Please do not upload this copyright pdf document to any other website. Breach of copyright may result in a criminal conviction.

This Acrobat document was generated by me, Colin Hinson, from a document held by me. I requested permission to publish this from Texas Instruments (twice) but received no reply. It is presented here (for free) and this pdf version of the document is my copyright in much the same way as a photograph would be. If you believe the document to be under other copyright, please contact me.

The document should have been downloaded from my website https://blunham.com/Radar, or any mirror site named on that site. If you downloaded it from elsewhere, please let me know (particularly if you were charged for it). You can contact me via my Genuki email page: https://www.genuki.org.uk/big/eng/YKS/various?recipient=colin

You may not copy the file for onward transmission of the data nor attempt to make monetary gain by the use of these files. If you want someone else to have a copy of the file, point them at the website. (https://blunham.com/Radar). **Please do not point them at the file itself as it may move or the site may be updated.**

It should be noted that most of the pages are identifiable as having been processed by me.

I put a lot of time into producing these files which is why you are met with this page when you open the file.

In order to generate this file, I need to scan the pages, split the double pages and remove any edge marks such as punch holes, clean up the pages, set the relevant pages to be all the same size and alignment. I then run Omnipage (OCR) to generate the searchable text and then generate the pdf file.

Hopefully after all that, I end up with a presentable file. If you find missing pages, pages in the wrong order, anything else wrong with the file or simply want to make a comment, please drop me a line (see above).

It is my hope that you find the file of use to you personally – I know that I would have liked to have found some of these files years ago – they would have saved me a lot of time !

Colin Hinson
In the village of Blunham, Bedfordshire.

TI-99/4 HOME COMPUTER

EIA RS232C PERIPHERAL

GENERAL SOFTWARE INTERFACE AND

OPERATIONAL SPECIFICATION

Consumer Group
Mail Station 5890
2301 N. University
Lubbock, Texas 79414

TEXAS INSTRUMENTS
INCORPORATED

TABLE of CONTENTS

Paragraph Title

SECTION 1

INTRODUCTION

This document describes the software interface and the operational specification of the RS232 peripheral as seen by a programmer using the BASIC language contained in the Texas Instruments 99/4 Home Computer. The RS232 peripheral will process OPEN, CLOSE, LIST, INPUT, PRINT, OLD, and SAVE commands. Any other commands will generate an ILLEGAL OPERATION error code.

The RS232 peripheral has been designed to recognize a SHIFT-C on the console as an abort command. This will terminate any pending or in progress operation and return back to the BASIC program with a DEVICE ERROR code.

1.1 Functional Capabilities

All functions are programmable from a BASIC program.

The hardware programable functions of the RS232 peripher are:

BAUD rates: 110, 300, 600, 1200, 2400, 4800, or 9600
Number of data bits: 7 or 8
Parity: none, odd, or even
Number of stop bits: 1 or 2

The software programable functions of the RS232 peripheral are:

* Turn off the automatic carriage return and linefeed, or linefeed only, after each VARIABLE length DISPLAY data type record.

* Disable the automatic echo of all read DISPLAY data type characters to the sending device.

* Enable the checking of parity for error detection.

The RS232 peripheral also contains a handshaking protocol with retransmit capability when doing an OLD or SAVE operation.

SECTION 2

APPLICABLE DOCUMENTS

Home Computer BASIC Language Specification
(Revision 4.1, 12 April 1979)

TI-99/4 Home Computer EIA RS232C Peripheral Detailed Software Functional Specification
(Version 2.0, Revised 28 March 1983)

SECTION 3

DEFAULT PARAMETERS

You may override any default condition by using the switch option as in the OPEN statement.

Device name RS232 is equivalent to RS232/1 and refers to port 1.

OPEN/INPUT/PRINT - 300 BAUD, 7 data bits, 1 odd parity bit, and 1 stop bit. All VARIABLE length DISPLAY data type records will automatically have carriage return and linefeed characters transmitted after them, unless this function is disabled. All characters received with DISPLAY data type will be echoed back to the sending device unless echoing is disabled. Parity will only be checked if enabled.

SAVE/OLD - 300 BAUD, 8 data bits, no parity bit, and 1 stop bit.

SECTION 4

RS232 PERIPHERAL BASIC LANGUAGE INTERFACE

The RS232 peripheral contains all the software necessary to interface the RS232 peripheral to the Home Computer file management system. The File Management subset implemented for the RS232 peripheral is comprised of the following INPUT/OUTPUT (I/O) routines:

OPEN, CLOSE, INPUT, PRINT, LIST, OLD, and SAVE.

The following sections describe actions taken by the RS232 peripheral software upon each I/O call. The RS232 peripheral will return with an ILLEGAL OPERATION error code for any other I/O call except OPEN, CLOSE, INPUT, PRINT, LIST, OLD, or SAVE.

4.1 OPEN Command

The OPEN command makes sure that all the OPEN attributes are valid, initializes the hardware to the desired conditions, and controls how the peripheral software functions.

The general format of the OPEN statement is:

OPEN #N:"<DEVICE NAME>"<FILE ATTRIBUTES>
or
OPEN #N:"<DEVICE NAME><SWITCH OPTION(S)>"<FILE ATTRIBUTES>

An OPEN statement with all default conditions looks like:

OPEN #1:XY
where X = "RS232.BAUD RATE=300.DATA BITS=7.PARITY=ODD"
or where X = "RS232.BA=300.DA=7.PA=O"
and where Y = ",UPDATE,SEQUENTIAL,DISPLAY,VARIABLE 80"
This is equivalent to: OPEN #1:"RS232"

NOTE

The character string between the two quotes (") has a maximum length of 255 characters and may be replaced by a string variable. Such as OPEN #N:A$<FILE ATTRIBUTES>.

#N is the integer file number that is to be assigned to th OPENed device. The value of N is between 1 and 255 inclusive.

<DEVICE NAME> is the name of the device that is to respond. The device names for the RS232 peripheral are: RS232, RS232/1, and RS232/2; where "RS232" is equivalent to "RS232/1". Ports 1 and 2 on the RS232 peripheral box are referred to by "/1" and "/2" respectfully.

<SWITCH OPTIONS> are the parameters from this section which are used to initialize the hardware and control how the software functions. Only the first two characters are actually need for switch recognition. You may override any default condition with these switch options. The switch options may be in any order desired.

NOTE

An OPEN statement has access to all switch options. An OLD or SAVE command may only use .BA, .PA, .CH, or .TW switch options, an attempt to use any other switch will generate an error.

Hardware Switch Options: [Default Value]

- .BAUD RATE - 110, [300], 600, 1200, 2400, 4800, or 9600
- .DATA BITS - 7 or 8 [7 for OPEN] [8 for LOAD/SAVE]
- .PARITY - ODD, EVEN, or NONE [O for OPEN] [N for LOAD/SAVE]
  (Only O, E, or N are actually needed)
- .TWO STOP BITS [1 stop bit]

Software Switch Options:

Defaults:
.NULLS[OFF] .CHECKPARITY[OFF] .ECHO[ON] .CR LF[ON] .LF[ON]

- .NULLS ON - Enable automatic NULLs after each CR
- .CHECK PARITY ON - Enables check for parity errors

SECTION 4

RS232 PERIPHERAL BASIC LANGUAGE INTERFACE

The RS232 peripheral contains all the software necessary to interface the RS232 peripheral to the Home Computer file management system. The File Management subset implemented for the RS232 peripheral is comprised of the following INPUT/OUTPUT (I/O) routines:

OPEN, CLOSE, INPUT, PRINT, LIST, OLD, and SAVE.

The following sections describe actions taken by the RS232 peripheral software upon each I/O call. The RS232 peripheral will return with an ILLEGAL OPERATION error code for any other I/O call except OPEN, CLOSE, INPUT, PRINT, LIST, OLD, or SAVE.

4.1 OPEN Command

The OPEN command makes sure that all the OPEN attributes are valid, initializes the hardware to the desired conditions, and controls how the peripheral software functions.

The general format of the OPEN statement is:

OPEN #N:"<DEVICE NAME>"<FILE ATTRIBUTES>
or
OPEN #N:"<DEVICE NAME><SWITCH OPTION(S)>"<FILE ATTRIBUTES>

An OPEN statement with all default conditions looks like:

OPEN #1:XY
where X = "RS232.BAUD RATE=300.DATA BITS=7.PARITY=ODD"
or where X = "RS232.BA=300.DA=7.PA=O"
and where Y = ",UPDATE,SEQUENTIAL,DISPLAY,VARIABLE 80"
This is equivalent to: OPEN #1:"RS232"

NOTE

The character string between the two quotes (") has a maximum length of 255 characters and may be replaced by a string variable. Such as OPEN #N:A$<FILE ATTRIBUTES>.

.ECHO OFF - Disable automatic echo and edit of received data

.CRLF OFF - Disable automatic carriage return and linefeed

.LF OFF - Disable automatic linefeed only

<FILE ATTRIBUTES> are also mentioned in the Home Computer BASIC Language Specification. The FILE ATTRIBUTES consist of the following:

Open Mode:

,INPUT - OPEN file for INPUT only

,OUTPUT - OPEN file for OUTPUT only

,UPDATE - (Default) OPEN file for INPUT and OUTPUT

,APPEND - Same as OUTPUT for the RS232 peripheral

File Organization:

,SEQUENTIAL - (Default)

,RELATIVE - Invalid for the RS232 peripheral

Record Type:
(N is the length of the records. If not given the RS232 peripheral defaults to 80 bytes [characters]).

,FIXED N - (INTERNAL data type Default) Fixed length records

,VARIABLE N - (DISPLAY data type Default) Variable length records

Data Type:

,DISPLAY - (Default) Usual for terminals and printers

,INTERNAL - Console software internal format. This data type is used to save record transfer time. See the Home Computer BASIC Language Specification for a description of how BASIC defines INTERNAL data type. See the TI-99/4 Home Computer EIA RS232C Peripheral Detailed Software Functional Specification for a description

of how the RS232 peripheral handles this data type.

4.2 CLOSE Command

The RS232 peripheral ignores the CLOSE command. However, BASIC does require that this command be executed so that the memory allocation for this OPENed device can be reallocated.

4.3 INPUT Command

The number of bytes (characters) that will be returned depends on the record type specified in the OPEN statement. For FIXED length or INTERNAL data type records, the RS232 peripheral will retain control until either the number of characters equals the logical record length (FIXED N where N is the logical record length), or the user types a SHIFT-C on the console and aborts the input. For VARIABLE length records with DISPLAY data type records, control will be retained until one of three conditions are met.

1. The detection of an ENTER or CARRIAGE RETURN (code decimal 13) character.

2. The logical record buffer size given in the OPEN has been filled, VARIABLE N, where N is the logical record size.

3. The user types a SHIFT-C on the console and aborts the input.

An extra degree of intelligence has been coded into the RS232 software for the case where the INPUT command is INPUTing from a terminal device using DISPLAY data type and the .ECHO OFF switch option is not given. This code recognizes the following special characters:

1. DELETE (decimal code 127) - Each time this character is received and DISPLAY data type records were OPENed, the previous character received will be deleted from the returned character buffer and echoed back to the terminal. This will mean that the characters will be displayed on the terminal in reverse order as they are deleted. If all the characters in the buffer have been

deleted a command to delete is just ignored.

2. CONTROL-R (decimal code 18) - This will cause the RS232 peripheral to echo back a carriage return, linefeed, and the current contents of the INPUT buffer. This feature would be used when some characters have been deleted and the user is not clear as to exactly what the current input line looks like.

4.4 PRINT Command

The number of bytes (characters) that will be PRINTed depends on the record type, the .CR and .LF switch options, and the data type. Carriage return and linefeed characters will be appended to all records unless either the .CR switch options is used, .LF switch option is used, or INTERNAL data type is specified. If the .CR switch option is used both the carriage return and the linefeed characters will not be appended to the output record. However, if the .LF switch option is used, the carriage return only is appended to the output record and the linefeed is not. The .LF switch should be used when exchanging text between two Texas Instruments series 99/4 Home Computers. INTERNAL data type does not output .CR or .LF characters and is treated as if it were fixed length records.

4.5 LIST Command

The way the BASIC LIST command actually functions is to automatically do an OPEN, multiple PRINTs, then a CLOSE which means that all the switch options that apply to the OPEN also apply to the LIST command.
>LIST "RS232" (Simple command: port 1, 300 BAUD, 7 data bits, 1 odd parity bit, and 1 stop bit. Automatic CR and LF will be added.)
>LIST "RS232/2.BA=9600" (Same as above, but port 2 and 9600 BAUD.)
>LIST "RS232/1.LF" (Same as LIST "RS232", but no automatic LF after CR.)

4.6 OLD Command

See section 5.0 for an operational explanation of how this command is to be used and how it functions. The switch options

.BA, .PA, .CH, and .TW described in section 3 are available the OLD command. This command is used to load a new progr image through the RS232 peripheral and is expecially designed for communication with another Texas Instruments 99/4 Home Computer using the SAVE command.

4.7 SAVE Command

See section 5 for an operational explanation of how this command is to be used and how it functions. The switch options .BA, .PA, .CH, and .TW described in section 3 are available to the SAVE command. This command is used to SAVE or exchange a program image through the RS232 peripheral and is especially designed for communication with another Texas Instruments 99/4 Home Computer using the OLD command.

SECTION 5

SAMPLE PROGRAMS AND COMMANDS

EXAMPLE 1: This will OPEN the RS232 peripheral for UPDATE mode using all the default parameters: Port 1, 300 BAUD, 7 data bits, 1 odd parity bit, and 1 stop bit. a VARIABLE length record will be INPUT from port 1, then the message "ENTER TEXT NOW " will be written to port 1 followed by the actual INPUT text as received by the BASIC interpreter enclosed in single quotes "".

NOTE

DISPLAY data type will be used which means that the DELETE, BACKSPACE, and CONTROL-R features described in section 3.3 can be used during the INPUT on the remote terminal.

```
100 OPEN #1:"RS232"
200 PRINT #1: "ENTER TEXT NOW "
300 INPUT #1:A$
400 PRINT #1:"'";A$;"'"
500 IF A$<>"" THEN 200
600 CLOSE #1
```

EXAMPLE 2: This program will OPEN port 1 and 2 and then copy one character at a time from port 1 to port 2. The parameters are: 9600 BAUD, 7 data bits, 1 odd parity bit, and 1 stop bit. Parity will be checked and a DEVICE ERROR code will be returned if an error occurs. The .EC switch option is needed to turn off the automatic edit and echo. The .CHECK PARITY switch option is needed to enable parity checking.

```
100 OPEN #1:"RS232/1.BA=9600.EC.CH",FIXED 1
200 OPEN #2:"RS232/2.BA=9600.EC.CH",FIXED 1
300 INPUT #1:A$
400 PRINT #2:A$
500 IF A$<>"" THEN 300
600 CLOSE #1
700 CLOSE #2
```

EXAMPLE 3: Variable length ASCII text messages may be communicated between two Home Computers by using the following

programs. The parameters are: 300 BAUD, 7 data bits, 1 o
parity bit, and 1 stop bit. Parity errors will be ignored. T
.LF and .EC switch options are needed to turn off the automatic
LF and echo.

```
(One user runs this program) [BAUD rate may be changed.]
100 OPEN #1:"RS232.EC.LF",VARIABLE 255
200 INPUT A$
300 PRINT #1:A$
400 INPUT #1:B$
500 PRINT B$
600 GOTO 200

(The other user runs this program) [BAUD rate may be changed.]
100 OPEN #1:"RS232.EC.LF",VARIABLE 255
200 INPUT #1:A$
300 PRINT A$
400 INPUT B$
500 PRINT #1:B$
600 GOTO 200
```

EXAMPLE 4: This program will OPEN file 1, 2, and 3, all on port 1. File 2 will accept variable length ASCII text input but will not echo it back to the terminal. File 3 will PRINT data and the cursor or carriage return will stay on the same line after it is printed. File 1 will automatically put a carriage return and linefeed characters after every output record. If t
response to "ENTER PASWORD " is "LEMEIN" then "PASWORD IS OK
will be printed, otherwise "TRY AGAIN" will be printed.

```
100 OPEN #1:"RS232"
110 OPEN #2:"RS232.EC"
120 OPEN #3:"RS232.CR"
130 PRINT #3:"ENTER PASWORD "
140 INPUT #2:A$
150 IF A$<>"PASWORD" THEN 180
160 PRINT #1:"TRY AGAIN"
170 GOTO 130
180 PRINT #1:"PASWORD IS OK"
190 GOTO 130
```

SECTION 6

PROCEDURE FOR PROGRAM EXCHANGE WITH OLD/SAVE COMMAND

The exchange of programs using the RS232 interface was designed for ease of program exchange through modems. All that is required to establish this program exchange link is for two users to call each other on the phone, place the phones on their respective modems, then:

The user on the SAVE side will type a minimum of:
SAVE "RS232", SAVE "RS232/1", or SAVE "RS232/2".

The user on the OLD side will type a minimum of:
OLD "RS232", OLD "RS232/1", or OLD "RS232/2".

It makes no difference who types first or how long it takes for the other user to respond once the first user has typed their part. However, establishing the link is more reliable if the SAVE command is typed before the OLD command. If the OLD command is typed first, noise on the line can be interpreted as a start of input from the SAVE which will cause the OLD side to hang up. If after both sides have typed their part and the link is not established within 15 seconds, the OLD side should type SHIFT-C on the consold and retype the OLD command which should create the link immediately. There is no timeout, however typing a SHIFT-C on the console will abort the operation at any time. Once the data link is established, the exchange of the program image is completely under program control.

When either OLD or SAVE commands are executed with the RS232 peripherals, a three digit number, initially 255, will be displayed at the center of the top line on the display screen. The value will stay 255 until the program exchange link is established at which time it will change to the remaining number of 256 byte records to be transferred. Each time a good record is exchanged, the number will count down toward zero. This countdown may then be used as a visual indication of the status of the program exchange and should decrement at one of the following rates:

110 BAUD = 23.5 seconds
300 BAUD = 8.6 seconds
600 BAUD = 4.3 seconds
1200 BAUD = 2.5 seconds
2400 BAUD = 1.1 seconds

4800 BAUD = 0.5 seconds
9600 BAUD = 0.3 seconds

The switch options .BA, .PA, .CH, and .TW described in Section 3.1 may be used with the OLD or SAVE commands.

```
SAVE "RS232.BA=600.PA=0.CH"
  and
OLD "RS232.BA=600.PA=0.CH"
```

About 8 minutes will be required to exchange a 14K program. A typical program of 3.5K will require about 2 minutes. In general each K (1024 bytes) transferred will require about 35 seconds at 300 BAUD. Under certain conditions about 7 seconds are required to establish the link after both users have typed their parts.

SECTION 7

INTERFACE RESTRICTIONS

Although both ports may be OPENed at the same time, only one port may be active. However the hardware does have the ability to buffer a single character. If a port is OPENed for INPUT and the character in the hardware buffer does not get accepted before the next data byte is received, an OVERRUN ERROR will occur. BASIC will see this as a DEVICE ERROR code on the next INPUT from the same port.

If a port is OPENed for INPUT and N characters are read, but N+2 characters were transmitted, the next time an INPUT is done from that port a DEVICE ERROR code will be generated. This is because the hardware can only buffer one character and two have come in which causes an OVERRUN ERROR.

Whenever 110 BAUD rate is selected, the number of stop bits should be set to two with the clause ".TW" or ".TWO STOP BITS".

SECTION 8

SPECIAL COMMENTS

For an INPUT command:

1. INTERNAL data type and use of the .ECHO OFF switch option will disable the remote edit feature.

2. When the .ECHO OFF switch option is not used, the .CRLF OFF and the .LF OFF determine if the echoed records are followed by automatic CR and LF characters.

3. .ECHO OFF overrides .CRLF OFF which overrides .LF OFF switch options.

4. When FIXED length DISPLAY type data records are specified, the RS232 peripheral retains control until the exact number of characters asked for are input. Carriage returns and any other control characters, except the delete character and the CONTROL-R, are treated just like any other ASCII character. The delete and CONTROL-R perform the editing functions described in section 3.3. A SHIFT-C can abort an INPUT command for maore INPUT data from the RS232 peripheral.

5. The .ECHO OFF switch option disables both the automatic echo of the INPUT characters to the sending device and the interpretation of the delete and CONTROL-R characters for editing purposes.

For a PRINT command:

1. The .ECHO OFF switch option has no effect on the output data.

2. The .CRLF OFF overrides the .LF OFF switch option.

3. The .CRLF OFF and the .LF OFF switch options have no effect when fixed length records are selected. The exact number of characters asked for is output exactly as given to the peripheral.

For both INPUT and PRINT commands:

1. The .NULLS ON and the .CHECK PARITY ON switch options function independent of the .ECHO OFF, .CRLF OFF, and the .LF OFF switch options.

2. A given RS232 peripheral port can be OPENed more than once at the same time with different software switch options so long as all the hardware switch options given are the same. See EXAMPLE 5, section 4.0 and SWITCH OPTIONS, section 3.1.

3. The syntax of the OPEN statement switch option string is free form with the following conditions:

   a. There can be no spaces between the Device Name, the period ".", and the first two characters of the first switch.

   b. Once past the first switch XX, "RS232.XX .YY", there can be any number of spaces until the next period ".". When the period is found the next two characters must be the first two characters of the switch option name.

   c. When a switch option has the form ".BAUD RATE = 9600" after the "BA" all characters are ignored until the "=". After the "=" spaces are ignored until the first character. After the second zero in 9600 all spaces are ignored until the next "." is found.

SECTION 9

ERRORS AND ERROR CODES

The meaning of the error codes are as follows:

X1 = BAD DEVICE NAME (Generated by BASIC interpreter)
X2 = BAD ATTRIBUTES
X3 = ILLEGAL OPERATION
X4 = OUT OF TABLE/BUFFER SPACE
X5 = DEVICE ERROR

OPEN:
CODE 02 - The switch option entry is in error. Incorrect first two characters of a switch, invalid BAUD rate, or incorrect number of data bits. RELATIVE record type specified in OPEN statement.
CODE 06 - Console peripheral frequency not compatible with RS232 peripheral frequency of 2.5 or 3.0 MHz.

CLOSE: No errors are generated during a CLOSE statement.

INPUT:
CODE 24 - INTERNAL data type record is too large to be read into the buffer space allocated.
CODE 26 - Some type of hardware error; either a framing, overrun, or parity error occurred. Abort of a pending or in progress operation with a SHIFT-C on the console keyboard.

PRINT: No errors are generated during a PRINT statement.

OLD:
CODE 52 - Attempt to use .EC, .CR, .LF, .NU, or .DA switch during OLD statement. See also CODE 02.
CODE 54 - The program is too large to be loaded.
CODE 56 - See CODE 26.

SAVE:
CODE 62 - Attempt to use .EC, .CR, .LF, .NU, or .DA switch during SAVE statement. See also CODE 02.
CODE 66 - See CODE 26.

MISC ERROR CODES:
CODE 43 - Attempt to execute a RESTORE command which is illegal.

CODE 73 - Attempt to execute a DELETE command which is illegal.
CODE 83 - Attempt to execute a SCRATCH command which is illegal.
CODE 93 - Attempt to execute a STATUS command which is illegal.

SECTION 10

DECIMAL CODED ASCII TABLE

DEC	C H A R A C T E R	DEC	CHAR	DEC	CHAR	DEC	CHAR
0	NULL (Blank)	32	SPACE	64	@	96	`
1	SOH (Start of Header)	33	!	65	A	97	a
2	STX (Start of Text)	34	"	66	B	98	b
3	ETX (End of Text)	35	#	67	C	99	c
4	EOT (End of Transmission)	36	$	68	D	100	d
5	ENQ (Enquiry)	37	%	69	E	101	e
6	ACK (Positive Acknowledge)	38	&	70	F	102	f
7	BEL (Bell)	39	'	71	G	103	g
8	BS (Backspace)	40	(	72	H	104	h
9	HT (Horizontal Tabulation)	41	)	73	I	105	i
10	LF (Linefeed)	42	*	74	J	106	j
11	VT (Vertical Tabulation)	43	+	75	K	107	k
12	FF (Form Feed)	44	,	76	L	108	l
13	CR (Carriage Return)	45	-	77	M	109	m
14	SO (Shift Out)	46	.	78	N	110	n
15	SI (Shift In)	47	/	79	O	111	o
16	DLE (Data Link Escape)	48	0	80	P	112	p
17	DC1 (Device Control 1)	49	1	81	Q	113	q
18	DC2 (Device Control 2)	50	2	82	R	114	r
19	DC3 (Device Control 3)	51	3	83	S	115	s
20	DC4 (Dev Ctl 4 - Stop)	52	4	84	T	116	t
21	NAK (Negative Acknowledge)	53	5	85	U	117	u
22	SYN (Synchronization)	54	6	86	V	118	v
23	ETB (End of Text Block)	55	7	87	W	119	w
24	CAN (Cancel)	56	8	88	X	120	x
25	EM (End of Medium)	57	9	89	Y	121	y
26	SUB (Substitute)	58	:	90	Z	122	z
27	ESC (Escape)	59	;	91	[	123	{
28	FS (File Seperator)	60	<	92	\	124	|
29	GS (Group Seperator)	61	=	93	]	125	}
30	RS (Record Seperator)	62	>	94	^	126	~
31	US (Unit Seperator)	63	?	95	_	127	DELET

TI-99/4 HOME COMPUTER

EIA RS232C PERIPHERAL

DETAILED SOFTWARE FUNCTIONAL SPECIFICATION

Consumer Group
Mail Station 5890
2301 N. University
Lubbock, Texas 79414

TEXAS INSTRUMENTS
INCORPORATED

Date: March 28, 1983

TABLE of CONTENTS

Paragraph Title

SECTION 7 INTERFACE RESTRICTIONS AND SPECIAL COMMENTS

LIST of FIGURES

SECTION 1

INTRODUCTION

References in this document to the "General Specification" are references to the TI-99/4 Home Computer EIA RS232C Peripheral General Software Interface and Operational Specification. The "General Specification" should be read before reading this document because an attempt is made to not duplicate information. This specification is written to the TMS9900 or Graphics Language code users, but the "General Specification" is written to the BASIC language user. The purpose of this specification is to document the file management interface to the RS232C peripheral DSR, comment on those features not described in the "General Specification", and to add more detail illustrating data formats.

This document describes an RS232C peripheral DSR for the TI-99/4 Home Computer. This peripheral is built around two TMS 9902s asynchronous communications controllers, which provide an interface between a microprocessor and two serial, asynchronous communications channels. This peripheral accepts the EIA Standard RS232C interface protocol. The circuit design includes two TMS 9902s allowing two ports (port 1 and port 2). Both ports may be OPENed at the same time, but only one at a time may be accessed from BASIC.

In addition to those functions described in the other document, one additional hardware option is available which can not be used by a BASIC language program caller. This is the Circular Interrupt Input Buffer option and was written for use by Graphics Language or TMS 9900 code application programs. See Section 6.0 in this document. This peripheral has been designed to operate at peripheral frequencies of 3.0 and 2.5 MHz. The byte at console ROM location HEX 000C determines the frequency:

HEX 30 = 3.0 MHz and HEX 28 = 2.5 MHz.

This is needed to obtain the correct BAUD rates at the two frequencies.

SECTION 2

APPLICABLE DOCUMENTS

File Management Specification for the TI-99/4 Home Computer
(Version 2.5, Revised 25 February 1983)

TMS9902 Asynchronous Communications Controller Specification Sheet

Home Computer BASIC Language Specification
(Revision 4.1, 12 April 1979)

TI-99/4 Home Computer EIA RS232C Peripheral General Software Interface and Operational Specification z1b
(Version 2.0, Revised 28 March 1983)

SECTION 3

RS232 INTERFACE DESCRIPTION

This section will describe the RS232C interface protocol. It contains a description of the individual interface signals, and the way in which they are used within the protocol.

The following two sections describe the RS232C interface signals as if the interface peripheral was connected to a modem (active) device. If the interface peripheral is connected to a terminal device, the roles of the interface peripheral and the connected device will be reversed.

3.1 RS232 Signal Description

The RS232C protocol accepted by the TMS 9902s involves 3 input signals and 2 output signals. These 5 signals are:

Signal	I/O	Description
CTS	I	Clear To Send
DSR	I	Data Set Ready
RTS	O	Request To Send
RIN	I	Receiver Serial Data Input
XOUT	O	Transmitter Serial Data Output

An individual description of each of the control signals is given below:

CTS - Clear To Send - Input from the modem device to RS232C interface. When set, it enables the transmit section of the RS232C interface unit.

DSR - Data Set Ready - Input from the modem device to RS232C interface. Indicates that the modem device is ready to receive data.

RTS - Request To Send - Output to the modem device. Indicates that the RS232C interface is ready for data-transmission to the modem device. This signal has to remain set during the actual transmission.

The RIN and XOUT signals are the actual transmission signa' between the modem device and the RS232C interface.

3.2 RS232 Protocol

The protocol used for the RS232C interface involves the control signals: RTS, CTS, and DSR; as described in section 3.1. The relationship between these three control lines is given in the following state diagram.

Figure 3-1 RS232C Transmission Cycle

Figure 3-1 describes the transmission cycle for the RS232C interface peripheral. Once the interface sets the RTS line, it has indicated that it is ready for transmission. The modem device then has to answer by setting the CTS line, indicati that it is ready for reception of the data-stream. The R. signal remains set during the entire actual transmission of the data.

Transmission from modem device to the RS232C interface peripheral is controlled by only one control line, the DSR line. This line indicates that the modem device is ready for transmission. One extra line, the DTR or Data Terminal Ready, which is defined in the standard RS232C protocol, but is not supported by the TMS 9902s, indicates to the modem device that is can start transmission. This line is continuously held high in the Home Computer RS232C peripheral.

SECTION 4

STANDARD I/O COMMAND HANDLING

The RS232 peripheral contains all the software necessary to interface the RS232 peripheral to the Home Computer file management system. This not only includes READ/WRITE routines, but also LOAD/SAVE routines. The File Management subset implemented for the RS232 peripheral comprises the following I/O routines: OPEN, CLOSE, READ, WRITE, LOAD, and SAVE. The following sections describe actions taken for each I/O call.

4.1 OPEN

See the "General Specification", Section 3.1. The only invalid attribute in the file attributes section of the PAB is specifying RELATIVE record type. All other file attributes are understood by this peripheral. An OPEN command with the most significant bit set in the operation code section of the file management PAB will enable the Circular Interrupt Input Buffer. See Section 6.0 in this document describing the Circular Interrupt Input Buffer. /p1(_CLOSE}) The RS232 peripheral ignores the CLOSE command. However, BASIC does require that this command be executed so that the memory allocation for this OPENed device can be reallocated.

4.2 READ

See the "General Specification", Section 3.3. When INTERNAL data type has been specified the first byte read is interpreted as the byte count of the remaining number of bytes to be read. Once this byte count has been read the READ command functions as if FIXED length records of this length were specified. This byte count is transparent to the caller of the DSR. N+1 bytes are received but the caller gets the last N bytes.

4.3 WRITE

See the "General Specification", Section 3.4. When INTERNAL data type has been specified the first byte of the record

actually transmitted will be the actual character count of th number of bytes in the record followed by the record. Th. actual character count is transparent to the caller of the DSR. N+1 bytes are transmitted for an N byte record.

4.4 LOAD/SAVE

See the "General Specification", Sections 3.6, 3.7, and 5.0. When LOADing or SAVEing through the RS232 interface over modems there is handshaking involved. This handshaking takes each data block, described below, and outputs one block at a time. The handshaking starts with the LOAD section continuously sending a SYN, synchronization character decimal 22, every ~7 seconds to the SAVE section which watches its input for data to start coming in. As soon as the SAVE section sees the SYN character it starts transmitting the program image as described below. The first data block transfered contains 2 bytes of program image byte count followed by a two byte CRC check code for these two bytes. After outputing each block the SAVE section waits for either an ACK (Positive Acknowledge decimal 6) or a NAK (Negative Acknowledge decimal 21). For a NAK the last block is retransmitted, but for an ACK the next data block is transmitted.

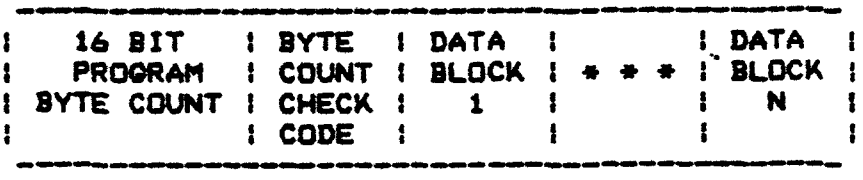

Figure 4-1 Program Image for LOAD/SAVE thru EIA RS232C Interface

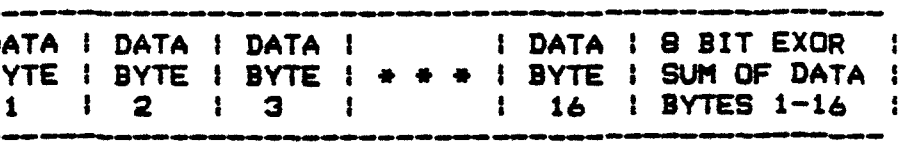

Figure 4-2 Data Block Format

The program transfer image consists of the program data image byte count and CRC check code for it followed by as many 256 byte data blocks as needed, the last of which will be variable length, if needed.

The format of a data block will be 256 data bytes followed by a two byte cyclic redundancy check code for the data bytes.

The two byte CRC check code is generated by the following TMS 9900 Assembly Language code:

```
 *
 * INPUTS:  R6 - MSbyte = character to add to CRC
 *          R9 - Current CRC check code value
 * OUTPUTS: R9 - Updated CRC check code value
 *
 * The CRC             16    12    5
 * polynomial is     X   +  X   +  X  +  1
 *
CRCALC MOV  R6,R1          Copy new data to work reg
       ANDI R1,>FF00       Clear LSbyte
       XOR  R1,R9          XOR new data with old CRC
       MOV  R9,R1          Move to scratch register
       SRL  R1,4           Shift right logical 4
       XOR  R9,R1          XOR CRC with scratch
       ANDI R1,>FF00       Clear LSbyte
       SRL  R1,4           Shift right logical 4
       XOR  R1,R9          XOR scratch with CRC
       SRC  R1,7           Shift right circular 7
       XOR  R1,R9          XOR scratch with CRC
       SWPB R9             Reverse bytes of CRC
       RT                  Return
```

SECTION 5

GRAPHIC LANGUAGE INTERFACE

See the File Management Specification for the TI-99/4 Home Computer and the "General Specification", Section 3.0, for a complete description of the PAB options syntax of the device name. The switch options described in the "General Specification", Section 3.1, have the same syntax for both BASIC and Graphics Language callers.

SECTION 6

CIRCULAR INTERRUPT INPUT BUFFER OPERATION

NOTE

This feature can not be used in a BASIC language program because of the common CPU RAM usage, locations 0-5.

This option is enabled by calling the RS232 peripheral DSR with an operation code of HEX 80, most significant bit set plus an OPEN operation code in the I/O opcode (byte 0) of the PAB. The normal processing for an OPEN command is executed but the receive interrupt is enabled as part of the OPEN.

The CPU RAM usage is as follows:

LOCATION	MEANING
0-1 (2 bytes)	Address of start of buffer area
2 (1 byte)	Length of buffer (1-255 bytes)
3 (1 byte)	Callers read offset value
4 (1 byte)	RS232 DSR write offset value
5 (1 byte)	Not Defined

When the RS232 gets an interrupt it attempts to store the input data byte to VDP memory address (word 0 + byte 4 offset + 1) address. When (byte 3 offset) = (byte 4 offset + 1), an overrun error is declared and the data byte at address (word 0 + byte 4 offset) is overwritten with a >FE as long as this condition is present. If an input hardware framing, overrun, or parity error occurs and there is not an overrun condition, a >FF is returned as the character code. When (word 0 + byte 4 offset + 1) > (byte 2), the write offset (byte 4) is set to zero and (word 0 + byte 4 offset) is used as the write address. This functions as a circular interrupt buffer.

A user of this feature should read data whenever (byte 3) <> (byte 4). Each time a data byte is read (byte 3) should be incremented by one. In order to read data, increment (byte 3) to the next offset and then use the address (word 0 + byte 3). When (word 0 + byte 3 offset + 1) > (byte 2), the read offset (byte 3)

is set to zero and (word 0 + byte 3 offset) is used as the re address.

When input is done in this manner the software switch options described in the "General Specification", Section 3.1, have no effect. However the hardware switch options control the input TMS 9902.

The output section of the port that is OPENed in this manner may still be used so long as the hardware switch options are the same. One OPEN statement can OPEN input for interrupt mode and output with switch options because the interrupt input ignores the software switch options.

SECTION 7

INTERFACE RESTRICTIONS AND SPECIAL COMMENTS

Even though the TMS 9902 can be programmed to have 1.5 stop bits, this option is not given as one of the software switch options.

Even though the TMS 9902 can be programmed to generate 5 and 6 data bits, this option is not given as one of the software switch options.

There is no name associated with the LOAD/SAVE operation because this would require additional protocol and overhead.

Since there is only one set of pointers in CPU RAM for the circular interrupt input buffer, only one port should be opened in this mode at a time. However, if desired and it is not necessary to know from which port the data came, multiple ports may be opened in this manner all of which will use the same pointers.

Because there is only one TMS 9900 microprocessor in the console which drives the peripheral DSRs and DSRs do not generally enable interrupts, only one event at a time can be serviced. This means for example, when the circular interrupt input buffer option is selected, no other DSR should be called or data could be lost if it were to come in.