

# MSX X-BASIC

X-BASIC also known as MSX-BASIC KUN , KUN BASIC or TURBO BASIC (Turbo-BASIC) is the run-time compiler of [BASIC](#), which is specifically good for writing [demos](#) and games due to its' relatively high performance.

[x-basic-2.1.rom](#), [Source](#)

[NestorBASIC](#) is a development of X-BASIC.

## KUN BASIC

### MSX-BASIC-KUN (BASIC COMPILER)

by J.Suzuki 1989  
this document & samples by Sho Endo  
translate to english by LASP  
this text file typed without changes by Nestor Soriano

MSX-BASIC-KUN is an incredible BASIC compiler. It will compile a BASIC program on memory in few seconds and execute it 15 to 100 times faster!! It can compile most of the statements and functions of MSX-BASIC and can handle strings and floating numbers. Once you see it, you'd feel you'd never need to learn the Z-80 machine language. Real time games, C.G., demo programs can be written by the ease of BASIC for machine language speed.

\*\*\* USAGE \*\*\*

#### 1. Settings & General knowledge

This compiler is sold in Japan as a cartridge for 4500 yen. You just set it in a slot to use it. Also, you can find this compiler in MSX2+ Sanyo machines, in ROM.

Now you are in BASIC mode as usual, except that two commands are available:

CALL RUN  
CALL TURBO ON/OFF

"CALL" can be written as "\_" (underscore). I will use that from now on.

\_RUN is the command to compile and execute the entire program on memory. If it finds an error it will stop and yield the message.

\_TURBO ON is the statement to define the beginning of the turbo block.  
\_TURBO OFF is the end of the block.

The turbo block is the part of the program you want to execute fast. When the entire program contains some uncompileable statements, you can define the block to be compiled using this set.

#### EXAMPLE

```
100 SCREEN 8:DEFINT A-Z
110 BLOAD"PICTURE",S
120 _TURBO ON
130 FOR X=0 TO 255
140 LINE(X,0)-(X,211),0
150 NEXT X
```

```

160 _TURBO OFF
170 GOTO 170

```

This program cannot be "\_RUN", because the "BLOAD" is one of the commands that cannot be compiled. If you "RUN" this, the part lines 130 through 150 will be executed fast.

As '\_RUN"FILE"' is not supported, you have to add \_TURBO ON and \_TURBO OFF at the beginning and the end if you want to RUN"FILE" and have the effect.

```

100 FOR I=0 TO 999
110 ...
.
.
.
890 'END OF THE PROGRAM

```

So, this can be \_RUN or add 10 \_TURBO ON and 900 \_TURBO OFF and RUN"FILE".

If you \_RUN a program containing "\_TURBO ON/OFF" it will be an error.

\_TURBO ON/OFF can not be written in a multi-statement lines.

\_TURBO ON/OFF can not be nested. But you may have many closed blocks in a program.

Variables and arrays are handled differently in and outside of the blocks. Once you are out of the block, variables and arrays used in the block are lost. Only, the integer types can be defined as common.

```

100 DEFINT A-Z:DIM C(2),D(2)
110 A=1:B=2:C(0)=3:D(0)=4
120 _TURBO ON(A,C())
130 DIM D(2)
140 PRINT A,B,C(0),D(0)
150 A=5:B=6:C(0)=7:D(0)=8
160 _TURBO OFF
170 PRINT A,B,C(0),D(0)

```

```

RUN
1 0 3 0
5 2 7 4
Ok

```

Floating numbers used by the compiler is a special format 3-byte value. It's accuracy is about 4.5 digits. Double precision is not available.

An array must be declared by a constant in the beginning.

This compiler parses the BASIC program in RAM and places the object code and variables in the remaining free RAM. So there is a limit of the size of the source program about 10K. Big arrays, string variables (each uses 256 byte), CLEAR ???,&H???? will make the situation tighter as you can imagine. The compiled objects can not be saved as independent programs.

Interrupts available, such as KEY(1) ON, OFF etc. But it will decrease the efficiency of the executed object's size & speed.

Some statements may not work correctly.

```

100 GOTO 130
110 A=3/2
120 END
130 DEFINT A-Z

```

```
140 GOTO 110
```

If you RUN this, A is 1. If you \_RUN this, A is 1.5. DEF??? will be effective when encountered during the execution in the case of interpreter, while it depends on the order of line number in the other case.

A little complicated string operation may cause easily a "String formula too complex" error. As this compiler has only one level of stack for it. Break a long string formula into multiple small ones, if so.

If you \_RUN an endless program, you can not stop it. Make a part to check keyboards.

```
100 GOTO 100 'Reset or power off to stop
```

```
100 IF INKEY$="" THEN 100
```

```
110 END
```

is better.

## 2. Difference from MSX-BASIC interpreter

List of statements, commands and functions that can not be compiled.

AUTO, BASE, BLOAD, BSAVE, CALL, CDBL, CINT, CLEAR, CLOAD, CLOAD?, CLOSE, CONT, CSAVE, CSNG, CVD, CVI, CVS, DEFFN, DELETE, DRAW, DSKF, EOF, ERASE, ERL, ERR, ERROR, EQV, FIELD, FILES, FPOS, FRE, GET, IMP, INPUT#, KEY LIST, LFILES, LINEINPUT#, LIST, LLIST, LOAD, LOC, LOF, LPRINT USING, LSET, MAXFILES, MERGE, MOTOR, MKD\$, MKI\$, MKS\$, NAME, NEW, ON ERROR GOTO, ON INTERVAL GOSUB (due to a bug), OPEN, PLAY, PRINT#, PRINT USING, PUT KANJI, RENUM, RESUME, RSET, SAVE, SPC, TAB, TRON, TROFF, WIDTH.

List of those with limits.

CIRCLE: Start, end angles and aspect ratio can't be specified.

COPY: Only graphic COPY is available.

DEFDBL: Same as DEFSNG.

DIM: Must come first in the program or in the turbo block.

INPUT: Can handle only one variable at the time.

KEY: ON KEY GOSUB, KEY(n) ON/OFF only.

LOCATE: x,y must be given in as a set. No cursor switch parameter.

NEXT: Variable names after the NEXT can not be omitted.

ON: ON STOP GOSUB, ON INTERVAL GOSUB not available.

PRINT: Commas work in a different way. No wrapping for digits.

PUT: PUT SPRITE only.

RUN: Variables won't be initialized.

SCREEN: Screen mode and sprite size only.

SET: SET PAGE only.

STOP: Same as END.

USR: Parameter type must be integer only.

VARPTR: File number can not be given as the parameter.

Otherwise there is no significant difference.

In general, I/O commands & functions, and editing commands can not be compiled. Of course they are available in the direct mode, and outside of the turbo block. You can edit, debug and save a program in MSX-BASIC and execute it by \_RUN.

If you want to use PRINT# to write characters on GRP:, use it outside of turbo block. Otherwise study the sample, "PRINT.TRB".

If you want to use PLAY, use BGM compiler, and get the sound by USR(n).

### 3. New features added

3 special commands are available by starting a remark line with some specific characters.

#### #I

Stands for INLINE. You can write a short machine-language routine.

```
100 DEFINT A-Z
110 K=1
120 '#I &H2A,K
130 '#I &HF3,&HCD,@150,&HFB
140 END
150 'SUB
160 RETURN
```

120 means LD HL,(K); K must be a simple variable of integer type.

130 means DI

CALL @150 ;Be careful, this line won't be RENUMed.

EI

#### #C

Stands for CLIP. In the screen modes 5 through 8 (except for PAINT, and CIRCLE), this will set clipping on and off.

```
10 SCREEN 8
20 '#C-
30 LINE(0,0)-(255,255) 'Y CLIPPED
40 IF INKEY$="" THEN 40
50 '#C+
60 LINE(0,0)-(255,255) 'NOT CLIPPED
70 IF INKEY$="" THEN 70
```

#### #N

Check if NEXT overflows.

```
10 FOR I%=0 TO &H7FFF
20 NEXT I%
```

This program will end up in a "Overflow error" in MSX-BASIC. And if \_RUN, it will be an endless loop. If #N+ is specified, it will end normally. This code will decrease the efficiency of the object, too. Better not use unless it's really necessary. To clear, specify #N-.

NOTE: In MSX-2+ Sanyo you can find a new command:

CALL BC

This command turn on the BASIC COMPILER options.

[Source](#)

## Links

[MSX-BASIC-KUN \(BASIC COMPILER\) | MSX2-Technical-Handbook](#)

 [MSX-BASIC](#)

<http://sysadminmosaic.ru/en/msx/x-basic/x-basic>

2023-05-01 13:34

