CENTRONICS A-D/D-A CONVERTER

J. Ruffell

To the many PC users who would like to interface their computers with the real world we present an analogue-to-digital and digital-to-analogue converter. The low-cost, versatile, unit with accompanying control software is unconventional in that it is connected to the PC's Centronics port, which is normally used for a parallel printer.

The use of standard interfaces for applications they are not intended for is widespread and goes back to the days of the first hobby computers. The advantages are obvious: there is no need to get to grips with the computer hardware, and the function of the peripheral is not dependent on extension connectors peculiar to the system. Thus, the software required to control the 'custom-made' peripheral is often hardware-dependent and obtained by rewriting the system-resident I/O routines, or accessing the relevant circuitry in a non-standard way, e.g., through bypassing the BIOS (basic input/output system).

Many modern PCs contain large gate arrays instead of individual I/O circuits. The A-D/D-A converter described may be used without the need of rewriting the control routines available for parallel I/O operations. Note, however, that this does not imply complete independence of the hardware, since the BIOS routines normally used for controlling the Centronics printer port are not suitable for the controlling the present converter board. Fortunately, the degree of hardware dependency is low and restricted to a few addresses in I/O routines. The control program available for the converter board should not, therefore, give problems on most MS-DOS computers. Note, however, that a number of older PCs have a printer port with incomplete handshaking. The absence of certain lines generally does not cause problems when a standard printer is used. The converter board, however, may require these lines for a number of functions.

Centronics port inputs

The block diagram of the A-D/D-A converter is given in Fig. 1. The operation of the circuit is based on the use of the output as well the input lines of the Centronics port. The latter are normally used to convey 'paper empty', 'busy' and other information from the printer to the computer. The converter, however, uses these inputs to convey digital data, such as the state of two comparators, to the computer. The two comparators enable two analogue input voltages to be compared with an analogue output voltage supplied by the DAC (digital-to-analogue converter). By writing a series of rising values to the DAC and monitoring the relevant comparator output, the computer is able to determine the value of the analogue input voltage applied to the board. Such an operation is generally referred to as successive approximation. In the present case, its advantage lies in the use of a single computer input only instead of a number equal to the conversion resolution in bits (in this case, eight). This is an important consideration since there are few inputs on a Centronics interface. A LED monitor circuit enables, by selection, either the state of the Centronics data lines or those of the digital outputs of the circuit to be indicated.

Circuit description

The actual circuit (see Fig. 2) is just as straightforward as the block diagram, although some details may create an impression of greater complexity. The eight databits on the Centronics port are fed direct to the inputs of the DAC, IC_{12}. Provided the PC port meets the drive specifications set out in Centronics standard, these lines are driven by open-collector (OC) outputs,

MAIN SPECIFICATIONS

- **8-bit D-A converter**
  - output voltage: −5 V to +5 V
  - total settling time: approx. 1 µs
  - three reference options:
    - REF-02 (+5 V; very stable)
    - TL317 (+5 V; low-cost)
    - external source
- **2-channel A-D converter**
  - DAC-based successive approximation
  - attenuators for adjustable input sensitivity
- **4 multi-purpose OC outputs**
  - $I_{C(max)}$=100 mA; $U_{CE(max)}$=30 V
- **3, 4 or 5 digital inputs**
  - switching threshold: approx. 2.5 V
  - CMOS and TTL compatible
- **LED indication for functional checks**
  - monitors either Centronics datalines or digital input/output lines
- **Supply voltage: ±12 V**

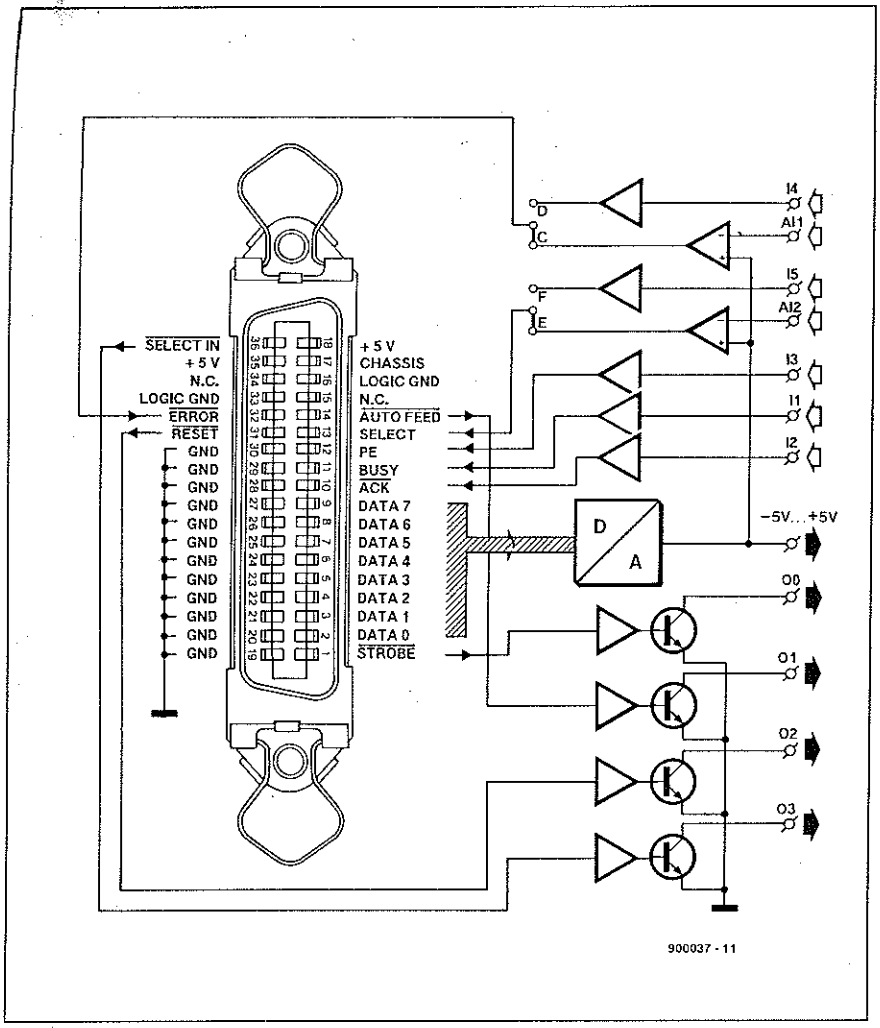

Fig. 1. Block diagram of the Centronics-compatible A-D/D-A card.

whence the use of pull-up network R_3. The output current supplied by the DAC is determined by the value of the dataword supplied by the computer. This current is converted into a voltage with a range of −5 V to +5 V with the aid of IC_9 and IC_{10}. The relation between the applied data and the output voltage, U_o, is expressed by

$$U_o = U_{ref}\,(\text{data}-128)/128 \quad [\text{V}]$$

The reference voltage, U_{ref}, is supplied either by an external source (via jumper B), or by an internal source. The internal source is either a REF-02 or a TL317; the REF-02 provides the highest stability and accuracy, but is more expensive than the TL317. The choice between these two devices is up to you.

Opamps IC_{11a} and IC_{11b} form the previously mentioned comparators. Their inputs are protected against overvoltage by two diodes (D_3-D_4/D_5-D_6) and a resistor (R_{45}/R_{46}). The input sensitivity may be adapted by modifying attenuators R_{47}-R_{48} and R_{49}-R_{50}. The indicated resistor values provide an attenuation of 2 times, which creates an input voltage range of −10 V to +10 V. In case one or both analogue inputs are not used, the associated input on the Centronics interface may be set to function as a digital input. This is achieved by means of a jumper (C–D; E–F).

Each digital input consists of a darlington transistor (T_1–T_5), a collector resistor and two base resistors. The inputs switch at about +2.5 V, which makes them suitable for driving both TTL and CMOS logic. The maximum input voltage is about +30 V.

The digital outputs, T_6–T_9, are of the open-collector type to allow direct connection to small loads, such as LEDS or relays. Note, however, that any relay coil must shunted by a diode to prevent the transistor being damaged by inductive voltage surges. The maximum voltage that can be switched by the output transistors is 30 V.

Two three-state buffers, IC_1 and IC_2, and a LED array, allow the operations of the ADC and the DAC to be checked visually. Depending on the connection made by jumper G–H, the LEDs indicate either the data applied to the DAC, or the state of the digital inputs on the Centronics port. Since the monitor circuit falls short of one line to provide an indication of SEL (select), this is taken over by two inverters, IC_{3a}-IC_{3b}, and LED D_{15}. The function of each LED is shown in Table 1.

LED	jumper G	jumper H
D7	00	D0
D8	01	D1
D9	02	D2
D10	03	D3
D11	I1	D4
D12	I2	D5
D13	I3	D6
D14	I4/AI1	D7
D15	I5/AI2	I5/A12

Table 1. LED function overview.

Construction

The converter is best constructed on the double-sided, through-plated printed-circuit board of which the component mounting plan is shown in Fig. 3 (the track lay-outs are not given because this through-plated board is virtually impossible to make without special equipment).

The construction itself is precision work, but none the less entirely straightforward with reference to the component overlay and the parts list. As already stated, the choice between the REF-02 and the TL317 is up to you: simply fit and omit the relevant components as indicated in the parts list. Connector K_1 is a standard 36-way Centronics socket with angled pins for PCB mounting. This type of connector is often used on matrix printers.

Control software and setting-up

A 360-KByte MS-DOS formatted 5¼-inch diskette is available for this project. The programs on the disk are helpful for adjusting and testing the converter board. A Turbo-Pascal source file is provided that contains the basic routines for the I/O operations with the A-D and the D-A converter. This program uses a set of default I/O register addresses, which may have to be changed depending on the computer used. It should be noted that the logic levels in the status-, data- and control-registers must correspond to those of the inputs and outputs. This point is made because the levels of the active-low lines are inverted either by hardware or software, again, depending on the type of computer. Obviously, the LED indicator array comes in handy here.

Adjustment

The first routine given in the form of a flow-chart is the adjustment procedure (see Fig. 4). All indicated voltages are measured with respect to the analogue ground potential. The relevant connection for the DVM may be found between the two analogue inputs and the analogue output.

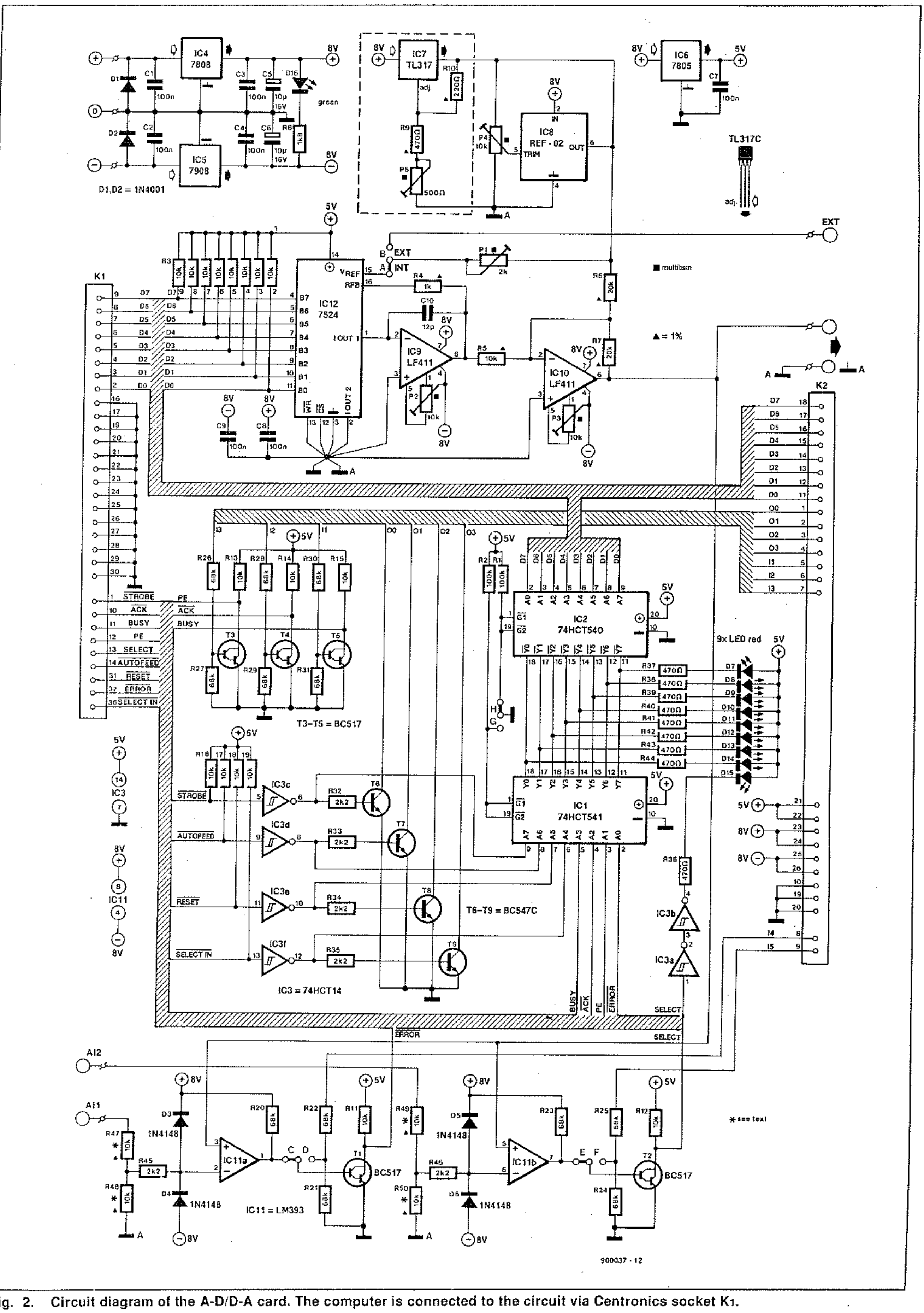

Fig. 2. Circuit diagram of the A-D/D-A card. The computer is connected to the circuit via Centronics socket K1.

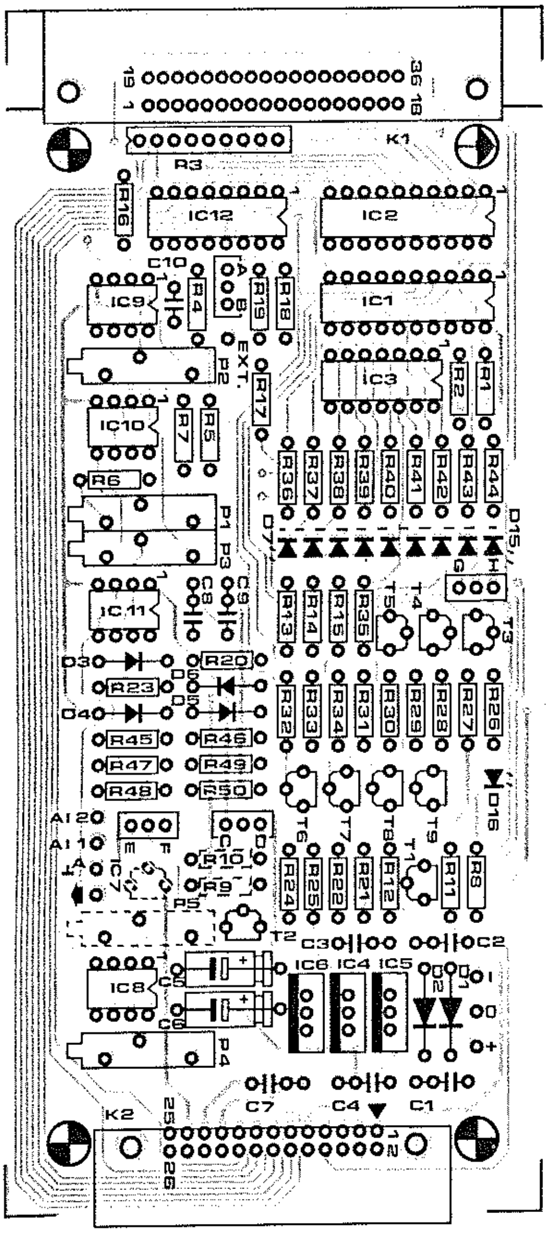

Fig. 3. Component mounting plan of the double-sided, through-plated PCB.

COMPONENTS LIST

Resistors:

2	100k	R1;R2
1	10k 8-way SIL	R3
1	1k0 1%	R4
5	10k 1%	R5;R47;R48;R49;R50
2	20k 1%	R6;R7
1	1k8	R8
1	470Ω 1%	R9 **
1	220Ω 1%	R10 **
9	10k	R11 – R19
12	68k	R20 – R31
6	2k2	R32 – R35;R45;R46
9	470Ω	R36 – R44
1	2k multiturn preset	P1
3	10k multiturn preset	P2;P3
1	10k multiturn preset	P4 ***
1	500Ω multiturn preset	P5 **

Capacitors:

6	100n	C1 – C4;C7;C8;C9
2	10μF 16 V	C5;C6
1	12pF	C10

Semiconductors:

2	1N4001	D1;D2
4	1N4148	D3 – D6
9	LED 3mm red	D7 – D15
1	LED 3mm green	D16
5	BC517	T1 – T5
4	BC547C	T6 – T9
1	74HCT541	IC1
1	74HCT540	IC2
1	74HCT14	IC3
1	7808	IC4
1	7908	IC5
1	7805	IC6
1	TL317	IC7 **
1	REF-02	IC8 ***
2	LF411	IC9;IC10
1	LM393	IC11
1	PM7524 (PMI) or AD7523 (Analog Devices)	IC12

Miscellaneous:

1	36-way Centronics socket for PCB mounting; angled pins	K1
1	26-way header; angled pins; with eject handles	K2
1	printed-circuit board	900037
1	control program on disk	1421

** not fitted when IC8 is used
*** not fitted when IC7 is used

First, adjust either P5 (reference: TL317) or P4 (reference: REF-02) until a reference of 5.00 V is obtained. Next, cancel the off-set voltage of IC9 by adjusting P2. Set the current that flows into the V_{ref} input by adjusting P1. Finally, adjust P3 to cancel the off-set voltage of IC10.

Writing data to the DAC is simple: in nearly all cases, this involves loading one register with the desired value. One statement,

```
PORT [DATAREG] := DATA;
```

is sufficient in Turbo Pascal. Since U_{ref} = 5 V, the relation between the value of the dataword and the resultant analogue output voltage is expressed by

$$U_o = 5\,(\text{data}-128)/128 \quad [\text{V}]$$

Data '0' therefore produces –5 V; data '80$_H$' 0 V; and data 'FF$_H$' +4.961 V. Since the circuit is capable of producing 0 V, the highest output voltage remains 39 mV below +5 V. A slightly different setting-up procedure allows you to reach +5 V, but this in turn makes it impossible to achieve 0 V, which can not be approximated at a difference better than 39 mV or ½LSB at data = 80$_H$. In practice, it is easier to state a voltage and calculate the corresponding DAC data from

$$\text{data} = 128\,(U_o/5+1)(-5 \le U_o \le 4.961)$$

where 'data' is rounded off to give a whole number. Both the checking of U_o and the required conversion computation to provide the necessary bit combination for the

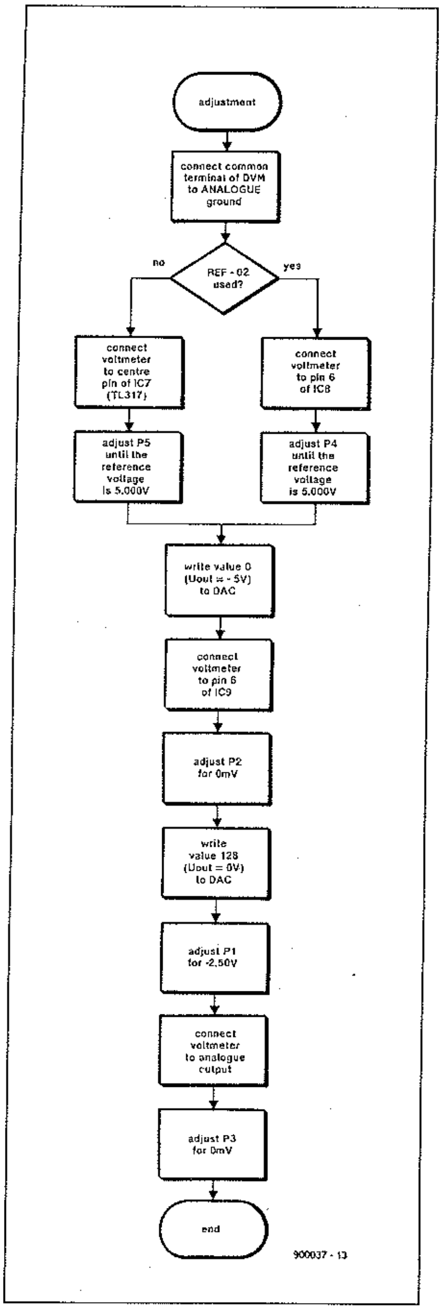

Fig. 4. Flow diagram of the adjustment procedure.

```
external variables:
Ui, input,
K (= input attenuation)

begin procedure A-D conversion
intermediate value := 0
bit position := 7 down to 0
    DAC := intermediate value OR (SHL 1 bit position)
    wait (DAC and comp. settling time)
    comparator number (input) <> 0 (Ui < UDAC)
    no \/ yes
        intermediate value := intermediate value OR (SHL 1
        bit position)
Ui := Uref * K * (intermediate value - 128)/128
end of procedure
```

Fig. 5. Flow diagram of the successive approximation routine

```
 Volume in drive A is ESS_1421
 Directory of  A:\

CENTR&ME EXE    41328   2-12-90   3:06p
CIOPUBLI PAS     4241   3-16-90  10:32a
CIOPUBLI TPU     2816   2-20-90   9:57a
EXAMPLE  EXE     8656   2-20-90   9:59a
CONFIG   DAT      386   3-16-90  10:35a
EXAMPLE  PAS     1697   3-16-90  10:39a
        6 File(s)    300032 bytes free
```

Fig. 6. Directory of the PC-MSDOS program disk, number 1421, for the project.

DAC may be included in the routine that controls the voltage setting.

A-D conversion

As already noted, this is effected on the basis of successive approximation. The unknown analogue voltage is approximated by comparing it to analogue voltages generated by the DAC, whose resolution causes the longest (worst-case) approximation to require the maximum number of steps, 2^8 or 256. The use of a different approach may reduce this number to 8, or the width (in bits) of the data input of the DAC. Figure 5 shows the flow-chart of the D-A routine. This procedure is invoked with two variables: the input channel and the attenuation (for which variable K must be greater than 1).

Variable 'data' is used for intermediate data storage, starting with value 0. Next, a for-next loop is entered. This is passed eight times, during which the A-D conversion is effected. A program within the loop checks for each bit whether this must become 0 or 1. The most significant bit is treated first with an intermediate value of 0 (all bits are 0). The sum of these two is obtained with the aid of an OR function and is subsequently written to the DAC. When the relevant comparator indicates that the voltage is too high, the corresponding bit in the intermediate value must remain at 0. When the voltage is too low, the same bit must become 1. All eight bits are treated in this manner by shifting the 1 to the left (SHL). After the eight steps, the input voltage may be calculated on the basis of the intermediate value.

Short wait times are inserted between the write operation to the DAC and the read operation to the comparator. This is done to allow for the response time of the ICs. This wait time is so short, however, as to make the use of standard time functions in the PC impossible, since their minimum delay of about 1 ms is much too long. Hence, a for-next loop is used. An obvious problem caused by this approach, dependency on the clock speed of the computer, may have to be resolved empirically with different loop repetitions. In many cases, a single repetition is sufficient to establish the required wait time. In that case, the loop may be replaced by one or more useful statements of your own. A statement like 'repeat until true' does effectively nothing but last a number of clock cycles and does not require a previously declared auxiliary variable.

Test program

The diskette supplied for this project contains the basic routines in a Turbo-Pascal unit, both in the form of compiled code and source text. Also on the disk is an auxiliary program for testing and adjusting the card. This program, CENTR&ME.EXE, searches for a file called CONFIG.DAT, which contains five numbers that indicate the printer port number, the attenuation on input channel 1, the attenuation on input channel 2, the reference voltage, and the number of iterations in the wait loop, in that order. This configuration file may be edited to individual requirement with the aid of any ASCII compatible word processor, like EDLIN or the one in SideKick or PCTools. The numbers are separated either by a comma or a space. The file also contains a few lines to explain the meaning of the numbers. These lines are comment and have nothing to do with the actual operation of the test program, which, incidentally, may be run without the A-D/D-A card connected to the computer. This is particularly useful to become acquainted with its structure and commands. ■