API functions		On Entry	On Exit
$00	System reset	A = Reset type	none
$01	Input character (waits)	none	A = Character input
$02	Output character (waits)	A = Character to output	A = Character output
$03	Input status	none	NZ if char avail
$04	Input line	A = Size, DE = Address	A = Length, DE = Address
$05	Input line default	none	A = Length, DE = Address
$06	Output line	DE = Address	none
$07	Output new line	none	none
$08	Get version details	none	Version info in A B C D E H L
$09	Claim jump table entry	A = Number, DE = Address	none
$0A	Delay in milliseconds	DE = Milliseconds	none
$0B	Output message	A = Number	none
$0C	Read jump table entry	A = Number	DE = Address
$0D	Select console in/out device	A = Device number	none
$0E	Select console input device	A = Device number	none
$0F	Select console output device	A = Device number	none
$10	Input char from console device	E = Device number	A = Character, NZ flagged if OK
$11	Output char to console device	A = Character, E = Device	NZ flagged if OK, else A = Char
$12	Poll idle events	none	none
$13	Configure idle events	A = Mode (0=Off, 1=On)	none
$14	Timer 1 control (A x 1ms)	A = Period, DE = Address	none
$15	Timer 2 control (A x 10ms)	A = Period, DE = Address	none
$16	Timer 3 control (A x 100ms)	A = Period, DE = Address	none
$17	Output port initialise	A = Port address	A = Output (zero)
$18	Write to output port	A = Output byte	A = Output byte
$19	Read from output port	none	A = Output byte
$1A	Test output port bit	A = Bit (0 to 7)	A = Output masked
$1B	Set output port bit	A = Bit (0 to 7)	A = Output byte
$1C	Clear output port bit	A = Bit (0 to 7)	A = Output byte
$1D	Invert output port bit	A = Bit (0 to 7)	A = Output byte
$1E	Input port initialise	A = Port address	A = Input byte
$1F	Read from input port	none	A = Input byte
$20	Test input port bit	A = Bit (0 to 7)	A = Input masked
$21	Set baud rate	A = Device, E = Rate code	NZ flagged if OK
$22	Execute command line	DE = Pointer to line	Z flagged if OK
$23	Get pointer to command line	none	DE = Pointer to line
$24	Skip delimiter in command line	DE = Pointer to line	A = Char, DE = Pointer
$25	Skip non-delimiter in command	DE = Pointer to line	A = Char, DE = Pointer
$26	Get Hex parameter from command	DE = Pointer	A = Status, DE = Ptr, HL = Value
$27	Get current console I/O devices	none	D = Output, E = Input
$28	Get top of free memory	none	DE = Top of free memory
$29	Set top of free memory	DE = Top of free memory	none
$2A	Read from banked RAM	DE = Address	A = Data byte read
$2B	Write to banked RAM	A = Data, DE = Address	A = Data byte written

API Monitor Command:

```
    API   <function number>   [<A>]   [<DE>]
    result displayed:   <A>   <DE>
```

API Assembler Instructions:

```
    LD    C, <function number>
    RST   $30
```

Self-test (at reset)

Test output is via LEDs on the system's status display port (eg. RC2014 or LiNC80 Digital I/O module)

Pass	Single sweep of lights followed by all lights off
Failed RAM	Continuous sweeping of lights
Failed serial	LED bit 0 stays on if serial module not found

Monitor command line interpreter

? / HELP		Display help
A	[<address>]	Assemble instructions
API	<function> [<A>] [<DE>]	Call API function
B	[<address>]	Breakpoint set or clear
BAUD	<device> <rate code>	Set baud rate
CONSOLE	<device identifier>	Select console device
D	[<address>]	Disassemble instructions
DEVICES		List devices detected
DIR		List files in the ROM
E	[<address>]	Edit memory
F	[<flag or condition name>]	Flags display or modify
FILL	<start> <end> <byte>	Fill memory
G	[<address>]	Go to program
I	<port>	Input from port
M	[<address>]	Memory display
O	<port> <data>	Output to port
R	[<register name>]	Registers display or edit
RESET		Reset monitor
S	[<address>]	Step one instruction

Monitor ROM may also include additional commands, such as "BASIC".

Jump Table

$00	Non-maskable interrupt handler	
$01	Restart $08, console character output	
$02	Restart $10, console character input	
$03	Restart $18, console input status	
$04	Restart $20, handler (not currently used)	
$05	Restart $28, breakpoint handler	
$06	Restart $30, applications programming interface (API) handler	
$07	Restart $38, mode 1 interrupt handler	
$08	Console input routine	
$09	Console output routine	
$0A	Reserved for get console input status	
$0B	Reserved for get console output status	
$0C	Idle event handler	
$0D	Timer 1 event handler	
$0E	Timer 2 event handler	
$0F	Timer 3 event handler	
$10	Device 1 input character	default = serial port channel A
$11	Device 1 output character	default = serial port channel A
$12	Device 2 input character	default = serial port channel B
$13	Device 2 output character	default = serial port channel B
$14	Device 3 input character	
$15	Device 3 output character	
$16	Device 4 input character	
$17	Device 4 output character	
$18	Device 5 input character	
$19	Device 5 output character	
$1A	Device 6 input character	
$1B	Device 6 output character	

Memory map

ROM	(minimum)	9 kbytes, 0x0000 to approx 0x2500 used
RAM	(minimum)	32 kbytes assumed from 0x8000 to 0xFFFF, 0xFC00 to 0xFFFF used

Flag and condition names

Flags:	*set*	*clear*
Zero	Z	NZ
Carry	C	NC
Negative	S	NS
Half carry	H	NH
Parity even	Pa	NP
Subtract	N	NN

Conditions:		
Zero	Z	zero set
Not zero	NZ	zero clear
Carry	C	carry set
Not carry	NC	carry clear
Negative	M	sign set
Positive	P	sign clear
Even	PE	parity set
Odd	PO	parity clear

Baud rate codes

Baud rate	*Rate codes*	
230,400	$1	$23
115,200	$2	$11
57,600	$3	$57
38,400	$4	$38
19,200	$5	$19
14,400	$6	$14
9,600	$7	$96
4,800	$8	$48
2,400	$9	$24
1,200	$A	$12
600	$B	$60
300	$C	$30

Either code can be used

Console devices

Console device	*Identifiers*	
#1 (eg. SIO A)	$1	$A
#2 (eg. SIO B)	$2	$B
#3 (device 3)	$3	$C
#4 (device 4)	$4	$D
#5 (device 5)	$5	$E
#6 (device 6)	$6	$F

Either identifier can be used

Small Computer Monitor ROM v1.3   Reference Sheet, edition 1.3.0   Stephen C Cousins   (www.scc.me.uk)