 10) % 10;
            p = 0b10000000;        // 小数点点灯
        }
        if( cir == 2){
            PORTD = 0b10000000;    // 100の桁点灯
            j = (i / 100) % 10;
        }
        x = segdat[ j];
        PORTB = x | p;             // ポートBに出力
        cir++;
        if(cir == 3)
            cir = 0;
    }

    void wait( unsigned char time ) // 2500クロック単位のwaitルーチン
    {
        unsigned char j,k;

        for( j= 0; j< time ; j++)
            for( k = 0; k < 250; k++) // 10 * 250 = 2.5 msec
                ;                    // WinAVR 20070525の場合,
                                    // 最適化なしで, 6命令10クロック
    }
}
```

