OST COMMUNICATIONS UTILITY

he host communications utility (HCOMM) for the 9520 Software Development System rovides the ability to transfer ASCII files between the 9520 and other devices. t be executed under either MP/M or CP/M on a 9520. The transfer is made ver an RS-232 type link. HCOMM responds to and manages the RS-232 control ines DTR and DSR. In addition, two software protocols (positive/negative cknowledge and XON/XOFF) are supported. The transferred files are composed of 7- or 8-bit ASCII characters in variable length records. Each record is normal- y terminated by a selectable end-of-record sequence. The entire file may be erminated by a selectable end-of-file sequence. Other devices HCOMM may commu- icate with may be another computer (through either a terminal or a non-terminal ort) or a piece of equipment (such as a PROM programmer). Virtually any device upporting RS-232 communications may be used.

eneral Operations

everal steps are required to prepare for and perform a file transfer. First, the ype of connection must be determined. This may be either to a terminal (console) port on another computer or to a non-terminal port on another computer or piece of equipment. Then, the physical link must be established. This equires knowledge of the signals originating from and expected by both the 9520 nd the other device. A special cable or a breakout box may be needed to make he correct connections. Next, each of the options controlling a file transfer ust be considered and correctly selected. The option selections depend on the perating parameters of the interface, the structure of the transferred file, he protocol used, and the type of connection. The options may also depend on he direction of the file transfer. After the options for a particular transfer av been selected, they may be saved in an options file and quickly loaded gain in a subsequent invocation of HCOMM. Finally, files may be transferred. o do this, knowledge of how to operate both HCOMM and the other device is equired.

hree examples are included in this document to describe HCOMM usage in detail. he examples are presented in a step-by-step manner, with each step following he description of the relevant HCOMM display. The displays are in the order hat an operator would normally consider them in performing a transfer. Gener- lly the following questions are answered for each step: Where was the infor- ation obtained that was required in this example? How was the information used o select an option or prescribe an action? And what, if anything, was entered n response to the HCOMM queries?

n this chapter, sets of example descriptions are preceded and followed by thick old lines. Examples of actual HCOMM displays and queries are preceded and ollowed by thin bold lines. Operator entries are presented in bold print. hree special operator entries are enclosed in angle brackets:

<cr>	carriage-return character - return key pressed
<^T>	terminal mode entry character - control and T keys pressed simultaneously
<^@>	null character - control and @ keys pressed simultaneously (shift may also be required for @ key)

tion of specific products, not produced or sold by Millennium, should not be nstrued as a recommendation, by Millennium, of their merits over other, simi- products. They have been chosen, as examples of actual transfers, merely to l trate some types of HCOMM usage, and how a user, faced with a similar task, is proceed.

---

xample #1 Intellec Series II, Model 230, running ISIS-II, V3.4

n this example, an Intel hex format file was transferred from the Intellec to e 9520. Intel hex format is a means of specifying object files using ASCII aracters. Each record contains an introductory character, a record length, an ddress, a record type, memory data, and a checksum. All items except the ntroductory character are represented in hexadecimal digits. The 9520 has tilities for manipulating and executing Intel hex format files.

xample #2 8002 Microprocessor Lab running TEKDOS, V3.1, and CP/M, V2.2

wo operations were performed in this example. In one, a TEKHEX format file was ansferred from the 9520 to the 8002's memory. TEKHEX is an object file ormat, developed by Tektronix, that is similar to the Intel hex format. In the ther operation, a text file was transferred from the 8002 to the 9520.

xample #3 UNIX Time-Sharing System

is example, text files were transferred in both directions between a com- ter running the UNIX system and the 9520. Communication with the UNIX system s through a modem connection over a dial-up telephone line.

---

voking and Using HCOMM

host communications utility may be invoked in either of two forms:

```
OA>HCOMM
```

:

```
OA>HCOMM filename
```

the first form the HCOMM options are initialized with their default values. the second form the specified options file is loaded to initialize the HCOMM tions (see display K). The filename may include an extension.

MM initially presents its menu. A menu item (display) may be selected by ing the appropriate letter (upper or lower case) and pressing the "return" . The console screen is then filled with the selected display. Some dis- are purely informational. Others control options. These present

information about the options as well as their current values. The operator may alter the values of the options in response to the display's queries. Entering a blank line (pressing the "return" key and nothing else) in response to any query leaves the value of the option unaltered. The menu is presented again after the operator has responded to all queries in a display. Other displays control file actions. In these a blank line immediately returns the operator to the menu. The remaining menu items allow the operator to request the menu be presented again, or to exit HCOMM. An example of the menu, display selection, and return to the menu is shown below.

```
File Transfer Utility - Asynchronous Communications Interface - Version 1.0

Menu:
     A    General information regarding protocol
     B    RS-232 link information
     C    DTR/DSR information
     D    Interface parameters options
     E    End-of-record sequence option
     F    End-of-file sequence option
     G    Abort sequence option
     H    ACK/NAK options
     I    XON/XOFF option
     J    Terminal mode options
     K    Save or load selected options to or from a file
     L    Transfer files from interface to disk
     M    Transfer files from interface to disk
     N    Display this menu
     O    Exit file transfer utility

In response to any query, the new options (blank or comma separated) must be
entered with a carriage-return.  A blank line leaves the options unaltered.
Striking CTL-C as the first character returns control to the operating system.

Int desired menu item letter  (Q <letter> - skips informative text): A <cr>
```

```
-al Information Regarding Protocol

File Transfer Utility is used to transfer ASCII files between the 9520 and
 devices over an RS232 type link.  The other device may be a computer,
JM programmer or anything else capable of meeting the interface and
col requirements.

 is transferred in variable length records terminated by an operator
table, end-of-record sequence.  The maximum record length is 255
acters (excluding the end-of-record sequence).

ous methods of synchronizing and acknowledging record transfers are
lable (DTR/DSR, ACK/NAK, and XON/XOFF).  Any or all of these protocols may
elected, but at sufficiently low transfer rates (2400 or less) none are
ired.  The DTR/DSR protocol is always selected, but it may be disregarde
ided certain hardware considerations are met (see DTR/DSR Information).

 transfers are terminated normally by an operator selectable, end-of-file
ence.  An abnormal termination is signalled by the transfer of an operator
table, abort sequence.

 blank line to return to menu:  <cr>
```

```
 Transfer Utility - Asynchronous Communications Interface - Version 1.0

 A    General information regarding protocol
 B    RS-232 link information
 C    DTR/DSR information
 D    Interface parameters options
 E    End-of-record sequence option
 F    End-of-file sequence option
 G    Abort sequence option
 H    ACK/NAK options
 I    XON/XOFF option
 J    Terminal mode options
 K    Save or load selected options to or from a file
 L    Transfer files from interface to disk
 M    Transfer files from interface to disk
 N    Display this menu
 O    Exit file transfer utility

sponse to any query, the new options (blank or comma separated) must be
ed with a carriage-return.  A blank line leaves the options unaltered.
ing CTL-C as the first character returns control to the operating system.

 desired menu item letter  (Q <letter> - skips informative text):
```

f the operator wishes to suppress the informative text of a display the quiet
pt n should be used. This is done by typing a "Q" or "q" followed by a blank
r comma and then the desired menu item letter. If the selected display con-
rols options their current values are presented before the display's queries.
he menu is not presented again on return from a display that was selected with
he quiet option. Only a reminder of how to request the menu is displayed, as
hown below.

---

nter desired menu item letter (Q <letter> - skips informative text): Q A <cr>

nter a blank line (or anything) to return to menu: <cr>

OTE: Item N displays the full menu

nt desired menu item letter (Q <letter> - skips informative text):

COMM operates in three modes. It is in entry mode whenever it is waiting for
he operator to type something in response to a query. The operating system
rovides a means to abort HCOMM, while in entry mode, by typing control-C as the
irst character of a response. A control-T entered as the first character of a
esponse puts HCOMM in terminal mode after the "return" key is pressed. When
erminal mode is exited the remainder of the line (following the control-T) is
aken as the response to the query, unless it is blank, in which case the query
s repeated. For more information on terminal mode see display J. Finally,
COM is in transfer mode when a file transfer is taking place. A description
f transfer mode may be found preceding the description of display L.

nitial Connection Decision

he first decision facing the user is how the 9520 is to be connected to the
ther device. The other device may treat the 9520 either as a terminal or a
on-terminal. Most computer systems provide utilities to display ASCII text
iles at a terminal and accept input from a terminal to be stored in a file.
he 9520 can be connected to these systems as a terminal and make use of these
tilities to transfer files. Also, some computer systems provide special utili-
ies that support communications through alternate I/O ports. The 9520 can be
onnected to these as a non-terminal. And of course, other equipment may be
erminal peripherals themselves so with these the 9520 must not be considered a
erminal.

---

ample #1 Intellec Series II, Model 230, running ISIS-II, V3.4

he Intellec had an alternate port (TTY) that was connected to a line printer.
his printer was known to utilize the type of RS-232 communications required by
COMM. By reading the associated manual, it was learned that the Intellec had
:ili ies for copying ASCII text files to its TTY port. Since the intended
ans r was only from the Intellec to the 9520, these capabilities were deemed
ff ient. It was decided, therefore, to connect the 9520 as a non-terminal to
he TTY port on the Intellec.

COMMUNICATIONS UTILITY

ple #2 8002 Microprocessor Lab running TEKDOS, V3.1, and CP/M, V2.2

eading the associated manual, it was learned that the 8002 had a terminal as well as an alternate remote I/O port. For the text file transfer either could have been used. On reading further, however, it was found that the EX format file transfer could only be accomplished with an 8002 utility that only the remote I/O port. It was decided, therefore, to use the remote I/O for both transfers and to connect the 9520 to it as a non-terminal.

mple #3 UNIX Time-Sharing System

only available connection to the UNIX system was through a modem. By ding the UNIX system manuals, it was learned that utilities existed for both iting a text file to a terminal and accepting input from a terminal to be ored in a text file. Therefore, text file transfers both to and from the UNIX stem were possible. The 9520 was necessarily connected to the modem as a rminal.

Est ishing the Physical Link

Display B - RS-232 Link Information

To establish the physical link, the user must connect the RS-232 like, line printer port (J12) on the 9520 to a similar port on the other device. HCOMM handles its port in a fashion similar to Data Communications Equipment (DCE). The other device's ability to manage and respond to these lines must be determined. It may be necessary to make a special cable, or use a breakout box, to cross connect certain lines. A detailed description of the RS-232 lines handled by HCOMM and the 9520 is presented in display B, as shown below.

S-232 Link Information

The interface to the 9520 involves an RS-232 type link. The pins connected, the lines assigned to them, the directions of the signals, and their uses are indicated below:

-	Protective Ground	(common)	
- TXD /	Transmitted Data	(to 9520)	- data transmitted from other device
- RXD /	Received Data	(from 9520)	- data received by other device
- RTS /	Request to Send	(to 9520)	- ignored
- CTS /	Clear to Send	(from 9520)	- same as DSR
- DSR /	Data Set Ready	(from 9520)	- controls other device transmission
-	Signal Ground	(common)	
- RLSD/	Rec. Line Sig. Det.	(from 9520)	- held marking (ON)
0 - DTR /	Data Terminal Ready	(to 9520)	- controls 9520 transmission

or more information on the "control" lines, see DTR/DSR Information.

nter a blank line (or anything) to return to menu: <cr>

---

xample #1 Intellec Series II, Model 230, running ISIS-II, V3.4

he Intellec was found to use the line printer attached to its TTY port in the ame manner as the 9520 used that printer. From this fact it was concluded that he Intellec TTY port was performing like DCE, as does the printer port on the 520. To connect the 9520 to the Intellec's TTY port, therefore, an RS-232 reakout box was used to connect the 9520's TXD pin to the Intellec's RXD pin. ince the data were only transferred from the Intellec to the 9520, the other ata line (9520's RXD to Intellec's TXD) was unnecessary, and therefore not con- ct Also, by examining the cable connecting the Intellec's TTY port to the ine printer, it was learned that the printer's DTR pin was connected to the nte ec's CTS pin. Since the printer was known to manage the DTR line correct- r (and hence that the Intellec had to respond to it correctly) the 9520's DSR in was connected to the Intellec's CTS pin. And, of course, signal ground was

connected. The connections that were used are tabulated below.

9520 pin		to		Intellec pin	
(TXD)	2	- - - -	3	(RXD)
(DSR)	6	- - - -	5	(CTS)
	7	-(common)-	7	

xample #2 8002 Microprocessor Lab running TEKDOS, V3.1, and CP/M, V2.2

y reading the associated manual, it was learned that the 8002 remote I/O port erforms as Data Terminal Equipment (DTE) and manased and responded to the DTR nd DSR control lines. Therefore, the 9520 was connected directly to the 8002 emote I/O port with a flat ribbon cable. The necessary connections are tabu- ated below.

9520 pin		to		8002 pin	
(TXD)	2	- - - -	2	(TXD)
(RXD)	3	- - - -	3	(RXD)
(DSR)	6	- - - -	6	(DSR)
	7	-(common)-	7	
(DTR)	20	- - - -	20	(DTR)

ample #3 UNIX Time-Sharing System

e modem performed like DCE, of course; therefore, a breakout box was used to ss connect the data lines. Also, by reading the modem's manual, it was rned that the modem required an active DTR signal from the connected termi- . The 9520's DSR pin was connected to the modem's DTR pin for this purpose. modem was also found to hold its CTS pin active when it was ready. This pin connected to the 9520's DTR pin. The connections that were used are tabu- ted below.

9520 pin		to		modem pin	
(TXD)	2	- - - -	3	(RXD)
(RXD)	3	- - - -	2	(TXD)
(DSR)	6	- - - -	20	(DTR)
	7	-(common)-	7	
(DTR)	20	- - - -	5	(CTS)

---

play C - DTR/DSR Information

primary protocol for controlling transfer synchronization and speed is two dy/not ready" control lines, DTR and DSR. The DSR line is held active when " is ready to receive and is inactive when HCOMM is not ready to receive. e other device cannot respond to this signal a software protocol must be cted to control the transfer of data to the 9520. However, no protocol i ssary if the data is sent to the 9520 at less than 280 characters/second. HCOMM will not transmit data unless the incoming DTR line is active. If

h: ther device does not manage this line correctly it must either be held
ctive at all times or remain disconnected. Also, in that event, a software
rotocol must be selected if the other device cannot accept the transferred data
s quickly as HCOMM sends them. This information is reiterated in display C,
s shown below.

TR/DSR Information

he interface's DTR line controls transmission from the 9520. The DSR line
ontrols transmission from the other device. Of the other two commonly used
ontrol signals, RTS is ignored and CTS is held the same as DSR (see RS-232
ink Information).

he 9520 will not transmit unless the other device indicates that it is ready
o receive by holding the DTR line in an active state. Similarly, the 9520
ill hold the DSR line in an active state only when it is ready to receive.
hus, these lines provide synchronization and speed controls if the other
evice is capable of managing and responding to them.

f the other device cannot manage and respond to these lines, synchronization
nd speed controls, if needed, must be accomplished with either an ACK/NAK or
ON/XOFF protocol, or both. Furthermore, the DTR line must be held in an
ct : state. This is done automatically in the 9520 hardware if the other
evice is not connected to the DTR line. If the other device is connected,
owever, it must be responsible for holding the DTR line in an active state.

nter blank line (or anything) to return to menu: <cr>

---

ample #1 Intellec Series II, Model 230, running ISIS-II, V3.4

om previous use of the line printer it was known that the Intellec's TTY port
rrectly responded to a control line signal. For this reason the 9520's DSR
n was connected as explained previously. Since the data were only transferred
om the Intellec to the 9520, the other control line (9520's DTR to Intellec's
S) was unnecessary, and therefore not connected.

ample #2 8002 Microprocessor Lab running TEKDOS, V3.1, and CP/M, V2.2

e DTR and DSR control lines were found to be correctly managed and responded
by the 8002 remote I/O port. (As will be explained later, however, an
K/NAK protocol was necessary to control synchronization for the TEKHEX format
le transfer.)

ple #3 UNIX Time-Sharing System

e the transfer speed and synchronization of the UNIX system could not be ged with the control lines over a modem connection, the control line con- ions were only required for correct operation of the modem.

ions Selection

Display D - Interface Parameters Options

Having established the physical link it is necessary to inform HCOMM of the interface parameters. There are four such: the size of each data character (either 7- or 8-bit ASCII), the selection of parity generation and checking for each character, the number of stop bits transmitted with each character, and the transmission rate (bits/second). In response to the query the user may enter a list of up to four letters, separated by blanks or a comma. The letters correspond to one option selected for each parameter and may be typed in any order. Provided there is no entry error, when the "return" key is pressed any new options entered are selected and the user is returned to the menu. The interface parameters options are controlled by display D, as shown below.

Interface Parameters Options

The interface may have one of the appropriate options, listed below, selected for each of its four parameters.

Data Size:	A - 7 bits	B - 8 bits	
arity:	C - None	D - Odd	F - Mark (7-bit data size only)
		E - Even	G - Space (7-bit data size only)
Stop Bits:	H - 1	I - 2	
Baud Rate:	J - 110	N - 2400	
	K - 300	O - 4800	
	L - 600	P - 9600	
	M - 1200	Q - 19200	

nterface parameters options:
ize: B - 8 bits Parity: D - None Stop Bits: H - 1 Rate: P - 9600

nter new options: <cr>

---

ote: The mark and space "parity" options are different from the others. The rest of the options are used to directly set up the interface hardware. The mark and space options, however, actually use 8-bit transmission and reception and use software to generate and check the most significant bit for the appropriate condition. If used with the 8-bit data size option they are the same as the no parity option.

---

mple #1 Intellec Series II, Model 230, running ISIS-II, V3.4

n the Intellec's use of the line printer it was known that three of its erface parameters had to be: 8 bit data size, no parity, and one stop bit. rate for the printer was 9600 baud. The Intellec TTY port was set to this e, however, by a special program. It was decided, instead, to use the port's ault rate for the Intel hex format file transfer. The name of the TTY port gested that its defalut rate might be 110 baud. This rate was tried and it rked. Since only the rate differed from the default initialization parame- s, only its new selection was entered and the other options were left unal- ed.

---

erface Parameters options:
e: B - 8 bits Parity: D - None Stop Bits: H - 1 Rate: P - 9600

er new options: ↓ <cr>

---

mple #2 8002 Microprocessor Lab running TEKDOS, V3.1, and CP/M, V2.2

reading the associated manual, it was learned that for the TEKHEX format file fer the interface parameters were: 8-bit data size, no parity, one stop , and 9600 baud rate. Since these are the default initialization values, no options were entered.

---

erface Parameters options:
e: B - 8 bits Parity: D - None Stop Bits: H - 1 Rate: P - 9600

er new options: <cr>

---

er, for the text file transfer, different interface parameters options were ected. From previous experience with CP/M it was known that the utility used transfer the text file would simulate 7-bit data with a space parity bit. default initialization parameters (8-bit data with no parity) could have n used without affecting the data, but the actual parameters were chosen tead, to provide a better check on the transfer.

---

erface Parameters options:
e: B - 8 bits Parity: D - None Stop Bits: H - 1 Rate: P - 9600

er new options: A G <cr>

---

x ple #3 UNIX Time-Sharing System

he interface parameters options for the UNIX system were determined from the ettings of the terminal being used. They were found to be: 8 bit data size, no arity, one stop bit, and a 1200 baud rate. Only the rate differed from the efault initializations so only that parameter was altered.

```
nterface parameters options:
ize:  B - 8 bits    Parity:  D - None    Stop Bits:  H - 1    Rate:  P - 9600

nter new options:  M <cr>
```

isplay E - End-of-record Sequence Option

he ASCII character data transferred by HCOMM is organized into variable length ecords that are terminated by an end-of-record sequence. The ASCII text files tored on the 9520's diskettes have variable length records terminated by a arriage-return/line-feed sequence. The purpose of display E, therefore, is to rovide the translation between the sequence used on the 9520's file system and hat being generated or expected by the other device. In some cases it may be ec sary to have an empty end-of-record sequence: with fixed length records, or example, or with variable length records that contain a count rather than a erminator. Such files may not be acceptable to many of the 9520's standard tilities but HCOMM may be able to transfer them for special purposes.

here may be up to 16 characters in the end-of-record sequence. They are each ntered, in order and blank or comma separated, as two-digit hexadecimal numbers epresenting their ASCII codes. An empty sequence is selected by entering "N" "n" alone on the line.

he HCOMM record buffer may contain up to 255 characters. If the record read om disk fills the buffer before a carriage-return/line-feed is encountered the ecord is sent to the other device without an end-of-record sequence. Likewise, f the buffer fills with characters from the other device before an end-of-ecord sequence is detected the record is written to the 9520 file without a arriage-return/line-feed. This will happen, particularly, when the end-of-ecord sequence is empty unless the entire file is less than 255 characters.

f the last record received by HCOMM is not properly terminated with an end-of-ecord sequence it is discarded. This is done so that, when the 9520 is con-ected as a terminal, the prompt that is normally sent to a terminal after a ansfer, by the other device, is not written to the 9520 file. This also means at when the end-of-record sequence is empty a last partial record containing alid data may be discarded. In this case, loss of data may be avoided by ving the other device send 254 pad characters to insure that all valid data is it n to the file. A text editor on the 9520 may then be used to remove any d characters that may be in the file.

Ind-of-record Sequence Option

he end-of-record sequence is the sequence of characters that terminates the ansferred records of ASCII characters. When stored on the disk of the 9520, his sequence is CR-LF (0D-0A hex). If the other device uses some other equence for this purpose, it must be made known to the File Transfer Utility o that the sequence can be correctly translated.

```
:-of-record sequence: 0D 0A

ter new end-of-record sequence
 (hex bytes or N if none): <cr>
```

---

mple #1 Intellec Series II, Model 230, running ISIS-II, V3.4

n previous use of the line printer it was known that the Intellec sent a riage-return/line-feed sequence at the end of each line. Since this is the ault initialization end-of-record sequence, no new sequence was entered.

```
-of-record sequence: 0D 0A

er new end-of-record sequence
 (hex bytes or N if none): <cr>
```

mple #2 8002 Microprocessor Lab running TEKDOS, V3.1, and CP/M, V2.2

eading the associated manual, it was learned that the 8002 expected a riage-return at the end of each record when transferring a TEKHEX format . That character was entered as the new end-of-record sequence.

```
-record sequence: 0D 0A

 new end-of-record sequence
ex bytes or N if none): 0D <cr>
```

a r, when selecting options for the text file transfer from the 8002, it was etermined by experimentation that the end-of-record sequence needed to be set o carriage-return/line-feed. The experiment was carried out by placing HCOMM n terminal mode and then transferring the text file from the 8002. This caused he text file to appear on the 9520's console. From this it could be seen that oth a carriage-return and a line-feed were being sent by the 8002 after each ine. Such a visual inspection could not rule out the possibility of other, nseen control characters, but a simple carriage-return/line-feed sequence was ried first and it worked.

```
nd-of-record sequence:  0D

nter new end-of-record sequence
   (hex bytes or N if none):  D A <cr>
```

xample #3 UNIX Time-Sharing System

t was known from previous use that the utility that accepted characters from a NIX system terminal and stored them in a text file expected a carriage-return fter each line. For the text file transfer from the 9520 to the UNIX system, herefore, that character was selected as the end-of-record sequence.

```
nd-of-record sequence:  0D 0A

nter new end-of-record sequence
   (hex bytes or N if none):  D <cr>
```

t was also known from previous use that the utility for writing text files to a NIX system terminal sent a carriage-return/line-feed after each line. Later, or the text file transfer from the UNIX system to the 9520, therefore, those wo characters were selected as the end-of-record sequence.

```
nd-of-record sequence:  0D

nter new end-of-record sequence
   (hex bytes or N if none):  D A<cr>
```

isplay F - End-of-file Sequence Option

rm ly, the end of the transfer is signalled by an end-of-file sequence. The SCII text files stored on the 9520's diskettes use SUB (1A hex) for this rpose when the last block of a file is a partial block. The purpose of isplay F, therefore, is to provide the translation between the character used the 9520's file system and whatever sequence is being generated or expected

e other device. In some cases it may be best, or even necessary, to have
..pty end-of-file sequence.

e end-of-file sequence may contain up to 16 characters. It is entered in the
.e manner as the end-of-record sequence.

---

nd-of-file Sequence Option

he end-of-file sequence is the sequence of characters that signals normal
rminations of the transferred files. When stored on the disk of the 9520,
is sequence is the character CTL-Z (1A hex). If the other devices uses some
er sequence for this purpose, it must be made known to the File Transfer
.lity so that the sequence can be correctly translated.

d-of-file sequence: 1A

.ter new end-of-file sequence
(hex bytes or N if none): <cr>

.mple #1 Intellec Series II, Model 230, running ISIS-II, V3.4

:m previous use of the line printer it was known that the Intellec sent no
.racters to signal the end of a file that had been copied to its TTY port. It
also known that the default initialization end-of-file sequence (1A hex) was
ikely to occur in the file to be transferred. If the default end-of-file
.ence had occurred in the file it could have caused HCOMM to close the file
aturely, thereby losing data. In order to prevent this an empty end-of-file
.ence was selected.

---

of-file sequence: 1A

new end-of-file sequence
ex bytes or N if none): N <cr>

---

x- le #2 8002 Microprocessor Lab running TEKDOS, V3.1, and CP/M, V2.2

y reading the associated manual, it was learned that the 8002 expected no end-f-file sequence after a TEKHEX format file transfer. Since the 8002 was interpreting the data, it recognized the end block defined in the format. Therefore, n empty end-of-file sequence was selected.

---

nd-of-file sequence: 1A

nter new end-of-file sequence
(hex bytes or N if none): N <cr>

---

ater, for the text file transfer from the 8002 to the 9520, the end-of-file e[illegible]nce was selected to be SUB (1A hex). This was done because the CP/M ti[illegible]ty used on the 8002 had the capability of sending this character, after the ext file, as a signal that the transfer was complete.

---

nd-of-file sequence: NONE

nter new end-of-file sequence
(hex bytes or N if none): 1A <cr>

---

xample #3 UNIX Time-Sharing System

or the text file transfer from the 9520 to the UNIX system, the UNIX system tility that was used expected to receive an EOT (04 hex) to signal the end of nput from a terminal. Therefore, that character was selected as the end-of-ile sequence.

---

d-of-file sequence: 1A

nter new end-of-file sequence
(hex bytes or N if none): 4 <cr>

---

or the text file transfer from the UNIX system to the 9520, the UNIX system tility that was used was incapable of sending a sequence to signal the end of e transfer. Therefore, when the options were being selected for this transr, an empty end-of-file sequence was selected.

---

d-of-file sequence: 04

ter new end-of-file sequence
(h bytes or N if none): N <cr>

T COMMUNICATIONS UTILITY

lay G – Abort Sequence Option

abort sequence may be received from the other device, or sent to it if the rator forces an abort from the 9520's console (see displays L and M). When abort occurs HCOMM immediately halts the transfer and discards any characters bsequently received from the other vice, the portion of the transfer received prior to the abort is written to e file.

e abort sequence may contain up to 16 characters. It is entered in the same ner as the end-of-record sequence.

---

rt Sequence Option

e abort sequence is the sequence of characters that signals abnormal rminations of the transferred files. It halts the file transfer and causes y subsequently transferred characters to be discarded. The sequence used by ne ther device, if any, must be made known to the File Transfer Utility so a it can be transferred and detected correctly.

rt sequence: 1B

er new abort sequence
(hex bytes or N if none): <cr>

---

mple #1 Intellec Series II, Model 230, running ISIS-II, V3.4

nough not verified, it was surmised that the Intellec would not respond to an rt sequence received by its TTY port, nor would it send such a sequence to nal a problem in sending the file. Therefore, an empty abort sequence was cted.

---

sequence: 1B

r new abort sequence
nex bytes or N if none): N <cr>

---

xample #2 8002 Microprocessor Lab running TEKDOS, V3.1, and CP/M, V2.2

y reading the associated manual, it was learned that the 8002 would send no bort sequence when it was receiving a TEKHEX format file; however, it would espond to a record that began with two slashes by causing succeeding characters n the record to be displayed on the 8002's console. Therefore, the abort equence had to begin with a carriage-return so that succeeding characters would e interpreted by the 8002 as the beginning of a record. Then a number of null haracters had to be sent, since it was known from previous experience that the 002 was slow to respond to a carriage-return. Next the two slash characters "//") had to be sent to signal an abort. Finally, a pound symbol ("#") was ent so that the fact that the transfer had been aborted would be visible on the 002's console.

---

bort sequence: 1B

nter new abort sequence
(hex bytes or N if none): D 0 0 0 0 0 0 0 0 0 0 0 0 2F 2F 23 <cr>

---

ikewise, when sending a text file to the 9520, the 8002 would not send an abort equence but would respond to an ETX (03 hex). Therefore, when the options for he text file transfer were being selected, that character was selected as the bor sequence.

---

bort sequence: 0D 00 00 00 00 00 00 00 00 00 00 00 00 00 2F 2F 23

nter new abort sequence
(hex bytes or N if none): 3 <cr>

---

xample #3 UNIX Time-Sharing System

t was learned by experimentation that the UNIX system would respond to an ETX 03 hex) at any time by aborting the currently executing program, but would send abort sequence. Therefore, an ETX was selected as the abort sequence for the ext file transfers in both directions.

---

ort sequence: 1B

ter new abort sequence
(hex bytes of N if none): 3 <cr>

..Y H - ACK/NAK Options

e ACK/NAK software protocol requires the receiver to respond, when a record is
nt to it, with either a positive (ACK) or negative (NAK) acknowledgement. The
quences used for acknowledgements are operator selectable. If this protocol
tion is selected HCOMM expects the appropriate sequences when sending and
enerates them when receiving. When sending, HCOMM sends a record and then
aits until receiving an acknowledgement. If it is positive then HCOMM sends
ne next record. If is it negative then HCOMM sends the same record again.
en receiving, HCOMM sends an acknowledgement when it receives a record. If
ere were no errors in the record (parity, overrun, etc.) than a positive
cknowledgement is sent. If any errors were detected then a negative acknowl-
dgement is sent.

record is normally terminated by an end-of-record sequence. However, for the
rposes of the ACK/NAK protocol, a full record buffer (255 characters) is al
nsidered a record. Therefore, ACK/NAK should only be used with a non-empty
d-of-record sequence, and none of the transferred records should contain more
an 254 characters, excluding the end-of-record sequence.

K/NAK Options

he file transfer protocol may prescribe a response from the receiver after
ch record is transferred. That response is either an acknowledgement (ACK)
a equest for retransfer (NAK). There are no timeouts in the protocol, so
is also possible for the receiver to control transfer speed by withholding
e response until ready for the next record.

spite the use of ASCII standard names, ACK and NAK are actually operator
ectable sequence of characters. The use of ACK/NAK does not conflict with
er protocol options unless the selected ACK/NAK characters are the same as
e other significant protocol characters (XON, XOFF, end-of-record sequence,
-of-file sequence, or abort sequence).

/NAK option: DISABLED
sequence: 06
sequence: 15

ble or disable ACK/NAK option (E or D): <cr>
er new ACK sequence
(hex bytes): <cr>
ter new NAK sequence
hex bytes): <cr>

---

ple #1 Intellec Series II, Model 220, running ISIS-II, V3.4

e the Intellec was capable of responding to a control line protocol, no
are protocol was needed. Hence, the ACK/NAK protocol was left in its
lt (disabled) initialization state.

```
CK..AK option:  DISABLED
CK sequence:  06
AK sequence:  15

nable or disable ACK/NAK option (E or D):  <cr>
nter new ACK sequence
  (hex bytes):  <cr>
nter new NAK sequence
  (hex bytes):  <cr>
```

xample #2 8002 Microprocessor Lab running TEKDOS, V3.1, and CP/M, V2.2

y reading the associated manual, it was learned that the 8002 required an CK/NAK protocol for the TEKHEX format file transfer. The positive acknowledge equence was learned to be the character "0" followed by a carriage-return. The egative acknowledge sequence was learned to be the character "7" followed by a arriage-return. Therefore, the ACK/NAK option was enabled and the sequences ere selected appropriately.

```
CK/NAK option:  DISABLED
CK sequence:  06
AK  equence:  15

nable or disable ACK/NAK option (E or D):  E <cr>
nter new ACK sequence
  (hex bytes):  30 D <cr>
nter new NAK sequence
  (hex bytes):  37 D <cr>
```

or the text file transfer from the 8002 to the 9520 no software protocol was equired; therefore, the ACK/NAK option was disabled. The sequences were left naltered since they have no effect when the protocol is not used.

```
CK/NAK option:  ENABLED
CK sequence:  30 0D
AK sequence:  37 0D

nable or disable ACK/NAK option (E or D):  D <cr>
nter new ACK sequence
  (hex bytes):  <cr>
nter new NAK sequence
  (hex bytes):  <cr>
```

xam..e #3 UNIX Time-Sharing System

he ACK/NAK protocol was left in its default (disabled) initialization state for he text file transfers in both directions.

ST COMMUNICATIONS UTILITY

```
:/NAK option:  DISABLED
.K sequence:  06
AK sequence:  15

nable or disable ACK/NAK option (E or D):  <cr>
nter new ACK sequence
  (hex bytes):  <cr>
nter new NAK sequence
  (hex bytes):  <cr>
```

isplay I - XON/XOFF Option

he XON/XOFF software protocol allows the receiver to suspend and resume the ransfer. When HCOMM's receive buffer becomes nearly full, it sends an XOFF haracter (13 hex) to the other device as a signal that the other device should op sending. While the transfer is thus suspended HCOMM continues processing e characters it has already received. Then, when the receive buffer becomes arly empty, it sends an XON character (11 hex) to inform the other device that may resume sending. HCOMM also responds to incoming XOFF and XON characters nen this protocol option is selected. When it receives an XOFF it sends at os two more characters before stopping, and sends no more until after it eceives an XON.

'XOFF Option

file transfer protocol may allow the receiver to temporarily suspend nsfer by sending an XOFF (13 hex) to the transmitter. To resume the nsfer the receiver sends XON (11 hex).

transmitter is presumed to be able to respond quickly (i.e. within two racter times) after receiving an XOFF. Similarly, the receiver is presumed e able to accept up to two additional characters after sending XOFF.

```
FF option:  DISABLED

e or disable option (E or D):  <cr>
```

xample #1 Intellec Series II, Model 230, running ISIS-II, V3.4

ince the Intellec was capable of responding to a control line protocol, no oftware protocol was needed. Hence, the XON/XOFF protocol was left in its efault (disabled) initialization state.

```
ON/XOFF option:  DISABLED

nable or disable option (E or D):  <cr>
```

xample #2 8002 Microprocessor Lab running TEKDOS, V3.1, and CP/M, V2.2

y reading the associated manual, it was learned that the 8002 did not manage an ON/XOFF protocol through its remote I/O port. Therefore, the XON/XOFF option as left in its default (disabled) initialization state for the transfers in oth directions.

```
ON/XOFF option:  DISABLED

nable or disable option (E or D):  <cr>
```

xample #3 UNIX Time-Sharing System

y reading the associated manuals, it was learned that the UNIX system used an ON/XOFF protocol for communicating with its terminals. Since the 9520 was onnected as a terminal, the XON/XOFF protocol was enabled for the text file ransfers in both directions.

```
ON/XOFF option:  DISABLED

nable or disable option (E or D):  E <cr>
```

splay J - Terminal Mode Options

he terminal mode options are selected to control HCOMM's actions when the 9520 connected to the other device as a terminal. These options are only in ffect when HCOMM is in the terminal mode. They have no effect during transfer ode

hen the automatic terminal mode option is enabled HCOMM immediately enters rminal mode after a transfer file name is entered (see display L or M). Then hen terminal mode is exited HCOMM is ready to begin the transfer immediately.

mple #1 Intellec Series II, Model 230, running ISIS-II, V3.4

ce the 9520 was not connected to the Intellec as a terminal none of the minal mode options were altered from their default initialization states.

```
tomatic terminal mode option:  DISABLED
rminal mode operation:  FULL-DUPLEX
Receive/echo filter" list:  NONE
erminal mode exit character:  1C
ransmitted exit sequence:  NONE

nable or disable automatic terminal mode option (E or D):  <cr>
ter new terminal mode operation (F or H):  <cr>
ter new "receive/echo filter" list
  (hex bytes or N if none):  <cr>
ter new exit character (hex byte):  <cr>
ter new transmitted exit sequence
  (hex bytes or N if none):  <cr>
```

ar le #2 8002 Microprocessor Lab running TEKDOS, V3.1, and CP/M, V2.2

ce the 9520 was not connected as a terminal and terminal mode was only used experimentation, none of the terminal mode options were altered from their fault initialization states for the transfers in either direction.

```
tomatic terminal mode option:  DISABLED
minal mode operation:  FULL-DUPLEX
ceive/echo filter" list:  NONE
minal mode exit character:  1C
nsmitted exit sequence:  NONE

ble or disable automatic terminal mode option (E or D):  <cr>
ter new terminal mode operation (F or H):  <cr>
ter new "receive/echo filter" list
  (hex bytes or N if none):  <cr>
ter new exit character (hex byte):  <cr>
ter new transmitted exit sequence
  (hex bytes or N if none):  <cr>
```

mple #3 UNIX Time-Sharing System

* a text file transfer from the 9520 to the UNIX system, HCOMM would not ...o respond quickly so the automatic terminal option was left in its default sabled) initialization state. The UNIX system operated with full-duplex nals so the operation option was also left with its default initialization. ...s surmised, and subsequently verified, that the UNIX system would send no

his ption is useful when HCOMM must be prepared to respond very quickly after eaving terminal mode (e.g. as in Example #3, transferring a text file to the 320). The terminal mode operation option determines whether or not HCOMM choes input characters locally while in terminal mode. It will echo locally hen HALF-DUPLEX is selected. When FULL-DUPLEX is selected, HCOMM assumes that he other device will echo all characters sent to it. The "receive/echo filter" s a list of up to 32 characters that will not be sent to the 9520's console uring terminal mode. This is important because many characters, notably con- ol characters (01-1F hex), can adversely alter the state of the 9520's con- ole. If any of the characters in the list are received they will be filtered ut. The "filter" also affects echoed characters, both locally echoed and those choed by the other device. Thus, it is possible to send control characters to he other device during terminal mode without affecting the 9520's console. The haracter used to exit terminal mode is operator selectable. It should be hosen so as not to conflict with characters that must be sent to the other evice during terminal mode. When this character is typed HCOMM returns to the ode it was in (either entry or transfer mode) when terminal mode was entered. he exit character is not sent to the other device. If it is necessary to send haracters to the other device when terminal mode is exited, they should be pecified in the transmitted exit sequence, which may contain up to 16 charac- ers. When the selected exit character is typed the transmitted exit sequence s sent to the other device just before terminal mode is actually exited.

---

erminal Mode Options

erminal mode may be entered: automatically, if the option is enabled, after he filename is entered in response to a transfer query; by responding to any uery with a CTL-T; or by striking CTL-T while in transfer mode. Loss of data y result if the last method is used while a transfer is actually occurring.

hen in terminal mode, characters struck at the 9520's console are transmitted thout interpretation, and those received are written to the 9520's console. he operation may be either full or half duplex. Also, it may be appropriate prevent certain received (and echoed) characters from being written. Such haracters, if any, must be specified in a "receive/echo filter" list.

hen the selected exit character is struck the selected exit sequence, if any, transmitted and normal mode resumes. The exit character should be selected as not to conflict with any character needed to control the other device.

utomatic terminal mode option: DISABLED
erminal mode operation: FULL-DUPLEX
"receive/echo filter" list: NONE
erminal mode exit character: 1C
ansmitted exit sequence: NONE

able or disable automatic terminal mode option (E or D): <cr>
ter new terminal mode operation (F or H): <cr>
ter new "receive/echo filter" list
(x bytes or N if none): <cr>
ter new exit character (hex byte): <cr>
te new transmitted exit sequence
(hex bytes or N if none): <cr>

---

acters that would adversely affect the 9520's console while HCOMM was being
ed in terminal mode. Therefore, the "receive/echo filter" list was left
pty. An EOT (04 hex) was selected as the terminal mode exit character since
is was more like the UNIX system usage than the default initialization exit
aracter. Finally, the transmitted exit sequence was left empty since none was
eeded for this transfer.

```
utomatic terminal mode option:  DISABLED
erminal mode operation:  FULL-DUPLEX
Receive/echo filter" list:  NONE
erminal mode exit character:  1C
 ansmitted exit sequence:  NONE

 able or disable automatic terminal mode option (E or D):  <cr>
 ter new terminal mode operation (F or H):  <cr>
nter new "receive/echo filter" list
  (hex bytes or N if none):  <cr>
nter new exit character (hex byte):  4 <cr>
nter new transmitted exit sequence
  (hex bytes or N if none):  <cr>
```

---

t was known that HCOMM would have to respond quickly when the UNIX system
tility was invoked to transfer the text file to the 9520. Therefore, the
t matic terminal mode option was enabled. The other options were left as they
d been previously selected, except for the transmitted exit sequence. A
rriage-return was selected for this option so that when the UNIX system util-
y invocation line had been typed, the exit character (EOT) could be typed to
use, nearly simultaneously, the execution of the utility and the exit from
rminal mode. In this manner, HCOMM would be quickly ready to receive the
rst characters sent from the UNIX system.

```
tomatic terminal mode option:  DISABLED
rminal mode operation:  FULL-DUPLEX
 eive/echo filter" list:  NONE
 minal mode exit character:  04
 nsmitted exit sequence:  NONE

 able or disable automatic terminal mode option (E or D):  E <cr>
 ter new terminal mode operation (F or H):  <cr>
 er new "receive/echo filter" list
  (hex bytes or N if none):  <cr>
 er new exit character (hex byte):  <cr>
 ter new transmitted exit sequence
  (hex bytes or N if none):  0D <cr>
```

---

is -y K - Saving and Loading Options Files

hen the options have been correctly selected for a certain type of transfer hey may be saved in a file on the 9520's diskettes. Subsequently, when using COMM for another transfer of the same type (i.e. to or from a particular evice), the file may be loaded, thus easing option selection. The option file ay be loaded with display K or by specifying it in the HCOMM invocation line.

---

ave or Load Selected Options To or From a File

ll of the options selected for: interface parameters, end-of-record sequence, nd-of-file sequence, abort sequence, ACK/NAK, XOFF/XON, and terminal mode may e saved in a file that can later be loaded into the File Transfer Utility.

n options file can be saved, after all desired options from other displays ave been selected, by responding to the queries below. It can then be loaded r specifying its name on the subsequent invocation line as:

```
FTU  options-file-name
```

in response to the queries below.

```
ave or load selected options to or from a file
  (S or L or blank line to return to menu): L <cr>
nter name of options file to be LOADED  (blank line will return to menu
ithout performing action):  <cr>
```

---

xample #1 Intellec Series II, Model 230, running ISIS-II, V3.4

fter the options were selected, as shown previously in this example, they were ved in an options file.

```
ave or load selected options to or from a file
  (S or L or blank line to return to menu):  S <cr>
nter name of options file to be SAVED  (blank line will return to menu
thout performing action):  FROMINT.OPT <cr>
```

---

xam) #2 8002 Microprocessor Lab running TEKDOS, V3.1, and CP/M, V2.2

rs) the options were selected for the TEKHEX format file transfer to the 02 s memory, as shown previously in this example. These options were saved in options file for subsequent use.

```
e or load selected options to or from a file
(S or L or blank line to return to menu): S <cr>
er name of options file to be SAVED (blank line will return to menu
thout performing action): HEXTOTEK.OPT <cr>
```

xt, different options were selected, as shown previously in this example, for e text file transfer from the 8002 to the 9520. These were saved in another tions file.

```
ve or load selected options to or from a file
(S or L or blank line to return to menu): S <cr>
ter name of options file to be SAVED (blank line will return to menu
thout performing action): FROMTEK.OPT <cr>
```

en, the first options file was loaded to prepare HCOMM for the TEKHEX format le transfer.

```
ve or load selected options to or from a file
(S or L or blank line to return to menu): L <cr>
te name of options file to be LOADED (blank line will return to menu
thout performing action): HEXTOTEK.OPT <cr>
```

mple #3 UNIX Time-Sharing System

options for the text file transfer from the 9520 to the UNIX system were cted, as shown previously in this example, then saved in an options file.

```
or load selected options to or from a file
S or L or blank line to return to menu): S <cr>
name of options file to be SAVED (blank line will return to menu
out performing action): TOUNIX.OPT <cr>
```

the previous options had been saved, the options for the text file transfrom the UNIX system to the 9520 were selected, as shown previously in this le, then saved in another options file.

```
or load selected options to or from a file
or L or blank line to return to menu): S <cr>
name of options file to be SAVED (blank line will return to menu
performing action): FROMUNIX.OPT <cr>
```

ransferring Data Files

COMM is in transfer mode when it is transferring files either to or from the ther device. As each logical record is detected by HCOMM (either end-of-record r full buffer) a period is written to the 9520's console. After a multiple of 4 periods have been written the total number is also displayed in decimal.

lso during a transfer, error messages may be displayed. Before any error essage is written the decimal number of logical records is written. Therefore, he number of the record in which the error occurred is one greater than the ecimal number preceding the message. There are three types of errors. Charac- er errors can occur on each character transferred. These are, for example, rrors in parity detection, framing errors, overrun errors, and interrupt buffer verflow. The number of the character, within the record, on which the error cc[redacted]red is also displayed with the error message. Another type, record errors, ccur in conjunction with a complete logical record. For example, the reception f an abort sequence, a NAK sequence, or the necessity of sending a NAK equence, are record errors. Finally, console overides, while not strictly rrors, are displayed in the same manner as errors.

console overide is a character struck at the 9520's console while HCOMM is in ransfer mode. All but three characters are ignored. A control-T will place COMM in terminal mode. This can be a dangerous procedure. If the other device s in the process of transferring necessary data to the 9520, these data will be en[redacted] to the 9520 console rather than written to the file. In other words, data an be lost. When terminal mode is exited HCOMM resumes the transfer. An scape struck while in transfer mode aborts the transfer. If a non-empty abort equence has been selected it is immediately sent to the other device. No ubsequent data are sent by HCOMM and all subsequently received data are dis- arded. The 9520 file is closed. If the transfer was from the other device, he data transferred prior to the abort will be in the file. A control-Z from he console forces HCOMM to act as if it had detected the end of the file. If he transfer is to the other device the end-of-file sequence is sent, provided [redacted] non-empty, and no more data are sent. If the transfer is from the other evice any further data are discarded. The 9520 file is closed. If the trans- er was from the other device, the data transferred prior to the forced end-of- ile will be in the file.

hen the transfer is terminated a message is written to the 9520 console. This essage contains the name of the 9520 file, the type of termination (either ompleted or aborted), and the number of characters and disk records in the 9520 ile. Another transfer may then be performed.

isplay L – Transfer Files from Disk to Interface

n this display the operator is queried for the name of a file. This is the le on the 9520 disk system that is read and sent to the other device. A blank ne in response to this query causes HCOMM to return to the menu. Only one le [redacted]y be sent at a time but after each transfer the operator is queried for nother.

T COMMUNICATIONS UTILITY

---

nsfer Files from Disk to Interface

en a file name is entered in response to the query below, transfer mode is tered, the file is opened, and (if the file exists) records are read from it d transferred to the interface. A period is written to the console after ch record is successfully transferred. Periodically, the number of records at have been transferred is displayed in decimal.

abnormal termination occurs when the abort sequence is detected from the terface or when the operator strikes ESC (1B hex) at the 9520's console. is halts the file transfer; and in the latter case, the abort sequence is ansferred to the interface, as well. Operator intervention at the other evice might be required if it is unable to respond to the abort sequence.

e transfer is normally terminated when the end of the file is encountered. he end-of-file sequence is then transferred to the interface and the file is osed. The operator may force this condition by striking CTL-Z (1A hex) at 9520's console; any part of the file remaining will not be transferred.

fter the transfer is terminated (normally or abnormally) the file name query s repeated. The operator may then name another file to be transferred.

nter name of file to be transferred to interface (blank line will return to nu without performing action): <cr>

---

xample #1 Intellec Series II, Model 230, running ISISII, V3.4

nce the transfer was only from the Intellec to the 9520, display L was not d in this example.

ample #2 8002 Microprocessor Lab

er the options for the TEKHEX format file transfer had been loaded the ansfer was performed. First, the following commands were entered at the 02's console:

```
> COMM E=L
COMM V3.0
<^@><cr>
```

COMM program is the TEKDOS utility for performing TEKHEX format transfers een the 8002's memory and other devices. Invoking it with the "E=L" argu- t meant that the other device (in this case the 9520) would not echo charac- and that they should be echoed locally by the 8002. The "<^@><cr>" command the method used to ready the 8002 to receive a TEKHEX format file to be e in its memory. Then, at the 9520's console, display L was selected and n me of the TEKHEX format file was entered in response to the transfer y.

```
nte. name of file to be transferred to interface   (blank line will return to
enu without performing action):  TEK.HEX <cr>
TU STATUS:  Transfer Mode
.....64
65

EK.HEX transfer complete - 2891 bytes transferred - 23 disk records
TU STATUS:  Entry Mode

nter name of file to be transferred to interface   (blank line will return to
enu without performing action):  <cr>
```

xample #3 UNIX Time-Sharing System

he text file transfer from the UNIX system was performed before this transfer. fter editing that file to remove the spurious blank line, HCOMM was invoked ith the command:

```
OA>HCOMM TOUNIX.OPT
```

his invocation initialized the options as they had been previously selected for he text file transfer from the 9520 to the UNIX system. Display L was selected nd control-T followed by a carriage-return was entered in response to the ansfer query. This placed HCOMM in terminal mode. Next, a carriage-return as struck to get the UNIX system prompt. Assured by the prompt that HCOMM was mmunicating properly with the UNIX system, the command line for the UNIX stem utility to accept input from the terminal (in this case the 9520) and ore it in a file was typed and entered with a carriage-return. After waiting few seconds to be sure that the utility was ready to accept characters, rminal mode was exited by striking the selected exit character (control-D). e transfer was then performed with the same file that had been previously ansferred from the UNIX system so that it could be compared with the original.

```
ter name of file to be transferred to interface   (blank line will return to
nu without performing action):  <^T><cr>
U STATUS:  Terminal Mode

P /dev/tty textfile

U STATUS:  Entry Mode
ter name of file to be transferred to interface   (blank line will return to
nu without performing action):  TEXTFILE <cr>
U STATUS:  Transfer Mode
.....64
.....78

XT  LE. transfer complete - 3040 bytes transferred - 24 disk records
U STATUS:  Entry Mode

ter name of file to be transferred to interface   (blank line will return to
nu without performing action):  <cr>
```

.splay M - Transfer Files from Interface to Disk

n this display the operator is queried for the name of a file. This is the .le that is created on the 9520 disk system and to which the data received from e other device are written. A blank line in response to this query causes COMM to return to the menu. Only one file may be received at a time but after ach transfer the operator is queried for another.

---

ansfer Files from Interface to Disk

en a file name is entered in response to the query below, transfer mode is tered, the file is created, and records are transferred from the interface henever it sends them) and written to the file. A period is written to the nsole after each record is successfully transferred. Periodically, the mber of records that have been transferred is displayed in decimal.

. abnormal termination occurs when the abort sequence is detected from the terface or when the operator strikes ESC (1B hex) at the 9520's console. bsequent transfers are discarded; and in the latter case, the abort sequence t nsferred to the interface. Operator intervention at the other device ght be required if it is unable to respond to the abort sequence.

e transfer is normally terminated when the end-of-file sequence is detected m the interface. The file is then closed. The operator may force this dition by striking CTL-Z (1A hex) at the 9520's console; any subsequent nsfers from the interface will be discarded.

er the transfer is terminated (normally or abnormally) the file name query repeated. The operator may then name another file to be transferred.

or name of file to be transferred from interface (blank line will return menu without performing action): <cr>

---

mple #1 Intellec Series II, Model 230, running ISIS-II, V3.4

ally, the transfer of the Intel hex format file from the Intellec to the 9520 performed. First, display M was selected and the name of the file to be ted on the 9520 entered in response to the transfer query. Then the follow- command was entered at the Intellec's console:

```
-COPY :F1:INTEL.HEX TO :TO:
```

command caused the named file to be copied to the Intellec's TTY port (and e, to the 9520). When the COPY completion message had been written at the llec's console it was known that the transfer was complete. At that time a l-Z was typed at the 9520's console to inform HCOMM that no more data

```
ust   be sent (i.e. forced an end-of-file condition since no end-of-file
sequence was sent by the Intellec).

-------------------------------------------------------------------------

Enter name of file to be transferred from interface  (blank line will return
to menu without performing action):  INTEL.HEX <cr>
FTU STATUS:  Transfer Mode
 ..............................................................64
 .............................................................128
 .............................................................192
 .............................................................256
 .............................................................320
 .............................................................384
 .............................................................448
 .............................................................512
 .............................................................576
 .............................................................640
 .............................................................704
 .............................................................768
 .............................................................832
 ..........................................................892
CONSOLE OVERIDE:  End-of-file forced

INTEL.HEX transfer complete - 39356 bytes transferred - 308 disk records
TU STATUS:  Entry Mode

nter name of file to be transferred from interface  (blank line will return
o menu without performing action):  <cr>

-------------------------------------------------------------------------
```

xample #2 8002 Microprocessor Lab running TEKDOS, V3.1, and CP/M, V2.2

hen the 8002 had been readied for the text file transfer to the 9520 (CP/M used
nstead of TEKDOS), HCOMM was invoked with the following command at the 9520's
onsole:

```
    0A>HCOMM FROMTEK.OPT
```

his invocation initialized the options with the options file that had been
reviously saved. Display M was then selected and the name of the file that was
o be created on the 9520 was entered in response to the transfer query. Then,
t the 8002's console, the following CP/M command was entered:

```
    A>PIP UP1:=TEXT.SRC,EOF:
```

his command caused the file "TEXT.SRC" to be copied to the device named "UP1:",
hich was the 8002's remote I/O port. The "EOF:" following the filename meant
hat a SUB (1A hex) would be appended to the file when it was copied. This
haracter was used to signal the end of the transfer to HCOMM.

T COMMUNICATIONS UTILITY

```
er name of file to be transferred from interface   (blank line will return
menu without performing action):  TEXT.SRC <cr>
 STATUS:  Transfer Mode
.....64
.....128
.....192
.....256
.....320
.....384
.....394

T.SRC transfer complete - 13924 bytes transferred - 109 disk records
 STATUS:  Entry Mode

ter name of file to be transferred from interface   (blank line will retur
menu without performing action):  <cr>
```

mple #3 UNIX Time-Sharing System

options were correct for the transfer of a text file from the UNIX system to 9520 and this transfer was performed first. Display M was selected and the e of the file that was to be created on the 9520 system was entered in se to the transfer query. HCOMM replied by automatically entering termi- mode. Next, a carriage-return was struck to set the UNIX system prompt. ured by the prompt that HCOMM was communicating properly with the UNIX sys- the command line for the UNIX system utility that wrote text files to a minal was typed. Instead of terminating the command line with a carriage- rn, however, the exit character (control-D) was struck. This caused the smitted exit sequence (carriage-return) to be sent to the UNIX system, eby causing the command line to be executed. At the same time (nearly) inal mode was exited and HCOMM was immediately ready to accept characters the UNIX system. When it was judged that the transfer was finished (HCOMM ed writing data), a control-Z was struck at the 9320's console. It was ssary to force an end-of-file condition because the UNIX system utility that used sent no character or sequence of characters to signal the end of the sfer.

```
nter name of file to be transferred from interface   (blank line will return
o menu without performing action):  TEXTFILE <cr>
TU STATUS:  Terminal Mode

cp textfile /dev/tty
TU STATUS:  Transfer Mode
.....64
.....79
ONSOLE OVERRIDE:  End-of-file forced

EXTFILE. transfer complete - 3042 bytes transferred - 24 disk records
TU STATUS:  Entry Mode

nter name of file to be transferred from interface   (blank line will return
o menu without performing action):  <cr>
```

ecause the transmitted exit sequence (carriage-return) had been echoed by the NIX system (as carriage-return/line-feed) after HCOMM had exited terminal mode, he file on the 9520 contained a first, blank line that the file on the UNIX ystem did not contain. HCOMM was exited and a text editor was used to remove he blank line from the 9520 file to make it identical to the UNIX system file.

se of the Quiet Option

he transfer of the text file from the UNIX system to the 9520 was repeated as n example of the use of the quiet option. After the necessary connections to he modem were made, HCOMM was invoked with the following command:

```
DA>HCOMM FROMUNIX.OPT
```

his initialized the options from the contents of the previously saved options ile for this transfer. The rest of the session is shown in its entirety below.

COMMUNICATIONS UTILITY

---

e Transfer Utility - Asynchronous Communications Interface - Version 1.0

u:

A	General information regarding protocol
B	RS-232 link information
C	DTR/DSR information
D	Interface parameters options
E	End-of-record sequence option
F	End-of-file sequence option
G	Abort sequence option
H	ACK/NAK options
I	XON/XOFF option
J	Terminal mode options
K	Save or load selected options to or from a file
L	Transfer files from interface to disk
M	Transfer files from interface to disk
N	Display this menu
O	Exit file transfer utility

response to any query, the new options (blank or comma separated) must be
tered with a carriage-return. A blank line leaves the options unaltered.
triking CTL-C as the first character returns control to the operating system.

ter desired menu item letter (Q <letter> - skips informative text): Q M <cr>

ter name of file to be transferred from interface (blank line will return
menu without performing action): TEXTFILE <cr>
J STATUS: Terminal Mode

= textfile /dev/tty
J STATUS: Transfer Mode
.....64
.....79
SOLE OVERRIDE: End-of-file forced

TFILE. transfer complete - 3042 bytes transferred - 24 disk records
STATUS: Entry Mode

er name of file to be transferred from interface (blank line will return
menu without performing action): <cr>

: Item N displays the full menu

r desired menu item letter (Q <letter> - skips informative text): O <cr>

Applications Note UK1.

Interfacing a Data I/O Prom Programmer to the 9520

May 1982

Introduction.

One of the major requirements in any microprocessor development laboratory is the ability of the user to program EPROMs and the like with prototype software. In the past, the preferred way of interfacing a prom programmer to Millennium equipment was via the Comm Link port of the 9508, using the various terminal mode options. At the time of writing, the communications protocols and options supported by the 9508 are fairly limited but adequate for most prom programmer requirements (e.g. Data I/O, Pro-log). Now that Millennium have entered the 16-bit emulation market, some other means of communicating with programmers is required, the 9516 having totally different characteristics to those of the 9508. This applications note is intended to demonstrate the ease with which a 9520 may be connected to a Data I/O prom programmer so that user programs may be easily 'blown'. It should also give a better understanding on the use of the file transfer utility (called COMM.COM or FTU.COM) when interfacing the 9520 to foreign hardware.

The Programmer

This example uses a Data I/O model 20B, although the communications setup is the same for any of their devices. It contains an amount of RAM which is used for intermediate storage of user code, which may be loaded from a device in the prom socket or from the communications link. The programmer provides a set of simple commands which can be used for the remote editing of the contents of the RAM, and for setting up various options relating to the type of device in the prom socket. The communications information given in the 20B manual is as follows:-

Connector pin assignments

Pin.No.	Signal	Description.
1	GROUND	In the RS232C environment this line is common for the -12V source and provides a safety ground connection to the RS232 compatible terminal.
2	Send Data	Transmits data.
3	Receive Data	Accepts data.
4	Request to send	This line is normally held high by the programmer. It is dropped to inhibit data transmission from the peripheral.
5	Clear to send.	A high level on this line allows the programmer to send data. A low level inhibits data transmission.
6		Not used.
7	Signal Ground	Provides a common signal connection.
8-25		Not used.

Serial I/O option switches give the choice of baud rates of 9600,4800,2400,1200,600,300 and 110, one or two stop bits and either odd, even or no parity.

Commands to the programmer are always terminated by a carriage return, and following completion of the command the programmer will reply with one of

four prompts. These are as follows:-

Response	Issued in case of
>	1.Entering remote control mode. 2.After successful command execution. 3.After a command has been aborted with the ESC character.
F	After failure to execute a valid command
?	After receiving an invalid command.
E	When a PROM is not installed correctly or the incorrect Prom code was selected.

For a full explanation of the commands the reader should consult the appropriate manual.

The 9520 Communications package.

The File Transfer Utility is a Millennium supplied program for the transfer of ASCII information between the 9520 and other intelligent devices. It supports a number of different communications protocols (ACK/NAK, XON/XOFF etc.) and gives the user sufficient flexibility and control over the Printer interface of the 9520 to make interfacing relatively simple. The program is entirely menu driven and explains each of the available options as it is invoked from the main menu. The following discussion of the various options available in the File Transfer Utility explains how to interface to a Data I/O PROM programmer and should provide sufficient understanding to enable the user to 'hook-up'to any foreign hardware.

The Interface.

In the following the prompts and comments typed by the FTU package will be printed in **bold type** for the sake of clarity.

The file transfer utility main menu contains 15 options, each indexed by a single letter as follows:-

A General information regarding protocol
B RS-232 link information
C DTR/DSR information
D Interface parameters options
E End-of-record sequence option
F End-of-file sequence option
G Abort sequence option
H ACK/NAK options
I XON/XOFF option
J Terminal mode options
K Save or load selected options to or from a file
L Transfer files from disk to interface
M Transfer files from interface to disk
N Display this menu
O Exit file transfer utility

It is recommended that the user first inspects each description displayed when an option letter is typed on the console followed by a carriage return. Once the user is clear what task each menu option performs, interfacing to foreign hardware is relatively simple. The steps to be considered are as follows:-

Option 'B'.

RS-232 Link Information

The interface to the 9520 involves an RS-232 type link. The pins connected,

the lines assigned to them, the directions of the signals, and their uses are indicated below:

```
 1  -       Protective Ground       (common)
 2  - TXD / Transmitted Data       (to 9520) - data transmitted from other device
 3  - RXD / Received Data          (from 9520) - data received by other device
 4  - RTS / Request to Send        (to 9520) - ignored
 5  - CTS / Clear to Send          (from 9520) - same as DSR
 6  - DSR / Data Set Ready         (from 9520)  -controls other device transmission
 7  -       Signal Ground          (common)
 8  - RLSD/ Rec. Line Sig. Det.    (from 9520) - held marking (ON)
20 - DTR / Data Terminal Ready     (to 9520) - controls 9520 transmission
```

For more information on the "control" lines, see DTR/DSR Information.

The following is an extract from the explanation of how the 9520 handles the hardware protocols, which is displayed when option 'C' (DTR/DSR Information) is chosen.

The 9520 will not transmit unless the other device indicates that it is ready to receive by holding the DTR line in an active state. Similarly, the 9520 will hold the DSR line in an active state only when it is ready to receive. Thus, these lines provide synchronization and speed controls if the other device is capable of managing and responding to them. If the other device cannot manage and respond to these lines, synchronization and speed controls, if needed, must be accomplished with either an ACK/NAK or XON/XOFF protocol, or both. Furthermore, the DTR line must be held in an active state. This is done automatically in the 9520 hardware if the other device is not connected to the DTR line. If the other device is connected, however, it must be responsible for holding the DTR line in an active state.

The above information gives the necessary explanation of signal line usage by the FTU for interfacing to another device. In the case of the Data I/O prom programmer it is fairly easy to determine that a cable should be made up connecting the devices as follows:-

9520		Model 20A
1	-	1
2	-	2
3	-	3
5	-	5
7	-	7
20	-	4

Note. The 9520 ignores Pin 4, so the cable can be constructed with a 4/20, 20/4 'crossover' connection so that either end may be connected to either device.

Having established the correct cabling connections between the 9520 and the other device, it will then be necessary to initialise the DART in the 9520 so that data sent by the 9520 may be 'understood' by the receiving device and vice versa. Option 'D' allows the user to select the speed and format of the data bytes.

Option D.

Interface Parameters Options

The interface may have one of the appropriate options, listed below, selected for each of its four parameters.

Data: **A - 7 bits** **B - 8 bits**

Parity: **C - None** **D - Odd** **F - Mark (7-bit data size only)**

E - Even G - Space (7-bit data size only)

Stop Bits: H - 1 I - 2

Baud Rate:
J - 110 N - 2400
K - 300 O - 4800
L - 600 P - 9600
M - 1200 Q - 19200

Interface Parameters Options:

A list of the current options is now displayed. To change the baud rate or some other parameter, simply type the letter corresponding to the option required after the prompt. For the Data I/O prom programmer these options must be selected to correspond with the switch settings on the RS232 board of the programmer itself.

Options E and F.

These two options are used for determining the format of the data. Option E informs the utility what character sequence normally terminates a line of ASCII text on the foreign hardware so that this can be converted into CR-LF for storage in 9520 disc files. Most systems use a CR-LF sequence to terminate records so this option will rarely be used.

The F option on the other hand informs the utility of the character sequence used to terminate a file. A Data I/O programmer does not use any characters to indicate specifically the completion of a transfer, it merely responds with its normal prompt '>' when the transfer is complete. The FTU should therefore be set up to acknowledge '>' as meaning end-of-file.

There is a small problem with Data I/O Model 20 programmers (and possibly others in the range) with using '>' to denote the end-of-file condition. When sending Tektronix type records, the programmer sends the end-of-file record without an end-of-record sequence (CR-LF) and follows it immediately with the '>' prompt i.e./00000000>. The FTU will disregard the end-of-file record in this case and it will have to be replaced by some means after exitting the utility. A version of FTU which avoids this problem is available from Millennium in England.

The following three options are not used when interfacing to the Data I/O devices, but for the sake of completeness the descriptions given by the FTU are included here.

Option G.

Abort Sequence Option.

The abort sequence is the sequence of characters that signals abnormal terminations of the transferred files. It halts the file transfer and causes any subsequently transferred characters to be discarded. The sequence used by the other device, if any, must be made known to the File Transfer Utility so that it can be transferred and detected correctly.

Option H.

ACK/NAK Options

The file transfer protocol may prescribe a response from the receiver after each record is transferred. That response is either an acknowledgement (ACK) or a request for retransfer (NAK). There are no timeouts in the protocol, so it is also possible for the receiver to control transfer speed by witholding the response until ready for the next record.

Despite the use of ASCII standard names, ACK and NAK are actually operator selectable sequences of characters. The use of ACK/NAK does not conflict with other protocol options unless the selected ACK/NAK characters are the same as some other significant protocol characters (XON, XOFF, end-of-record

sequence, end-of-file sequence, or abort sequence).

Option I

XON/XOFF Option

The file transfer protocol may allow the receiver to temporarily suspend the transfer by sending an XOFF (13 hex) to the transmitter. To resume the transfer the receiver sends an XON (11 hex).

The transmitter is presumed to be able to respond quickly (i.e. within two character times) after receiving an XOFF. Similarly, the receiver is presumed to be able to accept up to two additional characters after sending an XOFF.

Option J.

Terminal Mode Options

Terminal mode may be entered: automatically, if the option is enabled, after the filename is entered in response to a transfer query; by responding to any query with a CTL-T; or by striking CTL-T while in transfer mode. Loss of data may result if the last method is used while a transfer is actually occurring. When in terminal mode, characters struck at the 9520s console are transmitted without interpretation, and those received are written to the 9520s console. The operation may be either full or half duplex. Also, it may be appropriate to prevent certain received (and echoed) characters from being written. Such characters, if any, must be specified in a "receive/echo filter" list.

When the selected exit character is struck the selected exit sequence, if any, is transmitted and normal mode resumes. The exit character should be selected so as not to conflict with any character needed to control the other device.

For transfers to and from a device like the Data I/O programmers, The automatic terminal mode option should be enabled. Communication should be at half duplex, and the recommended exit character is ctrl-\ (1C hex) unless the user is communicating with the 9520 via a 9508, in which case some other control character must be used. The user should also set the transmitted exit sequence to 0D-0A. The receive echo filter list is not important in this application.

Option K.

Save or Load Selected Options To or From a File

This option is really self explanatory, all the user configurable options provided by the file transfer utility may be saved or loaded from a 9520 disk file.

Options L and M. File Transfers.

Transfer Files from Disk to Interface

When a file name is entered in response to the query below, transfer mode is entered, the file is opened, and (if the file exists) records are read from it and transferred to the interface. A period is written to the console after each record is successfully transferred. Periodically, the number of records that have been transferred is displayed in decimal.

An abnormal termination occurs when the abort sequence is detected from the interface or when the operator strikes ESC (1B hex) at the 9520s console. This halts the file transfer; and in the latter case, the abort sequence is transferred to the interface, as well. Operator intervention at the other device might be required if it is unable to respond to the abort sequence.

The transfer is normally terminated when the end of the file is encountered. The end-of-file sequence is then transferred to the interface and the file is closed. The operator may force this condition by striking CTL-Z (1A

hex) at the 9520s console; any part of the file remaining will not be transferred. After the transfer is terminated (normally or abnormally) the file name query is repeated. The operator may then name another file to be transferred.

Transfer Files from Interface to Disk

When a file name is entered in response to the query below, transfer mode is entered, the file is created, and records are transferred from the interface (whenever it sends them) and written to the file. A period is written to the console after each record is successfully transferred. Periodically, the number of records that have been transferred is displayed in decimal.

An abnormal termination occurs when the abort sequence is detected from the interface or when the operator strikes ESC (1B hex) at the 9520s console.

Subsequent transfers are discarded; and in the latter case, the abort sequence is transferred to the interface. Operator intervention at the other device might be required if it is unable to respond to the abort sequence.

The transfer is normally terminated when the end-of-file sequence is detected from the interface. The file is then closed. The operator may force this condition by striking CTL-Z (1A hex) at the 9520s console; any subsequent transfers from the interface will be discarded. After the transfer is terminated (normally or abnormally) the file name query is repeated. The operator may then name another file to be transferred.

Once the above options have been correctly set, the user need only set the programmer in remote control mode, and enter either the L or M options. After typing a file name in response to the query, the FTU will go into terminal mode, giving the user direct access to the programmer. Using the relevant commands the user can then set up the internal RAM of the programmer for uploading or downloading PROM contents. Terminating the programmer's 'O' command (if uploading to the 9520) or the 'I' command (in the case of downloading prior to programming a PROM) with a ctrl-\ will place the FTU into transfer mode and initiate the appropriate data transfer.

As you can see from the above example, interfacing to other devices via RS232 has been made considerably easier on the 9520 using the FTU package, making the 9520 extremely flexible in laboratory environments. This package should not be under-rated.

In the unlikely event that you have any problems with FTU please contact Steve Dearden at the Millennium UK office.