

Is the Interface OK?

T SURESH

National Institute of Oceanography, Goa 403 004, India.

When a peripheral device fails, software methods can be initially resorted to before the usual hardware test procedures are used. A test program is presented here that allows various peripherals, inter-faced to a Norsk Data computer, to be tested. The program illustrates the use of some essential monitor call routines, and can be used to develop driver routines for Norsk Data peripherals.

SYSTEM engineers often have to interface new devices to system. If it is a standard device listed by the system manufacturer, installation and test procedures usually exist, supported with necessary hardware and software. User required device not listed but with a compatible standard interface such as S-232C, RS-422 or IEEE 488 can also be interfaced with the system, provided user has considerable knowledge of the system and device. While interfacing such devices, primary aim should be to verify interface link by test data transfer between computer and device. Once this is established other software such as driver routines, application software could be developed by an experienced programmer.

Here a small program is presented, that is primarily used to test devices interfaced to computer. This software is developed on ND-100 CPU of Norsk Data, ND-520 supermini-computer in ND-FORTRAN with the help of various monitor call routines available with the system.

This software has been extensively used while interfacing new devices and also in identifying and isolating faults in serial interfaced (RS-232C/Current loop) devices such as terminal, plotter, digitizer, printer, PC, etc.

PROGRAM DESCRIPTION

As seen from Fig 1, at the beginning the port to which the device interfaced is defined.

If the program is suddenly aborted during its execution, it could leave the port in a reserved state by the same program. Thus as a safety measure, input and output parts of port are released for current use at the beginning.

Next it verifies whether the port is reserved for any other application. If so then system command should be used to set it free.

For testing input part of interface input buffer is emptied at the beginning. After this a watch is kept on the buffer to check for the arrival of any data. Data transmitted by the device is accepted by computer and displayed on the user terminal till the user defined break character arrived from the device.

For testing output part of interface, similarly output buffer is cleared at the beginning. A test pattern of ASCII character is generated and transmitted from the computer to device.

At the end of test, input and output parts are released. Complete listing of this program is given in Table 1.

APPLICATION

Most devices such as terminal, printer, etc, are connected to the computer via serial inter-

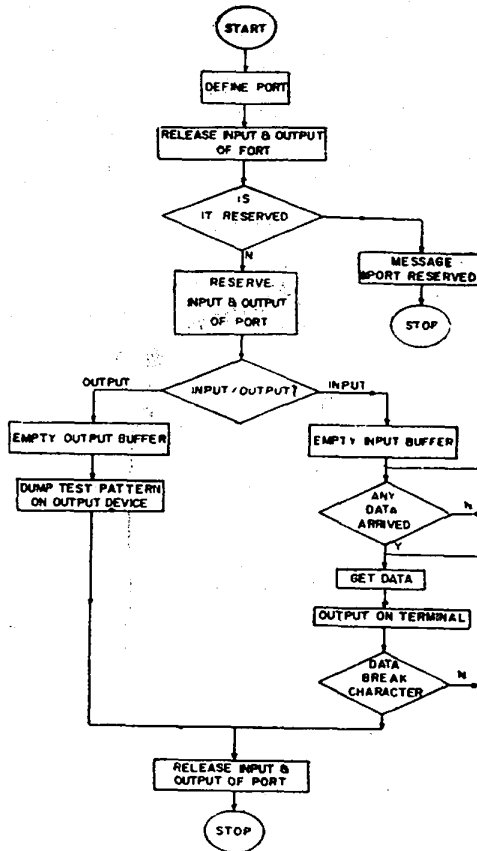


Fig. 1 Flow chart for testing input/output of an interfaced Device

TABLE 1 Program listing

PROGRAM TEST	
C	This program is used to test interface between ND Computer and a device interfaced.
C	Device is interfaced to a user defined port.
C	CHARACTER REPLY, BREAK INTEGER PORT NO, DATA
C	Define port WRITE (1,10) READ *, PORTNO
C	Release port it self reserved. CALL RELES (PORTNO, 0) CALL RELES (PORTNO, 1) IF (ISTAT. NE. 0) STOP 'Port reserved'
C	Reserve input part CALL RESRV (PORTNO, 0, 0)
C	Reserve output part CALL RESRV (PORTNO, 1, 0)
C	Select input or output for test WRITE (1, 20) READ *, REPLY
C	IF (REPLY. EQ. 'O') GO TO 99
C	Test input
C	Define break character WRITE (1, 30) READ *, BREAK
C	Clear input buffer CALL CIBUF (PORTNO)
C	Wait for data from device NODATA = ISIZE (PORTNO) DO WHILE (NODATA.NE.0) NODATA = ISIZE (PORTNO) WRITE (1, 40) NODATA ENDDO
C	Accept data—7 bit DATA = INCH (PORTNO) DATA = IAND (DATA, 177B) DO WHILE (CHAR (DATA).NE. BREAK) WRITE (1,50) CHAR (DATA) DATA = INCH (PORTNO) DATA = IAND (DATA, 177B) END DO GO TO 199
C	Test output
C	Clear output buffer 99 CALL COBUF (PORTNO) DATA = 32 DO FOR I = 1,90 CALL OUTCH (PORTNO. DATA) DATA = DATA + 1 END DO
C	End of test
C	Port released CALL RELES (PORTNO, 0) 199 CALL RELES (PORTNO, 1) 10 FORMAT ('\$, T5, 'Port No.')
	20 FORMAT ('\$, T5, 'Test Input or Output (I/O :')
	30 FORMAT ('\$, T5, 'Bread character:')
	40 FORMAT ('\$, T5, 'Amount of data in buffer : ', 14)
	50 FORMAT ('\$, T5, 'Data arrived is:', A)
	STOP END

faces, RS-232C or current loop. Users quite often face problems such as device not responding to 'commands or 'hanging' after a command is given, during on-line operation on computer. At such times if the system commands cannot solve the problem, it would not be possible to shut down the whole system to diagnose the fault. Nevertheless, with the system in operation, a series of hardware test using tools such as oscilloscopes and 'break-out boxes' could identify the fault, but it could be a time consuming and tedious task.

Thus as a first step, it would be advisable to run on-line tests program. On failure of such a test second step would be to check the interface lines for loose connection. If the fault still persists, the same software could be used to generate required signals to isolate any fault and rectify the same till the desired positive test results are obtained.

Illustrated here are a few but essential monitor call routines that can be used in a FORTRAN program with ease for I/O handling on Norsk Data Computer. The same monitor routines can be used to develop driver routines and interface a device with standard interfaces such as RS-232C, RS-422 or Current Loop.

CONCLUSION

This simple program described has been used extensively for testing serial interfaced devices. This software can be used to develop driver routines for other interface compatible devices to a Norsk Data computer with little modifications. Use of various monitor calls available for developing interface routines have been clearly illustrated here.

REFERENCES

1. Norsk Data, *SINTRAN III Real Time Guide*.
2. Norsk Data, *SINTRAN III Reference Manual*.
3. Norsk Data, *ND FORTRAN Reference Manual*.

APPENDIX A

Monitor Calls

Here it is assumed that data is transferred in ASCII format and thus data acquisition and transmission is on byte basis. A port indicates a piece of hardware interface of the computer, to which various devices are connected that can be accessed by I/O instructions *eg* terminal, communication channel, printer, plotter, etc.

Each internal or external port in a Norsk Data computer system has a logical device number. This number is fixed and for background programs running on a terminal, this is the terminal number. Two way devices are identified by one logical device number and while reserving input or output part must be specified. If both are to be reserved, two requests must be executed.

A brief description of various monitor calls used here are given below [1]. Parameters associated with these calls are defined as follows

- (a) LDN : Logical device number
- (b) ISTAT : Return value
- (c) IOF : Input/output flag
 - 0 = Input
 - 1 = Output
- (d) IRET : 0 = Set the port in waiting state if it is reserved
 - 1 = Return if port is reserved with ISTAT <
 - 0, otherwise return with ISTAT = 0
- (e) DATA : ASCII Decimal Equivalent (ADE) Value.

(i) Reserving and releasing device

The input and output or both parts of a two way port can be reserved. If the input part is reserved, data can be read from the device, whereas if the output part is reserved, data can be written to the device.

Reserving a device exclusively for calling program is given as follows

RESRV (LDN, IOF, IRET)

To verify whether a device has been reserved function WHDEV (LDN) can be used. If the return value is 0 then the device is free to use. A port must be free for use, if a device needs to be interfaced to that port. This can be achieved by monitor call RELES (LDN, IOF). (This routine may not prove effective if the part is reserved for specific job by system and in all such cases system commands should be used to release it).

(ii) Clearing the device buffer

Usually data are either read or written from a memory buffer associated with each port. Default size of this buffer is limited to 128 bytes, limiting the amount of data that can be written before it is read. This buffer is a ring buffer. If a program attempts a read operation while the buffer is empty, the program is normally set in a IOWAIT state till the data arrives. Input buffer is cleared by routine CIBUF (LDN). While output buffer is emptied by COBUF (LDN).

(iii) Amount of data available in the buffer

The number of bytes currently in the ring buffer is read through ISIZE. As there is one common buffer for the input and output part, the number of bytes that may

be written before the buffer is full is found by subtracting ISIZE from the size of the buffer (default is 128 bytes).

ISIZE = 0 implies that buffer is empty otherwise the number of bytes still available and read is returned. The difference between the ISIZE and the buffer size gives the number of bytes that may be written without entering a waiting state.

(iv) Reading and writing data

Data available in the buffer can be read on byte basis by function given as

INCH (LDN)

This function returns ADE value, similarly data can be output as follows:

OUTCH (LDN, DATA)

RECOGNITION OF THE GRAD IETE EXAMINATION (now AMIETE)

Copy of the Circular No. E (NG) II/90/RR-1/15 (R BE No. 230 of 1990) dated 5-12-90 from Railway Board is reproduced below :—

A representation has been received on the above subject from the Institution of Electronics and Telecommunication Engineers stating that a pass in the AMIETE Examination of the above Institution is recognised by the Government of India and Union Public Service Commission for the purpose of recruitment to supervisory posts and services under the Central Government as also by various State Governments and IITs/Engineering Colleges whereas it is not being recognised by some Railway Recruitment Boards.

The matter has been carefully considered by the Railway Board who have decided that professional and technical qualifications such as a pass in Sections 'A' and 'B' of the AMIE, Graduateship Examination of the Institution of Electronics and Telecommunication Engineers etc which are recognised by the Government as equivalent to degree in Engineering and for purposes of recruitment to Superior Services/posts under the Central Government should also be recognised for purposes of admission to competitive examinations conducted for recruitment to various posts on the Railways for admission to which an Engineering Degree of a Recognised University or equivalent has been prescribed as the basic educational qualification.