MSX RS-232-C

REFERENCE GUIDE

SVI
SPECTRAVIDEO

MSX

MSX RS-232-C

REFERENCE GUIDE

Published by
SPECTRAVIDEO INTERNATIONAL LTD.

First edition
First printing 1985
Printed in Hong Kong

Spectravideo International Ltd. shall not be liable in any event for claims of incidental or consequential damages resulting from the furnishing, performance, or use of this material.

Every effort has been made to supply complete and accurate information in this manual. Nevertheless, due to our never a ending commitment to improve both product design and performance, we reserve the right to change product specifications at anytime without prior notice.

TABLE OF CONTENTS

CHAPTER PAGE

INTRODUCTION

This manual describes the support offered by MSX hardware and software for RS-232-C communication. It is written for the experience programmer who seeks quick reference to those BASIC commands, BIOS function calls, and hardware addresses that are related to the RS-232-C interface. Knowledge of computers in general and BASIC in particular is assumed. A less technical treatment of the more commonly used features can be found in your User's Guide.

SYNTAX NOTATION IN REFERENCE SECTIONS

Wherever the format for a statement/command or 4 function is given, the following rules apply:

CAPS Items in capital letters must be input as shown.

<> Items in lowercase letters enclosed in angle brackets (<>) are to be supplied by the user.

[] Items in square brackets ([ ]) are optional.

... Items followed by an ellipsis (...) may be repeated any number of times (up to the length of the line).

{} Braces indicate that the user has a choice between two or more entries. At least one of the entries enclosed in braces must be chosen unless the entries are also enclosed in square brackets.

| Vertical bars separate the choices within braces. At least one of the entries separated by bars must be chosen unless the entries are also enclosed in square brackets.

All punctuation except angle brackets and square bracket (i.e., commas, parentheses, semicolons, hyphens, equal signs) must be included where shown.

Arguments to functions are always enclosed in parentheses. In the formats given for the functions in this book, the arguments are abbreviated as follows:

X and Y Represent any numeric expressions.

I and J Represent integer expressions.

X$ and Y$ Represent string expressions.

1. EXTENDED BASIC FOR RS-232-C COMMUNICATION

This section describes the specifications of the support for the RS-232-C communication interface on MSX computers.

1) Set up Communication Parameters

```
CALL COMINI [([<string exp>][,[<Rx baud rate>][,[Tx baud rate>][,[<time out>]]]])]
```

Initializes an RS-232-C port with the specified parameters. The <string exp> is a string that specifies the channel control parameters. See the detailed description below.

BAUD RATE

It is possible to set a different baud rate for transmitter and receiver. The possible baud rates are as follows:

50	110	600	1800	2400	4800	9600
75	300	1200	2000	3600	7200	19200

When only the receiver's baud rate is specified, the baud rate for transmitter assumes the same speed as the receiver. When only the transmitter's baud rate is specified, the baud rate the receiver is set to the default value.

If a negative value is specified, its absolute value is written to i8253 Timer/Counter directly.

TIME OUT

The RS-232C driver waits for the CTS (Clear To Send) to turn on or/and XON is received when the character is sent. The driver generates a time out error if the specified time has elapsed. The value for the time out error is specified in seconds. If zero (0) is specified, the driver does not generate a time out error, and the driver waits indefinitely.

STRING FIELDS

```
            "[0:] [8[N[1[X[N[N[N[N]]]]]]]]"
              |    | | | | | | | |
              |    | | | | | | | |
              +----+-+-+-+-+-+-+-+-- Channel Number
                   | | | | | | | |   When the system has more
                   | | | | | | | |   Than one channel, this
                   | | | | | | | |   Parameter specifies the
                   | | | | | | | |   Channel number, and it may be
                   | | | | | | | |   Omitted if the system has
                   | | | | | | | |   Only one channel. The
                   | | | | | | | |   Default value is 0.
                   +-+-+-+-+-+-+-+-- Data length
                     | | | | | | |   "5": 5 bits
                     | | | | | | |   "6": 6 bits
                     | | | | | | |   "7": 7 bits
                     | | | | | | |   "8": 8 bits
                     +-+-+-+-+-+-+-- Parity flag
                       | | | | | |   "E": Even parity
                       | | | | | |   "O": Odd parity
                       | | | | | |   "I": Ignore (Illegal when
                       | | | | | |        data length is 8 bits)
                       | | | | | |   "N": No parity
                       +-+-+-+-+-+-- Length of stop bits
                         | | | | |   "1": 1 bit
                         | | | | |   "2": 1.5 bit
                         | | | | |   "3": 2 bits
                         +-+-+-+-+-- XON/XOFF control
                           | | | |   "X": Enable control
                           | | | |   "N": Disable control
                           +-+-+-+-- CTS-RTS hand-shake
                             | | |   "H": Handshaking
                             | | |   "N": No handshaking
                             +-+-+-- Insert Line Feed to buffer
                               | |   when Carriage Return is
                               | |   received.
                               | |   "A": Insert Line Feed to
                               | |        buffer
                               | |   "N": Do not insert
                               +-+-- Send Line Feed after Carriage
                                 |   Return sent.
                                 |   "A": Do not send Line Feed
                                 |   "N": Send Line Feed
                                 +-- Shift-in/Shift-out control.
                                     Illegal when data length is
                                     Other than 7 bits.
                                     "S": Enable control
                                     "N": Disable control
```

Examples:

```
CALL COMINI ("0:7E1XHNNN"",600,1200,30)
CALL COMINI ('BN1',9600)
```

The default values for those switches are as follows:

```
"0:8N1XHNNN",1200,1200,0
```

Note that no previous value is taken as the default. If omitted, the above values are always assumed.

2) Open and Close Communication Channels

```
OPEN "COM[n]:"[FOR <mode>] AS [#] <file number>
```

This statement opens the RS-232-C channel for further processing. That is, a I/O buffer is allocated and the mode that will be used with the buffer is set. The RTS signal is also activated.

The <mode> is one of the following:

OUTPUT: Specifies sequential output mode
INPUT : Specifies sequential input mode

If the <mode> clause is not specified, the channelcan be accessed for both input and output and no EOFcharacter handling is done.

The <file number> is an integer expression whose valueis between one and the maximum number of filesspecified in a MAXFILES=statement.

The <file number> is the number that is associated withthe file for as long as it is OPEN and is usedby other I/O statements referring to the file.

An OPEN statement must be executed before I/O may be done to the file using any of the following statements. The OPEN statement must be executed before any statement or function requiring a file number:

```
PRINT #, PRINT # USING
INPUT #, LINE INPUT #
INPUTS$
```

Example:
OPEN "COM0: " AS #1

NOTE

Random access to RS-232-C channel is not possible. Logically, only sequential accesses are permitted.

CLOSE [[#]<file number>[,<file number>]]
Closes the channel and releases the associated buffer. If no <file number> are specified, all open channels are closed.

If the channel was opened in output mode, the EOF character ís sent.

3) Sequential Input and Output

After the channel is opened in input mode or file mode (open without <mode> clause), characters from communication channel can be sequentially input
or one of the following statements.

```
INPUT #n
LINE INPUT #n
INPUT$(#n,m)
```

After the channel is opened in output mode or file mode (open without <mode> clause), characters can be sequentially output to the communication channel by one of the following statements.

```
PRINT#n
PRINT#n USING
```

4) Program Load/Save Statements

```
SAVE "COM[<n>:]"[,A]
```

Sends a BASIC program to the communication channel. A [CTRL] - Z is treated as the end-of-file character.

The program is sent in ASCII format, whether the optional parameter, "A", is specified or not. No file name is allowed.

```
LOAD "COM[<n>:]"
```

Loads a BASIC program from the channel. A LOAD statement closes all open files and deletes the current program from memory. If the "R" option is specified, however, all data files remain OPEN and the program that is loaded is also executed. A [CTRL] - Z is treated as the end-of-file character.

```
MERGE "COM[<n>:]"
```

Merges lines from a program in ASCII format received through the communication channel into the program currently in memory.

If some of the line numbers of the program in memory match line numbers of the incoming (channel) program, the lines from the program from the channel replaces the matching lines. A [CTRL] - Z is treated as the end-of-file character.

After the MERGE command, the merged program willreside in memory, and control will return to BASIC atthe command level.

RUN "COM[<n>:]",R
Loadsa program from the channel into memory and runsit.

RUN closes all open files and deletes the current contents of memory before loading the designatedprogram. When the "R" option is specified, however,all data files remain OPEN.

5) Event Trap Control Statements

CALL COMON("[<n>:]")
Enables event trapping caused by incoming characterfrom the communication channel.

CALL COMOFF("[<n>:]")
Disables event topping caused by incoming characterfrom the communication channel. Thecommunicationbuffer is flushed.

CALL COMSTOP("[<n>:]")
Suspends event trapping caused by incoming characterfrom the communication channel.

CALL COM([<n>:],GOSUB <line number>)
Sets the line numbers for BASIC to trap wen character are received at the Communication channel.

When trap occurs, since CALL COMSTOP is automatically executed, received traps can never take place. The RETURN from the trap routine will automatically does CALL COMON unless CALL COMOFF has been explicitlyperformed inside the trap routine.

Event trapping does not take place when BASIC is notexecuting a program. When an error trap (resulting from an ON ERROR statement) take place, it automatically disables all trapping (including ERROR, STRIG, STOP, SPRITE, INTERVAL and KEY).

6) Miscellaneous ControlStatements

An OPEN statement must be executed before any one offollowing statementsmay be executed. The default channel number is 0 for all the following statements.

CALL COMBREAK (["<n>:"],<expression>)
Sends break characters specified by <expression.> to the channel specified by <n>. The range of the <expression> should be between 3 and 32767.

CALLCOMDIR(["<n>:"],<expression>)
Turns off the DIR signal when the <expression> is zero, otherwise turns on the DIR signal.

CALL COMSTAT(["<n>:"],<name of variable>)
Reads the status of the communication channel. The status returned by the hardware is assigned to the variable. The bit assignments are the follows:

BIT NO.	DESCRIPTION
15	Buffer Overflow Error 0: No buffer overflow l: Buffer overflow
14	Time Out Error (TMENBT) 0: No time out error occurred 1: Time out error occurred
13	Framing Error 0: No framing error occurred 1: Framing error occurred
12	Over Run Error 0: No over run error occurred 1: Over run error occurred
11	Parity error 0: Character has no parity error 1: Character has parity error
10	Control break key was pressed (BRONBT) 0: Control break key not pressed 1: Control break key was pressed
9	Not used: Reserved
8	Not used: Reserved
7	Clear To Send 0: False 1: True
6	Timer/Counter Output-2 0: Timer/Counter Output-2 negated 1: Timer/Counter Output-2 asserted
5	Not used: Reserved
4	Not used: Reserved
3	Data Set Ready 0: False 1: True
2	Break Detect 0: Not detected 1: Detected
1	Ring Indicator 0: False 1: True
0	Carrier Detect 0: False 1: True

7) Functions

EOF (<file number>)
Returns -l (true) if the EOF character is received. Otherwise, returns 0. Use EOF to test for end-of transmission during INPUT to avoid 'Input past end' errors.

LOC(<file number>)
Returns the number of characters received in the communication buffer. The size of the communication buffer is 255 characters.

LOF (<file number>)
Returns the size of the free space remaining in the communication buffer.

8) Terminal Mode

CALL COMTERM [("<n>:")]
Enters a terminal emulator mode. The channel should be closed when this statement is invoked. The function keys have a special use in the terminal mode as described below.

F-6: Toggles the literal mode on/off. In the literal mode, control characters are displayed, offset by 40H. As an example, a character whose code is O1H is displayed as "A".
Initial mode: Literal mode off

F-7: Toggles the Half/Full duplex modes. In Half duplex mode, the characters typed in are echoed to the screen as well as sent to the communication channel.
Initial mode: Full duplex

F-8: Turn on/off printer echo. When the printer echo is on, all characters sent to the screen are also echoed to the printer.
Initial mode: Printer echo off

9) Help Function (Optional)

```
CALL COMHELP[(<n>:)]
```

Prints out a brief description of parameters set by a COMINI statement on the screen as follows.

```
Initialize statement options

CALL COMINI ("
<Device# {0,1,2...9}}>:
<Character length {5,6,7,8}>
<Parity {E,0,1,N}>
<Stop bits {1,2,3}>
<XON/XOFF {X,N}>
<CTS handshaking {H,N}>
<Auto LF on receive {A,N}>
<Auto LF on transmit {A,N}>
<SI/SO {S,N}>"
,<Receiver baud rate>
,<Transmitter baud rate>

Default:

CALL COMINI("O:8N1XHNNN",1200,1200,0)
```

10) Behavior of Control Signals

	RESET	COMINI	OPEN	CLOSE
RTS:	Inactive	No effect	Active	Inactive
DTR:	Active	Active	No effect	No effect

The RTS signal is affected in the following cases:

1. OPEN statement executed: activated
2. CLOSE statement executed: inactivated
3. The remaining contents of the communication is lessthan 16 bytes and the CTS-RTS handshake is enabled:inactivated.
4. When inactive and the remaining contents of thecommunication buffer has more than one byte andCTS-RTS handshaking is enabled: activated.

DIR is affected by the CALL COMDTR and CALL COMINIstatements.

11) Handling of EOF

If a channel is opened in output mode, an EOF istransmitted when a CLOSE statement is executed.

2. EXTENDED BIOS CALLS HANDLING RS-232-C

The RS-232-C driver can be used by application programs using the "EXTENDED BIOS CALL" routine. Such programs can access the functions in the RS-232-C driver through the entry jump table with inter-slot call function provided in the BIOS. Programmers can determine the location of this table by using an EXTENDED BIOS CALL 0 and 1.

The RS-232-C driver is addressed by device number 8. The functions for the RS-232-C driver supported by extended BIOS calls are described below.

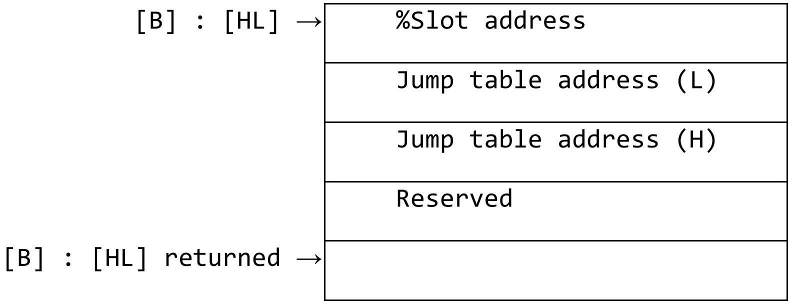
Build a Slot Address Table Entry to the Jump Table

Number: 0

Function: Builds entry address to the jump in the device driver pointed by [HL]
[B] = Slot address of table entry for the device driver
[HL] = Points to table entry for this device driver

Exit: [B] = Slot address of next table entry
[HL] = Points to next table entry

Description: The RS-232-C device driver call routine can issue this function call to determine the slot number and the location of the jump table to access the device driver for the device driver for the desired channel.



% The format of the slot address is same to the MSX standard notation, which is:

```
Bit # 76543210
      F000SSPP
      |   ||||
      |   ||||
      |   ||||
      |   ||||
      |   ||LL-- Primary slot (0-3)
      |   LL---- Secondary slot (0-3)
      L--------- 1 if secondary slot specified
```

Return Number Of Channels

Number: 1
Function: Returns the number of channels available to the device driver.
Entry: [A] = Contains number of RS-232-C channels so far.
Exit: [A] = Number of RS-232-C channels updated.

Description: This function is provided for each RS-232-C driver so as to find the channel number for the driver. Each driver can call this function to get the number of RS-232-C channels installed so far.

The device information byte indicates whether the following options are installed or not:

```
Bits 76543210
     ||||||||
     |||||||L-- Reserved
     ||||||L--- TxReady interrupt
     |||||L---- Sync/Break character detected
     ||||L----- Timer interrupt
     |||L------ Carrier detect
     ||L------- Ring indicator
     |L-------- Reserved
     L--------- Reserved
```

The RS-232-C driver has entries as follows. Application programs can use the RS-232-C driver by an 'inter-slot call' to those entries.

```
EXBTBL:  DEFB  DVINF8,0,0   ;  Device information
         JP    INIT         ;  Initialize RS-232-C port
         JP    OPEN         ;  Open RS-232-C port
         JP    STAT         ;  Read status
         JP    GETCHR       ;  Receive data
         JP    SNDCHR       ;  Send data
         JP    CLOSE        ;  Close RS-232-C port
         JP    EOF          ;  EOF code received
         JP    LOC          ;  Reports the number of
                               characters in the receiver
                               buffer
         JP    LOF          ;  Reports the number of free
                               spaces left in the
                               receiver buffer
         JP    BACKUP       ;  Back up a character
         JP    SNDBRK       ;  Send break character
         JP    DTR          ;  Turn on/off DTR 1 line
         NOENT              ;  Reserved for future
                               expansion
         NOENT
         NOENT
```

NOTE

The RS-232-C receiver is driven by the interrupt generated by the receiver ready. However, the inter-slot call handler disables the interrupt automatically. Thus, when control returns to the application program, it must enable an interrupt as soon as possible, or the RS-232-C receiver routine will lose some of the characters.

Description of each Extended BIOS Call

1) Initialize RS-232-C Port (INIT)

Entry: [HL] = Address of the parameter table
[B] = Slot address of the parameter table
Return: The carry flag is set if illegal parameters are set.
Modify: [AF]

Description:
Initializes the RS-232-C port with the specified parameters. This entry must be called before any other function calls are made. The parameters are similar to the COMINI expanded statement of BASIC. However, note that all the ASCII parameters must be specified with uppercase characters only.
(See section 1.1 CALL COMINI for details to BAUD RATEend TIME OUT)

[8]:[BH] →	Character Length '5' '8'	ASCII Characters
	Parity 'E','O','I','N'	
	Stop Bits '1','2','3'	
	XON/XOFF Control 'X','N'	
	CTR=RTS Handshaking 'H','N'	
	Auto LF for Receiver 'A','N'	
	Auto LF for Sender 'A','N'	
	SI/SO Control 'S','N'	
	Receiver Baud Rate (Low) 50-19200 (High)	Binary
	Transmitter Baud Rate (Low) 50-19200 (High)	
	Enc Out Counter 0-25	

3. RS-232-C INTERFACE

This section describes the hardware of RS-232 interface for MSX home personal computers

1) LSI Components

i-8251 Communication interface
i-8253 Programmable interval timer

8 Kbyte ROM storage for the support software.

2) Port Address

80H	R/W	8251 data port
81H	R/W	8251 command/status port
82H	R	Status sense port for CTS, Timer/Counter 2, RI and CD
82H	W	Interrupt mask register
84H	R/W	8253 counter 0
85H	R/W	8253 counter 1
86H	R/W	8253 counter 2
87H	W	8253 mode register

3) The Usage of Port at Address 82H

82H read - Get system status

DATA BIT	DESCRIPTION
D7	CTS (Clear To Send) 0 – CTS is asserted 1 – CTS is negated
D6	Timer/counter output-2 from i8253
D1	RI (Ring indicator) 0 – RI is asserted 1 – RI is negated
D0	CDC (Carrier detect) 0 – CDC is asserted 1 – CDC is negated

NOTE

The CTS signal is sensed through the port instead of through the 8251 because of a problem in CTS logic in some version of 8251. Software handling is thus made possible.

82H write - Interrupt mask register

DATA BIT	DESCRIPTION
D3	Timer interrupt from i8253 channel-2 1 - mask interrupt (initial value) 0 - enable interrupt
D2	Sync character detect/Break detect 1 - mask interrupt (initial value) 0 - enable interrupt
D1	Transmit data ready (TxReady) 1 - mask interrupt (initial value) 0 - enable interrupt
D0	Receive data ready (RxReady) 1 - mask interrupt (initial value) 0 - enable interrupt

4) The Usage of 8253 Timer Counter to General Baud Rate Clockfor 8251

A. Frequency of X'tal

The frequency of the crystal:
1.8432MHz

BAUD RATE (BAUD)	SCALE FACTOR AND ERROR (x16)
50	2304
75	1536
110	1047 110.0287 +0.3%
150	768
300	384
600	192
1200	96
1800	64
2000	58 1986.2 -0.7%
2400	48
3600	32
4800	24
7200	16
9600	12
19200	6

B. The Usage of Counter Channel

CH0 - Rx baud rate clock
CH1 - Tx baud rate clock
Ch2 - Used by application
(Generated interrupt optional)

5) The Connect of DB25 Connector

PIN	SIGNAL
1	Frame ground
2	Transmit data
3	Receive data
4	Request To Send
5	Clear To Send
6	Data Set Ready
7	Signal ground
8	Carrier detect
9	
10	
11	
12	
13	

PIN	SIGNAL
14	
15	
16	
17	
18	
19	
20	Data Terminal Ready
21	
22	Ring Indicator
23	
24	
25	

SVI
SPECTRAVIDEO

M-522