

RSX-180/280

KERMIT

VERSION 4.22

USER'S GUIDE

December 2021.

Table of Contents

1 RSX180/280 Kermit.....	3
1.1 Credits.....	3
1.2 License.....	4
1.3 Differences with Kermit-80.....	4
2 Overview of Kermit Operation.....	6
2.1 Kermit-180 Description.....	6
2.2 Local and Remote Operation.....	7
2.3 Kermit-180 Commands.....	7
3 Assembling Kermit-180 from the sources.....	16
3.1 Future Work.....	16

1 RSX180/280 Kermit

Program: Hector Peraza, see also Credits below
Language: Z80 Assembler, ZSM4
OS: RSX180 and RSX280
Version: 4.22
Date: December 4, 2021

Kermit-180 Capabilities at a glance:

Local operation:	Yes
Remote operation:	Yes
Login scripts:	Yes, limited
Transfer text files:	Yes
Transfer binary files:	Yes
Wildcard send:	Yes
File transfer interruption:	Yes
Filename collision avoidance:	Yes
Can time out:	Yes
8th-bit prefixing:	Yes
Repeat count prefixing:	No
Alternate block checks:	Yes
Terminal emulation:	Yes, limited
Communication settings:	If supported by the terminal driver
Support for dial-out modems:	No
Transmit BREAK:	If supported by the terminal driver
IBM communication:	Yes
Transaction logging:	No
Debug logging:	No
Session logging:	Yes
Raw file transmit:	Yes
Act as server:	Yes
Talk to server:	Yes
Advanced commands for servers:	Yes
Command/init files:	Yes
Command macros:	No
Local file management:	Yes
Handle file attributes:	No
Long packets:	No
International Character Sets:	No
Sliding Windows:	No
Printer control:	No

1.1 Credits

Kermit-180 is derivated from CP/M Kermit which, in turn, is the first of all the Kermit programs. CP/M Kermit was originally written by Bill Catchings of Columbia University in 1981. Over the years, contributions have been added by many people, including Charles Carvalho (ACC), Bernie Eiben (DEC), Nick Bush (Stevens Institute of Technology), John Bray (University of Tennessee), Bruce Tanner (Cerritos College), Greg Small (University of California at Berkeley), Kimmo Laaksonen (Helsinki University of Technology), Bertil Schou (Loughborough University), Jon Warbrick (Plymouth Polytechnic University), Brian Robertson (Aberdeen University), A.J. Cole (Leeds University),

John Shearwood (Birmingham University), Tony Addyman (Salford University), Godfrey Nix and Martin Carter (Nottingham University), Ian Young (Edinburgh University), Chris Miles (Manchester University), Richard Russell, Dave Roberts, and many, many others.

The latest known CP/M Kermit Version is 4.11, which is the work of Mike Freeman of the Bonneville Power Administration in Vancouver, WA, USA, with assistance from Russell Lang of Monash University in Australia, Jay S Rouman of Mt Pleasant MI, and others.

The current Kermit-180 Version 4.22 is the work of Hector Peraza.

This document is based on the original Kermit-80 documentation written by C. Gianone, Columbia University Center for Computing Activities New York, NY 10027, on April 23, 1991.

1.2 License

On July 1st 2011, Columbia University officially terminated the Kermit project and released all Kermit Software as Open Source under a Revised-3 Clause BSD License:

Copyright 2011, Trustees of Columbia University in the City of New York.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- + Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- + Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- + Neither the name of Columbia University nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For more information visit <http://www.columbia.edu/kermit/licensing.html>

1.3 Differences with Kermit-80

Kermit-180 is compatible with Kermit-80 at command level, except for a few exceptions:

- SERVER mode is supported.
- SET DIRECTORY-LISTING {LONG/BRIEF} replaces SET DIRECTORY-FILE-SIZE {ON/OFF} command. The long directory listing displays file sizes, attributes and creation date using a format similar to RSX180/280's PIP command.

- SET COLLISION {DISCARD/OVERWRITE/NEW-VERSION} The BACKUP option was removed, since RSX180/280 automatically creates a new version of an existing file, keeping the old one. The RENAME option was changed to NEW-VERSION and makes sense only when transferring files from a system that supports file versions (e.g. RSX-11M, VMS, or another RSX180 or RSX280 system) and the file being transferred has the same name, type and version number as a file on the local machine.
- CD command to change current device and directory, replaces CP/M's SET DEFAULT-DISK and SET USER commands.
- The current device and directory are displayed in the prompt.
- Argument to SET BUFFER-SIZE command is the number of 512-byte blocks, instead of 128-byte sectors.
- SET LINE added as synonym for SET PORT for those used to Kermit-11.
- DELETE added as synonym for the ERASE command.
- Added @ as synonym for TAKE command, for compatibility with RSX180/280 conventions.
- TAB auto-completes keyword like ESC does (for compatibility with modern Linux/Windows conventions.)
- Precise PAUSE timing using the system timer services.

2 Overview of Kermit Operation

Use the SET command to establish necessary communication parameters like PORT, SPEED, and PARITY. Use the CONNECT to establish a terminal connection to the remote computer. If you are dialing out with a modem, type the necessary dialing commands to the modem first. The dialing process can be automated to some extent using a TAKE command file containing INPUT, OUTPUT, and PAUSE commands. Then log in to the remote computer or service and conduct a session.

To transfer a text file, start the Kermit program on the remote computer and tell it to SEND the desired file (if uploading) or to RECEIVE (if downloading). "Escape back" to Kermit-180, usually by typing Ctrl-\ C (hold down the Control key and press the backslash key, and then type the letter C.) At the Kermit-180 prompt type RECEIVE (if you gave a SEND command to the remote Kermit) or SEND filename (if you gave a receive command to the remote Kermit).

To transfer a binary file, give the command SET FILE TYPE BINARY to the remote Kermit and SET FILE-MODE BINARY to Kermit-180 before issuing any SEND or RECEIVE commands.

Multiple files of the same type (text or binary) can be transferred in a single operation using "wildcard notation" (including special characters like asterisk in the filename).

When file transfer is complete, CONNECT back to the remote computer, use the EXIT command to exit from the remote Kermit program, finish your work on the remote computer, log out from it, escape back to Kermit-180 again, and EXIT from Kermit-180.

The remote Kermit may also be put into "server mode" to simplify these operations. Give the SERVER command to the remote Kermit, escape back to Kermit-180, and then issue SEND commands to send files (upload), GET filename commands to receive (download) files, REMOTE commands to request various other services (like directory listings) from the remote Kermit. When you are done, give a BYE command to terminate your remote session, or a FINISH command to tell the remote Kermit to return to its prompt so you can CONNECT back and conduct further business.

The opposite is also possible: Kermit-180 may also be put into "server mode" and GET and SEND commands issued from the remote Kermit. This simplifies operation with remote computers whose Kermit does not implement the SERVER command.

That's all there is to it.

2.1 Kermit-180 Description

Kermit-180 is able to run in both "remote" and "local" modes. It includes a terminal emulator for establishing a connection to a remote computer or service, and during file transfer in "local" mode it keeps the screen updated with the file name and the packet number, whether sending or receiving.

Kermit-180 is capable of precise timeouts on an input request, and can break deadlocks automatically. If, despite the timeout capability, the transmission appears to be stuck (and you can tell that this has happened if the screen fails to change for a while) you can type carriage return to have the micro do what it would have done on a timeout, namely NAK the expected packet to cause the foreign host to send it again (or, if the micro is sending, to retransmit the last packet). Micro/micro or micro/IBM-mainframe transfers could require this kind of manual intervention.

File transfers may be interrupted in several ways.

- | | |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Control-C | This will return you to Kermit-180 command level immediately, so that you can connect back to the remote system, or take any other desired action. |
| Control-X | When sending a file, this will terminate the sending of the current file with a signal to the Kermit on the other side to discard what it got so far. If there are more files to be sent, Kermit-180 will go |

on to the next one. When receiving a file, Kermit-180 will send a signal to the remote Kermit to stop sending this file. If the remote Kermit understands this signal (not all implementations of Kermit do), it will comply, otherwise the file will keep coming. In any case, the remote Kermit will go on to the next file in the group, if any.

- Control-Z** Like Control-X, except if a file group is being transmitted, this will stop the transmission of the entire group. If only a single file is being transmitted, it works exactly like Control-X.
- Carriage Return** If you type a carriage return Kermit-180 will resend the current packet. You may do this repeatedly, up to the packet retry limit (somewhere between 5 and 16 times) for a particular packet.

2.2 Local and Remote Operation

Kermit-180 by default assumes that all file transfers will occur over the terminal line that you are currently logged in on (TI:). This is known as REMOTE mode (the RSX180/280 is the remote system). This would be the desired case if you are running Kermit on a microcomputer such as a PC and are currently logged into the RSX180/280 system through the micro. However, if you wanted to dial out, say by an autodial modem, from RSX180/280 to another system, then you need to tell Kermit-180 to use some other terminal line. This would be called LOCAL mode (the RSX180/280 is the local system). The line can be altered with the SET PORT or SET LINE command (see section on SET and CONNECT).

On RSX180/280, privilege is required to alter settings on any other terminal line unless the line is set Public via the MCR SET /PUB command.

2.3 Kermit-180 Commands

Kermit-180 is an interactive program. It issues a prompt, you type a command. The process repeats until you give the EXIT command to leave the program.

Commands consist of keywords, filenames, and numbers. Keywords may be abbreviated to minimum unique length. "?" may be typed to request a menu of the available options for the current field at any point in a command. ESC may be typed at any point in a command to fill out the current keyword or filename; if sufficient characters have not been typed to identify the current field uniquely, Kermit-180 will sound a beep and allow you to continue from that point. Here are Kermit-180's commands:

- @filespec** Take characters and commands from the specified file as if they were entered from the keyboard. See the TAKE command for more details.
- BREAK** Send a BREAK condition to the remote computer. This is only possible if the terminal driver is capable of sending breaks. It is intended to be used with PAUSE, OUTPUT, etc and the TAKE command to do weird and wonderful things, like automatic logging on to a remote host.
- BYE** When talking to a remote Kermit Server, this command shuts down the server and logs it out, and also exits from Kermit-180 to RSX180/280 command level.

CD device:[directory]

This allows you to set the default device and unit and/or the directory as destination of file transfers. In addition, issuing this command causes Kermit-180 to switch to the specified device and directory. The colon must be included in the device name, and the directory may be enclosed in brackets. The selected device and directory appear in your Kermit-180 prompt, for instance

```
Kermit-180 SY0:[USER]>
```

- CONNECT** Establish a terminal connection to the computer, service, or device that is connected to the serial port, i.e. pass all typing to the serial port and display all input from the serial port on the screen.

Also, emulate a DEC VT52 to allow cursor control, screen clearing, etc., if VT52-EMULATION is ON (see below), in which case you should also set your terminal type on the remote host to VT52. The CONNECT command may be abbreviated by the single letter C.

Warning: VT52 emulation is only successful if your system or its attached terminal can do the same sort of functions as a genuine VT52. Things to beware of are cursor addressing, clear to end of page and end of line, clear screen, home cursor, and clear-and-home functions. The usability of VT52 emulation depends entirely on how many of the VT52 functions can be emulated by your micro or terminal.

The escape character differs from micro to micro; when you issue the CONNECT command, the micro will print a message telling you how to get back. The escape sequence is generally an uncommonly-used control character, like CTRL-backslash or CTRL-rightbracket, followed by a single letter "command":

- C Close Connection, return to Kermit-180 command level.
- S Display Status of connection, but maintain remote connection.
- ? List available single-character commands.
- 0 (zero) Send a null (0) character.
- B Send a BREAK signal. Most systems provide this function.
- P Toggle printer on or off. Allows you to copy whatever goes to the screen to the printer.
- S Temporarily suspend logging to the log file.
- Q Restart logging to the log file
- ^ (or whatever – a second copy of the escape character) Send the escape character itself to the remote host.

COPY source destination

Copy a named file to another file, either on the same drive or another drive.

DELETE *filespec* This command erases the specified file(s). The names of the files being erased are not displayed. This is the same as the ERASE command.

DIRECTORY This provides a directory listing of the specified files. If no files are specified, all files on the default disk are listed. File sizes in bytes, file attributes and file creation date are included. You may interrupt the listing at any time by typing Ctrl-C. The listing concludes with a display of the amount of free storage left on the disk. You can inhibit the display of file attributes by SET DIRECTORY-LISTING BRIEF.

ERASE *filespec* This command erases the specified file(s). The names of the files being erased are not displayed. This is the same as the ERASE command.

EXIT Quit back to the system. QUIT is a synonym for EXIT.

FINISH Like LOGOUT, but shuts down the remote server without logging it out. Leaves you at Kermit-180 command level; subsequent CONNECT commands will put you back at host system command level.

GET filespec [local_filespec]

When Kermit-180 is talking to a Kermit Server on the host, you should use the GET command to request the server to send files to you, for example:

```
get hlp:k*.hlp
```


You may specify a local filename if you want to save the remote file under a different filename. Limitation: If you request an alternate block check type using the SET BLOCK command, the GET command will not communicate it to the remote server. If you want to have type 2 or 3 block checks done when getting files from the server, you have to issue the appropriate SET BLOCK command to the remote Kermit before putting it in server mode.

HELP List all these commands, with a short description on what the commands do. A question mark will do the same. If you have already typed a command but do not know what the parameters are, type a space (to indicate the end of the command) and a question mark. You will be informed of what Kermit can expect at that stage.

INPUT *seconds text*

Setup a text line and time delay for the Kermit-180 program to expect from the host, then wait up to the given number of seconds for text to be sent to your RSX180/280 system.

LOG *filespec*

When CONNECTed to a foreign host as a terminal, log the terminal session to the specified diskette file. This functionality depends to some extent on the remote host's ability to do XON/XOFF flow control, and does not guarantee a complete transcript (after all, that's what the Kermit protocol is for). The log file is closed when the connection is closed by typing the escape character followed by the single-character command "C".

It is possible to temporarily suspend logging during connect state. Typing an escape sequence can turn file logging on (<escape-character> R for Resume) or off (<escape-character> Q for quiet).

Re-entering connect state will re-open the previously opened log file and append to that file.

LOGOUT

Like BYE, but leaves you at Kermit-180 command level.

OUTPUT *text*

Send the text to the remote computer (or modem) as if you had typed it. This simply copies the string to the correct line, and assumes all appropriate parameters have been set to be used, e.g. speed, parity etc. It is intended for use in TAKE command files.

PAUSE *seconds*

If this command is issued, the Kermit-180 program will wait for the given number of seconds before proceeding with another command. This is intended for use in TAKE commands, where you may want to pause for a while before proceeding with the rest of the TAKE file.

PRINT

Print a file to the console and printer. Output to the printer is buffered by the Kermit-maintained printer buffer. This routine is identical to TYPE but characters are echoed to the printer as well as to the screen. Suspending and canceling output is as described in TYPE.

QUIT

Synonym for EXIT.

RECEIVE *filespec*

Receive file(s) from the remote Kermit, and save them under the names provided in the file headers supplied by the remote host. If a local filespec is given, the file is saved under the given filename. If the names aren't legal, use as many legal characters from the name as possible (see the description of SET FILE-WARNING below). If there's a conflict, and FILE-WARNING is ON, warn the user and try to build a unique name for the file by adding "&" characters to the name. RECEIVE can be abbreviated to the single letter R.

REMOTE *command*

Send a command to a remote Kermit server. The results are sent back to your screen. When two arguments are required and specify less than two in the command, you will be prompted for the missing arguments. REMOTE commands include:

REMOTE CD [*directory*]

Ask the remote server to change its default directory. If no directory is specified, the server changes to its login directory.

REMOTE COPY *file1 file2*

Ask the remote server to copy *file1* to *file2*.

REMOTE RENAME *file1 file2*

Ask the remote server to rename *file1* to *file2*.

REMOTE DELETE *filespec*

Ask the remote server to delete the named file or files.

REMOTE DIRECTORY [*filespec*]

Ask the remote server to display a directory listing of the given files or, if the *filespec* is omitted, all the files in the current device or directory.

REMOTE DISK-USAGE

Ask the remote server to display information about its disk usage (such as free or used space).

REMOTE ERASE *filespec*

Same as REMOTE DELETE.

REMOTE FINISH

Same as FINISH.

REMOTE HELP

Ask the remote server to display a list of the commands it can respond to.

REMOTE HOST *command*

Ask the remote server to have its operating system execute the given command.

REMOTE KERMIT *command*

Ask the remote server to execute the given Kermit command, given in the server Kermit's command syntax.

REMOTE LOGIN *user password*

Log in to a remote Kermit server which has been set up to require a username and password.

REMOTE MESSAGE *text*

Send the text to the remote server for display on its screen (useful with MS-DOS Kermit servers).

REMOTE SET *parameter value*

Ask the remote server to set the given parameter to the given value, for example REMOTE SET FILE TYPE BINARY. Type REMOTE SET ? to see a list of the REMOTE SET options.

REMOTE SPACE

Same as REMOTE DISK-USAGE.

REMOTE STATUS

Ask the remote server to provide a status report.

REMOTE TYPE *file*

Ask the remote server to display the named file on the micro's screen.

REMOTE WHO [*user*]

Ask the remote server for a list of users who are logged in, or if a user is specified, for a report on the named user.

RENAME *file1 file2*

Rename local RSX180/280 *file1* to *file2*.

SEND *filespec*

Send file(s) specified by *filespec* to the remote Kermit. The filespec may contain RSX180/280 wildcards. SEND may be abbreviated to the single letter S.

SERVER

Puts a remote Kermit-180 in "server mode", so that it receives all further commands in packets from the local Kermit. The Kermit-180 server is capable (as of this writing) of executing the remote server commands listed below. Any nonstandard parameters should be selected with SET commands before putting Kermit-180 into server mode, in particular the file type.

SEND *filespec*

Send file(s) specified by *filespec* to the remote Kermit-180 server.

GET *filespec [local_filespec]*

Requests the remote Kermit-180 server to send files to the Local machine.

FINISH

Tells the remote Kermit server to exit without logging out of the remote system. You can then CONNECT back to the remote Kermit program.

REMOTE DIRECTORY [*filespec*]

The names of the files that match the given file specification will be displayed on your screen, perhaps along with size and date information for each file. If no file specification is given, all files from the current directory will be listed.

REMOTE CWD [*directory*]

Change Working Directory. If no directory name is provided, the server will change to the default or home directory. The directory specification may be preceded by a device name.

REMOTE SPACE

Display information about disk usage in the current device.

SET *parameter [value]*

Set the specified parameter to the specified value. Possible parameter settings:

AUTORECEIVE ON (or OFF)

Allows several files to be received without having to type RECEIVE on the receiving machine. The routine simply looks for activity on the serial line, and if so fudges a RECEIVE command. The packet sent by the sender will be lost.

BLOCK-CHECK-TYPE *option*

The options are:

1-CHARACTER-CHECKSUM

Normal, default, standard 6-bit checksum.

2-CHARACTER-CHECKSUM

A 12-bit checksum encoded as two characters.

3-CHARACTER-CRC-CCITT

A 16-bit CCITT-format Cyclic Redundancy Check, encoded as 3 characters.

BUFFER-SIZE *value*

This allows you to set a buffer size during transfer of data. On some systems it takes so long that the remote end times out while the local system is reading or writing to disk.

The size is the number of disk blocks (nominal) and can be from 1 (512 bytes) to the maximum value determined by how much memory was available to the program on startup.

COLLISION *value*

What to do when a file arrives that has the same name, type and version as an existing file. DISCARD means to discard and reject the incoming file. OVERWRITE means to overwrite the existing file. NEW-VERSION means to write a new version of the file.

DEBUG ON (or OFF)

Enables/disables displaying of packets on the screen during file transfer. Not performed if the QUIET option has been set for the terminal (SET TERMINAL QUIET)

DEFAULT *device:[directory]*

This is the same as the CD command, and allows you to set the default device and unit and/or the directory as destination of file transfers. In addition, issuing this command causes Kermit-180 to switch to the specified device and directory. The colon must be included in the device name, and the directory may be enclosed in brackets. The selected device and directory appear in your Kermit-180 prompt, for instance

```
Kermit-180 SY0:[USER]>
```

DIRECTORY-LISTING LONG (or BRIEF)

By setting DIRECTORY-LISTING to BRIEF you can get an abbreviated listing of your disk drive. File sizes and attributes are not shown, and up to three files are shown on a line. Setting this option LONG will show the size, attributes and the creation date of each file. In addition, the LONG option will list the free space remaining.

ESCAPE Change the escape character for virtual terminal connections. Kermit-180 will prompt you for the new escape character, which you enter literally.

FILE-MODE *option*

Tells Kermit-180 what kind of file it is sending, so that Kermit can correctly determine the end of the file. SET FILE BINARY means to send all the file blocks, including the last block in its entirety; SET FILE ASCII is used for text files, and transmission stops when the first Control-Z is encountered anywhere in the file (this is the CP/M convention for marking the end of a text file).

SET FILE-MODE DEFAULT tells Kermit to attempt to determine the file type by examining the file being transmitted. If a Control-Z appears before the last block of the file, it is assumed to be BINARY; if, when the first Control-Z is encountered, the remainder of the file contains only control-Z's, it is assumed to be a text file. Unfortunately, not all programs fill the remainder of the last record of a text file with Control-Z's, so this algorithm is not always successful.

If ASCII transmission is used on a binary file, any 8th bits set will be stripped and a warning sent to the console. When the first control-Z is encountered, the file is assumed to be at the end, even if it is not.

FLOW-CONTROL ON (or OFF)

Sets XON/XOFF flow control on or off. If set ON the host is expected to respond to an XOFF or XON sent by Kermit-180. If set off, no flow control is assumed and any XON/XOFF is ignored.

IBM ON (or OFF)

Allow the transfer of files to and from an IBM mainframe computer. This makes Kermit-180 wait for the IBM turnaround character (XON), ignore parity on input, add

appropriate parity to output, and use local echoing during CONNECT. As distributed, Kermit-180 uses MARK parity for IBM communication. If you don't give this command, IBM mode is OFF. Since IBM VM/CMS Kermit does not have timeout capability, SET IBM ON also turns on the Kermit-180 timer automatically.

LINE *portname*

This is the same as the SET PORT command, and allows you to switch between different communication ports. Normally, you can connect to any free terminal device.

LOCAL-ECHO ON (or OFF)

When you CONNECT to a remote host, you must set LOCAL-ECHO ON if the host is half duplex, OFF if full duplex. OFF by default.

LOGGING ON (or OFF)

Cease or resume logging whenever connect mode is entered. This is really only applicable after a LOG command is no longer required.

NO-EXIT This command is applicable only for Kermit initiated with a command tail. For example, if Kermit was initiated by:

```
KERMIT SEND HELLO;NO-EXIT
```

Kermit would first seek out and execute the KERMIT.INI file (if present), then send file HELLO to a remote system. Usually Kermit would exit back to the system, but NO-EXIT over-rides this. STAY is a synonym for NO-EXIT.

PARITY *option*

Sets parity for outgoing characters to one of the following: NONE, SPACE, MARK, EVEN, or ODD. On input, if parity is NONE, then the 8th bit is kept (as data), otherwise it is stripped and ignored. The parity setting applies to both terminal connection and file transfer. If you set parity to anything other than none, Kermit-180 will attempt to use "8th bit prefixing" to transfer binary files. If the other Kermit is also capable of 8th bit prefixing, then binary files can be transferred successfully; if not, the 8th bit of each data byte will be lost (you will see a warning on your screen if this happens).

PORT *portname*

This is the same as the SET LINE command, and allows you to switch between different communication ports. Normally, you can connect to any free terminal device.

PRINTER ON (or OFF)

Turns copying of CONNECT session to printer on and off. It is also possible to toggle the printer on/off from the connect state, by typing <escape character> followed by P.

RECEIVE *parameter* [*value*]

Set a RECEIVE parameter.

PAD-CHAR

Set the PAD character to use while receiving files. Currently a dummy, as for SET SEND PAD-CHAR.

PADDING [*value*]

Set the number of PAD characters to use while receiving files. Same as SET SEND PADDING.

START-OF-PACKET [*value*]

Set the default start of Packet character for receiving files. Apply the same rules and considerations as for SET SEND START-OF-PACKET.

PACKET-LENGTH *number*

Tell the other Kermit the longest packet length CP/M Kermit is willing to receive during file transfer. The maximum length is 94, which is also the default length.

SEND *parameter* [*value*]

Set a SEND parameter.

PAD-CHAR

Set the Pad character to be used while sending files. It is currently a dummy entry, and does not do anything.

PADDING [*value*]

Set the number of PAD-CHARS to be used while sending files. This too does nothing.

START-OF-PACKET

Set the default start of packet character to another character than control-A. This may be necessary on systems (including intervening networks) that trap control-A characters. Choose a control character not otherwise used, ie not carriage return (13D, 0DH), line feed (10D, 0AH), tabs (09D, 09H), backspace (08H), and bell (07H) or any other used between you and your remote system.

SPEED *value*

Change the baud rate of the communications port. This command only works if supported by the terminal driver. *value* is the numeric baud rate (300, 9600, etc.) desired. Type SET SPEED followed by a question mark for a list of supported baud rates.

TACTRAP

Set the TAC intercept character. If you are attached to a TAC it will swallow the intercept character (commercial AT sign by default) so Kermit sends it twice. With this command you can set the intercept character (ie the one to send twice) to another character.

TERMINAL *option*

Select one of the following terminal characteristics:

OFF sets emulation off, and its up to the attached terminal to respond to escape sequences sent from the remote host system.

DUMB Like off, but carriage return and line feed characters are the only control characters accepted. All other control characters are simply ignored. (Really a "Glass TTY").

EXTERNAL

Emulation is provided for by a routine in the system dependent part of Kermit. Attempting to set this option without having an externally supplied routine will return a "Not Implemented" error.

OFF All characters are passed directly to the terminal without any interpretation by Kermit.

VT52 When connected as a terminal to a foreign host, the micro emulates a VT52. VT52 emulation is set by default, except on micros that already have terminal functionality built in, such as the DEC VT180 and DECmate (these act as VT100-series terminals).

QUIET Do not display any file transfer information onto the console. This mode is useful if you console takes a long time to update the display. Only the file name is displayed. **DEBUGging** information is not displayed even if selected.

REGULAR

Inverse of **QUIET**. All packets etc displayed, as usual.

TIMER ON (or OFF)

Enable or disable the timer. The timer is off by default, because in the normal case Kermit-180 is communicating with a mainframe Kermit that has its own timer. Mainframe Kermit timers tend to be more precise or adaptable to changing conditions. You should **SET TIMER ON** if you are communicating with a Kermit that does not have a timer. You should **SET TIMER OFF** if you are communicating over a network with long delays.

WARNING ON (or OFF)

Warn user of filename conflicts when receiving files from remote host, and attempt to generate a unique name by adding "&" characters to the given name. **ON** by default, which is equivalent to **SET COLLISION RENAME**.

SHOW Display all settable parameters. You will get a page or so of the status and all parameters that can be set using the **SET** command.

STATUS The same function as **Show**.

STAY Equivalent to **SET NO-EXIT**.

TAKE *filespec* Take characters and commands from the specified file as if they were entered from the keyboard. This is useful if you want to set up a batch job. A command file can send, get, receive, set functions etc automatically. A **TAKE** command can be interrupted with **^C**.

An automatic "**TAKE KERMIT.INI**" is executed from the default drive when Kermit-180 is loaded. This can be used to set defaults of band rate, parity, filetype, default drive etc.

If **KERMIT.INI** does not exist, control is given directly to the user.

TRANSMIT *filespec turnaround*

Send the specified file to the system on the other end of the connection as though it were being typed at the terminal, one line at a time. Each line sent is terminated with a carriage return, and any line feeds are stripped from the file sent. After each line has been sent Kermit waits for a character string from the host (e.g. a carriage return). If not specified, a carriage return is assumed. No Kermit protocol is involved. An asterisk (star) is sent to the console for every line sent, to indicate how the transfer is progressing. This is useful for sending files to systems that don't have a Kermit program. During transmission, you may type one of these single-character commands:

Control-C

Cease transmission, and drop into terminal emulation mode.

CR (carriage return)

Re-transmit the previous line.

TYPE *filespec* Type a local file or files on your screen. A **Ctrl-C** will cancel the command and return to the Kermit prompt. A **Ctrl-X** will cancel the current file and go on to the next one, if any. Typing **Ctrl-S** while the file is being displayed will suspend the output, typing **Ctrl-Q** will resume output.

VERSION Show the name, edit number, and edit date of several of the modules that make up Kermit-180.

3 Assembling Kermit-180 from the sources

Kermit-180 is built in from the following files:

KDEF.INC	Definitions for KERMIT
KMIT.MAC	Initialization, main loop, miscellaneous commands (BYE, EXIT, LOG, PAUSE, HELP and TAKE)
KCOM.MAC	Second part of commands (SET and SHOW).
KPKT.MAC	KERMIT protocol handler (SEND, RECEIVE, LOGOUT, and FINISH commands)
KREM.MAC	REMOTE routines
KSER.MAC	SERVER routines
KTT.MAC	The transparent commands (TRANSMIT, CONNECT)
KCMD.MAC	The command parser
KSYS.MAC	RSX180/280-specific routines and local file handling commands (DIR, ERA, USER, TYPE, PRINT, COPY)
KUTL.MAC	Utility routines and data
KDAT.MAC	Global data space

Unlike Kermit-80, there are no customizable system-dependent files since all the communication is done via the system device drivers, and RSX180/280 programs are not allowed to access I/O ports directly anyway.

There is a jump entry in the KSYS.MAC file to allow users to add their own terminal emulator (a leftover of Kermit-80). If you write the code for such an emulator, you must load this jump address with the address of your emulator, and SET TERMINAL EXTERNAL from within Kermit. All characters will be passed to this routine during connect mode.

The files are assembled separately with ZSM4 and the resulting relocatable object files combined into the final task image file with the TKB linker. The provided BUILD.CMD command file automates the build steps under RSX180 or RSX280. Note that you will still need the system include files and the standard SYSLIB library, which are normally found in the [SYSTEM] directory of the OS distribution. If they are in another directory then you will have to edit the BUILD.CMD file accordingly.

If you are doing the development system on an PC, then you can assemble the files using a CP/M emulator such as John Elliott's ZXCC and the supplied Makefile. You can transfer later on your assembled files to the target system using the vol180 utility supplied with the RSX180 or RSX280 development tree.

Note that the current version of Kermit-180 is built by default as a privileged task since it needs to set the characteristics of the terminal used for communications to slave with echo disabled, and those are privileged operations for terminals other than the user's current one. This has the undesirable side effect that Kermit-180 can access any file in the system, and in any directory, regardless of the current user privileges. That is not a problem as long as Kermit-180 is not INStalled, since only privileged users can run a privileged task.

3.1 Future Work

Work that needs to be done in future releases includes:

- Addition of missing features, e.g. compression of repeated characters during packet transmission, transmission of file attributes (particularly size, so that "percent done" can be displayed for both incoming and outbound files.)

- Command macros, more advanced login scripts.
- Implement more Server commands.
- etc.