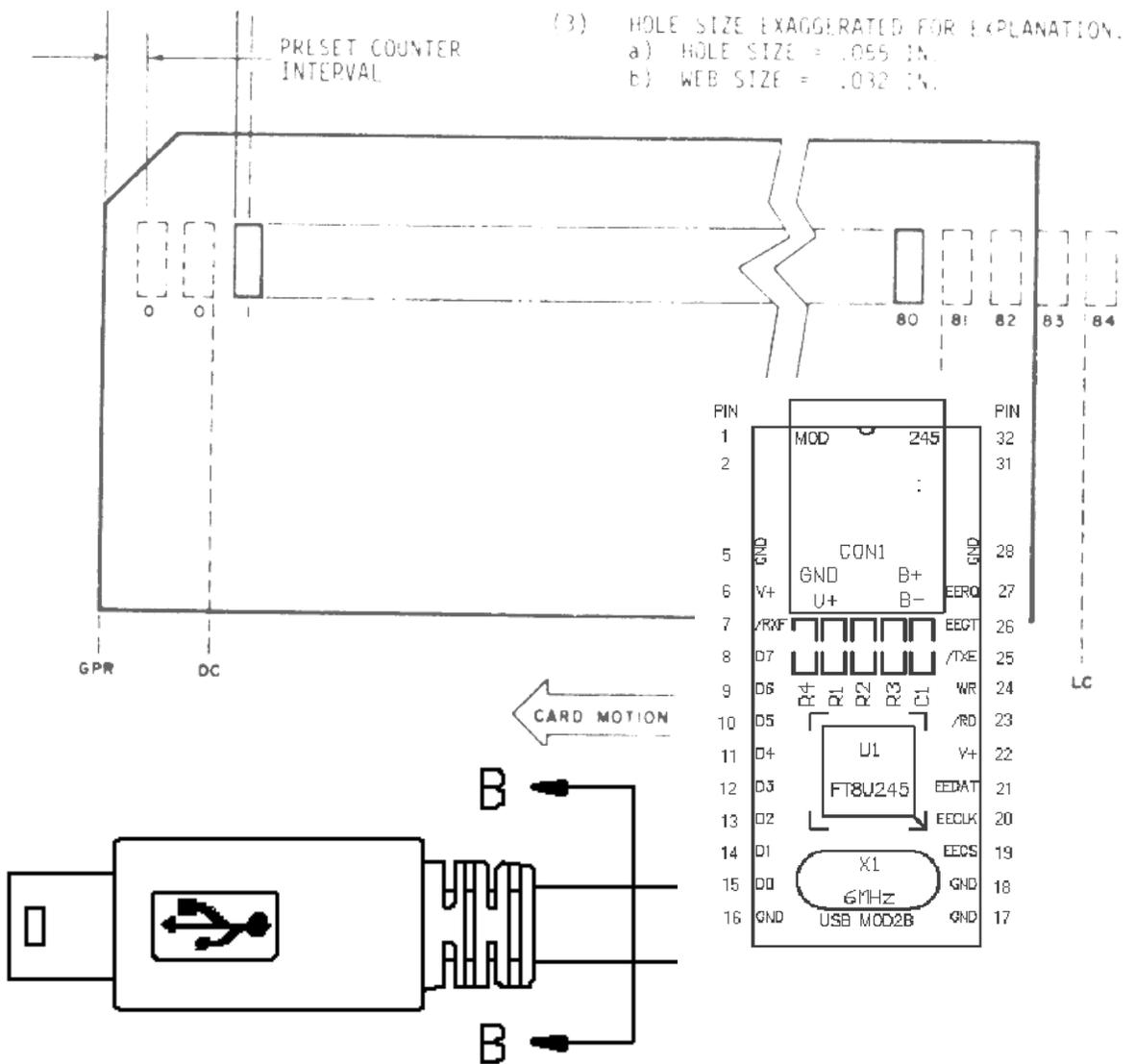


CardRead

Documation Card Reader to a PC through a USB Interface

Version 1.3 - 20 October 2004



Notices

➔ Software, documentation, interface hardware design and firmware
©Copyright 2003-2004 Quarterbyte Systems, Inc.

➔ For information and/or maintenance, please contact

Brian Knittel
Quarterbyte Systems, Inc
817 Delaware St, Suite B
Berkeley, CA 94710
(510) 559-7930
brian@quarterbyte.com

➔ Hardware, Software and Firmware licensed under the terms of the GNU General Public License, available at <http://www.gnu.org/copyleft/gpl.txt>.

Documentation licensed under the GNU Free Documentation License, available at <http://www.gnu.org/copyleft/fdl.txt>.

➔ **BECAUSE THE SOFTWARE, HARDWARE, FIRMWARE AND DOCUMENTATION ARE LICENSED FREE OF CHARGE, there is**

NO WARRANTY—Quarterbyte Systems, Inc. does not warrant that operation of the CardRead software, interface hardware or firmware will be uninterrupted or error free or that all documentation or program defects will be corrected. All express or implied conditions, representations, and warranties including, without limitation, any implied warranty or condition of merchantability, fitness for a particular purpose, noninfringement, satisfactory quality or arising from a course of dealing, law, usage, or trade practice, are hereby excluded to the extent allowed by applicable law

and

NO LIABILITY FOR CONSEQUENTIAL DAMAGES—Quarterbyte Systems, Inc. expressly disclaims all liability for any indirect or consequential damages whatsoever (including, without limitation, damages for loss of business profits, business interrupted, loss of business information, or any other pecuniary loss) arising out of the use of, or inability to use, this product, even if Quarterbyte Systems, Inc. been advised of the possibility of such damages. In no event shall Quarterbyte Systems, Inc. be liable for costs of procurement of substitute products or services, or for any lost profits, or for any consequential, incidental, direct or indirect damages, however caused and on any theory of liability, arising from the use of this software or hardware.

➔ OK, now in English: Use at-your-own-risk.

➔ Copying, modification and distribution of the hardware, software, firmware and documentation is permitted in any medium, provided these notices are preserved.

Table of Contents

Notices.....	i
Revision History.....	iii
1. Introduction.....	1
1.1. The Interface Box.....	1
2. Installation.....	1
2.1. Installing the USB Driver.....	1
2.2. Configuring the USB Driver.....	2
2.3. Installing CardRead.....	3
2.4. Configuring CardRead.....	3
3. The CardRead Interface.....	4
4. Data Formats.....	7
4.1. Binary Format.....	7
4.2. ASCII Format.....	7
4.3. Specifying Alternate Encodings.....	9
5. Reading Card Decks.....	10
6. Reading Multiple Decks.....	12
6.1. Specifying Autosep Cards.....	12
6.2. Using the Autosep Feature.....	13
6.3. Using the Next File feature.....	13
7. Verifying Decks.....	14
7.1. Verifying Multiple Decks.....	15
8. Viewing Binary Decks with DeckView.....	16
Appendix 1. CDROM Contents.....	18
Appendix 2. Interface Schematic.....	20
Appendix 3. Interface Protocol.....	23

Figures

Figure 1. Use the Device Manager to find the interface's COM port number.....	2
Figure 2. The configuration dialog lets you specify the COM port associated with the Documentation/USB Interface.....	3
Figure 3. The Default CardRead display.....	4
Figure 4. With the Display box checked, CardRead displays the last card read as a graphic.....	6
Figure 5. The “Configure” item on the System Menu activates the configuration dialog.....	7

Figure 6. Binary Data Format..... 7

Figure 7. Hollerith-to-ASCII Conversion Tables 8

Figure 8. Select a folder and enter a filename in the “Save As” dialog..... 11

Figure 9. DeckView lets you view and edit binary card deck files 16

Revision History

Version	Date	Remarks
1.0	15-Aug-03	Initial release
1.1	25-Nov-03	Added choice of encoding table to "Data" section. Added interface to DeckView, so that View button works with binary decks. Added "Verify" option to "End of File / Autosep" section
1.2	30-Jul-04	Added "stop-on-error" option so that reading can be resumed after a data check, necessitated by odd deck at the Computer History Museum that had punches in column 81. This causes the Documentation reader to halt with a data check. Unchecking Stop On Error lets user press Start on the card reader to continue reading this type of deck. Changed name of Port item on system menu to Configure. Added DeckView to the distribution.
1.3	20-Oct-04	Added option to enter custom encoding tables with lists of row-punch values rather than requiring use of hex values. Added Text format option to DeckView's Save As dialog.

1. Introduction

CardRead works with the Documation/USB Interface to read card decks to disk files on a PC. The program should work under Windows 98 and above, although as of this writing it has only been tested on Windows 2000 and XP.

CardRead has the following features:

- Cards can be saved as raw binary punch data, or as ASCII text
- Verification mode compares cards to files stored on disk
- Autoseparation mode allows one input deck to be split into multiple files

The requirements are:

- A computer running Windows 98, ME, 2000 or XP, with a free USB port
- A USB “A/B” cable
- The Documation/USB Interface box
- A Documation M series card reader

The Interface hardware is described in Appendix 2. The source code for the interface firmware and for CardRead itself is provided on the CDROM. The Interface protocol is described in Appendix 3.

- ➔ The interface described in this manual is designed for a reader with the standard Documation interface, with a 38-pin connector and active-low TTL signalling levels. Other interfaces are possible; an interface has been built for a Hewlett-Packard OEM variant that uses differential signalling and a 50-pin connector.

1.1. The Interface Box

The Documation/USB Interface box has a female Centronics Parallel connector and a female USB “B” socket on the back. The Centronics connector must *not* be connected to a PC parallel port. It is to be connected only to a Documation M-series card reader using the custom cable described in Appendix 2. The USB port connects the interface to the PC.

The front of the interface box has a pushbutton switch that signals End of File (ignore the “TEST” label over it, an artifact of the box’s former life). Press this button when the last cards of a given file are being read. When the hopper is emptied, CardRead will close the output file. A LED indicates that End of File has been signalled. While it is lit, you may press the button again to toggle it off.

2. Installation

There are two steps involved in installing CardRead: the USB driver must be installed, and the CardRead programs must be copied to the hard disk.

2.1. Installing the USB Driver

CardRead communicates with the USB interface using the virtual COM port (VCP) device driver provided by the USB interface chip's manufacturer. The driver is provided on the CD containing this document (see Appendix 1, page 18). If it is not, you may obtain it from www.ftdichip.com/FTWinDriver.htm. Download the “Virtual COMP Port (VCP) Drivers for

Windows 98 / ME / 2000 / XP.” If you are using the downloaded drivers, unzip the downloaded files into a folder on your hard disk.

Full driver installation instructions may be found in the Application Notes provided on the CD in folder \usbdriver. There are two files:

- AN232-03.pdf for Windows 98/ME
- AN232-05.pdf for Windows 2000/XP

The approximate procedure for Windows 2000/XP is:

1. Log on as a Computer Administrator.
2. Connect the interface box to the computer.
3. When the Windows Hardware Wizard prompts for a device driver, direct it to the \usbdriver folder on the CDROM, or the directory into which you unzipped drivers downloaded from ftdichip.com.

2.2. Configuring the USB Driver

Whenever the interface box is connected, the USB driver creates a virtual COM port. You'll need to know which COM port was created. To find out on Windows 2000 and XP, follow these steps:

1. Right-click My Computer and select Manage.
2. Select Device Manager in the left-hand pane, and expand the “Ports” item in the right-hand pane, as shown in Figure 1. The entry for “USB Serial Port” will indicate the COM port for the interface.
3. If you wish to change the port, right-click the “USB Serial Port” entry and select Properties. Select the Port Settings tab and click Advanced. Change the COM port to the desired number.

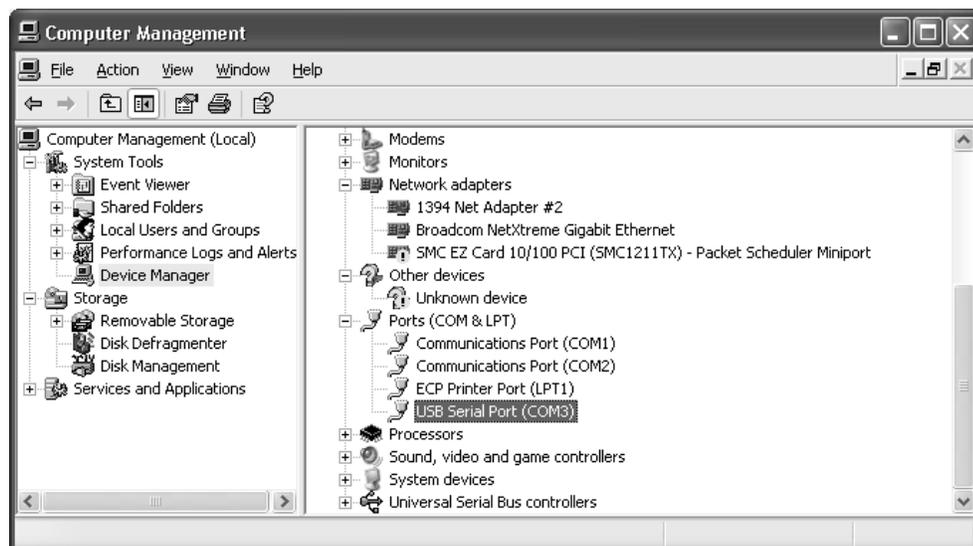


Figure 1. Use the Device Manager to find the interface's COM port number.

The Application Notes on the CD give full instructions for Windows 98, ME, 2000 and XP.

2.3. Installing CardRead

Install the CardRead software by copying CardRead.exe and DeckView.exe to any folder on your hard drive. You might copy them to C:\Program Files\CardRead.

Create a desktop shortcut to cardread.exe by right-clicking the desktop, selecting New, Shortcut, and locating the CardRead.exe file. Repeat this procedure to create a shortcut to DeckView.exe

- ➔ If you intend on reading decks in Binary mode and plan on using the same filename extension for all binary decks (for example, .bin, .dat or .card), you may wish to associate DeckView with this extension, using Windows Explorer's Tools > Folder Options > File Types menu. This will let you view card deck files in DeckView by double-clicking them.

2.4. Configuring CardRead

CardRead must be told which COM port the interface is using. To do this, follow these steps:

1. Start CardRead, then click the System Menu in the upper left-hand corner of the title bar (or press Alt+Space).
2. Click Configure. In the Select Port dialog, enter the COM port name used by the interface, as shown in Figure 2. Click OK.

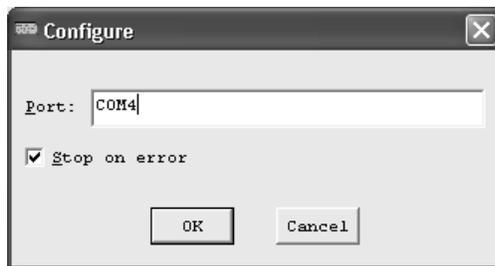


Figure 2. The configuration dialog lets you specify the COM port associated with the Documation/USB Interface

3. The Stop on Error option controls CardRead's behavior when a read check error is reported by the card reader. When this option is checked, CardRead stops reading and closes the output file. If unchecked, CardRead will continue reading cards to the same output file. This option should usually be left checked.

If CardRead cannot contact the interface, it will display one of the following messages:

- **ERROR - COMn:: The system cannot find the file specified**—Indicates that the specified COM port does not exist. The interface may be unplugged or the port number may be entered incorrectly.
- **Interface failed to respond after 10 tries**—The interface is malfunctioning, or you may have entered the COM port number for a modem or other serial device. If this message does not disappear within a few seconds, check the configuration setting.

CardRead attempts to contact the interface every few seconds, so if the problem is that the interface was unplugged, you can simply plug it in and proceed.

- ➔ When CardRead is trying to contact the interface and is experiencing problems, the user interface may respond sluggishly.

- ➔ CardRead stores its configuration settings in the Registry under HKEY_CURRENT_USER\Software\Quarterbyte\Cardread, so you will only need to enter the Port information the first time a given user runs CardRead.

3. The CardRead Interface

The CardRead program initially displays the window shown in Figure 3.

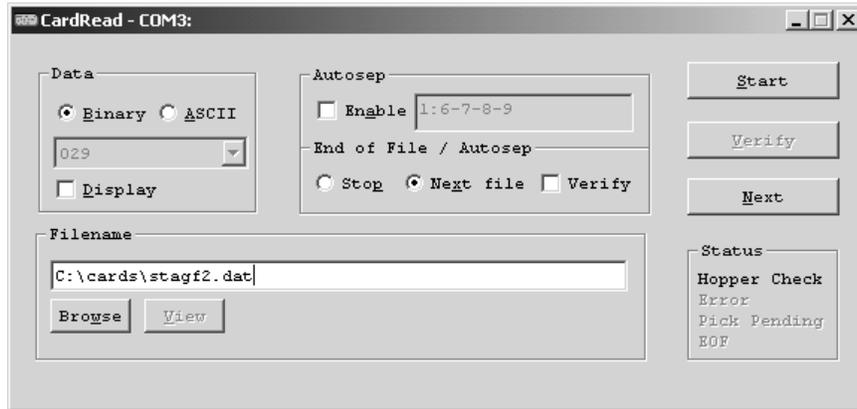


Figure 3. The Default CardRead display.

The window is divided into several sections, which are described below:

Section	Control	Description
Data	Binary	Tells CardRead to save raw binary data. The binary data format is described in Section 4.1, “Binary Format,” (page 7).
	ASCII	Tells CardRead to save ASCII text. The Hollerith to ASCII conversion table is listed in Section 4.2, “ASCII Format” (page 7).
	Encoding	Selects the Hollerith-to-ASCII encoding table to use when ASCII mode is selected. Default choices are 029, 026 Fortran and 026 Commercial. Custom encoding tables can be created; see Section 4.3, "Specifying Alternate Encodings" (page 9).
	Display	When checked, each card is displayed as a graphic as it is read. The card's text, if any, is also displayed.
Autosep	Enable	When checked, indicates that a specially punched card indicates a division between card decks. “Autosep” cards are not saved to the PC.
	(input field)	Specifies the punches that indicate an autosep card. The default is a 6, 7, 8 and 9 punch in column 1 with no other punches on the card.

Section	Control	Description
		The format for this field is described in Section 6.1, “Specifying Autosep Cards” (page 12).
End of File / Autosep	Stop	Instructs CardRead to stop reading cards when you press the End of File button or when it encounters an autosep card.
	Next file	Instructs CardRead to close the current file, increment the filename, and continue reading cards when you press the End of File button or when it encounters an autosep card.
	Verify	Instructs CardRead to re-read and verify the same file when you press the End of File button or when it encounters an autosep card. If the verification is successful, CardRead then increments the filename and prepares to read the next file.
Filename	(input field)	Specifies the name of the file to be created or verified.
	Browse	Displays the standard Windows “Save As” dialog, which you may use to select the output folder and filename.
	View	Displays the current file in DeckView or Notepad, depending on the Binary/ASCII Data setting.
Buttons	Start / Stop	Starts reading cards into the selected file using the current conversion settings, or halts reading and closes the current file.
	Verify	Re-reads cards and compares them to the selected file, to ensure that they were read correctly.
	Next	Like “Go,” but adds 1 to the current filename before starting. For more information about name incrementing, see Section 6.3, “Using the Next File feature” (page 13).
Status	Hopper Check	Indicates that the card reader input hopper is empty or that the output stacker is full.
	Error	Indicates that there was a problem reading the last card. Usually indicates a mechanical problem with the reader, but may indicate a card with a torn leading or trailing edge.
	Pick Pending	Indicates that the card reader is ready to read a

Section	Control	Description
		card.
	EOF	Indicates that you have pressed the End of File button on the interface.
Remark		A single line of text may appear under the Filename box. This is used to display errors and informational messages.

When the “Display” option is checked, the CardRead window expands to show the last card read, as shown in Figure 4. This display also shows the autosep card when you are editing the Autosep specification.

Finally, there is one more or less hidden interface item in the System Menu, activated by clicking the upper left corner of the window's title bar or by pressing Alt+Space, as shown in Figure 5. The menu has an item titled “Configure,” which displays the configuration dialog. This dialog is discussed in Section 2.4, “Configuring CardRead” (page 3).

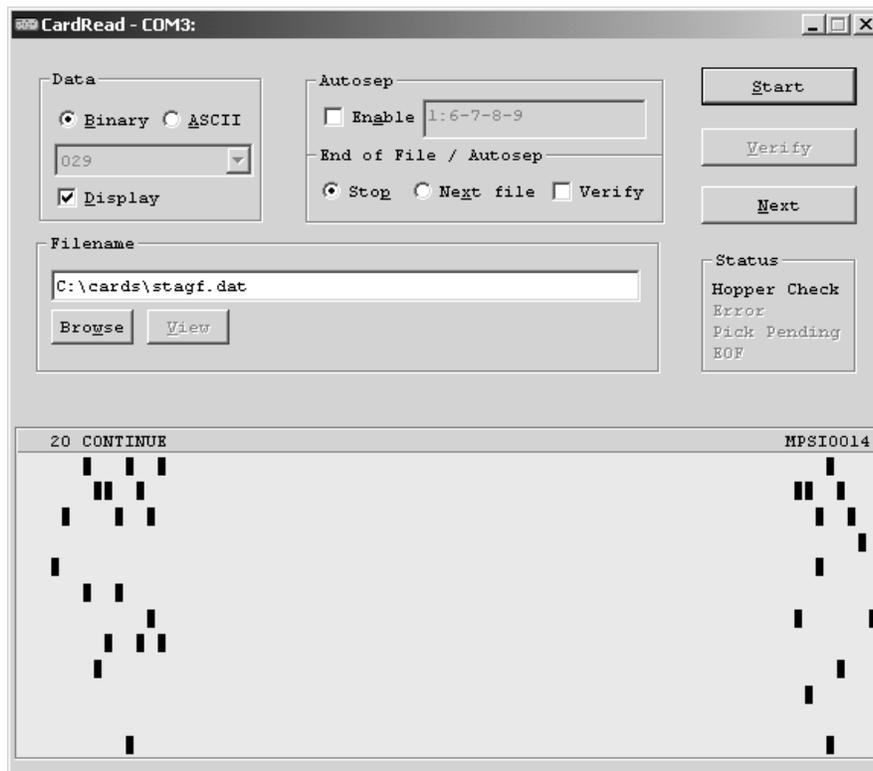


Figure 4. With the Display box checked, CardRead displays the last card read as a graphic.

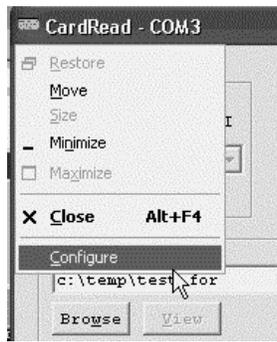


Figure 5. The “Configure” item on the System Menu activates the configuration dialog.

In addition to the GUI interface, the interface box has a pushbutton labeled “End of File” (EOF). This button tells CardRead that when the hopper has been emptied, the current deck is finished. CardRead’s response depends on the “Stop/Next File” setting:

- If Stop mode is selected, the current file is closed and CardRead stops reading.
- If Next File mode is selected, CardRead increments the filename and waits to read more cards, as described in Section 6.3, “Using the Next File feature” (page 13).

Two pushes in succession tell CardRead to stop reading,

4. Data Formats

CardRead can save cards as ASCII text or as raw binary punch data. You can choose the format by selecting either the “ASCII” or “Binary” radio button in the upper left corner of the window.

4.1. Binary Format

The binary format is the one used by the IBM 1130 simulator in the SIMH “Computer History Simulation Project” (simh.trailing-edge.com) The file consists of a series of 80 word (160 byte) records, each representing one card, written in Intel “little-endian” format, least significant byte first. Each word represents one card column, with the data left-aligned as illustrated in Figure 6. The low four bits in each word are zeroes.

	MSB								LSB							
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Punch:	12	11	0	1	2	3	4	5	6	7	8	9	-	-	-	-

Figure 6. Binary Data Format

4.2. ASCII Format

ASCII format converts cards to standard MS-DOS text files, with each card represented by a line of ASCII characters followed by a carriage-return/line-feed pair. Trailing blanks are eliminated.

The program allows you to specify the encoding used on the punched cards by making a selection in the dropdown box under the Binary / ASCII radio buttons. The default choices are 029, 026 FORTRAN and 026 Commercial, although you can specify additional encodings, or alter the default encodings, as discussed in the next section.

These default tables are shown in Figure 7. The “Code” columns contain Hollerith character codes in hexadecimal, and the “Char” columns contain the corresponding ASCII characters.

Figure 7. Hollerith-to-ASCII Conversion Tables

029 Encoding							
Code	Char	Code	Char	Code	Char	Code	Char
0000	(space)	5000	J	00A0	=	A100	e
8000	&	4800	K	0060	"	A080	f
4000	-	4400	L	8820	¢	A040	g
2000	0	4200	M	8420	.	A020	h
1000	1	4100	N	8220	<	A010	i
0800	2	4080	O	8120	(D000	j
0400	3	4040	P	80A0	+	C800	k
0200	4	4020	Q	8060		C400	l
0100	5	4010	R	4820	!	C200	m
0080	6	3000	/	4420	\$	C100	n
0040	7	2800	S	4220	*	C080	o
0020	8	2400	T	4120)	C040	p
0010	9	2200	U	40A0	;	C020	q
9000	A	2100	V	4060	¬	C010	r
8800	B	2080	W	2420	,	6800	s
8400	C	2040	X	2220	%	6400	t
8200	D	2020	Y	2120	_	6200	u
8100	E	2010	Z	20A0	>	6100	v
8080	F	0820	:	B000	a	6080	w
8040	G	0420	#	A800	b	6040	x
8020	H	0220	@	A400	c	6020	y
8010	I	0120	'	A200	d	6010	z

026 FORTRAN Encoding							
Code	Char	Code	Char	Code	Char	Code	Char
0000	(space)	0010	9	4400	L	2080	W
8000	+	9000	A	4200	M	2040	X
4000	-	8800	B	4100	N	2020	Y
2000	0	8400	C	4080	O	2010	Z
1000	1	8200	D	4040	P	0420	=
0800	2	8100	E	4020	Q	0220	'
0400	3	8080	F	4010	R	8420	.
0200	4	8040	G	3000	/	8220)
0100	5	8020	H	2800	S	4420	\$
0080	6	8010	I	2400	T	4220	*
0040	7	5000	J	2200	U	2420	,
0020	8	4800	K	2100	V	2220	(

026 Commercial Encoding							
Code	Char	Code	Char	Code	Char	Code	Char
0000	(space)	0010	9	4400	L	2080	W
8000	&	9000	A	4200	M	2040	X

Code	Char	Code	Char	Code	Char	Code	Char
4000	-	8800	B	4100	N	2020	Y
2000	0	8400	C	4080	O	2010	Z
1000	1	8200	D	4040	P	0420	#
0800	2	8100	E	4020	Q	0220	@
0400	3	8080	F	4010	R	8420	.
0200	4	8040	G	3000	/	8220	<
0100	5	8020	H	2800	S	4420	\$
0080	6	8010	I	2400	T	4220	*
0040	7	5000	J	2200	U	2420	,
0020	8	4800	K	2100	V	2220	%

Notes: The characters \neg and ϕ are saved as bytes with hexadecimal values AC and A2, respectively, following the MS-DOS encoding standard. The 026 Commercial punch 8220 sometimes printed the symbol \neg but often printed ϕ , so the default conversion table converts this punch to ϕ .

4.3. Specifying Alternate Encodings

The default encodings listed in the previous section are built into CardRead. You can add additional encoding choices and/or replace the build-in encoding tables by creating a file named CARDCODE.TBL placed in the same directory as CARDREAD.EXE. The structure of cardcode.tbl is as follows:

- Blank lines and lines starting with /, *, # or ; are ignored as comments.
- A new encoding scheme is introduced by a line containing the name of the scheme enclosed in square brackets, for example, [029] or [My Special]. It's best if the length of the name is 15 characters or less, so it will display properly in the dropdown box. More than one table can be specified in the file.
- After the name there may be up to 150 encoding entries. Each entry appears on a separate line, and consists of a specification for the punch value, one or more blanks or tabs, then a specification for the corresponding ASCII value.

The punch value can be specified in either of two ways: as a four character hexadecimal number listing the hollerith code left justified in 16 bits (that is, in CardRead's binary format), *or* as a row-punch list in the format used by the Autosep field (see "Specifying Autosep Cards" on page 12). For example, the entries 8010 and 12-9 are equivalent. The blank punch must be specified as the hexadecimal value 0000.

The ASCII code can be specified as either a single non-white ascii character *or* by a two-digit hexadecimal number specifying the ASCII value. Note that the space character must be specified as the hexadecimal value 20.

- The character or character code may optionally be followed by whitespace and comment text.
- The ordering of the hollerith/ascii entries within a given code table is not important, although it can't hurt to place the most common codes near the start of the table, as it is searched sequentially.

For example, the 029 encoding table might start like this:

```
[029]
0000 20      (space)
8000  &
```

```
4000 -
2000 0
1000 1
0800 2
0400 3
```

and so on.

CardRead comes with a sample `cardcode.tbl` file that adds the Extended BCD (EBCD) encoding.

You can replace any of the built-in encodings by creating tables in `CARDCODE.TBL` using the built-in tables' names "029", "026 FORTRAN" or "026 Commercial". (You must specify the entire table if you do so, not just the characters whose encodings you wish to alter). If you use other names for the tables, these names will appear as additional choices in the dropdown list.



To help you construct encoding tables, the correspondence between hexadecimal Hollerith values and card punch rows is listed below. However, starting with Version 1.3, you can enter punch-row lists and do not need to use hexadecimal values.

<u>Hex Value</u>	<u>Punch Row</u>
8000	12
4000	11
2000	0
1000	1
0800	2
0400	3
0200	4
0100	5
0080	6
0040	7
0020	8
0010	9

5. Reading Card Decks

Before reading cards, you should be familiar with the Documentation M-series Card Reader.



You may want to read section 3 of the Documentation manual provided in PDF form on the CDROM. It's for an M200 reader, but the operating instructions apply to all of the M-series readers.

To read a card deck, follow these steps:

1. Select Binary or ASCII data format.
2. If you wish to see the cards as they are read, check Display.
3. Uncheck Autosep.
4. Under "End of File / Autosep," select Stop.
5. Enter the name of the file you wish to create, using a full drive and path specification. You may find it easier to click Browse and use the standard Windows "Save As" dialog to select a folder. Be sure to enter the desired filename at the bottom of the Save As dialog, as shown in Figure 8. When you click Save, the full path to the selected file will be placed in the filename field in the CardRead window.
6. Click Start.

7. Place your deck in the card reader and press the RESET button on the card reader. (Actually, you can click Start and press RESET in either order).
8. The cards will be read through. A counter just under the filename will show the number of cards read.

If there are more cards than will fit in the reader, process them in groups. After each group, empty the stacker, place the next group in the hopper and press RESET. All of the cards will be read into the same file.

9. When the last card has been read, inform CardRead in any of the following ways:
 - Press the End of File (EOF) button on the interface box. (Actually, you can press EOF at any time while the last hopper full of cards is being read),
 - Click Stop in the CardRead window, or
 - Press the Esc key on they keyboard.

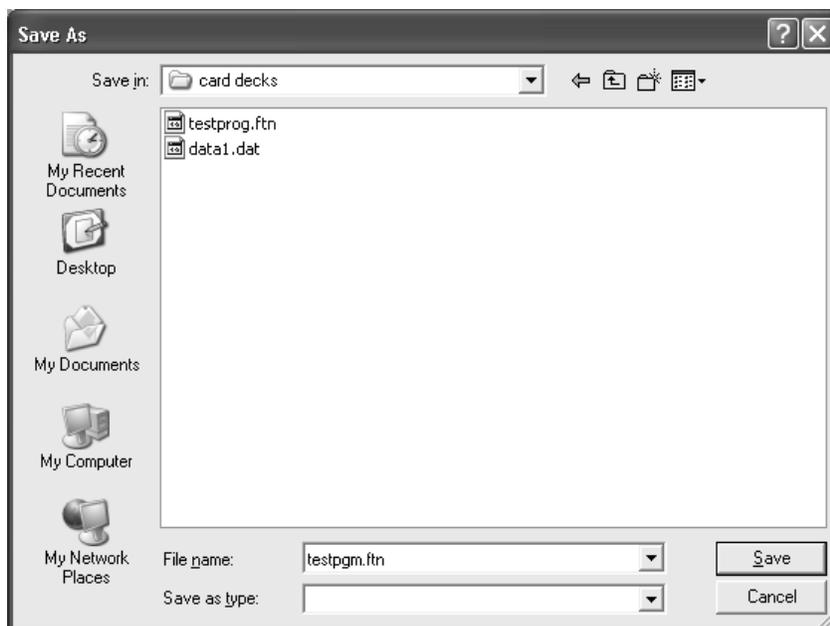


Figure 8. Select a folder and enter a filename in the "Save As" dialog.

Cardread will close the output file and display the total number of cards read on the Remark line.

- ➔ If you have just read a deck in ASCII mode, you can click View to open the file in Notepad.
- ➔ If you have read a deck in Binary mode, and the program file DeckView.exe has been placed in the same directory as CardRead.exe, you can click View to open the file in DeckView.
- ➔ If there is an error reading the cards or writing the output file, by default, CardRead will stop reading cards, close the output file, and will display a message on the Remark line.

- ➔ We have encountered decks that have punches beyond column 80. On encountering an extra punch, the Documentation reader stops with a Read Check error. To read this sort of deck, open the Configuration dialog and uncheck Stop On Error. You will still need to press the Start or Reset button on the card reader after every card with the extra punch, but at least you will be able to read the deck into one file. Remember to re-check Stop On Error after reading the troublesome deck.

6. Reading Multiple Decks

If you have many card decks to read, each into a separate file, you may wish to use CardRead's autoseparation and "next file" (autoincrement) features to simplify the chore. Here's what these two features do:

- The *autosep* feature lets you define a special card that indicates the end of a deck. This separator card signals CardRead that it has reached the end of a group of cards. It is not stored in the output file.
- The *next file* feature automatically adds a digit to the current filename and proceeds to read the next deck. For example, if you'd just read file "test.dat", the next file will be "test1.dat", followed by "test2.dat" and so on.

These two features can be used independently.

6.1. Specifying Autosep Cards

To use the Autosep feature, you will need to choose a set of punches that will never appear in the card decks you're reading. The default autosep card is a 6-7-8-9 punch in column 1 with no other punches*. You can select any other punch pattern that you wish.

You can specify the autosep card using row and column numbers, or as literal text, in the following format:

<code>row[-row]*</code>	row punches; rows are 12, 11, 0, 1, 2, ..., 8 and 9
<code>"xxx"</code>	literal text

Multiple entries may be separated by commas to indicate data for successive columns. By default, the first entry is placed in column 1. An entry may be preceded by a column number and a colon (:) to specify a particular column.

- ➔ When you edit the autosep specification, CardRead automatically enables "Display" mode and displays the specified card as you type.

Here are some examples (chosen to illustrate the formatting technique rather than to suggest separator patterns):

`6-7-8-9`

Punches 6, 7, 8 and 9 in column 1

`12-11-8-9, 12-11-8-9, 79:12-11-8-9, 12-11-8-9`

A group of four punches in each corner of the card

`"// SEPARATOR"`

* This was the end-of-job separator used by the BKY operating system for the CDC-6400 and CDC-6600 at U.C. Berkeley and Lawrence Berkeley Laboratory.

Literal text, starting in column 1

12-11-0-1-2-3-4-5-6-7-8-9, 70:"END OF DECK"

All 12 punches in column 1, with END OF DECK at the far right.

When you have selected a separator punch format, make separator cards using a keypunch, Wright manual punch, or a high-speed punch.

➔ Punch your separator cards on a brightly colored card stock, and if possible, use stock that has the corner cut on the opposite side from the cards you're reading. This will make it easier to locate and remove them after reading the decks.

 **Warning:** If your separator cards have punches that don't correspond to standard printing characters, don't duplicate them in a printing 026, 029 or 129 IBM keypunch, or you may damage the printing mechanism. Turning the print switch off will *not* prevent the damage.

6.2. Using the Autosep Feature

To read compound decks with separator cards, perform the following steps:

1. Insert a single separator card between each group of cards. Stack the cards up into a single compound deck.
2. Select Binary or ASCII data format.
3. Check Autosep.
4. If necessary, enter the correct separator punch specification. Verify that the display shows the card accurately.
5. Under “End of File / Autosep,”
 - Select Stop if you want to manually change the filename for each sub-deck.
 - Select Next File if you want to use the automatic Next File feature, which is discussed in section 6.3 (below).
6. Enter the name of the first file you wish to create, using a full drive and path specification.
7. Click Start.
8. Place your deck in the card reader and press the RESET button.
9. If you selected Stop mode, each time the reader stops, change the output filename and click Start to read the next group of cards.

If you selected Next File mode, the reader will just plow through all of the cards until the supply is exhausted.

10. When the last card has been read, press EOF, type Esc or click Stop.

6.3. Using the Next File feature

CardRead makes it easy to process groups of cards into a series of files that have the same base name and an ascending number appended to each successive filename. You may wish to do this when you have a large number of cards to read—it might be more efficient to read all the cards as quickly as possible, and then sort out the deck names later.

You can use the Next File feature in either of two ways:

- You can read the decks separately. Click Start to process the first deck, and press the EOF button or click Next to process each subsequent deck. Click Stop or press EOF a second time when you have finished reading all decks.
- You can combine the decks with separator cards and read them together. Use the Autosep feature along with the “Next File” setting to automatically increment the filename at each separator card.

When you click Next, or when CardRead encounters an autosep card in “Next File” mode, CardRead modifies the filename in this way:

- It skips any drive or path specification.
 - It scans the filename from left to right looking for a numeric digit. If it finds one, it counts the number of successive digits to determine the number length. It evaluates the digit(s) as a decimal number, adds one, and replaces them with the incremented number, using the same number of digits. Leading zeroes are preserved. For example, “FILE0001.DAT” would become “FILE0002.DAT”. If the new number requires more digits than the original, more digits are used. Thus, “FILE9999.DAT” would become “FILE10000.DAT”.
 - If no digit is found, CardRead inserts the digit 1 before the file extension. For example, “TEST.DAT” would become “TEST1.DAT”. If there is no extension, it appends the digit 1 to the end of the filename.
- ➔ In Next File mode, if CardRead encounters two successive separator cards with no intervening data cards, it will stop reading at that point. You can use this feature to read some decks with one base name, and then change the base name for remaining cards.

7. Verifying Decks

If you are concerned about read errors, you may use CardRead to verify a deck against the stored file. While this won't protect against systematic errors in reading (for example, mechanical problems that produce the same incorrect data every time a deck is read), it will protect against random errors that result in different errors each time the deck is read.

To verify a single deck, follow these steps:

1. Load the deck in the reader and press RESET.
 2. Select Binary or ASCII mode, as appropriate.
 3. In the Filename field., enter the filename under which the deck was stored.
 4. Click Verify.
 5. If the deck contains more cards than fit in the hopper, you may load additional cards and press RESET after each batch has run through.
- ➔ If you verify the deck immediately after reading it, you can skip steps 2 and 3.

It should not be necessary to press EOF when the deck is finished, as CardRead should encounter the end of the PC file at the same time the hopper empties. CardRead will display one of the following status messages:

- **verify OK**—indicates that the deck matched the data stored in the PC file.

- **VERIFY FAILED: Too many cards**—indicates that the PC file ended but the hopper is not empty.
- **VERIFY FAILED in col *n*, file had "xxxxxx", read "xxxxyxx"**— indicates that the character read from column *n* differed from the one stored in the file. A few characters on either side of the indicated column are displayed from both the file and the card (ASCII mode only).
- **VERIFY FAILED - Data differ in column *n***—indicates that the data read from column *n* differed from that stored in the file (Binary mode only).

In the event of a failure, the reader will stop with the problem card at the top of the stacker.

- ➔ If the program runs out of cards but doesn't display one of these messages, the cards that were read match those in the file, but somehow the second reading has come up short some number of cards. In this case, press EOF or click Stop to cancel the failed verification.
- ➔ If you are verifying a binary deck and encounter an error, turn on Display mode. CardRead will show the punch differences in red. A solid red rectangle indicates a punch on the card that was not in the file, and an open red rectangle indicates a punch in the file that was not on the card.
- ➔ If you can obtain two Documentation readers, accuracy may be considerably improved by reading decks with one reader and verifying them with the other.

7.1. Verifying Multiple Decks

If you are reading multiple decks and want to verify them, there are two ways that CardRead can assist you, depending on whether or not you are using separator cards. Use one of the following two procedures to minimize your effort:

If you are using autosep cards:

1. Enter the first filename, select Binary or ASCII mode, check Autosep, and select "Next File" to enable autoincrementing.
2. Click Start.
3. Load the deck(s) and press RESET to begin processing them.
4. When all of the decks have been read, press EOF or click Stop.
5. Re-enter the first filename.
6. Click Verify.
7. Reload the cards and press RESET to begin processing them.

In this case, verification will continue through each file as the autosep cards are encountered.

If you are *not* using autosep cards, you should read and verify each deck separately. Follow this procedure:

1. Enter the first filename, select Binary or ASCII mode, uncheck Autosep, and select "Verify" to enable automatic verification-then-next-file.
2. Click Start.
3. Load a deck and press RESET to start reading it.

4. When all of the cards have been read, press EOF or click Stop.
5. Reload the deck and press RESET to start verifying it.
6. If the verification fails, CardRead will stop after encountering a problem. If the verification is successful, CardRead automatically increments the filename and prepare to read another deck.
7. Repeat steps 3 through 7 for each successive deck. When the last deck has been read and verified, click Stop.

8. Viewing Binary Decks with DeckView

The DeckView utility lets you view and edit cards decks stored in CardRead's binary data format. The DeckView window is shown in Figure 9.

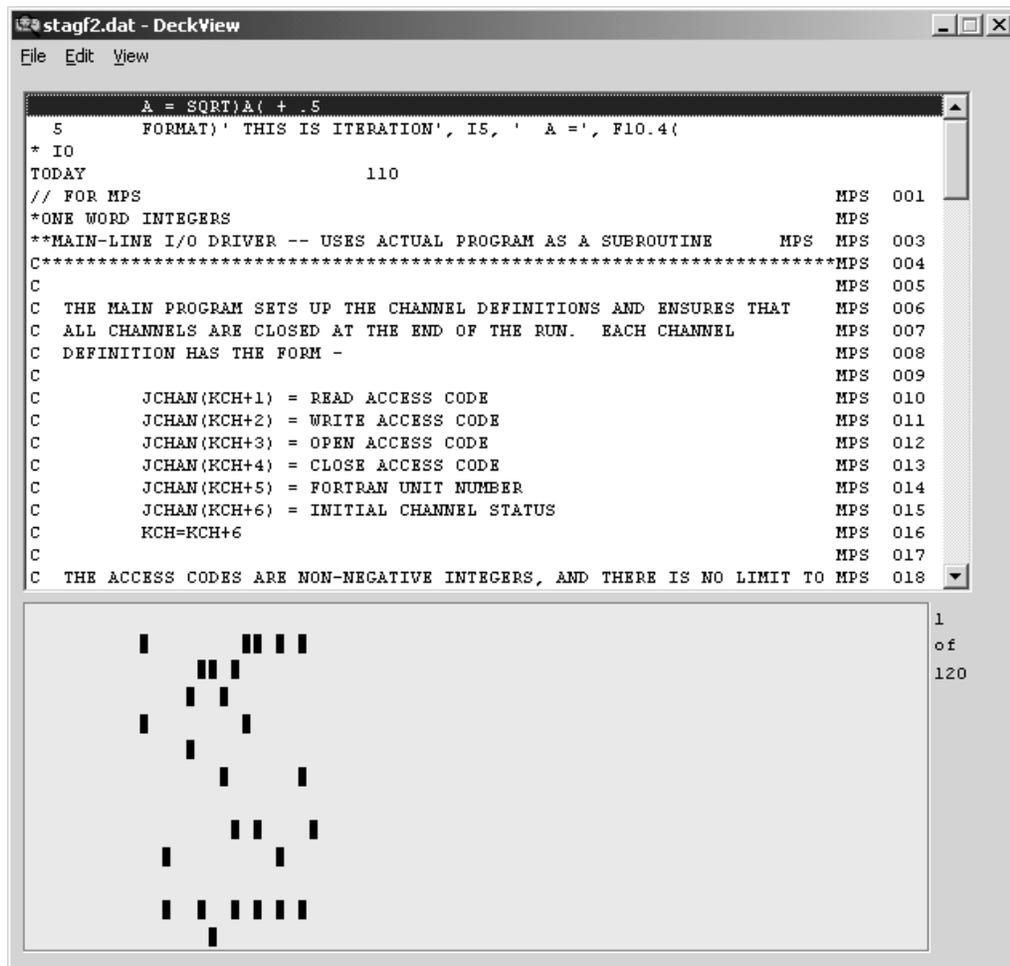


Figure 9. DeckView lets you view and edit binary card deck files

The program should be reasonably self-explanatory. The menu items are described here:

Menu	Item	Description
F ile	<u>N</u> ew	Opens an additional DeckView window.
	<u>O</u> pen	Opens a binary deck file.

Menu	Item	Description
	<u>S</u> ave	Saves the current deck file with the current filename.
	<u>S</u> ave As	Saves the current deck file into a file of your choosing. The Save As dialog also has a "Save As Type" option to let you save a copy of the deck as a text file.
	<u>E</u> xit	Terminates DeckView.
<u>E</u>dit	<u>C</u> ut Ctrl+X	Moves the selected line(s) to the clipboard. The data can be pasted as text into another program, if desired.
	<u>C</u> opy Ctrl+C	Copies the selected line(s) to the clipboard.
	<u>P</u> aste Ctrl+V	Pastes the text in the clipboard into the deck file above the line containing the cursor. The text is encoded using the currently selected encoding scheme. The text can come from another program, if desired.
	<u>D</u> elete Del	Deletes the selected lines.
	<u>F</u> ind	Lets you locate a specified text string.
	Find <u>N</u> ext F3	Repeats the search starting with the line below the cursor.
<u>V</u>iew	<u>G</u> oto Ctrl+G	Moves the cursor to a specific line number.
	next <u>S</u> equence Ctrl+S	Interprets card columns 73-80 as a sequence number field and moves the cursor to the next interruption in the sequence. Helpful in locating missing cards, edits and program groups within a combined deck.
	<u>E</u> ncoding	Lets you select a binary to ASCII encoding scheme. DeckView uses the same encoding schemes and configuration file as CardRead (see page 7).

Some important features are noted here:

- You can scroll up and down in the usual way.
- Ranges of text can be selected by holding down the Shift key
- You can cut and paste text between DeckView and other programs, or within DeckView files.
- You can select an encoding scheme from the View menu. Deckview uses the same encoding schemes and customization settings as CardRead.

Appendix 1. CDROM Contents

This section lists the contents of the accompanying CDROM. It includes source code for CardRead and the interface firmware, schematics, and references for the card reader and interface components.

\

cardread.doc	CardRead Manual (this document)
cardread.pdf	A .pdf version may be provided as well
cardread.exe	CardRead executable program
deckview.exe	DeckView executable program

usbdriver\

AN232-03.pdf	AppNote: Installing FT8U232/245 Devices Under Windows 98
AN232-05.pdf	AppNote: Installing FT8U232/245 Devices Under Windows 2000
	URL: www.ftdichip.com/FTApp.htm
.	FTDI VCP Device Driver for Windows 98/ME/2000/XP
	URL: www.ftdichip.com/FTWinDriver.htm

sources\

cardread.c	CardRead Source Code
cardread.h	
cardread.rc	
cr_encoding.c	
cr_encoding.h	
cardread.ico	
cardread.mak	
deckview.c	DeckView Source Code
deckview.h	
deckview.rc	
deckview.ico	
deckview.mak	
museumread.src	Interface Firmware Source Code for standard interface and for HP interface.
hpread.src	
museum schematic.doc	Interface Schematic for standard and HP interfaces. Standard interface uses inverted TTL signals, HP uses differential line drivers.
hp schematic.doc	

references\

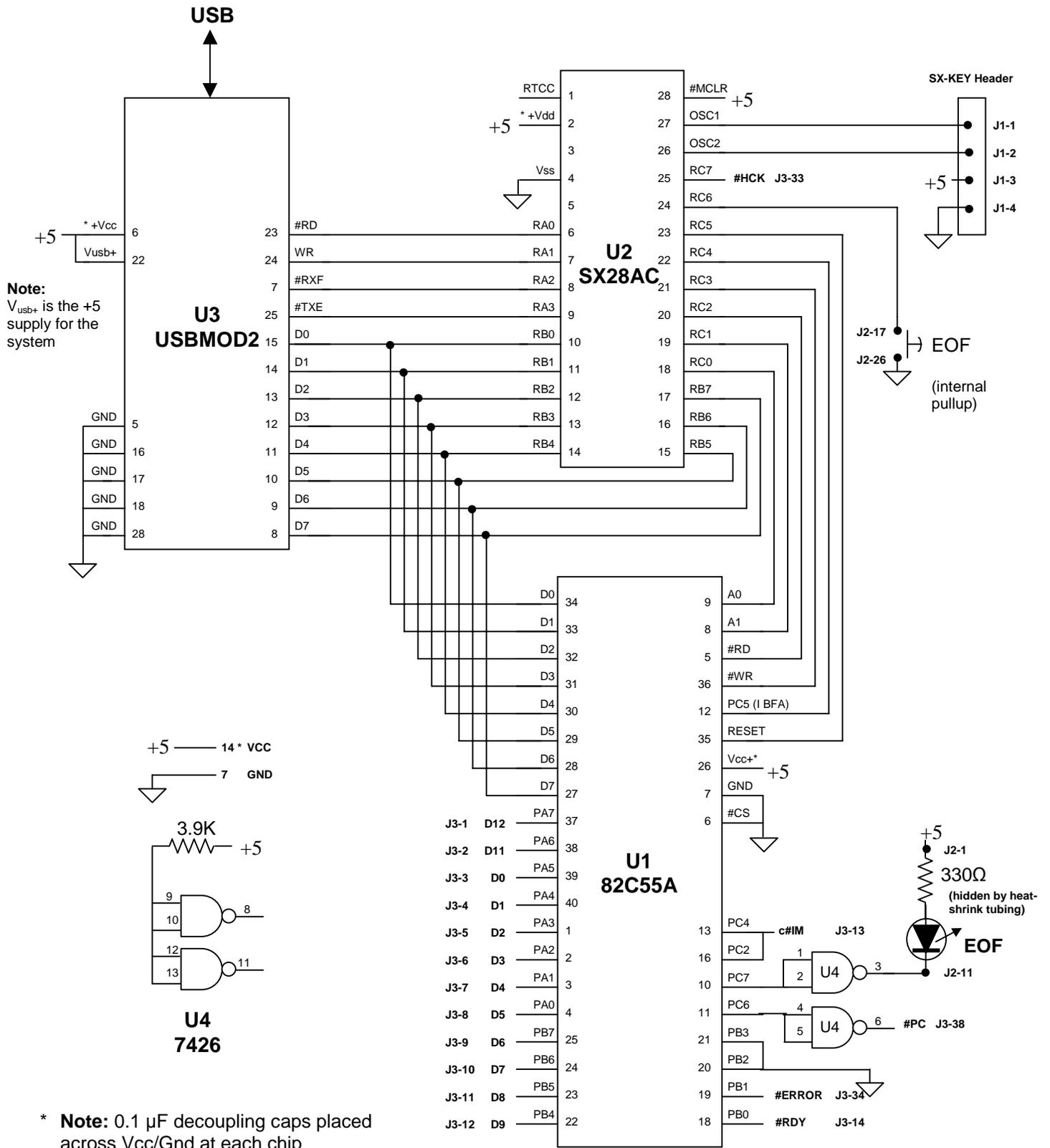
references\

Documation_M200.pdf	Documation M-Series Card Reader Manual URL: www.spies.com/~aek/pdf/documation/Documation_M200.pdf This is not quite the right manual for the Museum's M600, but it's very close and tables in the manual list differing parameter values for each model. Note: Brian Knittel (email: brian@quarterbyte.com) has an extender card that you can borrow if you need to troubleshoot the reader's plug-in circuit boards.
SX-DDS-SX2028AC-16.pdf	Ubicom SX-28AC Microcontroller Datasheet URL: www.ubicom.com/products/sx/sx.html
Setup SX-Key Editor.exe	Parallax SX-Key Development Software URL: www.parallax.com/sx/downloads.asp
SX_Key_Manual.pdf	Parallax SX-Key In-Circuit Programming/Debugging Interface URL: www.parallax.com/sx/downloads.asp (Note: as of 8/13/03, the manual is not up-to-date with the software)
USBMOD2DS.pdf	Gigatechnology USBMOD2 USB/Parallel Interface Module URL: www.gigatechnology.com/usbmodprod.html Note: there are some inaccuracies in the "Features" section, e.g. the transmit and receive buffer sizes are reversed. Refer to the FTDI data sheet (below) for interface specifications.
ds245b10.pdf	FTDI FT8U245BM USB FIFO Interface Chip Datasheet URL: www.ftdichip.com/FTProduct.htm
23125604.pdf	82C55A CHMOS Programmable Peripheral Interface Datasheet URL: www.intel.com/design/periphrl/datashts/231256.htm



Here's an idea: an ambitious person might build the interface with an SX-48 or SX-52 microcontroller, omit the 82C55A, and use the FT8U245M chip directly rather than using the USBMOD2 carrier. With a custom-built circuit board, the entire interface could be built into the 38-pin connector. Alternately, you might consider modifying the accompanying interface circuit to use a USBIO24 module (see e.g. www.dontronics.com) based on the Ubicom SX48, rather than the current design's SX28, USBMOD2 and 82C55A. This would be much simpler to build, and would let you view all of the Documation's status bits all the time. The USBIO24 uses a 5 MHz clock, not quite fast enough to both watch for the Documation's 1 μ s data strobe pulse *and* watch for incoming USB commands. However, if you connect the strobe to a port B pin you can have the strobe generate an interrupt. Firmware changes would be required; the current design already uses the interrupt for timing.

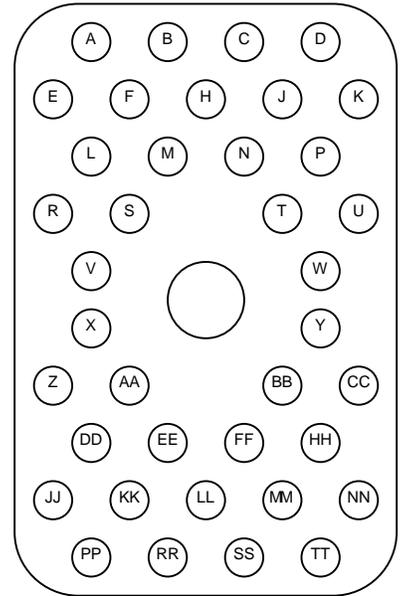
Appendix 2. Interface Schematic Logic



Cables and Connectors

Card Reader Signal	EDAC 516-038-000-301 Connector Pin	Wire	Cntrnics Pin	J2 Header pin*
D12+	A	brn	1	1
D12-	E			
D11+	B	brn/wht	2	2
D11-	F			
D0+	C	red	3	3
D0-	H			
D1+	D	red/wht	4	4
D1-	J			
D2+	K	org	5	5
D2-	P			
D3+	L	org/wht	6	6
D3-	R			
D4+	M	yel	7	7
D4-	S			
D5+	N	grn	8	8
D5-	T			
D6+	U	grn/wht	9	9
D6-	W			
D7+	V	lime	10	10
D7-	X			
D8+	Y	wht	11	11
D8-	CC			
D9+	Z	gry	12	12
D9-	DD			
IM+	AA	blk	13	13
IM-/GND	EE	pink	19-30, 33	21-32, 35
RDY+	BB	blk/wht	14	14
RDY-	FF			
ERROR+	HH	blu	32	34
ERROR-	NN			
HCK+	JJ	blu/wht	31	33
HCK-	PP			
MOCK+	KK			
MOCK-	RR			
BSY+	MM			
BSY-	TT			
PC+	LL	violet	36	38
PC-	SS			

View from BACK of plug on cable or face of socket on reader



View from BACK of J2

D12	1	21	Gnd
D11	2	22	"
D0	3	23	"
D1	4	24	"
D2	5	25	"
D3	6	26	"
D4	7	27	"
D5	8	28	"
D6	9	29	"
D7	10	30	"
D8	11	31	"
D9	12	32	"
#IM	13	33	#HCK
#RDY	14	34	#ERROR
		35	Gnd
	16	36	
	17	37	
	18	38	#PC
	19	39	
	20	40	

* **Notes:** ribbon cable pin 1 has a blue stripe. Row 1-20 is the one with the blank pin. The control and status signals are all active low. The #PC control signal is meant to be driven by an open collector driver.

Back side of header J3

Signal	Wire	Pin		Wire	Signal
LED +	ORG	1	14	ORG	Power
Power	BRN	2	15		
		3	16		
			17	WHI	Switch
		5	18		
		6	19		
		7	20		
		8	21		
		9	22		
LED -	YEL	10	23		
		11	24		
		12	25		
		13	26	BRN	Switch

Registers

82C55A Parallel Interface Controller

PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
in							
D12	D11	D0	D1	D2	D3	D4	D5

PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
in	in						
D6	D7	D8	D9	0	0	#ERROR	#RDY

PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
out	out	in	in	in	in	in	in
EOF LED	PC	IBF	--	--	--	--	--

SX28AC Microcontroller

RA - USB control

	RA3	RA2	RA1	RA0
	in	in	out	out
	#TXE	#RXF	WR	#RD

RB - data bus - read/write

RB7	RB6	RB6	RB4	RB3	RB2	RB1	RB0
i/o							
bus D7	bus D6	bus D5	bus D4	bus D3	bus D2	bus D1	bus D0

RC - 82C55 control, mostly

RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
in	in + pullup	out	in	out	out	out	out
#HCK	EOF button	RESET	IBF	#WR	#RD	A1	A0

Appendix 3. Interface Protocol

Windows programs communicate with the interface through a virtual serial port driver. From the Windows side, the port is treated like a normal Windows serial port, and the standard Communications API functions can be used. The definitive version of the protocol is in the interface firmware source code. The version current at the time of this writing is included here for convenience.

```
; -----  
; Protocol (send through USB using Virtual COM Port driver)  
; -----  
;  
; Single-character ASCII commands sent from host computer to the interface  
; are followed with a response as indicated in the description of each command.  
; The Reset command can be issued at any time. The remaining commands can be  
; issued while a pick is pending but not started, or when no pick is pending,  
; but not while a card is in the process of being read and its data transmitted,  
; that is, not between = and [status]. Commands issued during data transmission  
; are ignored.  
;  
; The commands are listed below.  
;  
; CMD DESCRIPTION  
; --- -----  
; R Reset (can be issued any time)  
; Response: [status]  
;  
; Resets the interface. If the card reader is ready, one card will be read  
; but ignored, due to an unavoidable pulse on the pick-command line.  
;  
; S Status check  
; Response: [status]  
;  
; Note: while a pick is pending, some status bits may not be accurate.  
; See the description of [status], below.  
;  
; P Pick one card  
; Response: [status]  
;  
; then, after reading card (which could be quite some time later)  
; = [card data] [status]  
;  
; or, if EOF button is pressed with hopper empty before a card is read  
; ! [status]  
;  
; After reading data, be sure to check the status byte for  
; the error bit, which would indicate a light or dark check -- bad news.  
; Status is sent 1.75 msec after last card column, to give time for  
; hopper check or read check to be set by the reader. Ready, however,  
; will usually not be back on by the time the status byte is sent.  
; For maximum picking rate, if hopper empty and EOF are not set, issue  
; another pick command immediately after receiving the status byte, if  
; even if ready is not set. You can poll for status afterward to  
; see if error comes up (if you care).  
;  
; X Cancel pending pick  
; Response: [status]  
;  
; C Clear EOF lamp. (Best to call only when hopper is empty)  
; Response: [status]  
;  
; (any unrecognized command)  
; Response: ?  
;  
; NOTE: The program talking to the card reader interface must be sure to read all  
; data from one card before a new pick is issued, otherwise the interface may lock  
; up. There is insufficient RAM in the interface to buffer more than one card.  
; If the FIFO overflows, columns will be missed, and the card data will not  
; be received completely. To ensure recovery from an overrun lockup, the calling  
; program should set a timeout of no less than 200 msec after receiving the =; if  
; 161 bytes aren't received in this time, issue an R reset command and recover.  
;  
; NOTE: the firmware accepts commands sent in upper or lower case. To see if  
; the interface is present, have your program on the PC send a lowercase 's'.
```



```

;
; -----
; [card data] - 160 bytes of data in the following format
; -----
;
; byte 0 [ 6 7 8 9 . . . .] column 1 (the four low bits are 0)
; byte 1 [12 11 0 1 2 3 4 5]
; byte 2 [ 6 7 8 9 . . . .] column 2 ( " )
; byte 3 [12 11 0 1 2 3 4 5]
; ...
; byte 158 [ 6 7 8 9 . . . .] column 80 ( " )
; byte 159 [12 11 0 1 2 3 4 5]
;
; The 160 bytes of card data are followed by a status character.
;
; When read as 16-bit integers in x86 word order, the card data appear
; in the format used by the IBM 1130 simulator:
;   MSB                                                                                               LSB
; +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
; | 12 | 11 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |   |   |   |   |   |   |   |   |
; +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
;
; The low four bits are zero.

```