

RS232C Extended BIOS Call specification

September 18th, 1984

December 24th, 1985

1.0 GENERAL DESCRIPTION

The RS232C driver can be used by application programs using the "EXTENDED BIOS CALL" mechanism. Such user can access each function in the RS232C driver through the entry jump table with inter-slot call function provided in the BIOS. The user is able to know the location of this table by EXTENDED BIOS CALL 0 and 1. Refer to the document "The specification of the EXTENDED BIOS CALL" for details.

1.1 EXTENDED BIOS CALL ENTRY TABLE

The RS232C driver has entries as follows. A application program is able to use RS232C driver by 'inter-slot call' to those entries.

For I/O port only type of RS-232C interface.

EXBTBL:	DEFB	DVINFB	; device information
	DEFB	0	; reserved for future expansion
	DEFB	0	
	JP	INIT	; initialize RS232C port
	JP	OPEN	; open RS232C port
	JP	STAT	; Read STATUS
	JP	GETCHR	; receive data
	JP	SNDCHR	; send data
	JP	CLOSE	; close RS232C port
	JP	EOF	; tell EOF code received
	JP	LOC	; reports number of characters in the receiver buffer
	JP	LOF	; reports number of free space left in the receiver buffer
	JP	BACKUP	; back up a character
	JP	SNDBRK	; send break character
	JP	DTR	; turn on/off DTR line
	NOENT		; reserved for future expansion
	NOENT		
	NOENT		

Multi channel type RS-232C cartridge

EXBTBL:

DEFB	DVINFB	
DEFB	1	; version number
DEFB	0	; reserved for future expansion
JP	INIT	; initialize RS232C port
JP	OPEN	; open RS232C port
JP	STAT	; Read STATUS
JP	GETCHR	; receive data
JP	SNDCHR	; send data
JP	CLOSE	; close RS232C port
JP	EOF	; tell EOF code received
JP	LOC	; reports number of characters in the ; receiver buffer
JP	LOF	; reports number of free space left ; in the receiver buffer
JP	BACKUP	; back up a character
JP	SNDBRK	; send break character
JP	DTR	; turn on/off DTR line
JP	SETCHN	; set channel number
NOENT		; reserved for future expansion
NOENT		

NOTE

The RS232C receiver is driven by the interrupt generated by receiver ready. However, the inter-slot call handler disables interrupt automatically. So, when control returns to the application program, it have to enable interrupt as soon as possible. Otherwise, RS232C receiver routine loses incoming characters.

2.0 DESCRIPTION OF EACH EXTENDED BIOS CALL

2.1 Initialize RS232C Port (INIT)

Entry: [HL]= address of the parameter table
[B] = slot address of the parameter table
Return: carry flag is set if illegal parameters are contained
Modify: [AF]

Description:

To initialize the RS232C port with specified parameter. This entry must be called before any other function calls are made. The parameters are similar to _COMINI expanded statement of BASIC. However, note that all the ascii parameters must be specified with upper case characters only.

BAUD RATE

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

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

When negative value is specified, its absolute value is written to i8253 timer/counter directly.

TIME OUT

The RS232C driver waits till the CTS (Clear To Send) signal is turned on and/or XON is received when the character is to be sent. The driver will generate time out error when it waits for them, if specified time passed. Its time is specified in this value at second. If 0 specified then the driver doesn't generate time out error and waits forever.

[B]:[HL]-->	Character length '5'~'8'	\	
	Parity 'E','O','I','N'		
	Stop bits '1','2','3'		
	XON/XOFF control 'X','N'		>-- ASCII character
	CTR-RTS hand shake 'H','N'		
	Auto LF for receive 'A','N'		
	Auto LF for send 'A','N'		
	SI/SO control 'S','N'	/	
	Receiver baud rate (low)	\	
	50~19200 (high)		
	Transmitter baud rate (low)		>--Binary
	50~19200 (high)		
	Time out counter 0~255	/	

2.2 Open RS232C Port (OPEN)

Entry: [HL]= address of FCB (must be located higher address than 8000H)

[C] = buffer length (32~254)

[E] = open mode, one of following:

open mode	meanings
1	<input> mode
2	<output> mode
4	<raw> and <input/output> mode

Return: carry flag is set if any error occurred.

Modify: [AF]

Description:

Opens RS232C port with specified FCB (File Control Block). Open must be made before any I/O operations take place. Each received character occupies two bytes in the buffer. One is received character code itself and another is error status of the received character. And extra 9 bytes are necessary as a working storage for the file control. Note that the buffer length passed by [C] specifies number of characters, so the actual length of buffer is $[C] \times 2 + 9$ bytes. And this buffer area can be accessed without slot handling whenever the RS232C driver is called (including the timing when the interrupt from the receiver generated).

[HL]-->	9 bytes for file control
	[C] X 2 bytes receiver buffer

2.3 Read Status (STAT)

Entry: None

Return: [HL]= status data.

Modify: None

Description:

Returns various status information and error code of the character just read from the buffer (not the character just received).

BIT NO.	Description
15	Buffer over flow error 0 - no buffer over flow 1 - buffer over flow
14	Time out error 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 - the character hasn't parity error 1 - the character has parity error
10	Control break key was pressed 0 - control break key wasn't 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 is negated 1 - timer/counter output-2 is asserted
5	Not used, reserved
4	Not used, reserved
3	Data Set Ready 0 - false 1 - true
2	break detect 0 - not yet detect 1 - detect
1	Ring Indicator 0 - false 1 - true
0	Carrier Detect 0 - false 1 - true

2.4 Get A Character From The Receive Buffer (GETCHR)

Entry: None

Return: [A] = character received
sign flag is set if any error occurred.
carry flag is set if the character is an EOF code when
port is opened for input mode.

Modify: [F]

Description:

Get a character from the receiver buffer. Returns backed
up character if any.

2.5 Send A Character To The RS232C Port (SNDCHR)

Entry: [A] = character to send

Return: carry flag is set if control break key was pressed
zero flag is set if time out error occurred during
waiting for XON or/and CTS signal.

Modify: [F]

Description:

Send specified character to RS232C port. The character flow
control by XON/OFF characters and/or CTS (Clear To Send) line
signal is handled if initialized so. Time out error will
be generated when specified time passed during waiting for
permission for transmission, and the character will not be
sent.

2.6 Close The RS232C Port (CLOSE)

Entry: None

Return: carry is set if any error occurred.

Modify: [AF]

Description:

Closes the RS232C port. The buffer is released, and a EOF
code is sent if the port was opened for <output> mode. RTS
signal is turned to a inactive state.

2.7 Check For The EOF Code (EOF)

Entry: None

Return: [HL]= -1, carry flag is set, if the next character is EOF code.

= 0, carry flag is reset, if the next character is not EOF.

Modify: [AF]

Description:

Tests whether the next character is EOF code or not. Returns 0 if no character.

2.8 Returns A Number Of Character In The Receive Buffer (LOC)

Entry: None

Return: [HL] = number of character in the receiver buffer

Modify: [AF]

Description:

Returns number of valid character in the receive buffer. This value includes number of backed up character. Characters after EOF code are ignored if opened in <input> mode, but occupies buffer space although.

2.9 Returns Number Of Free Space In The Receive Buffer (LOF)

Entry: None

Return: [HL]= number of free space

Modify: [AF]

Description:

Returns a number of free space in character in the receiver buffer.

2.10 Back Up A Character (BACKUP)

Entry: [C] = character to back up

Return: None

Modify: [F]

Description:

Backs up a character in the special buffer. Last backed up character will be lost if any.

2.11 Send Break Character (SNDBRK)

Entry: [DE] = number of break character to send
Return: carry flag is sets if control break key was pressed
Modify: [AF], [DE]

Description:

Transmit specified number of break characters. Aborts if Control-Break key is pressed during the transmission and returns with carry flag set.

2.12 Turn On/off DTR Line (DTR)

Entry: [A] = 0 if turn off
[A] = not 0 if turn on
Return: None
Modify: [F]

Description:

DTR (Data terminal Ready) line is turned on when power-on/reset initializing or INIT routine is called.

3.0 MULTIPLE CHANNEL TYPE RS-232C CARTRIDGE

3.1 Set Channel Number (SETCHN)

Entry: [A] = channel number.
Return: Carry flag is set if the channel is not in the cartridge. Otherwise the channel is in the cartridge.
Modify: [AF], [BC]

Description:

The channel number is set with 0 when power-on/reset initializing.

MSX MEMORY ORGANIZATION INFORMATION

ASCII Corp. Dec. 21, 1985
Edited 31st Jan 1986

(All information contained herein is proprietary to ASCII MSFE)

1.0 MAIN-RAM

FFFF	Reserved for secondary slot select register (4 bytes)	
FFFC		
FFF8	Reserved for future extension (4 bytes)	
FFF7	Main ROM slot address	
	for NEW VDP VDP register save 8 - 23 (16 bytes)	
FFE7		
FFD9	XXXINT program (14 byte)	
FFD4	ENAINIT HOOK (5 byte)	
FFCF	DISINT HOOK (5 byte)	
	Extended BIOS Call Entry (5 byte)	
FFCA		
	RAM HOOK AREA (560 byte) 112 Hooks	
FD9A		HOKJMP
	System Work Area (2586 byte)	
F380		INIDAT
	User Area	
8000		
	RAM DISK area	
0000		

2.0 RESERVED AREA FOR RS232C BUFFER

FAF5	RSIQ +OFFSET		
FAF5	DPPAGE	0	Display page
FAF6	ACPAGE	1	Active page
FAF7	AVCSAV	2	AV control port save
FAF8	EXBRSA	3	Extended Basic ROM slot address
FAF9	CHRCNT	4	Character counter for Roma-kana
FAFA	ROMA	5	Character save for
FAFB		6	Roma-kana (2byte)
FAFC	MODE	7	Roma-kana conversion mode switch/VRAM size
FAFD		8	Not used, reserved
FAFE	XSAVE	9	MOUSE/CAT/LIGHT PEN
FAFF		10	X - SAVE (2 byte)
FB00	YSAVE	11	MOUSE/CAT/LIGHT PEN
FB01		12	Y - SAVE (2 byte)
FB02	LOGOPR	13	Logical operation code (Lower 4 bits)

FB03	TOCNT	14	byte data	
FB04	RSFCB	15	Address of RS232C (low)	Data area for RS-232C driver
FB05		16	(high)	
FB06	RSIQLN	17	byte data	
MEXBIH		18	'RST 30H' (0F7H)	
		19	byte data	
		20	(low)	
		21	(high)	
		22	'RET' (0C9H)	
OLDSTT		23	'RST 30H' (0F7H)	
		24	byte data	
		25	(low)	
		26	(high)	
		27	'RET' (0C9H)	
OLDINT		28	'RST 30H' (0F7H)	
		29	byte data	
		30	(low)	
		31	(high)	
		32	'RET' (0C9H)	
DEVNUM		33	byte offset	
DATCNT		34	byte data	> must be contiguous
		35	byte pointer	
		36	byte pointer	
ERRORS		37	byte data	
FLAGS		38	bit boolean	
ESTBLS		39	bit boolean	
COMMSK		40	byte data	
LSTCOM		41	byte data	
LSTMOD		42	byte data	
		43		> Device driver can not use
		63	Used by system	

3.0 MAIN-ROM

7FFF		Jump instruction	
		to the BDOS entry	
7FFD			
+-----+			
01B6		SLOT.MAC and other	
		BIOS codes	
+-----+			
0171		90 - 24 bytes hole	
+-----+			
015C		BIOENT.MAC	
		-Added bios entries	
+-----+			
0000		BIOENT.MAC	
		-Bios entry jumps	
+-----+			

4.0 BATTERY BACK-UP RAM IN CLOCK CHIP (RP5C01)

4.1 MODE 2

Address	Contents / Related statement	
0	ID nibble (1010 indicates clock RAM is initialized)	
1	Adjust X (-8 to 7)	SET ADJUST
2	Adjust Y (-8 to 7)	SET ADJUST
3	Screen mode 0 or 1, interlace mode <==2bits left	SET SCREEN
4	Width Lo	SET SCREEN
5	Width Hi <==1bit left (0-127 is enough)	SET SCREEN
6	Foreground color (0-15)	SET SCREEN
7	Background color (0-15)	SET SCREEN
8	Border color (0-15)	SET SCREEN
9	Key click, Key ON/OFF, Printer mode, Cassette speed	SET SCREEN
10	Beep sound select, volume (0-3,0-3)	SET BEEP
11	Title color (0-3) <==2 bits left	SET TITLE
12	Country code (0-15)	

Country code table	
0	Japan
1	USA
2	INTERNATIONAL
3	UK
4	FRANCE
5	GERMANY
6	ITALY
7	SPAIN
8	ARABIA
9	KOREA
10	RUSSIA
11	
12	
13	
14	
15	

4.2 MODE 3

Address	Usage
0	ID nibble 0:Title, 1:Password, 2:Prompt, 3-15:not defined
1	Lo1---+
2	Hi1
3	Lo2
4	Hi2
5	Lo3
6	Hi3
7	Lo4
8	Hi4
9	Lo5
10	Hi5
11	Lo6
12	Hi6 ---+

6 byte string data for above usage.

The RAM usage for PASSWORD	
Address	Usage
0	usage id = 1
1,2,3	usage id = 1,2,3
4,5,6,7	encoded password
8	flag key cartridge exists = 0 - No key cartridge exists = 1 - cartridge exists, no password is set = 2 - cartridge exists, password is set
9,10,11,12	key cartridge value