

G9006 DA コンバーター制御事例

概要

G9006 のこの事例は、ワンチップマイコンを使用して DA コンバーターを制御するものです。

ワンチップマイコンには PIC12F1822 (Micro Chip 製)、DA コンバーターには TLV1549 (TI 製) を使用し、ワンチップマイコンに書き込んだソースを提供します。

動作確認方法

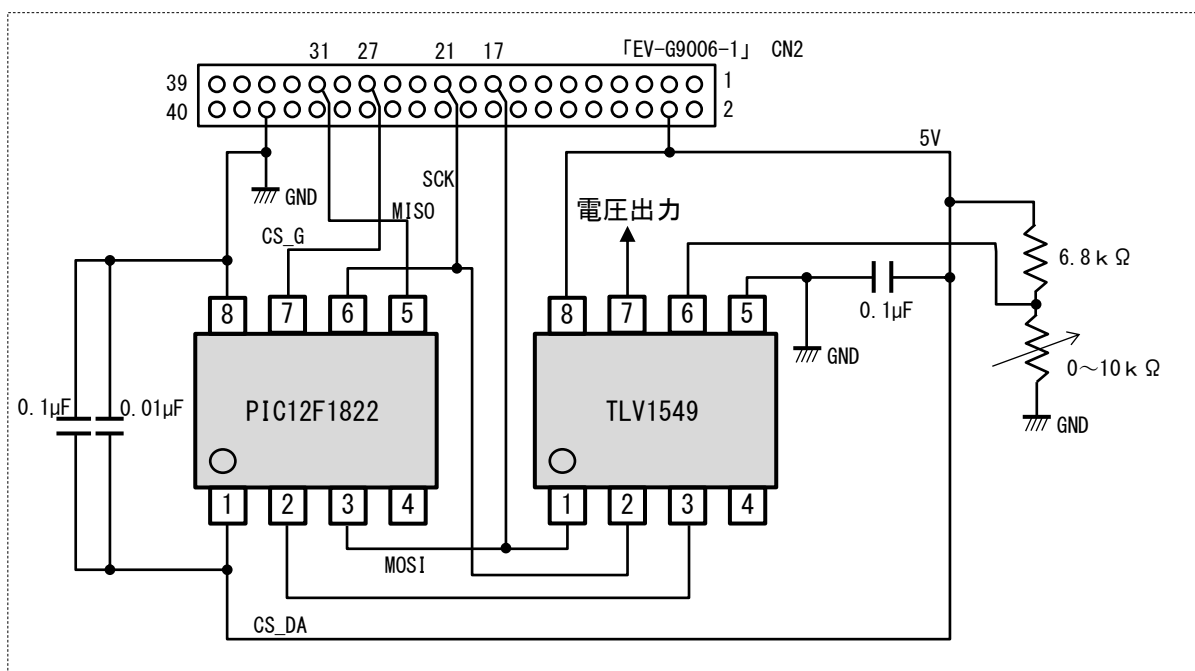
PUSB3601 などと接続し、サイクリック通信を開始してください。

G9006 はモニターモードで動作し、デバイス番号 32 (20h) のポート 0 と 1 の値を読み出して DA コンバーターへ与えます。

G9006 AD コンバーター制御事例の回路を、Motionnet 上で接続すれば、AD コンバーターからのデータを自動的に出力できます。

回路事例

この事例は次の回路を使用して開発およびテストされました。



この回路を、G9006 評価ボード「EV-G9006-1」の CN2 へ接続します。

電源は「EV-G9006-1」の USB 端子から 5V を供給してください。

また「EV-G9006-1」のジャンパー「JP02」の 1、2、3 番ピンのすべてをショートさせてください。これで USB 端子からの 5V が、CN2 の 3、4 番ピンに供給されます。

PIC ソース

```

#include <stdio.h>
#include <stdlib.h>
#include <pic.h>
#include <htc.h> // delay?
// コンフィギュレーション 1 の設定
#pragma config CPD = OFF, BOREN = ON, IESO = OFF, FOSC = INTOSC, FCMEN = OFF
#pragma config MCLRE = OFF, WDTE = OFF, CP = OFF, PWRTE = ON, CLKOUTEN = OFF
// コンフィギュレーション 2 の設定
#pragma config PLLEN = ON, STVREN = ON, WRT = OFF, BORV = LO, LVP = OFF
// delay 用に必要(クロック 32MHz を指定)
#define _XTAL_FREQ 32000000

////////////////////////////////////
#define G96_CS      RA0
#define ADC_CS      RA5

//Reg
#define RADD      0x00
#define RENV1     0x02
#define RENV2     0x04
#define PORT      0x06
#define Tick      0x08
#define RSH0      0x0C
#define RSH1      0x0E
#define RSHR      0x10
#define RBCS      0x12
#define RIDN      0x14
//commanf
#define CDVON     0x10
#define CDVOF     0x11
//Memory adres(32d)
#define DEV32     0x80

/*****
指定した時間(numx10ms)だけウエイトを行う処理関数
*****/
void Wait(unsigned int num)
{
    int i ;

    for (i=0 ; i<num ; i++)
    {
        __delay_ms(10) ;    // 10ms 一時停止
    }
}

/*****
initialize
*****/
void init(void)
{
    OSCCON = 0b01110000 ;    // 8MHz の 4 通倍 = 32MHz
    ANSELA = 0b00000000 ;    // アナログは使用しない (すべてデジタル I/O に割当てる)
    TRISA = 0b00001100 ;    // RA2 ピン以外を全て出力に割当てる (RA3 は入力専用)
    PORTA = 0b00100001 ;    // 出力ピンの初期化 (RA0 (G96_CS) と RA5 (ADC_CS) は High、それ以外は LOW)

    // SPI モードの設定と初期化
    SDOSEL = 1 ;            // 1:RA4 を SDO ピンに設定 / 0:RA0 を SDO ピンに設定
    SSEL = 1 ;             // 1:RA0 を SS ピンに設定 / 0:RA3 を SS ピンに設定
    SSP1CON1 = 0b00110001 ; // SPI (bit5)=1 (Enables serial port and configures SCK, SDO, SDI and SS)
                            // CKP (bit4)=1 (Idle state for clock is a high level)
                            // SPI Master mode
                            // clock = FOSC/16
    SSP1STAT = 0b00000000 ; // CKE (bit6)=0 (SPI Clock Edge Select bit)
    SSP1IF = 0 ;           // Interrupt is not pending
    SSP1IE = 1 ;          // SPI の割込みを許可する
    PEIE = 1 ;            // 周辺装置割込み有効
    GIE = 1 ;             // 全割込み処理を許可する
    SSP1BUF = 0;
}

```

```

/*****
SPI_WriteRead
*****/
unsigned char SPI_WriteRead(unsigned char TransDat)
{
    SSP1BUF = TransDat;
    while(!BF) {};//while (BF==0);

    return SSP1BUF;
}

/*****
G9006 Set reg
*****/
void SetRegG9006(int RegNum,
                unsigned char write_1,
                unsigned char write_2
                )
{
    //
    G96_CS = 0;
    SPI_WriteRead(0x00);//Reg-Write command, Add=00
    SPI_WriteRead(RegNum);//Address
    SPI_WriteRead(write_1);//data
    SPI_WriteRead(write_2);//data
    G96_CS = 1;
}

/*****
G9006 Set com
*****/
void SetComG9006(int Comm)
{
    //
    G96_CS = 0;
    SPI_WriteRead(0x20);//Command-Write, Add=00
    SPI_WriteRead(Comm);//command
    G96_CS = 1;
}

/*****
G9006 Get Mem
*****/
void GetMemG9006(int MemAdd,
                unsigned char *read_L,
                unsigned char *read_H
                )
{
    //
    G96_CS = 0;
    SPI_WriteRead(0x50);//PortArea-Read command, Add=00
    SPI_WriteRead(MemAdd);//Address
    *read_L = SPI_WriteRead(0);//dummy
    *read_H = SPI_WriteRead(0);//dummy
    G96_CS = 1;
}

/*****
* Get ADC
*****/
void SetADC(unsigned char write_H,
            unsigned char write_L
            )
{
    unsigned char dat_1, dat_2;
    //
    ADC_CS = 0;
    SPI_WriteRead(write_H);
    SPI_WriteRead(write_L);
    ADC_CS = 1;
}

```

```

/*****
MAIN
*****/
int main(int argc, char** argv)
{
    union data
    {
        char Cdat[2];
        short Sdat;
    } dat;
    unsigned char dat_L, dat_H;

    init(); // PIC initial

    Wait(22); //Wait G9006 reset end

    //G9006 initial
    SetRegG9006(RADD, 0x80, 0x00); // RADD=0x0080 (DN=0, MONI=1, PMD=0, RSV=0)
    SetComG9006(CDVON); // start

    while(1)
    {
        //Get G9006-port
        GetMemG9006(DEV32, &dat_L, &dat_H);
        dat.Cdat[0] = dat_L;
        dat.Cdat[1] = dat_H;
        //TLV1549 取得値を TLV5618 用に調整
        dat.Sdat = dat.Sdat - 4;
        dat.Sdat = (dat.Sdat << 2) & 0xFFFC;
        dat.Sdat = (dat.Sdat & 0x3FFF) | 0x4000; // Set DAC_B fast
        SetADC(dat.Cdat[1], dat.Cdat[0]);
    }
}

```

以上