TI-99/4 HOME COMPUTER

EIA RS232C PERIPHERAL

DETAILED SOFTWARE FUNCTIONAL SPECIFICATION

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

Date: March 28, 1983
Version 2.0

TEXAS INSTRUMENTS
INCORPORATED

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 zlb
(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 signals 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, indicating that it is ready for reception of the data-stream. The RTS 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 the number of bytes in the record followed by the record. This 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.

16 BIT PROGRAM BYTE COUNT	BYTE COUNT CHECK CODE	DATA BLOCK 1	* * *	DATA BLOCK N

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

DATA BYTE 1	DATA BYTE 2	DATA BYTE 3	* * *	DATA BYTE 16	8 BIT EXOR SUM OF DATA BYTES 1-16

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 rea 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.