

Keywords	
'	Single line comment
/* ... */	Multi line comment
ADC FREQUENCY [freq]	This changes the sampling frequency of the ADC conversion without having to close and re-open
ADC CLOSE	Releases the pins to normal usage
ADC OPEN freq, n_channels[, interrupt]	This allocates up to 4 ADC channels for use and sets them to be converted at the specified frequency
ADC START array1!() [,array2!() [,array3!() [,array4!() [,C1min] [,C1max] [,C2min] [,C2max] [,C3min] [,C3max] [,C4min] [,C4max]	This starts conversion into the specified arrays. The arrays must be floating point and the same size. 'Cxmin' and 'Cxmax' will scale the readings.
ADC RUN array1% (), array2%()	Runs the ADC continuously in double buffered mode.
ARRAY ADD in(), value, out()	This adds (or appends for strings) the value 'value' to every element of the matrix in() and puts the answer in out().
ARRAY INSERT targetarray(), [d1] [,d2] [,d3] [,d4] [,d5], sourcearray()	This is the opposite of ARRAY SLICE
ARRAY SET value, array()	Sets all elements of array() to value

Keywords (cont)	
ARRAY SLICE sourcearray(), [d1] [,d2] [,d3] [,d4] [,d5] destinationarray()	This command copies a specified set of values from a multi-dimensional array into a single dimensional array.
ARC x, y, r1, [r2], a1, a2 [, c]	Draws an arc of a circle with a given colour and width between two radials (defined in degrees)
AUTOSAVE [CRUNCH APPEND]	Enter automatic program entry mode. CRUNCH removes all comments and whitespace, APPEND appends to the end of the existing program
BIT (var%, bitno) = value	Sets the specified bit <i>bitno</i> in the 64-bit integer variable <i>var%</i> to <i>value</i>
BLIT READ [#]b, x, y, w, h	Will copy a portion of the display to the memory buffer '#b'. 32 buffers are available ranging from #1 to #32.
BLIT WRITE [#]b, x, y [,mode]	Will copy the memory buffer '#b' to the display. 32 buffers are available ranging from #1 to #32.
BLIT LOAD [BMP] [#]b, fname\$ [,x] [,y] [,w] [,h]	Will load a blit buffer from a 24-bit bmp image file.
BLIT CLOSE [#]b	Will close the memory buffer '#b'

Keywords (cont)	
BLIT FRAMEBUFFER from, to, xin, yin, xout, yout, width, height [,colour]	Copies an area of a specific 'from' framebuffer N, F, or L to another different 'to' framebuffer N, F, or L.
BLIT MEMORY address, x, y [,col]	Copies an area of memory treated as a packed array of colour nibbles to the current graphical output as specified by FRAMEBUFFER WRITE
BLIT COMPRESSED address%, x, y [,col]	Acts the same as BLIT MEMORY but assumes the data is compressed and ignores the top bit in the width and height
BLIT x1, y1, x2, y2, w, h	Copy one section of the display screen to another part of the display
BOX x, y, w, h [, lw] [,c] [,fill]	Draws a box on the display with the top left hand corner at 'x' and 'y' with a width of 'w' pixels and a height of 'h' pixels
BYTE (var\$, byteno)=value	Sets byte <i>byteno</i> in string <i>var%</i> to integer value <i>value</i> . <i>value</i> can be in the range 0- 255.
CALL usersubname\$ [,user-subparameters,..]	Call subroutine <i>usersubname\$</i>
CAT S\$, N\$	Appends <i>N\$</i> to <i>S\$</i>



Keywords (cont)

CHAIN <i>fname\$</i> [<i>cmdline\$</i>]	Invokes program <i>fname\$</i> and keeps the variables intact
CHDIR <i>dir\$</i>	Changes the current working directory to <i>dir\$</i>
CIRCLE <i>x</i> , <i>y</i> , <i>r</i> [, <i>lw</i>] [, <i>a</i>] [, <i>c</i>] [, <i>fill</i>]	Draw a circle centered at <i>x</i> and <i>y</i> with a radius of <i>r</i> on the display
CLEAR	Delete all variables
CLOSE [#] <i>fnbr</i> [, [#] <i>fnbr</i>]	Closes the file(s) with the specified file number(s)
CLS [<i>color</i>]	Clears the screen
CMM2 LOAD RUN	Loads and or runs a program from disk using the CMM2 program loading mechanism
COLOR <i>fore</i> [, <i>back</i>]	Sets the default color for commands that display on the screen
COLOUR MAP <i>inarray%()</i> , <i>outarray%()</i> [, <i>colormap%()</i>]	Generates RGB888 colors in <i>outarray%</i> from color codes (0- 15) in <i>inarray%</i>
CONFIGURE <i>cfg</i>	Configures a board as per the <i>cfg</i> specified equivalent of OPTION RESET
CONFIGURE LIST	Lists all the various configurations available for the firmware version
CONST <i>id</i> = <i>expression</i> [, <i>id</i> = <i>expression</i>][, ...]	Creates constant identifier <i>id</i> and sets it to the value of <i>expression</i> . This value cannot be changed once set.

Keywords (cont)

CONTINUE	Resume running a program that has been stopped by an END statement, an error, or CTRL-C
CONTINUE DO FOR	Skip to the end of a DO/LOOP or a FOR/NEXT loop
COPY <i>fname1\$</i> TO <i>fname2\$</i>	Copy file <i>fname1\$</i> to new file <i>fname2\$</i>
COPY <i>fname\$</i> TO <i>dirname\$</i>	Wildcard copy. The bulk copy is triggered if <i>fname\$</i> contains a '*' or a '?' character. <i>dirname\$</i> must be a valid directory name and should NOT end in a slash character
CPU RESTART	Will force a restart of the CPUs. This will clear all variables and reset everything
CPU SLEEP <i>n</i>	Will cause the processors to sleep for <i>n</i> seconds
CSUB <i>name</i> [<i>type</i>][, <i>type</i>] [, ...]	Defines the binary code for an embedded machine code program module written in C or ARM assembler.
END CSUB	Ends the CSUB definition
DATA <i>constant</i> [, <i>constant</i>][, ... [READ	Stores numerical and string constants to be accessed by
DATE\$ = "- DD-MM-- YY[YY]" "- DD/MM/YY[- YY]" "YYYY-MM- DD" "YYYY/- MM/DD"	Set the date of the internal clock/calendar

Keywords (cont)

DEFINEFONT # <i>Nbr</i>	This will define an embedded font which can be used alongside or to replace the built in font(s) used on an attached LCD panel.
END	Ends the font definition
DEFINEFONT	
DEVICE BITSTREAM <i>pinno</i> , <i>n_transitions</i> , <i>array%()</i>	Generate an extremely accurate bit sequence on pin <i>pinno</i>

Predefined Read Only Variables

MM.VER	Returns the firmware version number
MM.CMD-LINE\$	Returns any command line arguments passed to the current program
MM.DEVICE\$	The device or platform that MMBasic is running on
MM.DISPLAY	Returns a 1 if a display is configured, else 0
MM.ERRNO	A non-zero value means an error occurred
MM.ERRMSG\$	The error message
MM.FLAGS	Returns the system flags register
MM.FON-THEIGHT	Returns the height of the current font
MM.FON-TWIDTH	Returns the width of the current font
MM.HRES	Returns the horizontal resolution of the configured display
MM.VRES	Returns the vertical resolution of the configured display

Predefined Read Only Variables (cont)

MM.HEIGHT	Returns the number of characters high the configured display is with the current font
MM.WIDTH	Returns the number of characters wide the configured display is with the current font
MM.HPOS	Returns the current horizontal position
MM.VPOS	Returns the current vertical position
MM.INFO\$(-AUTORUN)	Returns the current setting of OPTION AUTORUN
MM.INFO-O(ADC)	Returns the number of the buffer currently ready to read when using ADC RUN (1 or 2). Returns 0 if nothing ready.
MM.INFO-O(ADC DMA)	Returns 1 if the ADC DMA is active
MM.INFO-O(BOOT)	Tells you the reason of the last system restart
MM.INFO-O(BOOT COUNT)	The number of restarts since the last flash ram format
MM.INFO\$(-CPU-SPEED)	Returns the current CPU speed
MM.INFO\$(-LCDPANEL)	Returns the value of the configured LCD panel, or a blank string if none
MM.INFO(L-CD320)	Returns 1 if the configured LCD panel can run OPTION LCD320
MM.INFO\$(-SDCARD)	Returns the status of the SD card
MM.INFO\$(-CURRENT)	Returns the name of the current program

Predefined Read Only Variables (cont)

MM.INFO-O\$(-PATH)	Returns the path of the current program
MM.INFO-O(DISK SIZE)	Return the size of the active drive
MM.INFO-O\$(-DRIVE)	Returns active drive "A:" or "B:"
MM.INFO-O(E-XISTS FILE fname\$)	Returns 1 if the file specified exists, returns -1 if fname\$ is a directory, otherwise returns 0.
MM.INFO-O(E-XISTS DIR dirname\$)	Returns 1 if the directory exists, else 0
MM.INFO-O(FREE SPACE)	Returns the amount of free space available on the current drive
MM.INFO-O(F-FILESIZE file\$)	Returns the size of file <i>file\$</i> or 0 is not found

ALL variables start with MM., e.g. MM.VER, MM.FLAGS, etc.