SGS ATES

DESIGN NOTE

DN314

N A N O C O M P U T E R   N C – Z
S O F T W A R E   R O U T I N E S

ABSOLUTE ADDRESS LOCATIONS CORRESPONDING
TO NC – Z, REL. 2.1

The NC-Z software in 2K Bytes used on the NBZ80 Nanocomputer has a number of subroutines that can be called by user programs.

- Keyboard
- Display
- Serial interface

Also included in this Design Note is an example of the use of the Display for limited alphanumeric symbols and a list of the corresponding hexadecimal display codes.

NOVEMBER 1979

NC-Z software contains a number of subroutines. These can be called using the Z80

CALL XXXX

Instruction where XXXX is the absolute address.
In order to give revision flexibility to the software however all of the absolute addresses are referred to by labels. The absolute address of these labels in rel. 2.1 of the NC-Z software is given below. A listing of the software is not available.

	LABEL	DESCRIPTION	ADDRESS	
DATA	IN MODE	Flag to indicate input tape format	0FAB	
	BLCKCNT	Counter for input characters in each block	0FAC	
	BLKLENGHT	Data constant defining no. of characters in block	0FAD	
	BAUDRT	16 bit constant defining the baud rate.	0FAE	
	BAUDRT+1		0FAF	
	LEDH	7 segment display code storage area for	0FB8, 0FB9	
	ADD7	KB display. Selector LED's, 4 digits Data	0FBA - 0FBD	
	DATA7	and 4 digit Address fields.	0FBE - 0FC1	
	DATAL		0FE2	
	DATAH	16 bit data and address storage space for	0FE3	
	ADDL	binary input to be converted for enventual	0FE4	
	ADDH	7-segment display	0FE5	
SUBROUTINE ENTRY				
	KBSCAN	Call for keyboard input scan	F8DB or	FBD0
	DISPL	Call for 7-segment display drive	F909	FBD3
	NULL	Call for output of NULL character to TTY/CASS	F96E	FBD6
	TTYO	Call for output of character to TTY/CASS	F970	FBD9
	TTYI	Call for input character from TTY/CASS with check for record formatted tape sync.	F9AA	FBDF
	TTYI~~0~~1	Call for input of character from TTY/CASS with free formatted tape	FA09	FBDC
	BAUD	Call for delay for baud rate	F9F2	FBE5
	BAUDHF	Call for 1/2 baud delay	F9FE	FBE8
	ASCII	Call to send ASCII hex equivalent of one binary byte in (HL) to TTY	FA10	FBEB
	ASCIB	Call to send ASCII hex equivalent of one binary byte in B to TTY	FA12	FBEE
	BYTE	Call to read 2 characters of ASCII and convert them to a binary byte in E	FA2D	FBF1

LABEL	DESCRIPTION	ADDRESS	
CONVDI	Call to convert 16 bit Data & Address words to 7-segment code for display (DISPL)	FA7C or	FBF6
TTYI2	Call for input of a byte trasmitted by two 5 bits characters	F9C3	FBF2

MEMORY MAP.

The Nanocomputer can address 64K memory bytes and all address decoding is absolute.

The NBZ80 board carries 4K RAM located at 0-4K (decimal) and 2K EPROM located at 62-64K. F800 — 1000 H [illegible]

The entry to the NC-Z 2K EPROM is made by a hard wa~~y~~e jump when the RESET key is pressed.

The 4K RAM is mostly available for user programs, except for the locations

0038H 0039H 003AH	the RST38H (op. code FF) instruction jumps here: this causes a further jump to software breakpoint routine which saves the CPU registers. This instrution is used when a break-point is set and can also be used is a user program to cause a return to NC-Z, saving the CPU status.
0066H 0067H 0068H	The NMI input on the CPU, connected the key-board BREAK key cause a jump to the NC-Z operating system and saves the CPU registers.

The other locations used by NC-Z are 85 (decimal) locations (0FAB to 0FFF) at the top of the RAM for data space and CPU register save locations.

The stack pointer is initialized to 0F00 by NC-Z.

SUBROUTINES

KBSCAN: CALL F8DBH

This routine scans the 28 key keyboard of the Nanocomputer and gives two outputs

- is a key pressed?	CARRY FLAG = 0, YES = 1, NO
- which key was pressed?	code as shown below is returned in the Registers A and C.

A & C content (Hex)	Key pressed
ØØ - ØF	Ø - F
10	
11	
12	ST
13	LA
14	2ND
15	SS
16	INC
17	LD
18	ARS
19	GO
1A	BRK
1B	DP

Note that the BREAK and RESET keys are not software scanned but directly connected to the Z80 CPU:

BREAK = NMI , jump to 0066H to execute routine to save CPU status.

RESET = RESET, reset CPU & MC-Z initialize

KBSCAN uses AF and BC registers and 2 levels of stack.

DISPL: CALL F909H

This subroutine takes the 7-segment driver codes stored in locations LEDH, ADD7 and DATA7 and SCANS the display of selector LEDS, Address and Data 7-segments once; for a continuous display the user must form a loop and repeatedly call DISPL.

The call should be made at least every 10ms for good display brightness.

The 7 Segment drive codes are stored in 10 bytes:

LABEL	First Address	Number of Bytes
LEDH	OFB8	2
ADD7	OFBA	4
DATA7	OFBE	4

The assignment of the bits to the display is as follows

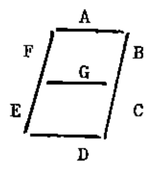
LABEL	LOCATION	b7	b6	b5	b4	b3	b2	b1	b0
LEDH	OFB8	BRK	I/O	MEM	PC	SP	ERR	ARS	—
	OFB9	IR	AF	BC	DE	HL	IX	IY	—

if a bit = 1 the LED is ON.

LABEL	LOCATION	DISPLAY DIGIT
ADD7	ØFBA	Left digit
	ØFBB	" "
	ØFBC	" "
	ØFBD	Right digit
DATA7	ØFBE	Left digit
	ØFBF	" "
	ØFCØ	" "
	ØFC1	Right digit

and the segments of the display are assigned to the bits as follows.

b7	b6	b5	b4	b3	b2	b1	b0
A	B	C	D	E	F	G	-



The segments are ON if the bit = 1

For example to display the number "2" in the left hand digit of the Address display load.

b7 b0

1101101X (X = don't care) = DA or DB

into the location OFBA and CALL DISPL.

The display can also be masked to switch off individual digits, for this feature see CONVDI subroutine.

DISPL uses the AF and HL registers and 2 levels of stack, the BC register is saved by the subroutine on the stack and restored on returning to the calling program.

There is a list of Hex codes corresponding to all possible displays at the back of this design note (Appendix 1) and an example (Appendix 2).

TTYO: CALL F970

This subroutine outputs the code in C register to the TTY serial terminal or the Audio Cassette depending on the position of the TTY/CASS switch on the keyboard. The serial character is sent to I/O port 4 bit 4.

To output a meaningful character the code in the C register must be in ASCII and the b7 (parity bit) set or reset as required.

The serial output is one start bit, 8 bits of Register C, and 2 stop bits.

TTYO uses the registers AF and saves BC, it uses a 4 levels of stack.

BAUD: CALL F9F2

The speed of trasmission is determined by a delay routine BAUD and the content of two RAM locations BAUDRT+1; it is initialized as 600 baud (for cassette load/dump) but can be changed as follows

	BAUDRT	BAUDRT+1
Baud rate	OFAE	OFAF
600	9A	ØØ
300	35	Ø1
110	35	Ø3

BAUD uses F register and saves BC; it uses 2 levels of stack

BAUDHF: CALL F9FE

This routine returns half the delay of BAUD.

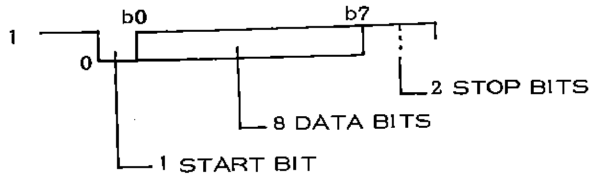
Teletype or Cassette data /program Input

3 routines are available in NC-Z for inputting a character from a serial terminal or from a tape cassette.

TTY CALL FAØ9

The subroutine inputs a serial character from I/O port 4 bit 7 the CPU Registers A and C, and resets (Ø) the parity bit 7.

The routine reads a character at the baud rate fixed by the BAUD subroutine, the serial format is



In order to read a sequence of characters the user must write a looping program since each call to TTYI1 inputs only a single character.

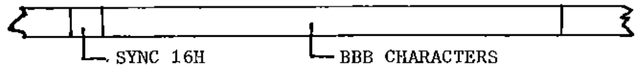
TTYI1 uses registers A, B, C, and 3 levels of stack.

TTYI CALL F9AA

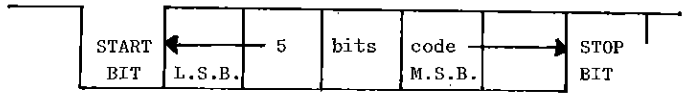
This routine which also uses TTYI1 is intended to read data or program files from tape cassettes or TTY paper tape readers.

The format of the data on the tape can be of two basic kinds:

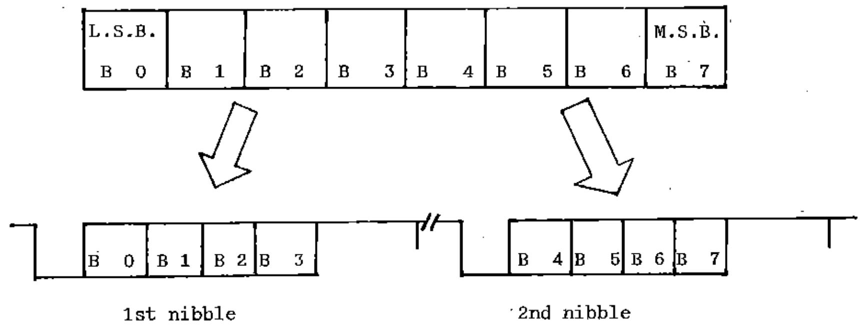
A, Record format



in wich each record is composed by 5 bits asyncronous character whose serial format is

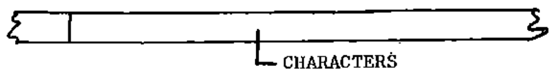


each record begins with sync character (16) followed by BBB 8 bits characters, each of which is described by two 5 bits characters as follows:



The 5th bit is always setted to one.
A record format is dumped by the SGS-ATES MO-Z Monitor software and has BBB = 82 decimal characters per record.

B.   Free format



in which data is 8 bits wide and continuous with possible start and end file marks imposed by the file structure. The SGS-ates NC-Z uses free format.

Three locations in RAM are used to allow reading both types of tape, and should be initialized to the following value:

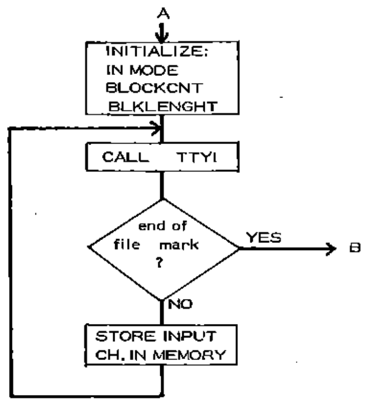
IMMODE	OFAB	=	ØØ	Free format
		=	FF	Record format
BLOCKCNT	OFAC	=	ØØ	Record format with 16H sync character
		≠	ØØ	Free format (that is, set to any non-zero value)
BLKLENGHT	OFAD	=	81	Decimal for reading SGS-ATES MO-Z monitor dumped tapes in Record format
		=	BBB-1	for reading other tapes in Record format with BBB characters per record.

Summary table of initialization for TTYI sybroutine.

FORMAT	INMODE	BLOCKCNT	BLKLENGHT
FREE	ØØ	not zero	xx
RECORD (82 ch)	FF	ØØ	81
(BBB ch)	FF	ØØ	BBB-1

xx = don't care.

For a user program to read a tape a flow diagram (A-B) as shown below can be used.



The SGS-ATES load programs respond to "CR-LF-any character other than: (colon)" as a file end mark.

The SGS-ATES tapes are recorded at 600 baud.

The routine TTYI returns the input character in Register C and A, uses register A, B, C and 4 levels of stack.

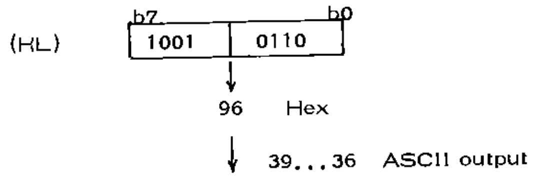
TTYI2 CALL F9C3

Reads from the serial line a byte trasmitted by two 5 bits characters; the byte is loaded in ACC and register C. Registers B, F are destroyed; 5 levels of stack are used.

ASCII CALL FA10

This routine outputs two ASCII characters which are the hexadecimal content of the memory location pointed to by HL register; it uses the TTYO routine.

For example



The routine increments HL for each call to simplify outputting blocks of memory and decrements DE which can be used as a byte counter.

A checksum is calculated in A' register as the binary sum of output characters: it must be initialized to zero if used.

The routine uses AF, AF', B, HL and DE registers and the registers of TTY0; it uses 6 levels of stack.

The speed of transmission is set by the subroutine BAUD and the BAUDRT flag.

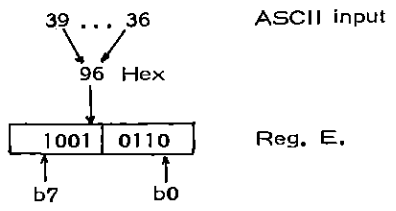
ASCIB: CALL FA12

This is the same as ASCII except the character to be output is in the B register and HL is not incremented.

BYTE: CALL FA2D

This subroutine is the opposite to ASCIB, it reads two ASCII characters and converts them from hexadecimal to a single 8 bit binary byte.

For example



The result in is the E register. There is no check for valid hex characters.

A checksum is calculed in register A' which is the binary addition of input bytes: A' must be initialized to zero if used.

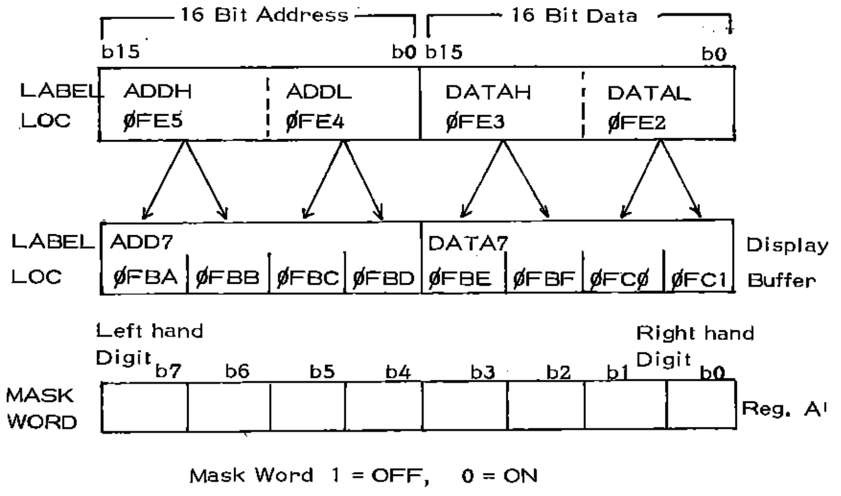
BYTE uses AF, AF' and E register and those used TTYI; it uses 7 levels of stack.

BYTE does not distinguish between Record format or Free format tapes and always calls TTYI (not TTYI1, so the user must initialize the values of INMODE BLOCKCNT an BLKLENGHT flags (see TTYI).

CONVDI: CALL FA7C

This subroutine converts two 16 bit words in memory, containing data to be displayed on the Address and Data 7 segment displays into 8 bytes of 7 segment display code.
Furthermore any digit in the display can be masked off or on with a control word.

The relation ship of the 16 bit words and 8 bytes is:



To use CONVDI to convert ADDH, L and DATAH, L to display codes, the Register HL must point to LEDH + 1 = 0FB9 which is the location just prior to the display buffer for the 7 segment display (see subroutine DISPL).

CONVDI uses registers HL, FA, AF' & BC, register DE is saved, it uses 3 levels of stack.

A P P E N D I X   1.

Hex codes for use in the display routine DISPL.

Letter		Hex Code
A	A	EE
b	b	3E
C	C	9C
D	]	F0
d	d	7A
E	E	9E
F	F	8E
G	G	BC
H	H	6E
h	h	2E
l	l	60
i	i	20
J	]	F0
K	K	4E
L	L	1C
M	M	EC, E0
m	m	2A, 22
N	N	EC
n	n	2A
O	O	FC
o	o	3A
P	P	CE
R	R	AC
r	r	0A
S	S	B6

Note that M and W use two bytes.

Letter		Hex Code
t	t	IE
U	U	7C
u	u	38
v	u	38
W	W	7C, 70
w	w	38, 30
Y	Y	4E
Z	Z	DA
=	=	12
-	-	02
-	-	10
o	o	3A
?	?	CA
\	\	4A
/	/	26
I	I	0C
2	2	DA
3	3	F2
4	4	66
5	5	B6
6	6	BE
7	7	EO
8	8	FE
9	9	F6
0	0	FC
EOT		01

A P P E N D I X   2.

When using DISPL remember that the buffer space LEDH, ADD7 and DATA7 is used any time there is a display on the Nanocomputer, it is used by the NC-Z operating system.

For this reason the user is recommended to create his own buffer and move the data to LEDH, ADD7 and DATA7 when a display is required.

This in an example

```
                ORG     100H
INIT :          LD      B, FFH          ; delay between display
DEL :           DEC     B               ; calls
                JP      NZ, DEL         ;
                LD      HL, 0200H       ; user buffer at 0200H
                LD      DE, 0FB8        ; DISPL buffer at 0FB8
                LD      BC, AH          ; 10 bytes to move
                LDIR                    ; move the block
                CALL    F909H           ; call DISPL
                JP      INIT

                ORG     0200H           ; data bytes
                DEFB    00H
                DEFB    00H
                DEFB    B6H
                DEFB    BCH
                DEFB    B6H
                DEFB    02H
                DEFB    EEH
                DEFB    1EH
                DEFB    9EH
                DEFB    B6H
                END.
```