APPENDIX E
Disablement for the Teletype delay

To disable the Teletype delay (see Section 12), which is highly advisable if the NIC-80 is no longer being controlled from a Teletype, is very simple. Proceed as follows -

(a) Cut the connection coming from pin 8 of IC30.

(b) Connect pin 3 of IC31 to earth.

APPENDIX D
Upgrade for the NIC-80 RS232 board

The information reproduced below was kindly provided at no cost by Nicolet. The upgrade described has been carried out on the six Lancaster NIC-80s with complete success. It is well worth implementing in view of the very substantial increase in speed and performance that is achieved: the maximum baud rate is thereby raised from its standard value of 2400 baud to 19200 baud, on both channels A and B.

APPENDIX C
The NIC-80 RS232 B-channel

Many NIC-80s were shipped with only the A-channel RS232 link, used by the Teletype or other terminal, operative. However, it is usually very straightforward to get the B-channel running, as internal provision already exists for this in all cases that we have come across. All that is necessary is to complete the wiring in accordance with the schematics, which involves -

(a) Add the extra UART and associated chips and small components to the RS232 PCB from slot 11 (assuming that they are not already fitted). It will be found that empty positions are ready waiting for them, and it is simply a matter of soldering them in.

(b) Add the additional RS232 "D-socket" to the back of the NIC-80, a few inches above the existing A-channel socket.

(c) Connect the socket to appropriate pins on the underside of slot 11, simply paralleling the wiring of the A-channel in accordance with the schematics.

APPENDIX B
Layout and operation of NIC-80 KERMIT

For users who wish to modify the sourcecode, some very brief notes on the operation of NIC-80 KERMIT may be helpful. The programme layout when in use is as follows -

0 - 2777	I/O disk buffer (0-1777 for floppy)
3000 - 5777	disk directory (4000 - 5777 for floppy)
6000 - 7777	FPP72, DIRFUN overlays, DEMON monitor head
100000 - 100777	input buffer for dumb terminal mode
101000 - 101777	output buffer for dumb terminal mode
102000 - 102777	output packet construction
103000 - 103777	input packet construction
104000 - 105777	text for command summary, notes, etc
106000 - 107777	work routines for receive mode
110000 - 117777	command block, controlling routines
112000 - 113777	work routines for send, packet construction and deconstruction
114000 - 115777	filename parsing, NICBUG, BOOTS
116000 - 117777	disk routines, general utilities, arrow display

When first loaded, FPP72 occupies 102000 - 103577 but it is immediately moved to its operating position in 6000 - 7577 as soon as the programme starts; this arrangement allows KERMIT to be stored conveniently as a single file.

The heart of the dumb terminal emulator mode lies in a routine called RSDUTY on page 110000. In effect, this simulates an interrupt system in software. It inspects, in turn, each of the four RS232 flags and takes action accordingly: there are no separate wait loops for the individual devices. The advantage of this approach is that is almost impossible to take the NIC-80 unawares, e.g. for it to miss an incoming B-channel character while in a wait loop servicing the keyboard or printing a character. Note that, even at 9600 baud, the time taken for a character to arrive or be sent is $\sim$ 1 ms, which is a lot longer than the NIC-80 instruction time of $\sim$ 4-6 μs. Thus, rather than waiting while a character is being received/transmitted/entered/printed, it is much better for the NIC-80 to be doing other things.

The packet send/receive modes also use RSDUTY. However, when receiving, the system watches out for the arrival of a $\uparrow$A signifying the start of a packet, and then diverts the incoming data stream to the packet input buffer at 103000. The output packets are constructed and sent from the buffer at 102000. The current ouput packet is preserved there until it has been acknowledged, and so it can easily be re-sent if the first copy gets corrupted or lost.

Note that, although the appearance of the "@" prompt for filename entry seems at first sight to imply use of the usual DCI routine, the routine actually employed is the more sophisticated one held on page 114000, enabling the use of jokers/wildcards. (It is a slightly modified version of the keyboard input routine extracted from PNICL).

APPENDIX A
Construction of NIC-80 KERMIT

NIC-80 KERMIT can be supplied on 8-inch floppy disk, on a Diablo 30 front-loading disk cartridge, or on paper tape. Future versions or upgrades of the programme can be sent by electronic mail or ftp to a suitable local computer over Internet but, until the user has a version of KERMIT on his NIC-80, he or she will probably have no way of downloading the programme from the machine used for receiving mail. Thus the initial provision will almost certainly need to be on one of the media mentioned above. Users who wish to modify KERMIT to suit their own local situation, or to add additional features, should ask for the sourcecode. What follows is a description of how to construct the executable programme from the sourcecode.

There are two sections of code to be assembled. The first, which provides the main part of the programme, is KERM**.A, where the ** are two numeric digits (i.e. *not* following the Nicolet joker/wildcard convention for filenames) representing the development number of the version in use. The second, very short, piece of code is called BOOTS.A and provides the spare bootstraps (for the disk-drives or high-speed paper tape) for use in emergencies. Both KERM**.A and BOOTS.A are assembled with ASM, creating binaries KERM**.C and BOOTS.C which can be loaded with LOADER. The sequence of commands for constructing NIC-80 KERMIT is then -

```
*RUN LOADER
@BOOTS.C:M
*STO BOOTS 115366 - 115476; 7600
```

(Once BOOTS has been assembled and stored, it of course need not be re-assembled when a future version of KERMIT is being constructed).

```
*RUN LOADER
@KERM**.C:M
*LOA FPP72 102000
*LOA NICBUG 114632
*LOA BOOTS
*STO KERMIT 102000-117777; 110000
```

where FPP72 is the standard Nicolet NIC-80 floating point package and NICBUG is the octal debugging programme. The above description assumes that a Diablo-30 is in use. To construct the programme on a NIC-298/299 floppy system, the procedure is identical except that the code is assembled with FASM and loaded with FLOAD.

13. Emergencies

It has been found prudent to keep some spare bootstraps in core, just in case the monitor head at 7600 should somehow become corrupted. In the Lancaster system, the relevant code (BOOTS) is usually held on page 114000, just above NICBUG (which is also incorporated into almost all programmes). Thus, in the event of a system crash while (or just after) KERMIT is being run, after which the NIC-80 declines to start properly at 7600, it is worth attempting to start it at one of the following addresses using the switch register:

115367 to read a new monitor head from a NIC-298/299 floppy disk
115414 to read a new monitor head from a NIC-294/Diablo hard disk
115436 to read in the NICOLODEON paper tape on the high-speed reader

If all of these fail, it may still be worth starting at 114700 to try to raise NICBUG, which provides a much quicker way of entering code to test the system than by using the switches.

For NIC-80 systems fitted with a ROM bootstrap loader it is, of course, usually better to use it in preference to the resident disk bootstraps.

P V E McClintock 5 May 1994

The K and E commands can be used to turn off (kill) or restore the echo from the remote system, assuming that this is operating a full-duplex protocol, and that it is running under Unix.

11. Miscellaneous commands (M, Q, D, P, T)

An exit to the Nicolet Demon monitor can be effected by entry of either a Q or an M. One can return to the programme again by typing GO 110000 to the DEMON * prompt.

Entry of a D will cause the disk directory to be printed, without needing to leave the programme.

The P command will be for setting the various KERMIT parameters. It is not yet implemented, but parameters can readily be modified if need by use of NICBUG (B) in conjunction with a listing of KERMIT.

The T command is a toggle that activates or de-activates the RS232 B-input.

12. Hardware considerations

The NIC-80 was an extraordinary machine for its era, and has proved remarkably reliable (apart from contact problems) in service. Even after some twenty years of active use, it still performs very well the kinds of tasks for which it was designed. However, for efficient operation of KERMIT, a few minor changes are needed.

First, many NIC-80s were delivered without an operative B-channel, even though most of the internal circuitry for the B-channel was in place. Appendix C describes the actions needed to complete the job.

Secondly, it is assumed that the NIC-80 is being controlled from a standard VDU ASCII terminal, rather than the original ASR33 Teletype. As a result, the baud rates on both the A and B channels of RS232 can be raised above the original value of 110.

Thirdly, the UART chip used by the NIC-80 for RS232 communications can be operated at very much higher speeds than were permitted by the original crystal. Appendix D describes how the RS232 PCB can easily be upgraded. Note that this is a properly engineered *official Nicolet upgrade*, based on information that was provided by Don Parker.

Finally, there is a need for a more subtle modification if full advantage is to be taken of the higher baud rates. There is a cunning piece of circuitry in the NIC-80 that looks for carriage returns and implements a $\sim$ 100 ms delay whenever one is detected. During this period all of the RS232 circuitry is effectively paralysed, both for input and output. Consequently, there can sometimes be considerable loss of incoming data. Fortunately, a trivial modification to the RS232 PCB will eliminate the delay facility, which presumably originated in the need to wait for a mechanical return in the old Teletype before it would accept the next character: see Appendix E. (This problem, although easily dealt with once it had been identified, took a long time to understand; it caused considerable mystification, partly because the delay facility does not seem to be documented other than in the NIC-80 schematics).

With the small modification described above, the Lancaster NIC-80s are normally operated with 19200 baud set on their RS232 channels-A, and 9600 baud on their channels-B, and seem to be entirely reliable at these speeds.

significances of the packet indicator symbols are as follows -

s - a packet has been sent
S - the same packet has been re-sent
r - a good (uncorrupted) packet has been received
w -the packet received has the wrong sequence number
ℓ - the packet received has the wrong length specified
c - a checksum error in the received packet
n - a NAK (negative acknowledgement) has arrived
N -a NAK to the *next* packet has arrived
t - receiver has timed-out while awaiting a packet
T -a time-out occurred during reception of a packet

Note that, because of the order in which KERMIT performs validity checks of an incoming packet, not all relevant symbols will be printed: only one error is noted for each invalid/corrupted packet. Thus, if the same packet arrives a second time, only a w will be printed showing that the sequence number is wrong, regardless of whether or not the length and checksum of the packet are correct; this will normally be followed by an S to indicate retransmission of the NIC-80s last packet.

With the I facility toggled on, the previous four outgoing packets are preserved at octal 200 intervals in addresses 102000-102777. The previous four incoming packets are preserved within 103000-103777. They can be inspected by invocation of NICBUG under the B command as described above, and it is then usually fairly obvious where the problem lies.

9. Help commands (H, N)

A command summary can be obtained by entry of an H, and some succinct notes on operation are printed after an N. The latter are intended mainly to help anybody who has acquired the NIC-80 KERMIT programme, but does not have a copy of these instructions.

10. Convenience commands (L, U, E, K)

There are four commands that simply reproduce, automatically, what the user could otherwise do manually in dumb terminal mode (see Section 4). They are particular to the system at Lancaster University, but could readily be modified for use under other conditions. If there seems to be any demand, these commands could all be made user-configurable in a future version of KERMIT.

The L command effects an automatic login, once KERMIT has been configured with the username and password by means of the U command. Note that the U command should be entered for a *newly loaded* copy of KERMIT. This ensures that, when the configured version is re-stored (automatically) on the disk, the copy of FPP72 held initially at 102000 will be intact. There are three stages to the login process. First, KERMIT makes contact with a PAD, watching out for the ">" PAD prompt. Secondly, it calls up the Lancaster Sequent Symmetry, coded as lancs.cent1, and looks out for the ":" prompts corresponding to the standard Unix "Username:" and "Password:" queries, which it answers appropriately. Thirdly, it watches out for a ">" prompt from operating system showing that the login has been completed.

convention. One or more "?"s can be used to replace one or more characters when entering the filename and/or extension (and the usual Nicolet "*" can also be used in place of the first or second triad of characters). All files whose names match the corresponding mask, where the "?" is assumed to represent any character, will then be sent to the external system.

If necessary, transfer of the current file can be aborted with a ↑X, and transfer of a file series can be aborted with a ↑Z, provided of course that the external system supports these facilities. Alternatively, a ↑G, can be entered to force an unconditional return to the command block. If the packet exchange somehow becomes stuck during file transfer, entry of a carriage return will cause the last outgoing packet to be re-sent, and may possibly get the transfer started again.

6. Receiving files from the external system (R)

To receive a file, after one has logged in and activated the Kermit on the remote system, the normal procedure is as follows. Place the remote Kermit in send mode, e.g. with command "send filename" where filename is the name of the file that is to be sent to the Nicolet, return to the local KERMIT with a ↑G, and enter the R receive command. Once the transfer has started, the progress arrow will be seen advancing across the screen as described above for the S command; the arrow will be fainter, however, because it is displayed only during sending packets which, for receive mode, are relatively short acknowledgements of data packets received.

If a file with the same name as the incoming one already exists on the Nicolet's disk, the usual "DELETE:" query will be printed and must be answered with a Y if the original file is to be overlaid. Multiple transfers of incoming files can be performed provided that the remote Kermit supports this facility.

The ↑Z, ↑G and CR commands are available for aborting or attempting restart a file transfer, as described in the previous section.

7. Transfer of files between two Nicolets

To effect direct file transfer by KERMIT between two NIC-80s, a slightly different procedure to that described above under Sections 5 and 6 would obviously have to be used. KERMIT would have to be run on both machines, giving the commands locally in both cases. One of them would be put into receive mode with an R, and the S command would then be used to send files from the other one.

However, it is equally easy, and somewhat faster, to transfer files between two NIC-80 systems by means of PNICL. This programme uses the same packet-exchange protocol as FILMOV on NIC-1180 and NIC-1280 Dexter systems. Thus PNICL can be used to transfer files between the NIC-80 and other types of Nicolet system, as well as between two NIC-80s. It is not compatible with KERMIT.

8. Diagnostic facilities (B, I)

If KERMIT seems not to be functioning properly in relation to a particular external system, there are some simple facilities to help identify the problem. Entry of a B invokes the standard NICBUG NIC-80 octal debugging programme, with its usual facilities for placing breakpoints and for examining and modifying the contents of specified addresses.

Entry of an I toggles on (or off) the packet-indicator printing facility and also causes the last few packets sent/received to be preserved for examination using NICBUG. The

Filenames on the NIC-80 are all held in the form of packed 6-bit ASCII, with a maximum of six characters before the extension. The names are converted and transmitted to the external system as 7-bit ASCII, in lower case to conform with unix conventions. When a file is received by the NIC-80, its name is converted to the packed 6-bit format. Characters after the sixth are ignored. So also are any points (periods) unless these are followed by an upper-case or lower-case “A”, “B” or “C”; which are treated as filename extensions and converted to upper-case if they are not that already. An “A” extension implies that the incoming file will be treated as a text file. It will be converted and stored in standard form as packed 8-bit ASCII, which can subsequently be read or edited using one of the NIC-80 editors (e.g. NED or FNED).

4. Dumb terminal mode (C, ↑G)

Following the C (connect) command, the system enters its dumb terminal mode. Any characters entered at the keyboard are transmitted over the RS232 B-channel to the external system; any incoming characters received on the B-channel are sent to the terminal and printed or displayed on the VDU screen. Thus it is straightforward to login to the remote system, in the usual way, if it is e.g. a mainframe or workstation. Once the remote system has been prepared for receiving or sending a file (see sections 5 and 6 below), control can be returned to the KERMIT command block by entry of a ↑G.

In dumb terminal mode, conventional (XON/XOFF) flow-control is operative. Thus the incoming data stream can be stopped by entry of a ↑S (assuming that the remote system follows the same protocol), and restarted again with a ↑Q. Likewise, the remote system can interrupt and restart the data stream from the NIC-80.

In practice, large quantities of data (including login messages etc) often arrive very fast immediately after logging in to a mainframe. To avoid loss of characters caused by buffer-overflow in the Nicolet under such conditions, it is important that the baud-rate setting of channel-A should be equal to or higher than that of channel-B: see also Appendices C-E.

5. Sending files to the external system (S)

To send a file to an external system the usual procedure is as follows. First login, using dumb terminal mode as described above. Activate the remote Kermit and set it into receive mode, e.g. by typing “receive”. Return to the NIC-80 KERMIT with a ↑G, and enter the S send command. The system will ask for the name of the file to be sent, which should be entered at the @ prompt. If the user does not respond to the prompt, the system will eventually time-out and jump back to the command block. The same is true at all points where user-input is awaited *except* in the command block itself.

Once the two Kermits have made contact, and transfer of the file has started, progress can be monitored by watching the arrow on the CRT display as it advances from left to right across the screen. In the case of text files, which are always filed as complete disk records but usually only occupy part of them, progress of the arrow will tend to underestimate the proportion of the file that has been transferred; the transfer progress of core-image files will be indicated precisely.

On completion of the file transfer, the system will ask whether any more files are to be sent. If the user answers with a Y, the @ prompt will appear for entry of the next filename; and if he or she answers with an N, control will return to the KERMIT command block.

Multiple file-transfers can often be effected conveniently by means of a joker/wildcard

H - help: print command summary
I - toggle indicator symbol printing and packet archiving on/off
K - kill (switch off) external echo
L - login on remote system
M - return to DEMON monitor (same as Q)
N - print notes on operation
P - set KERMIT parameters (future implementation)
Q - quit, return to DEMON monitor (same as M)
R - receive programmes or data from external system and file them on disk
    ↑G - unconditional return to command block
    ↑X - abort transfer of current file
    ↑Z - abort transfer of file series
    CR - retransmit last packet, and continue
S - send programmes or data to external system and
    file it on disk (with ↑G, ↑X, ↑Z and CR as for R command)
T - toggle RS232 B-input off/on
U - configure username and password (for L command)

3. Data formats on Nicolet and external systems

Files sent from the Nicolet are always sent as text files to the external system, but in two different formats, depending on whether or not they start off as text files or core-image files.

Text files, which have a .A extension to the filename, are held on the NIC-80 in the form of packed 8-bit ASCII, with five characters per pair of 20-bit Nicolet words. Before transmission to the external system they are unscrambled, and then sent as a sequence of 7-bit ASCII characters (i.e. with Nicolet's bit-7 removed) either in the usual Nicolet upper-case, or in lower-case, as appropriate. The file will therefore appear on the external system in readable form and, if necessary, can be edited there before being returned to the same (or a different) Nicolet machine.

Non-text (core-image) files are sent to the external system in the form of decimal integers, one per 20-bit word, separated by commas, with a carriage-return/line-feed after every eighth number. This is a convenient format for FORTRAN, and is readily converted to whether other format may be needed by any particular plotting package. Thus, core-image files such as specta and other forms of averaged data can be readily be analysed, manipulated edited and plotted on the external system. Nicolet programmes are also core-image files, but obviously mean nothing outside the Nicolet. Nonetheless, they can be transmitted, held on an external system, and sent to a different Nicolet or returned to the original one, where they are reproduced in identical form as 20-bit integers.

All files sent to an external system, either text or integer, have a first line added consisting of three decimal integers. These carry relevant directory information, and represent respectively: the length of the file, in 20-bit words; the load address in the Nicolet; and the programme starting address. One or more of these numbers will be used by the receiving Nicolet to indicate progress and to make an appropriate directory entry once the file has been stored. Where, for some reason, a non-Nicolet file (or one not previously transferred by Kermit) is to be sent to a Nicolet, a first line containing three appropriate numbers *should be edited into the start of the file before it is sent.*

Instructions for NIC-80 KERMIT

1. Introduction

The NIC-80 KERMIT provides a basic implementation of the packet-exchange protocol described in Frank da Cruz's *Kermit: a File Transfer Protocol* (Digital Press, Bedford, Massachusetts, 1987). It is applicable to Nicolet data-processors in the NIC-80 series (LAB-80, MED-80, NMR-80, BNC-12, etc), provided that they have the full 12 K of internal memory installed and that they operate in conjunction with either a floppy (NIC-298, NIC-299) or hard (NIC-294/Diablo 30) disk drive. Users of these systems will be able to utilize KERMIT to transfer programmes and data between their Nicolet systems and external computers, using the NIC-80 RS232 B-channel. Thus they will be able, for example, to send averaged spectra or data to a mainframe or PC for further analysis or plotting using modern plotting packages, and to exchange programmes and data-files with each other by e-mail on a world-wide basis.

The present version of NIC-80 KERMIT is a relatively simple one. It uses standard sized packets; and innovations such as sliding windows, repeat-prefixing, or 8th-bit-prefixing have not been implemented. Nonetheless, it seems to function satisfactorily and is likely to open a new dimension to many Nicolet users.

The sections that follow describe the individual commands and facilities. Appendix A explains how to construct the executable programme from its source-code, Appendix B gives a very brief description of how the programme works, Appendix C describes how the RS232 B-channel can easily be added to a NIC-80 which does not already have it wired up (even though the internal circuitry usually exists) and Appendices D and E describe some minor modifications of the Nicolet hardware that vastly enhance the speed at which KERMIT can reliably operate.

Comments, amendments, reports of bugs detected, and other suggestions should be sent to

P V E McClintock
School of Physics and Materials,
Lancaster University
Lancaster, LA1 4YB, UK.
Tel: +44-524-593073
Fax: +44-524-844037
E-mail: pya007@lancaster.ac.uk

from whom copies and updates of NIC-80 KERMIT can also be obtained.

2. Command summary

Once the programme has been installed on the disk it can be started with a *RUN KERMIT command. The KERMIT commands, described in more detail in the sections that follow below, are -

B - call NIGBUG (octal debugging programme)
C - connect to external system (dumb terminal mode)
    ↑G - return to command block
D - do a directory listing
E - set external echo (for a Unix machine)