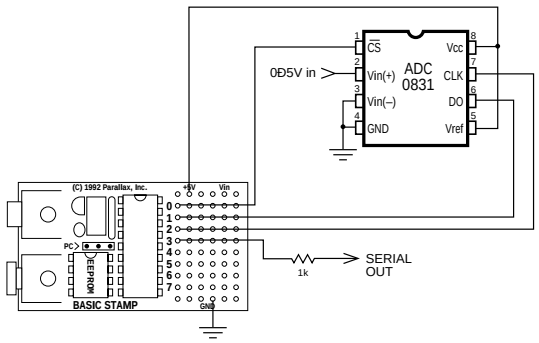
Introduction. This application note presents the hardware and software required to interface an 8-bit serial analog-to-digital converter to the Parallax BASIC Stamp.

Background. The BASIC Stamp's instruction pot performs a limited sort of analog-to-digital conversion. It lets you interface nearly any kind of resistive sensor to the Stamp with a minimum of difficulty. However, many applications call for a true voltage-mode analog-to-digital converter (ADC). One that's particularly suited to interfacing with the Stamp is the National Semiconductor ADC0831, available from Digi-Key, among others.

Interfacing the '831 requires only three input / output lines, and of these, two can be multiplexed with other functions (or additional '831's). Only the chip-select (CS) pin requires a dedicated line. The ADC's range of input voltages is controlled by the VREF and VIN(–) pins. VREF sets the voltage at which the ADC will return a full-scale output of 255, while VIN(–) sets the voltage that will return 0.

In the example application, VIN(–) is at ground and VREF is at +5; however, these values can be as close together as 1 volt without harming the device's accuracy or linearity. You may use diode voltage references or trim pots to set these values.



Schematic to accompany program AD_CONV.BAS.

How it works. The sample program reads the voltage at the ’831’s input pin every 2 seconds and reports it via a 2400-baud serial connection. The subroutine conv handles the details of getting data out of the ADC. It enables the ADC by pulling the CS line low, then pulses the clock (CLK) line to signal the beginning of a conversion. The program then enters a loop in which it pulses CLK, gets the bit on pin AD, adds it to the received byte, and shifts the bits of the received byte to the left. Since BASIC traditionally doesn’t include bit-shift operations, the program multiplies the byte by 2 to perform the shift.

When all bits have been shifted into the byte, the program turns off the ADC by returning CS high. The subroutine returns with the conversion result in the variable data. The whole process takes about 20 milliseconds.

Modifications. You can add more ’831’s to the circuit as follows: Connect each additional ADC to the same clock and data lines, but assign it a separate CS pin. Modify the conv subroutine to take the appropriate CS pin low when it needs to acquire data from a particular ADC. That’s it.

Program listing. This program may be downloaded from our Internet ftp site at ftp.parallaxinc.com. The ftp site may be reached directly or through our web site at http://www.parallaxinc.com.

```
' PROGRAM: ad_conv.bas
' BASIC Stamp program that uses the National ADC0831 to acquire analog data and
' output it via RS-232.

Symbol   CS      =   0
Symbol   AD      =   pin1
Symbol   CLK     =   2
Symbol   S_out   =   3
Symbol   data    =   b0
Symbol   i       =   b2

setup:   let pins = 255                    ' Pins high (deselect ADC).
         let dirs = %11111101              ' S_out, CLK, CS outputs; AD
                                           ' input.

loop:    gosub conv                        ' Get the data.
         serout S_out,N2400,(#b0,13,10)    ' Send data followed by a return
```

```
                                        ' and linefeed.
        pause 2000                      ' Wait 2 seconds
        goto loop                       ' Do it forever.

conv:   low CLK                         ' Put clock line in starting state.
        low CS                          ' Select ADC.
        pulsout CLK, 1                  ' 10 us clock pulse.
        let data = 0                    ' Clear data.
        for i = 1 to 8                  ' Eight data bits.
        let data = data * 2             ' Perform shift left.
        pulsout CLK, 1                  ' 10 us clock pulse.
        let data = data + AD            ' Put bit in LSB of data.
        next                            ' Do it again.
        high CS                         ' Deselect ADC when done.
        return
```

BASIC Stamp I Application Notes