

PCに12ビット・アナログ・インタフェースを与えるSO-8パッケージのマイクロパワーADCおよびDAC - デザインノート 138

Kevin R. Hoskins

はじめに

LTC[®]1298 ADCとLTC1446 DACを使用すれば、2チャンネルのアナログ入力/出力を簡単かつ安価に、しかも低電力でコンパクトにPCコンピュータに追加できます。LTC1298とLTC1446は、この種類では最初のSO-8パッケージの2チャンネル・デバイスです。図に示すアプリケーションはPCデータ収集用のものですが、この2つのコンバータは他の多くのアナログ/I/Oアプリケーションにも、最も小さなサイズで最低電力のソリューションを提供します。

小型マイクロパワーADCおよびDAC

LTC1298は12ビットADC、2チャンネル・マルチプレクサ、および入力S/Hを備えています。5V電源、最大変換速度で動作した場合でも、消費電流はわずか340μAです。内蔵の自動シャットダウン機能により、低いサンプリング・レートでの消費電力をさらに低減します(1kspsで30μAまで)。またシリアル・インタフェースにより、リモート・オペレーションが可能です。

LTC1446は最初のSO-8パッケージ12ビット・デュアル・マイクロパワーDACです。1mV/LSB (0V ~ 4.095V)レール・トゥ・レール出力、2.048Vリファレンス内蔵、およびシンプルで安価な3線シリアル・インタフェースを特長としています。DOUTピンにより、シリアル相互接続数を増やすことなく、複数のLTC1446をカスケード接続できます。LTC1446は、5V電源動作時にわずかに1mA (標準)しか消費しません。

PC 2チャンネル・アナログ/I/Oインタフェース

図1に示す回路は、DTR、RTS、CTS、TXの4本のインタフェース・ラインを使用して、PCのシリアル・インタフェースに接続されます。DTRはシリアル・クロック信号の送信、RTSはDACおよびADCへのデータ転送、CTSはLTC1298からの変換結果の受信に使用され、TX上の信号で入力データを受信するデバイスとしてLTC1446またはLTC1298を選択します。LTC1298およびLTC1446は、消費電力が低いためシリアル・ポートから回路の電源を得ることができます。TXラインとRTSラインはダイオードD5とD6を通して、コンデンサC4を充電します。LT[®]1021-5は電圧を5Vに安定化します。DACへのデータ送信後またはADC変換完了後に、TXラインとRTSラインをロジック'H'に復帰させることによって、LT1021-5に一定の電力を供給します。

486-33 PC使用時のスループットは、LTC1298の場合が3.3kspsそしてLTC1446で2.2kspsでした。この値は回路構成によって異なります。

リスト1は、ADCのCH0から変換結果を読み込むか、または両方のDACチャンネルにデータ・ワードを書き込む方法を示したCコードです。

LT LTC、LTはリニアテクノロジー社の登録商標です。
RAIL-TO-RAILはモトローラ(株)の登録商標です。

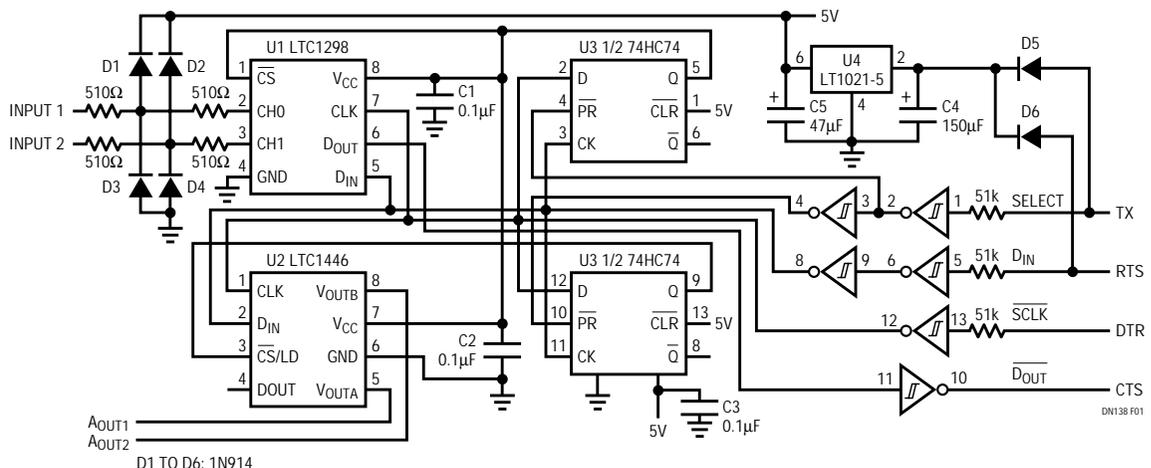


図1. シリアル・ポート通信を用いた、SO-8のLTC1298およびLTC1446によるPC用の簡単な低消費電力2チャンネル・アナログ・インタフェース

まとめ

これらSO-8パッケージのデータ・コンバータは、データ収集システム設計に先例のない省スペース、省電力、および経済性を実現可能です。多種多様な携帯用、バッテリー駆動、および小型サイズが要求される製品に最適な12ビット・アナログ・インタフェースを形成します。きわめて応用しやすく、最小限の受動サポート部品と相互接続しか必要ありません。

リスト1.Cコードによるアナログ・インタフェース設定プログラム

```
#define port 0x3FC /* Control register, RS232 */
#define inprt 0x3FE /* Status reg. RS232 */
#define LCR 0x3FB /* Line Control Register */
#define high 1
#define low 0
#define Clock 0x01 /* pin 4, DTR */
#define Din 0x02 /* pin 7, RTS */
#define Dout 0x10 /* pin 8, CTS input */
#include <stdio.h>
#include <dos.h>
#include <conio.h>
/* Function module sets bit to high or low */
void set_control(int Port,char bitnum,int flag)
{
    char temp;
    temp = inportb(Port);
    if (flag==high)
        temp |= bitnum; /* set output bit to high */
    else
        temp &= ~bitnum; /* set output bit to low */
    outportb(Port,temp);
}
/* This function brings CS high or low (consult the
schematic) */
void CS_Control(direction)
{
    if (direction)
    {
        set_control(port,Clock,low); /* set clock high for Din to
be read */
        set_control(port,Din,low); /* set Din low */
        set_control(port,Din,low); /* set Din high to make CS goes
high */
    }
    else {
        outportb(port, 0x01); /* set Din & clock low */
        Delay(10);
        outportb(port, 0x03); /* Din goes high to make CS
goes low */
    }
}
/* This function outputs a 24 bit(2x12) digital code to
LTC1446L */
void Din_(long code,int clock)
{
    int x;
    for(x = 0; x<clock; ++x)
    {
        code <<= 1; /* align the Din bit */
        if (code & 0x1000000)
        {
            set_control(port,Clock,high); /* set Clock low */
            set_control(port,Din,high); /* set Din bit high */
        }
        else {
            set_control(port,Clock,high); /* set Clock low */
            set_control(port,Din,low); /* set Din low */
        }
    }
}
```

```
        }
        set_control(port,Clock,low); /* set Clock high for
DAC to latch */
    }
}
/* Read bit from ADC to PC */
Dout_0
{
    int temp, x, volt =0;
    for(x = 0; x<13; ++x)
    {
        set_control(port,Clock,high);
        set_control(port,Clock,low);
        temp = inportb(inprt); /* read status reg. */
        volt <<= 1; /* shift left one bit for serial
transmission */
        if(temp & Dout)
            volt += 1; /* add 1 if input bit is high */
    }
    return(volt & 0xffff);
}
/* menu for the mode selection */
char menu()
{
    printf("Please select one of the following:\na: ADC\nb:
DAC\nq: quit\n\n");
    return (getchar());
}
void main()
{
    long code;
    char mode_select;
    int temp,volt=0;
    /* Chip select for DAC & ADC is controlled by RS232 pin
3 TX line. When LCR's bit 6 is set, the DAC is selected
and the reverse is true for the ADC. */
    outportb(LCR,0x0); /* initialize DAC */
    outportb(LCR,0x64); /* initialize ADC */
    while((mode_select = menu()) != 'q')
    {
        switch(mode_select)
        {
            case 'a':
                outportb(LCR,0x0); /* selecting ADC */
                CS_Control(low); /* enabling the ADC CS */
                Din_(0x680000, 0x5); /* channel selection */
                volt = Dout_0;
                outportb(LCR,0x64); /* bring CS high */
                set_control(port,Din,high); /* bring Din signal high
*/
                printf("\ncode: %d\n",volt);
                break;
            case 'd':
                printf("Enter DAC input code (0 - 4095):\n");
                scanf("%d", &temp);
                code = temp;
                code += (long)temp << 12; /* converting 12 bit to 24 bit
word */
                outportb(LCR,0x64); /* selecting DAC */
                CS_Control(low); /* CS enable */
                Din_(code,24); /* loading digital data to DAC */
                outportb(LCR,0x0); /* bring CS high */
                outportb(LCR,0x64); /* disabling ADC */
                set_control(port,Din,high); /* bring Din signal high */
                break;
        }
    }
}
```

リニアテクノロジー株式会社

162-0814 東京都新宿区新小川町 1-14 NAOビル5F
TEL(03)3267-7891 FAX(03)3267-8510
http://www.linear-tech.co.jp

dn138f 1299 6K • PRINTED IN JAPAN


© LINEAR TECHNOLOGY CORPORATION 1996