

## MMBasic V3.2 Quick Reference

## MMBasic V3.2 Quick Reference

## MMBasic V3.2 Quick Reference

## Literals / Variables

**Literals**

"Strings"; Numbers are decimal or represented as:  
 &Hnn Hex Literal, e.g. &H3C (60)  
 &Bnn... Binary Literal, e.g. &B00100011 (35)  
 n.nE+n Scientific, e.g. 1.6E+4 (16000)

**System Variables**

MM.HRES Horizontal Screen Resolution  
 MM.VRES Vertical Screen Resolution  
 MM.VER Firmware Version  
 MM.DRIVE\$ Current Default Drive  
 MM.FNAME\$ Current Default File  
 MM.ERRNO Last Error Code  
 MM.I2C Last I2C Result Code  
 MM.OW Last 1-Wire Result Code

**User Variable Names**

Numeric a-z[\_a-z0-9.\_] (max.32 chars)  
 String a-z[\_a-z0-9.\_]\$ (max. 32 chars)

## Operators

**Arithmetic**

^ \* / Exponent, Multiply, Divide  
 MOD \ Modulus (remain.), Integer Divide  
 + + - Add, String Concat., Subtract

**Logical**

= <> Equality, Inequality  
 > < Greater Than, Less Than  
 <= or =< Less Than Equal To  
 >= or => Greater Than Equal To  
 AND OR Conjunction, Disjunction,  
 NOT XOR Logical Inverse, Exclusive OR

**Error**

0 No error  
 1 No SD card found  
 2 SD card is write protected  
 3 Not enough space  
 4 All root directory entries are taken  
 5 Invalid filename  
 6 File does not exist  
 7 Directory does not exist  
 8 File is read only  
 9 Cannot open file  
 10 Error reading from file  
 11 Error writing to file  
 12 Not a file

## Codes

## Codes

13 Not a directory  
 15 Directory not empty / Hardware error  
 16 Flash Memory write failure

**File Open Mode**

INPUT Read Only  
 OUTPUT Write (Overwrite if exists)  
 APPEND Write (Append if exists)

**Format String**

% [flags] [width] [.prec] type

flags: - Left justify

0 Use 0 for the pad char, not space.  
 + A plus sign is shown for positive values.  
 space Space as sign, unless negative.

width:min. chars to output, less causes padding,  
 more causes expansion.

prec: no. of fraction digits for **e**, or **f** type, or the max.  
 no of significant digits for **g** type. Precede by a  
 dot (.) if used.

type: **g** format for the best presentation  
**f** format with decimal point and digits  
**e** Format in exponential format  
**G** exponential output with uppercase E  
**F** exponential output with uppercase E.

If format specification not specified "%g" is assumed.

**I2C**

0 No error  
 1 Received NACK response  
 2 Command timed out  
 4 Received general call addr. (slave mode)

**Pin Config**

0 None  
 1 Analog In [Pins 1-10]  
 2 Digital In [Pins 1-10 @ 3.3v, Pins 11-20 @ 5v]  
 3 Frequency In [Pins 11-14]  
 4 Period In [Pins 11-14]  
 5 Count In [Pins 11-14]  
 6 Interrupt LOW→HIGH [Pins 1-20]  
 7 Interrupt HIGH→LOW [Pins 1-20]  
 8 Digital Out [Pins 1-20]  
 9 Open Collector In [Pins 11-20]

**Assignment**

CLEAR  
 DATA  
 DIM variable(elements...)

## Commands / Statements

ERASE variable  
 LET variable =  
 LOCAL variable[, variable]  
 READ variable[, variable]...  
 RESTORE

**Editor**

AUTO [start] [, increment]  
 DELETE line  
 DELETE -lastline  
 DELETE firstline [- lastline]  
 EDIT [file|line-nbr] | (full-screen)  
 LIST [line]  
 LIST -lastline  
 LIST firstline [- lastline]  
 RENUMBER [first] [,incr] [,start]

**External Pins**

PIN(pin) = value  
 SETPIN pin, cfg  
 SETPIN pin, cfg, line

**File System**

CHDIR dir\$  
 CLOSE [#]nbr [, [#]nbr]  
 CLOSE CONSOLE  
 COPY source\$ TO target\$  
 DRIVE drive\$  
 FILES [search\_pattern\$]  
 INPUT #nbr, list of variables  
 KILL file\$  
 LINE INPUT #nbr, string-variable\$  
 LOAD file\$  
 MERGE file\$  
 MKDIR dir\$  
 NAME old\$ AS new\$  
 OPEN fname\$ FOR mode AS [#]fnbr  
 OPEN "comspec" AS [#]fnbr  
 PRINT|? @@(x,y[,m])| expr [[,;|expr]...  
 RMDIR dir\$  
 SAVE [file\$]  
 SAVEBMP file\$

**Flow Control**

CONTINUE  
 DO <statements> LOOP [UNTIL expression]  
 DO WHILE expression <statements> LOOP  
 ELSE

## MMBasic V3.2 Quick Reference

## MMBasic V3.2 Quick Reference

## MMBasic V3.2 Quick Reference

## Commands / Statements

```

ELSEIF expression THEN
ENDIF
END [SUB | FUNCTION]
EXIT [FOR | FUNCTION | SUB]
FOR count=start TO end [STEP inc.]
FUNCTION fn_name (arg1[, arg2, ...])
GOSUB
GOTO
IF expression THEN
IRETURN
NEXT [count_var][,count_var]...
ON variable GOTO|GOSUB line[,line,...]
PAUSE nbr
RETURN
SUB sub_name (arg1[, arg2, ...])
I2C/1-Wire
I2CEN speed,timeout [,int_line]
I2CDIS, I2CSDIS
I2CSEND addr,opt,len,data[,data...]
I2CRCV addr,hold,rcvlen,rcvbuf
[,sendlen,data[,data...]]
I2CSEN addr,mask,opt,send_int,rcv_int
I2CSSEND sendlen,data[,data...]
I2CSRCV rcvlen,rcvbuf,rcvd
NUM2BYTE number,var1,var2,var3,var4
NUM2BYTE number,array(x)
OWREAD pin,flag,len,data[,data...]
OWRESET pin [,presence]
OWSEARCH pin,srchflag,ser[,ser...]
OWWRITE pin,flag,len,data[,data...]
Keyboard
INPUT ["prompt string";]
LINE INPUT [prompt$,],string$
Miscellaneous/Configuration
COPYRIGHT
DATE$ = "DD-MM-YY" or "DD/MM/YY"
CONFIG COMPOSITE PAL | NTSC
CONFIG VIDEO ON | OFF
CONFIG CASE TITLE | UPPER | LOWER
CONFIG KEYBOARD US | BE | FR | GE | IT
CONFIG TAB 2 | 4 | 8
ERROR [error_msg$]
MEMORY
OPTION BASE 0 | 1
OPTION ERROR ABORT | CONTINUE

```

## Commands / Statements

```

OPTION Fnn str$
OPTION PROMPT str$
OPTION USB ON | OFF
OPTION VIDEO ON | OFF
POKE hiword, loword, val
RANDOMIZE nbr
REM Comment or ' Comment
RUN [line] [file$]
NEW
SETTICK period, line
TIME$ = "HH:MM:SS"
TIMER = msec
TROFF / TRON
Screen
CLS
CIRCLE (x, y) ,r [,c [,F]]
FONT #nbr [,scale] [,reverse]
FONT LOAD file AS #nbr
FONT UNLOAD #nbr
LINE [(x1 , y1)] - (x2, y2) [,c [,B[F]]]
PIXEL(x,y)
PRINT / ? expr [[,;]expr]...
PRINT @(x,y[,m]) expr [[,;]expr]...
Serial I/O
CLOSE CONSOLE
OPEN "COMSPEC" AS CONSOLE
XMODEM SEND file$
XMODEM RECEIVE file$
Sound/PWM/Pulse
SOUND frequency,duration
SOUND frequency,duration,dutycycle
PULSE pin, width

```

## Functions

```

Date, Time, Timer
DATE$
TIME$
TIMER
File System
CWD$
DIR$ ([fspec [,type]])
EOF ([#]nbr)
INPUT$ (nbr,[#]fnbr)
I2C/1-Wire
BYTE2NUM(array(x))
BYTE2NUM(arg1,arg2,arg3,arg4)
OWCRC8 (len, cdata [,cdata...])

```

## Functions

```

OWCRC16 (len, cdata [,cdata...])
I/O
LOC ([#]nbr)
LOF ([#]nbr)
PIN (pin)
SPI (rx, tx, clk [,data [,speed] ])
Math / Number
ABS (nbr)
CINT (nbr)
EXP (nbr)
HEX$ (nbr)
LOG (nbr)
PI
SGN (nbr)
SQR (nbr)
TAN (nbr)
ATN (nbr)
COS (nbr)
FIX (nbr)
INT (nbr)
OCT$ (nbr)
RND (nbr)
SIN (nbr)
STR$ (nbr)
Memory
PEEK (hiword,loword)
Screen
POS
PIXEL(x,y)
String / Character
ASC (str$)
BIN$ (nbr)
CHR$ (nbr)
FORMAT$ (nbr [,format$])
INSTR ([start,] search$, pattern$)
LEFT$ (str$, nbr)
LEN (str$)
LCASE$ (str$)
MID$ (str$, start [,nbr])
RIGHT$ (str$, nbr)
SPACE$ (nbr)
STRING$(nbr, val|str$)
TAB(nbr)
UCASE$ (str$)
VAL (str$)
INKEY$

```

## Info

This Quick Reference is a guide to the Maximite Family version of MMBasic. For full syntax, usage, examples, and DOS variations, refer to the MMBasic Language Manual V3.2 at:  
<http://geoffg.net/maximite.html>