DESIGN NOTES

Micropower ADC and DAC in SO-8 Give PCs a 12-Bit Analog Interface – Design Note 138

Kevin R. Hoskins

Introduction

Adding two channels of analog input/output to a PC computer is simple, inexpensive, low powered and compact when using the LTC®1298 ADC and LTC1446 DAC. The LTC1298 and the LTC1446 are the first SO-8 packaged 2-channel devices of their kind. While the application shown is for PC data acquisition, the two converters provide the smallest, lowest power solutions for many other analog I/O applications.

Small, Micropower ADC and DAC

The LTC1298 features a 12-bit ADC, 2-channel multiplexer, and an input S/H. Operating on a 5V supply at maximum conversion rate, the LTC1298 draws just 340µA. A built-in auto shutdown further reduces power dissipation at reduced sampling rates (to 30µA at 1ksps). The serial interface allows remote operation.

The LTC1446 is the first 12-bit dual micropower SO-8 packaged DAC. It features a 1mV/LSB (0V to 4.095V) rail-to-rail output, internal 2.048V reference and a simple, cost-effective, 3-wire serial interface. A D_{OUT} pin allows cascading multiple LTC1446s without increasing the number of serial interconnections. Operating on a supply of 5V, the LTC1446 draws just 1mA (typ).
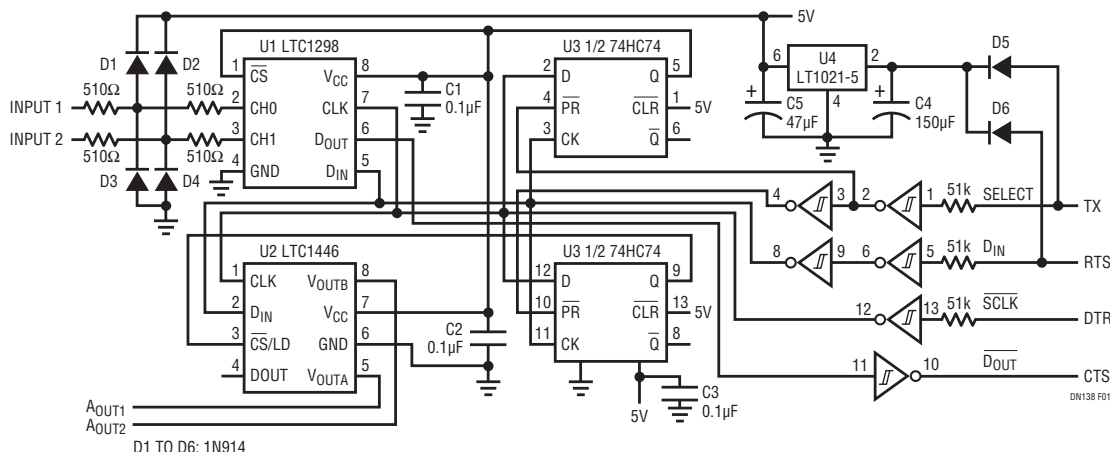
PC 2-Channel Analog I/O Interface

The circuit shown in Figure 1 connects to a PC's serial interface using four interface lines: DTR, RTS, CTS and TX. DTR is used to transmit the serial clock signal, RTS is used to transfer data to the DAC and ADC, CTS is used to receive conversion results from the LTC1298, and the signal on TX selects either the LTC1446 or the LTC1298 to receive input data. The LTC1298's and LTC1446's low power dissipation allow the circuit to be powered from the serial port. The TX and RTS lines charge capacitor C4 through diodes D5 and D6. An LT®1021-5 regulates the voltage to 5V. Returning the TX and RTS lines to a logic high after sending data to the DAC or completion of an ADC conversion provides constant power to the LT1021-5.

Using a 486-33 PC, the throughput was 3.3ksps for the LTC1298 and 2.2ksps for the LTC1446. Your mileage may vary.

Listing 1 is C code that prompts the user to either read a conversion result from the ADC's CH0 or write a data word to both DAC channels.

Figure 1. Communicating Over the Serial Port, the LTC1298 and LTC1446 in SO-8 Create a Simple, Low Power 2-Channel Analog Interface for PCs

Conclusion

These SO-8 packaged data converters offer unprecedented space, power and economy to data acquisition system designers. They form a very nice 12-bit analog interface to a wide variety of portable, battery-powered and size constrained products. They are extremely easy to apply and require a minimum of passive support components and interconnections.

Listing 1. Configure Analog Interface with this C Code

```c
#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_()
{
    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 & 0xfff);
}
/* menu for the mode selection */
char menu()
{
    printf("Please select one of the following:\na: ADC\nd:
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_();
        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;
     }
  }
}
```

For applications help,
call (408) 432-1900