Interfacing to the Parallel Port

While there is a choice of ports on the PC for amateur use and direct interfacing, the parallel port is probably the simplest. With eight data wires, several control wires and bidirectional capability, it offers a convenient way to get information in and out of a PC. The examples in the next two sections use an older software language, *BASIC* or *GWBASIC* to get information in and out. Newer languages can be used, but several varieties of *BASIC* are available on the Internet at no cost. The learning curve for someone who has never used a programming language is very short — usually a matter of a few minutes. The two examples that follow interface single chip analog-to-digital and digital-to-analog converters to the parallel port of a PC.

Single-Chip Dual-Channel A/D

In this analog world, often there is need to measure an analog voltage and convert it to a digital value for further processing in a PC. This single chip converter and accompanying software performs this task for two analog voltages.

Circuit Description

The circuit consists of a single-chip A/D converter, U2, and a DB-25 male plug (**Fig A**). Pins 2 and 3 are identical voltage inputs, with a range from 0 to slightly less than the supply voltage V_{CC} (+5 V). R1, R2, C3 and C4 provide some input isolation and RF bypass. There are four signal leads on U2. DO is the converted data from the A/D out to the computer; DI and CS are control signals from the computer, and CLK is a computer-generated clock signal sent to pin 7 of U2.

The +5-V supply is required. It may be obtained from a +12-V source and regulator U1. Current drain is usually less than 20 mA, so any 5-V regulator may be used for U1. The power supply ground, the circuit ground and the computer ground are all tied together. If you already have a source of regulated 5 V, U1 is not needed.

In this form the circuit will give you two identical dc voltmeters. To extend their range, connect voltage dividers to the input points A and B. A typical 2:1 divider, using 50-kΩ resistors, is shown in the figure. Resistor accuracy is not important, since the circuit is calibrated in the accompanying software.

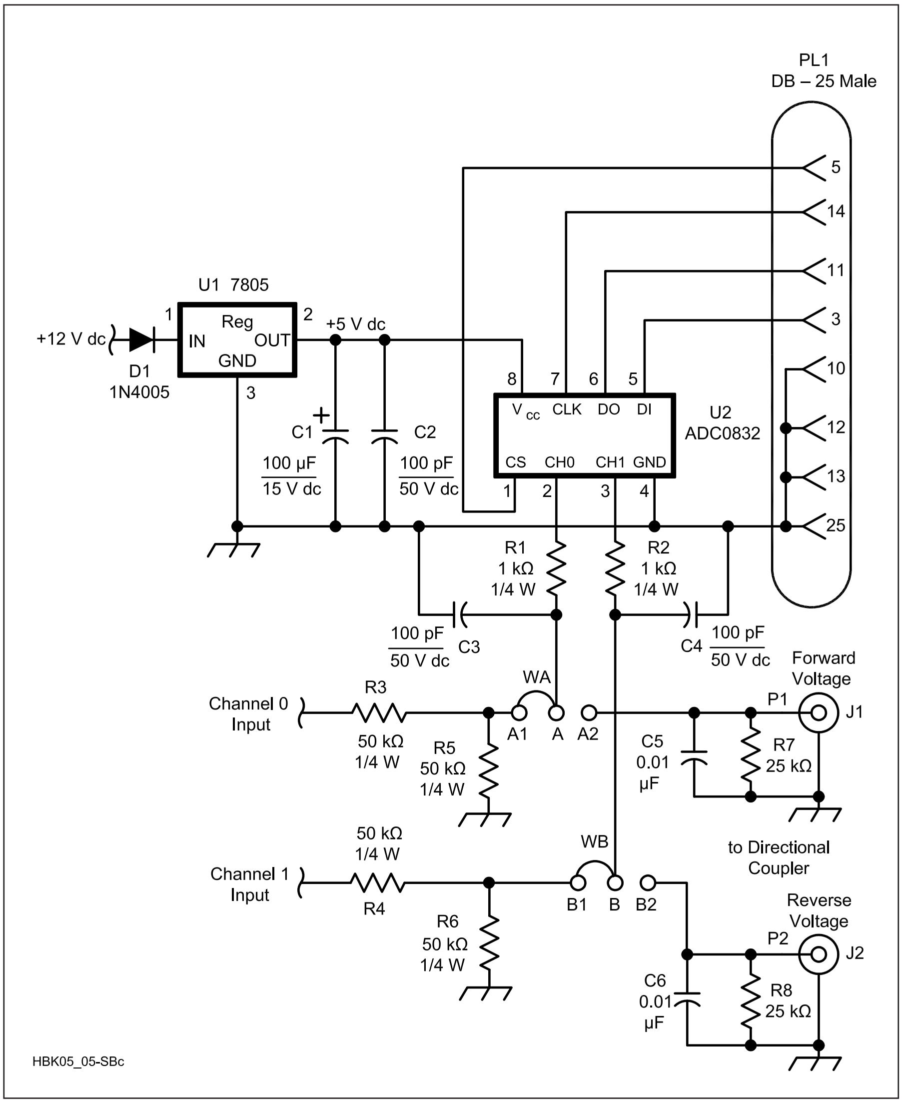

Fig A — Only two chips are used to provide a dual-channel voltmeter. PL1 is connected through a standard 25-pin cable to your computer printer port. U2 requires an 8-pin IC socket. All resistors are ¼ W. You can use the A/D as an SWR display by connecting it to a sensor such as the one shown in Chapter 19 of this *Handbook* (Tandem Match Wattmeter project). A few more resistors are all that are needed to change the voltmeter scale. The 50-kΩ resistors from 2:1 voltage dividers, extending the voltmeter scale on both channels to almost 10 V dc.

Software

The software, *A2D.BAS*, can be found on the *Handbook* CD. It includes a voltmeter function and an SWR function. It is written in *GWBASIC* and saved as an ASCII file. Therefore, you can read it on any word processor, but if you modify it, make sure you resave it as an ASCII file. It can be imported into *QBasic* and most other *BASIC* dialects.

The program was written to be understandable rather than to be highly efficient. Each line of basic code has a comment or explanation. It can be modified for most PCs. The printer port used is LPT1, which is at a hex address of 378h. If you wish to use LPT2 (printer port 2), try changing the address to 278h. To find the addresses of your printer ports, run *FINDLPT.BAS* (also included on the *Handbook* CD).

A2D.BAS was written to run on computers as slow as 4.7-MHz PC/XTs. If you get erratic results with a much faster computer, set line 1020(CD=1) to a higher value to increase the width of the computer-generated clock pulses.

The software is set up to act as an SWR meter. Connecting points A and B to the forward and reverse voltage points on any conventional SWR bridge will result in the program calculating the value of SWR.

Initially the software reads the value of voltage at point A into the computer, followed by the voltage at point B. It then prints these two values on the screen, and computes their sum and difference to derive the SWR. If you use the project as a voltmeter, simply ignore the SWR reading on the screen or suppress it by deleting lines 2150, 2160 and 2170. If the two voltages are very close to each other (within 1 mV), the program declares a bad reading for SWR.

Calibration

Lines 120 and 130 in the program independently set the calibration for the two voltage inputs. To calibrate a channel, apply a known voltage to input point A. Read the value on the PC screen. Now multiply the constant in line 120 by the correct value and divide the result by the value you previously saw on the screen. Enter this constant on line 120. Repeat the procedure for input point B and line 130.

D to A CONVERTERS — CONTROLLING ANALOG DEVICES

The complement to A/D converters is D/A (digital-to-analog) converters. Once there is a digital value in your PC, a D/A will provide an analog voltage proportional to the digital value. Normally the actual value is scaled. As an example, an 8-bit converter allows a maximum count of 255. If the converter is set up with a +5 V dc reference voltage, a maximum value digital value of 255 would result in a D/A output value of 5 V. Lower digital inputs would give proportionally lower voltages.

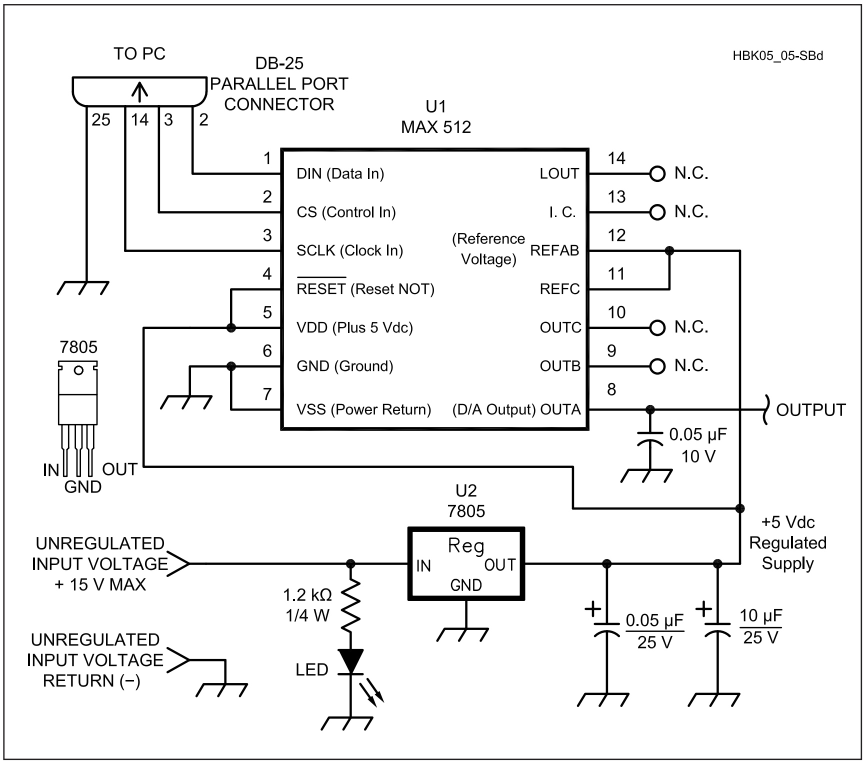

Fig B — Only three wires and a ground lead are needed to connect the converter to your PC.

Circuit Description

This project is the complement of the parallel port A/D converter described earlier. It takes a digital number from the computer, and converts it to a voltage from 0 to 5 V dc. Only one chip, the MAX 512, is required. It operates from a 5-V supply and is connected to the computer by a standard DB-25 parallel port connector. The chip may be ordered from Digi-Key, Allied Electronics and other ham suppliers as MAX512CPD-ND. The voltage regulator in **Fig B** provides the 5 V source required to power the chip.

Software

The software needed to run the chip, *D2A.BAS*, can be found on the *Handbook* CD. It is about 60 lines long, fully commented and written in *GWBASIC*, so it may be readily modified. The parallel port address is defined on line 105 as PORTO=&H378. Your computer may use a different address. To find the correct address, run *FINDLPT.BAS*.

The program takes the value AIN from the keyboard (line 230), converts it to a number between 0 and 255, and then sends it out as a serial word to the DIA chip. If you would like to use the project with another program, use your other program to set AIN to the value you want to generate, and then run this program as a subroutine.

At the end of the program is the clock pulse subroutine. In the event your computer is too fast for the converter chip, you can stretch the clock pulses by changing CD in line 5010 to a value greater than the default value of 1.

Applications

This circuit provides the capability of setting a voltage under computer control. It can be calibrated to match the power supply and the actual chip used. Tests with several chips showed an error of 25 mV or less over the range of 0 to 5 V dc output. — *Paul Danzer, N1II*