DIAB DATABOARD

SBC-DOS /SBC-Basic

General

SBC-DOS with SBC-Basic from DIAB is a software package which comprises a disk operating system as well as an advanced Basic. The package has been designed for and is fully adapted to the DataBoard single board computer 1006.

Taken together, the SBC-DOS and the SBC-Basic form a powerful and very economical combination of program development tools. In fact, with this combination application programs can be produced right within the target system thereby eliminating the need for expensive development systems. The final application program may be run in a disk based system, which will be advantageous when program modifications will be required, or in a PROM based system. In both cases, the final application will handle mass storage units such as floppy disks, hard disks or bubble memories.

SBC-DOS is a fully compatible enhancement of the UFD-DOS, the operating system which is employed in the ABC800 personal computers from Luxor, Sweden. Likewise, the SBC-Basic is a new improved version of the Basic II, also used in the Luxor range of personal computers and now also available for the IBM PC.

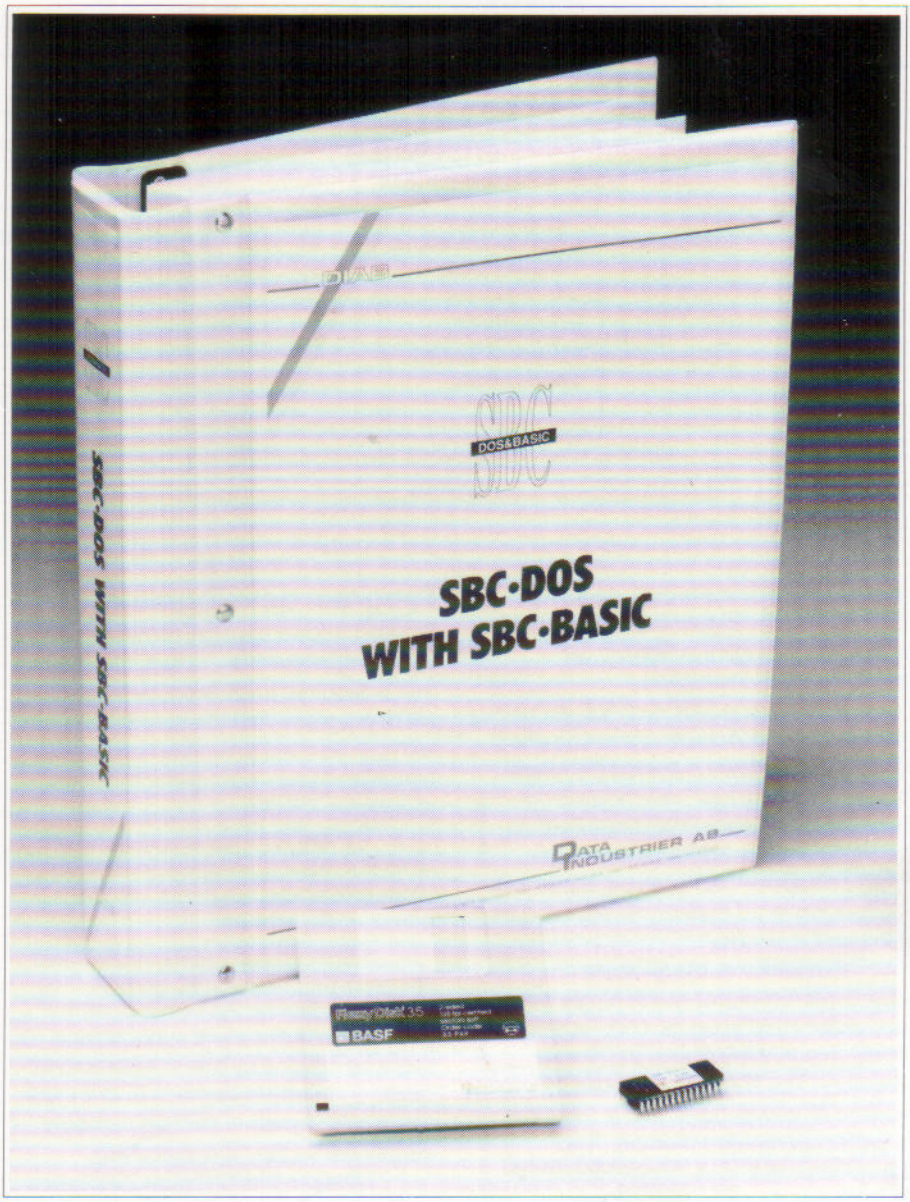
Software layout

The complete software package contains the SBC-DOS and SBC-Basic and a set of utility programs which will simplify system maintenance, application development, promming, disk initialization, copying, conversions, etc.

The SBC-Basic interpreter and the operating system consist of:

1. SBC-Basic interpreter.
2. SBC-Basic command handler (decoding and executing commands like LIST, RUN, etc).
3. SBC-DOS disk operating system which handles various types of mass storage units – floppy disks, hard disks and bubble memories.
4. INTCOM which is an interrupt and communication handler. INTCOM is used in certain special functions and to handle the V24/RS232 channels of the single board computer. INTCOM is also used to obtain file access in an external RAM/EPROM memory.

The four modules occupy a total of 32 kbytes and come ready to use in a 27256 EPROM chip.



The final system program can be made very compact as it normally employs only the SBC-Basic interpreter and possibly those parts of SBC-DOS and INTCOM that the application might require.

SBC-Basic – developed for data retrieval and control

SBC-Basic is a complete and very advanced Basic based upon DIAB's Basic II. However, SBC-Basic has a number of improvements in bit and interrupt handling making it an ideal programming language in applications involving process control and data retrieval. For instance, SBC-Basic is designed to perform direct bit manipulations on integer variables and SBC-Basic thus has instructions to set, reset and to test bits.

The INTCOM module supports three types of direct interrupt handling in Basic. With the ON INTR...GOSUB statement, an interrupt signal from an I/O module on the DataBoard bus might lead to immediate execution of Basic subroutines.

Time controlled processes can be performed by invoking subroutines with ON TIMER...GOSUB. And in order to simultaneously handle a control task

DIAB 606

DATA INDUSTRIER AB

Box 2029, S-183 02 Täby, Sweden. Phone +46 8 768 06 60. Telex 10978. Telefax +46 8 792 05 61. Teletex 2401-812 6002.

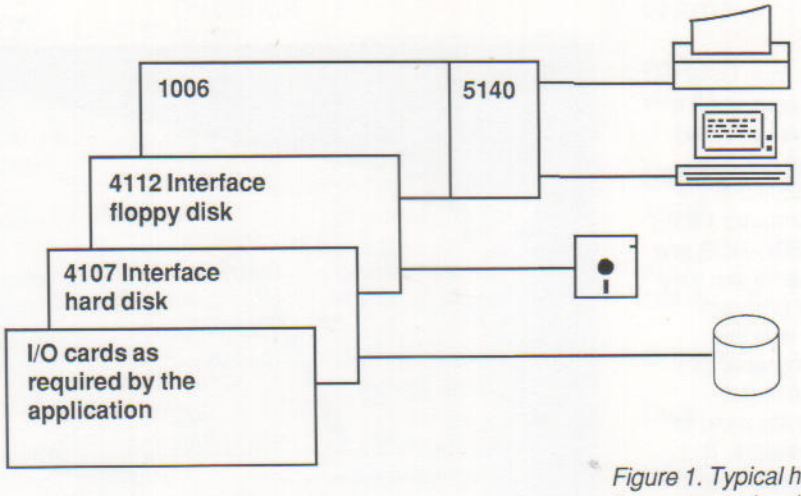


Figure 1. Typical hardware configuration for program development in an SBC-DOS environment.

and an operator's terminal, single input characters may trigger subroutine execution through the ON KEY...GOSUB instruction.

When using CMOS-RAM, variables declared COMMON will not be affected by power-down provided the single board computer has been hooked up with the DataBoard Power Fail Module 5059. The same applies if the single board computer has been equipped with battery powered memory chips, such as Mostek ZeroPower.

In addition to the 64 kbyte onboard internal memory, the single board computer 1006 accepts a non-volative RAM chip (NVRAM) as a parameter memory. The NVRAM will retain its information at power-down and is therefore a perfect parameter memory. The NVRAM on the 1006 single board computer holds 64 variables and has full program support from the SBC-Basic.

The single board computer 1006 handles up to 256 kbyte external memory. By using the functions included in INTCOM, the external memory can be handled in a file structured fashion which means that the application program will read and write in this memory instead of the floppy disk. Thus, the external memory will act as a sequential file. This buffer function is a very effective facility when running fast data handling programs. Programs and data may be mixed in the external memory.

The two internal serial channels on the 1006 has interrupt controlled communication with an 80 character buffer each. Transmission parameters may be defined from the application program.

Program development

Figure 1 shows a typical development system configuration. In fact, this is all that is required to develop, debug and test a program. And the very same system can be used to run the final application even if you would prefer to put the program in PROM.

The terminal is used for programming and editing. Program listings and cross references are printed on the printer or the screen.

The application program may use the terminal or the printer for error messages, etc. During the development phase, subroutines and subfunctions might be run one by one in a command mode, a step-by-step mode or in a trace mode (which means that line numbers are printed as they are executed).

The flow chart in figure 2 indicates how program development and implementation is performed.

Once the program is finalized, the volume of the program is normally reduced with the aid of the NOREM utility which removes all remark statements. Another utility, SQUEZ, which is an option, can be used to further reduce the program volume. This utility also features program encryption.

SBC-DOS in disk based systems

If the application program is to be run in a disk based application, a utility is employed to start an arbitrarily choosen program at power-up (auto start).

When used in a disk based application, SBC-DOS and SBC-Basic can change between different programs (CHAIN) as well as read and write data in mass storage units. Reading and writing is performed with sequential files or organized with search keys in an ISAM database (included in option package1).

SBC-DOS in PROM based systems

The application program can also be transferred to EPROM, which is one of the major features of SBC-DOS.

The final program will then be converted to a prommable format by using the BACABS utility. The CONFIG utility is used to insert the options required in the system. Following these steps, the program is ready for promming. During the development phase, the most extensive Basic interpreter has normally been employed and at this stage you may replace the interpreter with a less extensive one. Program and data can also be located in PROM in an external memory, with a maximum volume of 256 kbyte.

SBC-DOS option package 1

The SBC-DOS option package 1 contains software support for high resolution graphics (HR Graphics), GPIB control, asynchronous communication (D-LINK) and the ISAM database.

The high resolution graphics requires the use of the two DataBoard modules 5120 and 5121. The colour graphics has a resolution of 512x240 or 512x380 pixels depending upon the

monitor employed. The HR option adds a set of instructions for drawing lines, vectors, circles, rectangles, text, etc to the SBC-Basic.

The GPIB concept requires that the DataBoard module 4027 is added to the hardware configuration.

D-LINK is a modern and economical communication concept based upon an asynchronous low speed network where several computers of different types can be interconnected. D-LINK may be used to transfer data from a number of PROM based systems to a disk based system for further data processing.

The options can be used in systems including floppy disks, hard disks or bubble memories in which case the required options will be loaded at power-up. If the options are used in a system where the application program is located in an EPROM, then the configuration utility CONFIG allows for option selection. The operator simply selects which options shall be included whereupon CONFIG creates a file which is ready for promming.

SBC-DOS development package

The SBC-DOS development package makes it possible to write and test programs in Z80 assembler. In short, the development package consists of:

ASMZ	advanced MACRO assembler supporting DIAB and Zilog mnemonics
ED	line orientated editor
TRACE	debugger
ESTAB	linker

The kit also includes programs handling object files created by the assembler and other utilities which simplify mixing of SBC-Basic programs and machine code.

Ordering information

Product	Order No
SBC-DOS with SBC-Basic	8297
SBC-DOS option package 1	8298
SBC-DOS development package	8275

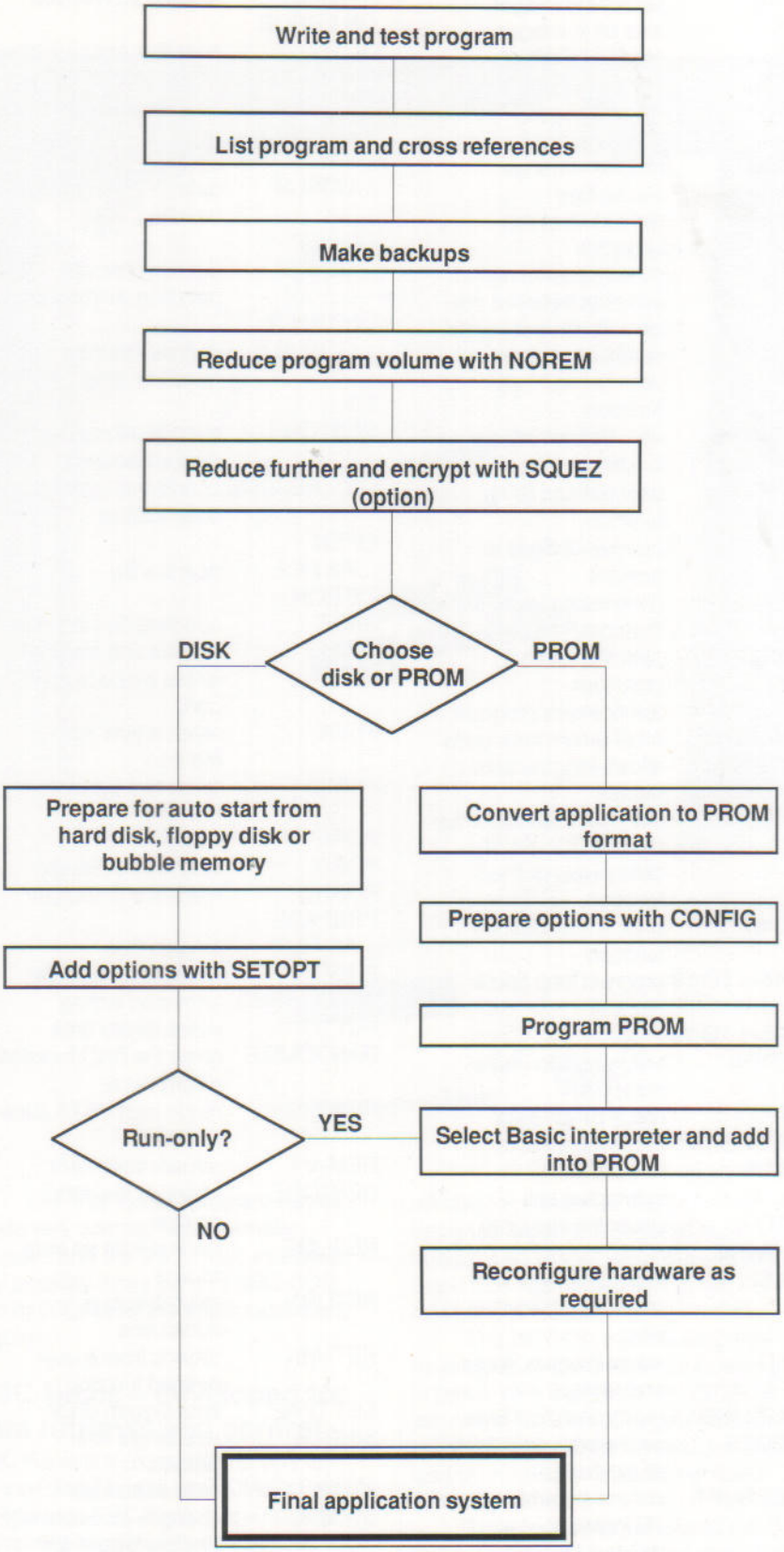


Fig 2. Program development within target system

Instructions

BITC	clears bit in integer
BIT%()	tests bit in integer
BITS	sets bit in integer
BYE	starts SBC-DOS command interpreter
CALL	machine code subroutine call
CHAIN	program change
CLOSE	closes files
COMMON	defines common variables
CUR	cursor positioning
CVT	converts between integers, floats and strings
DATA	stores internal data
DEF FNA	user defined float function
DEF FNA%	user defined integer function
DEF FNA$	user defined string function
DIGITS	number of digits to present
DIM	dimensions vectors and matrices
DOUBLE	sets double float precision
END	terminates a program
ERRCODE	most recent error code
EXTEND	allows long variable names
FLOAT	sets float calculation as default
FN	calls a user defined function
FNEND	ends a user defined function
FOR I=1 TO 10	program loop (increment 1)
FOR I=1 TO 10 STEP 0.5	program loop (increment 0.5)
GET	reads binary data
GET...COUNT	reads binary data
GOSUB	unconditional subroutine call
GOTO	unconditional jump
IF...THEN ...ELSE	test statement
INP	reads a byte from an I/O port
INPUT	reads integers, floats and strings
INPUT LINE	reads complete lines
INTEGER	sets integer calculation as default
INTERRUPT	control statement for I/O interrupt
KILL	delete a file
LET	assignment
NAME...AS	changes names of files
NEXT	ends a FOR...TO loop
NO EXTEND	inhibits long variable names
NO TRACE	disables trace mode
NVRAM	writes to NVRAM
NVRAM()	reads from NVRAM
ON ERROR GOTO	specifies error handling
ON...GOTO	conditional jump
ON...GOSUB	conditional subroutine call
ON INTR ...GOSUB	defines interrupt handling, I/O
ON KEY ...GOSUB	defines interrupt handling, keyboard
ON TIMER ...GOSUB	defines interrupt handling, time
ON... RESTORE	conditional restorage of data statements
ON...RESUME	conditional return from error routine
OPEN ...AS FILE	opens a file
OPTION BASE	specifies first index in vectors and matrices
OUT	writes a byte to an I/O port
PEEK	reads a byte from memory
PEEK2	reads two bytes from memory
POKE	writes in memory
POSIT	sets the file pointer
POSIT()	reads the file pointer
PREPARE ...AS FILE	creates files
PRINT	writes text and data
PRINT USING	formatted writing
PUT	writes binary data
RANDOMIZE	gives the RND function a start value
READ	reads next DATA statement
REM or !	remark statement
RESTORE	changes the data pointer
RESUME	returns from an error routine
RETURN	returns from a subroutine
RETURN...	returns from a user defined function
SET TIME	sets system clock
SINGLE	sets single float precision
STOP	halts execution
SWAP%	changes places of high and low byte in an integer
SYS	gives system status
TAB	moves cursor to specifed column
TIME$	reads system clock
TRACE	enables trace mode
VAROOT	gives the address to a variable root
VARPTR	gives the address to a variable area
WHILE	conditional program loop
WEND	ends a WHILE loop

String functions

ADD$	ASCII arithmetics, addition
ASCII	ASCII value of a string
CHR$	converts ASCII values to a string
COMP%	ASCII arithmetics, comparison
DIV$	ASCII arithmetics, division
HEX$	converts integers to hexadecimal string
INSTR	searches for substring
LEFT$	gives part of a string
LEN	gives length of a string
MID$	gives part of a string
MUL$	ASCII arithmetics, multiplication
NUM$	converts decimal value to a string
OCT$	converts integer to an octal string
RIGHT$	gives part of a string
SPACE$	gives a string containing spaces
STRING$	gives a string containing ASCII characters
SUB$	ASCII arithmetics, subtraction
VAL	converts a string to a decimal value

Mathematical functions

ABS(X)	absolute value of X
ATN (X)	arctangens of X
COS (X)	cosine of X
EXP (X)	natural exponential of X
FIX (X)	truncated value of X
INT (X)	integer of X
LOG (X)	natural log of X
LOG10 (X)	log 10 of X
MOD(X,Y)	X modulo Y
PI	π (3,14.....)
RND	a random figure 0<=RND<1
SGN (X)	sign of X
SIN (X)	sine of X
SQR (X)	square root of X
TAN (X)	tangens of X