

Change log for Micromite MMBasic Ver 5.03.02

Bug Fixes:

- Fixed a bug which sometimes caused VAL() to throw an error (thanks to Robert Rozee).
- Fixed the definition of getConsole() in CFunctions.h (used for embedded C code only).

Change log for Micromite MMBasic Ver 5.03.01

New Features:

- The format of the version numbers has changed slightly. The minor version number will be two digits and rather than using a letter to indicate a minor release (ie, Ver 5.3A) two digits will be appended to the version number (eg, version 5.03.01). MM.VER will return the version as x.yyyy (eg, 5.0301) which is compatible with the previous behaviour.
- The LIBRARY SAVE command will now add the contents of the program memory to the library while retaining its previous contents (ie, it no longer requires that the library space be cleared first). Repeated use of the LIBRARY SAVE command allows the contents of the library to be built up in stages which is handy if large fonts are being loaded into the library. LIBRARY DELETE works as before (ie, clears the whole library space).
- Optimised the transfer of data to SPI based LCD display panels - screen updates are now 50% faster. Thanks to *matherp* and *Nathan* on The Back Shed forum for showing how it could be done.
- Extended the commands and functions that use a serial I/O file number (eg, EOF(), LOC(), INPUT\$(), etc). Now a file number of zero (eg, #0) will refer to the console.
- Extended the PORT() function so that it will also return the state of output pins (similar to the way that PIN() works).
- Extended ON ERROR SKIP so that the number of commands to skip error checking can be optionally specified (ie, ON ERROR SKIP nn). The default is one, as before.
- Added the ability for a C Routine to add a custom touch driver for LCD display panels.
- On exiting the editor (with colour coding turned on) the editor will restore the previous colour settings used by the terminal emulator.
- Implemented the TRIANGLE drawing command. For the Micromite Plus it is a built in command, for the standard MX170 Micromite it is a CSub which can be embedded in the BASIC program (the CSub is included in the firmware zip file).
- Implemented the GUI RESET LCDPANEL command. This will reinitialise the LCD panel and is useful if the panel had been previously powered down to conserve power.
- Extended the FOR...NEXT loop to accept a floating point number for the STEP value. Note that this should be used carefully as repeatedly adding a fractional number to the floating point variable used in the FOR loop will accumulate rounding errors.
- New features in the Micromite Plus only:
 - Implemented the PLAY command. This will play stereo WAV files stored on the SD card using the I/O pins PWM 2A and PWM 2B. You can also generate precise sine waves with selectable frequencies from 1 Hz to 20 KHz. See the section Sound Output in the updated *Micromite Plus Manual*.
 - On the 64-pin Micromite Plus moved the PWM2A output from pin 47 to pin 32.

- Implemented transparent text and the BLIT command. Transparent text is invoked by using a background colour of -1 and the BLIT command allows the background image to be saved and then restored allowing moving objects to be displayed over a background image. See the section Basic Drawing Features in the updated *Micromite Plus Manual*.
- Implemented the PIXEL() function for reading the colour of a pixel on an LCD screen capable of supporting transparent text.
- Implemented the GETSCANLINE() function. For LCD panels using the SSD1963 controller this function returns the current scan line that is being refreshed. Using this a BASIC program can time its writes to the LCD panel to avoid flickering.
- Implemented a new USB stack (called M-Stack) ported to the Micromite by Peter Mather (*matherp* on the Back Shed forum). The functionality is the same but the speed of data transfer (particularly XModem) is faster.
- Added the EVAL() function. This will take a string which represents an expression, evaluate it and return the resultant value.
- The GUI LED can now be instructed to flash, ie, turn on for a short time then automatically turn off. This is accomplished by setting the value of the LED (using CtrlVal(#ref) =) to the number of milliseconds that it should remain on. As before, setting its value to one will turn it permanently on and zero will turn it off.
- Added the GUI AREA control. This will define an invisible area of the screen that is sensitive to touch and can be used as the basis for creating custom controls that are assembled and managed by the BASIC program.
- The Number Box and Text Box controls will now call the subroutine MM.KEYPRESS (if it exists) after most key presses. The control's reference number (an integer) and the caption on the key (a string) are supplied as parameters and using these the subroutine can inspect the current contents of the control and change it on the fly using CtrlVal(#ref).
- Added the decimal point and degree symbols to Font #6. The degree symbol is the back quote character (60 hex or 96 decimal).
- All strings used in GUI controls can now display multiple lines by using the tilde character (~) to separate each line in the string. For example, a push button's caption can be "ALARM~TEST" and this would be displayed on the button's face as two lines.
- The separator character between lines in a MsgBox has been changed to the tilde character (~) to match the above feature (previously it was the vertical bar).
- When calibrating a touch screen the accuracy of the calibration will be reported on the console and the LCD display panel.
- The state of the SD Card Card Detect and Write Protect pins can now be read using the PIN() function. The same for the touch IRQ pin.
- The PS2 keyboard will now start up with Num Lock off.
- For radio buttons and check boxes the touch sensitive area has been extended to include their associated caption.
- The OPTION CONTROL nn command can be used to change the number of GUI controls allowed (the default is still 100).
- Improved the speed of loading an image when using LOAD IMAGE. Thanks to *matherp* and *Nathan* on The Back Shed forum.
- Increased the maximum number of subroutines and functions in a program to 200.

The TRIANGLE, PLAY and BLIT commands, the PIXEL() and GETSCANLINE() functions and the transparent text functionality were written by *Nathan* and/or *matherp* on [The Back Shed](#) forum.

Bug Fixes:

- Fixed a bug introduced in Beta 3 which caused an MIPS exception on the MX170 version when reading from a serial port.
- Fixed a bug which caused VAL() to sometimes return a float when it should have returned an integer (thanks to Robert Rozee for the fix).
- Fixed a bug which caused PEEK and POKE to corrupt the variable table if used to access a non existent variable.
- Fixed a bug where, if the x and y axis of touch were reversed (which is allowed), you would get a warning message, even if your calibration was accurate.
- Fixed a bug which could generate an error when opening a COM port if the previous open had failed due to an invalid file number.
- Fixed a bug which caused errors when a user defined function was embedded in an expression passed to another user defined function.
- Corrected a bug which caused the OPTION RESET command to not reset all options.
- Fixed a bug which caused ON ERROR SKIP or IGNORE to clear all variables if an error was ignored in the main program (ie, not in a subroutine).
- The read only variable MM.OneWire is now reset when a program starts running.
- Worked around an issue with the PIC32 that caused the sleep current to increase by ten times at random times during sleep – now the chip will remain in deep sleep with a current of about 40µA (MX170 version). Updated the *Micromite User Manual* (page 15) with hints on how to reduce the sleep current.
- In the long strings CFunction library fixed an issue with SInstr() which occurred if the search string was longer than the long string.
- Bug fixes in the Micromite Plus only:
 - Fixed a bug where a file error would return the value of 16 for MM.ERRNO.
 - Fixed an inconsistency with the SPINBOX control which could be caused when the initial displayed value was not within the max/min range.
 - Corrected a bug where the Number Box and Text Box controls did not correctly set the TOUCH(LASTREF) value when exiting the control.
 - Fixed a bug in the Radio Button and LED controls where the control was not drawn on the centre of the X and Y coordinates and the radius was treated as the diameter. Unfortunately fixing this bug will cause the Radio Button and LED controls in programs written for 5.2 to require adjustment.
 - Corrected a bug which caused the editor's cursor to erase the underscore character on an LCD screen when it is acting as the console.
 - The Number Box and Text Box controls will now correctly erase the background to the current background colour.
 - Fixed an issue where radio buttons inside a frame on one page could be controlled by a frame on another page.
 - When a control is disabled it will now correctly respect the current background colour.

- Restored the OPTION ERROR CONTINUE/ABORT command. This has been largely superseded by the ON ERROR command but can still be used by old programs.

Change log for Micromite MMBasic Ver 5.2

- Implemented the ability to manage program errors. ON ERROR IGNORE will ignore all errors, ON ERROR SKIP will ignore an error in the next statement only and ON ERROR ABORT (the default) will abort on any error. MM.ErrNo will be non zero for an error and MM.ErrMsg\$ will contain the error message. Both can be cleared by ON ERROR CLEAR.
- Micromite Plus:
 - Implemented the MsgBox() function. This will display an ad hoc dialog box with program defined message and buttons. When the user selects one of the buttons any on screen controls will be restored and the function will return the number of the button pressed.
 - Implemented the GUI SETUP and PAGE commands which allows the GUI controls to be organised into pages that can be quickly hidden or displayed. This makes it much easier to write programs which need to switch between various screen displays.
 - The GUI DEFAULT command has been removed as its functionality is now provided by the GUI SETUP command.
 - You can now disable, hide or even delete a control while it is active (ie, from within the touch down interrupt). You can also change its colour and other attributes.
 - CPU speeds of 20MHz, 10MHz and 5MHz can now be selected.
 - Fixed a bug that would intermittently cause an exception or hang when switching the CPU speed from 60MHz or lower to 80MHz or higher.
 - Removed the LOAD FONT command (DEFINEFONT achieves the same outcome).
 - Changed the error message when trying to open SPI2 and it is already in use for a SPI display, touch or the SD card.
- Replaced the back quote (`) character with the degree symbol (°) in all built-in fonts except Font #6.
- Added the functions ACOS() and ASIN().
- CFunction changes (for both the Micromite and Micromite Plus):
 - Improved the CSub/Crunction syntax so that the type of parameters and the returned value can now be specified in the CSub/CFunction definition.
 - Implemented the ability for a CFunction to execute a BASIC subroutine from within the CFunction. This subroutine can contain nested calls to other subroutines and functions. Amongst other things this allows any MMBasic command to be run by a CFunction.
 - In the CallTable[] used by CFunctions added pointers to SaveProgramToFlash, error(), ProgFlash vartbl and varcnt. These will allow CFunctions to read/write program memory, use the standard error routines built into MMBasic and lookup (and even change) variables defined in the BASIC program.
 - Implemented the ability for CFunctions to hook into the Interrupt 2 handler.
- The RTC SETTIME command will accept a full year (eg, 2016) or an abbreviated year (eg, 16) and it will also automatically set the Micromite's internal clock.
- Increased the internal stack space used by MMBasic. This allows much more complex expressions which previously caused an "Expression is too complex" error.

- If an error or conflict due to an OPTION setting occurs at startup a suitable message will be displayed and the offending OPTION setting will be cleared. Previously the Micromite could be stuck in an endless series of reboots.
- Fixed a number of bugs in Str\$() that caused incorrect formatting. This function now always places the sign after any padding characters.
- Changed CPU SLEEP to make the sleep time more accurate and update the Timer correctly.
- Changed the exponentiation operator so that it traps a floating point overflow.
- Corrected a bug which caused PWM channel 1 to not operate correctly on power up.
- Fixed a bug that could cause the editor to lose track of the cursor's position.
- Corrected two bugs that caused a corrupted program when some specific line numbers were used on an otherwise empty line.
- Fixed a bug where closing COM2 did not release the Tx line.
- Fixed a bug which prevented a CFunction from returning a float.
- Fixed a bug which cause the wrong date/time to be returned when the OPTION RTC pins were the same as the normal I²C pins.
- Fixed a bug which caused small negative numbers to print without the negative symbol. This also affected the STR\$() function.
- Fixed a bug that caused strange error messages in complex programs with nested subroutines that in turn used nested FOR loops or DO loops that were often exited with END SUB or END FUNCTION (phew!).
- Saved variables will be automatically cleared when the NEW command is used but not when EDIT, AUTOSAVE and XMODEM are used.
- Corrected an issue that could cause the Timer and other internal clocks to lose time when repeatedly switching CPU speeds.
- Corrected a bug in the BOX and RBOX commands which drew a box with dimensions in error by one pixel.
- Fixed a bug that caused the program saved in the library to be truncated when the LCase\$(), Len() or Mid\$() functions were used in the argument list of a user defined subroutine.
- Modified the SERVO command so that it can accept rapid updates while still generating a valid servo signal.
- Fixed a bug which caused the RTC command to close the I²C channel if it was open before the command was used.
- Updated the IR command in the Micromite User Manual to document the fact that the bits returned in 'dev' and 'key' for the NEC protocol are reversed

Change log for Micromite MMBasic Ver 5.1

- Modified the way that VAR SAVE works:
 - It can now be used repeatedly adding to the list of saved variables. Variables that had been saved previously will be updated with their new value, any new variables (not previously saved) will be added to the saved list for later restoration.
 - Arrays can also be saved. They are specified with empty brackets. eg, var()
 - A new command VAR CLEAR can be used to erase all saved variables.

- The saved variables will be automatically cleared when a new program is loaded or a program is edited using EDIT.
- Implemented the TRACE command. TRACE ON and TRACE OFF work the same as TRON and TROFF (now superseded). Micromite Plus only: TRACE LIST nn will list the trace of the last nn lines executed (note that TRACE ON is not required for this to work).
- Micromite Plus only: A new command OPTION RTC allows the I/O pins for the RTC command to be specified (they may be any pair of I/O pins). If this command is used MMBasic will automatically get the correct time from the RTC on power up or restart.
- Added the ability for CFunctions to use Timer 1 and Timer 5 (on the Micromite Plus).
- If an argument to a CSub or CFunction resolves to a float it will be passed as a float. Previously it was converted to an int.
- MM.VRes and MM.HRes will return zero if a display is not configured (previously it caused an error).
- Text characters can be drawn partially off and on the screen and in that case the part on the screen will be drawn correctly (previously the character was ignored).
- MMBasic will now throw an error if a variable is declared twice using DIM or LOCAL.
- For compatibility with Microsoft BASIC a trailing comma in a print statement will now suppress the automatic generation of carriage return and line feed characters.
- When drawing text on the LCD a space character in font #1 will be drawn as a blank character. Previously it was drawn as a blank rectangle.
- The LOG() function will now throw an error if passed a negative argument.
- Standard (MX170) Micromite:
 - An interrupt previously set on the touch IRQ pin can now be cancelled.
 - The console input buffer has been increased to 128 bytes.
 - Corrected an issue that caused some characters to be lost when a large program was loaded using AUTOSAVE.
- Micromite Plus:
 - An error will now be thrown if an active GUI control is deleted or hidden.
 - Added the 'Can' button (cancel the change) to the number and text boxes. Changed the caption on the delete button from '<<<' to 'Del'.
 - The caption for a frame can now be updated using CTRLVAL(ref) =
 - Swapped the up down arrows on the spin box.
 - Fixed an issue where the spin box could freeze on repeated up/down button presses.
 - Using CLS (clear screen) will also automatically hide any visible GUI controls.
- Fixed a bug where a bracket in a string argument to a function could cause an error.
- Fixed a bug that sometimes caused LIBRARY SAVE to corrupt the library.
- Fixed a bug which caused labels in the library to be hidden.

Change log for Micromite MMBasic Ver 5.0

- Added support for ILI9341 based LCD panels. These are low cost colour LCD displays with 2.2 inch to 2.8 inch diagonal size and 240x320 pixels. From within MMBasic the programmer can change fonts and colours, draw text, lines, boxes, circles, etc.
- ILI9341 based LCD panels also include a touch controller and MMBasic provides full support including touch detection and reporting the touch coordinates in pixels.
- Support for many additional LCD panels can be loaded as embedded C modules. These are distributed in the "Embedded C Modules" folder in the Micromite firmware zip file.
- Additional fonts can be loaded as part of a BASIC program. These are distributed in the "Embedded Fonts" folder in the Micromite firmware zip file.
- The Library feature has been added. This allows BASIC code and CFunctions to be copied into a hidden area where they essentially become extensions of the MMBasic language. Programs saved to the Library area are also compressed saving on program flash space.
- Using the Library feature it is now possible to have a section of BASIC code which is run immediately before the main program is run. This allows constants and hardware features to be set up regardless of the program that is loaded into main memory.
- The functions MM.PROMPT and MM.STARTUP (if they exist) will be run immediately before the prompt is displayed and when the Micromite first starts up. This can be useful for changing the prompt, setting defaults and loading hardware drivers.
- The built in editor's output can be configured to be colour coded. When this is enabled keywords, text, numbers, comments, etc will be displayed in different colours.
- Implemented OPTION LIST and OPTION RESET commands. The first will list all saved options that have been changed from their defaults while the second will reset them all to their original defaults.
- Introduced numerous changes to CFunctions to allow them to load display drivers, hook into some MMBasic internal functions and more. A large range of embedded C modules are available including some that add additional serial, I²C and SPI ports to MMBasic. These are distributed in the "Embedded C Modules" folder in the Micromite firmware zip file. The folder also contains a tutorial and the tools needed to create embedded C modules.
- Added the commands CONTINUE FOR and CONTINUE DO to the BASIC language. These will cause MMBasic to skip to the end of the loop and continue program execution.
- Interrupts can now be set on any I/O pin up to a maximum of ten simultaneous interrupts. Because any pin can be used the INT designation has been removed from the pinout diagrams.
- Weak pullups can be specified on the I²C data and clock lines by appending the option ",PU" to the end of the I2C OPEN command.
- Renamed the DS18B20() function and DS18B20 START command to TEMPR() and TEMPR START. Also, renamed the DHT22 command to HUMID. The old names can be still used but will be removed in some future version.
- Fixed a bug which caused the internal clock to run fast when CPU 5 was used.
- Fixed a bug which caused the PULSE command to throw an error when PULSE pin, 0 was used to terminate a currently running pulse.

Change log for Micromite MMBasic Ver 4.6c

- Removed the ability to read the internal reference using PIN(0). Due to issues in the PIC32MX170 chip this function would occasionally cause the Micromite to restart without warning.
- Changed the reset function which can be invoked at startup (originally shorting the console Tx and Rx pins). The new routine requires a stream of exclamation marks on startup. See the section "Micromite Special Features" in the user manual for details.
- Modified the RESTORE command so that the starting line number for the READ command can be specified.
- Corrected a bug which could cause a processor exception when a leading + or – was used to prefix a constant number.

Change log for Micromite MMBasic Ver 4.6b

- Extended the IR command so that it will work with either NEC or Sony remote controls. The NEC protocol support includes both standard and extended addressing. The IR command syntax is exactly the same as in previous versions, the only change is that it will automatically detect and decode whatever remote control protocol (NEC or Sony) is being used at the time.
- Extended the RTC command to work with Maxim/Dallas chips including the DS1703, DS3231 and DS3232. This is in addition to the Philips PCF8563 which is already supported. The RTC command syntax is exactly the same as in previous versions, the only change is that it will automatically detect and use whatever chip is connected to the I²C pins.
- Added the RTC SETREG and RTC GETREG commands to set and read the registers in a real time clock chip. These commands are not necessary for normal operation but they can be used to manipulate special features of the chip (alarms, output signals, etc). They are also useful for storing repetitive temporary information in the chip's battery backed RAM when the SAVEVAR command is not suitable.
- Changed the OPTION LINES command to OPTION DISPLAY. The new command can set both the number of lines on the display (as did the old command) but also the width of the display in characters. This means that the EDIT command can now be used with the VT100 ASCII Video Terminal (<http://geoffg.net/terminal.html>) in composite video mode.
- The watchdog flag (MM.WATCHDOG) is now cleared by the RUN and NEW commands.
- Fixed a bug which caused the DHT22 command to return an invalid reading at low temperatures.
- Corrected a bug which caused the SPI READ command to return incorrect data when used immediately after a SPI WRITE command.
- Fixed a bug that sometimes caused a comparison between two integers to return an invalid result when the value of either integer was greater than 30 bits (ie, +/- 1073741823).

Change log for Micromite MMBasic Ver 4.6

- **IMPORTANT:** Ver 4.6 (and later versions) will only work with the PIC32MX170F256 or PIC32MX270F256 series of chips (see the Micromite MKII manual for the details). This is different from the previous version which ran on the PIC32MX150 series of chips.

- **IMPORTANT: Only one hex file is supplied.** The Micromite MkII hex file will work with either the 28 or 44-pin versions of the MX170 chip and will automatically configure itself to deal with the different pin outs. This is unlike previous versions that required a different hex file for each chip.
- Compared to version 4.5 on the PIC32MX150 series of chips the free flash memory for program storage is now 60K (up from 20K). The free RAM available for variables and buffers is now 52K (up from 22K).
- The extra flash memory in the Micromite MkII has also been used to optimise MMBasic for speed (rather than size as with previous versions). As a result a typical BASIC program will now run about 40% faster at the same CPU speed.
- An additional data type (64-bit integers) is now available. These can be used to store and manipulate numbers as large as 19 digits or ± 9223372036854775807 without loss of accuracy (as against floating point which has a limit of 6 to 7 significant digits). Integer arithmetic is also about 25% faster than floating point. Integers can be specified by adding the suffix "%" to a variable's name (eg, count%, j%, etc) and by using constants without a decimal point. See the section "Defining and Using Variables" in the User Manual for more details.
- Implemented the full SELECT...CASE structure according to the ANSI Standard for Full BASIC (X3.113-1987) or ISO/IEC 10279:1991 (also used by Microsoft in Visual Basic). This provides for tests like: CASE -11 TO 4, 8 TO 12, 51, 23, IS > MaxNbr
It also allows an unlimited number of CASE statements and unlimited nesting of SELECT...CASE structures within other SELECT...CASE statements.
- Implemented the ability to embed a compiled C or assembler program (called a CFunction) into a BASIC program. This provides the ability to perform particular high speed processing or gain access to the specialised features of the PIC32 chip using an efficient compiled language.
- Implemented the DHT22 command to read the temperature and humidity from a DHT22 temperature/humidity sensor. This device is also known as the RHT03 or AM2302. See the section "Measuring Temperature and Humidity" and the DHT22 command in the User Manual for details.
- Added a new command OPTION EXPLICIT. This will prevent MMBasic from automatically creating a variable when first referenced. Placing this command at the start of a program will require that every variable be explicitly declared using DIM or LOCAL which in turn will draw attention to misspelt variables (a common programming error).
- The default type for a variable which is not explicitly defined can be set with OPTION DEFAULT type where 'type' can be either FLOAT, INTEGER, STRING or NONE. See the "OPTION DEFAULT" entry in the manual.
- Floating point variables can now be specified with the suffix "!". ie, float! = 2.34
- The DIM command has been extended to allow the variable's type to be set without needing a suffix. ie, DIM STRING str1, str2, name, etc. It will also allow the Microsoft convention of: DIM str AS STRING, nbr AS INTEGER, etc
- The DIM command can also initialise a variable when it is created. For example:
DIM STRING str = "Hello World".
- Arrays can also be initialised when declared using the DIM command. The syntax is:
DIM var(d1, d2, etc) = (init1, init2, init3, init 4, etc)
See the DIM command for more details and examples.

- Added the CONST command to define a constant. This will act like a variable but cannot be changed. For example: `CONST TestPin = 5.`
- Implemented OPTION CLOCKTRIM to fine tune the frequency of the internal oscillator which is used as the basis for all timing (date, time, PWM, serial I/O, etc).
- Added the >> and << operators to shift right and shift left the bits in an integer by a certain number of bits. See the section "Operators and Precedence" in the User Manual.
- Added the ability for the CPU SLEEP command to put the processor to sleep for a specified number of seconds (from one second to 10 days) without any external circuitry.
- The SETPIN command has been enhanced with an option to specify the gate time to be used when making frequency measurements. Now the gate time can be any number from 10mS to 100 seconds. See the SETPIN command for more details.
- The SETPIN command and PIN() function have been enhanced to allow the period measurement function (PIN) to return the average period over a number of cycles.
- The SETPIN command now has an option to select an internal pull up or pull down resistor on an I/O pin when configured as a digital input. If the option is not selected the input will float (as before). See the SETPIN command for more details.
- Added the option "OC" to the SETPIN x, DOUT command. This option will set the output to open collector. The old SETPIN x OOUT command will still be recognised for backwards compatibility.
- The PIN() function will now return the value on the output of a pin configured as a standard digital output. This is in addition to its previous ability to read the output of a pin configured as open collector.
- When used to read pin 0 the PIN() function will measure the voltage of the internal voltage reference (nominally 1.2V). Note that pin 0 does not have to be configured via SETPIN. Because the reading will vary with the supply voltage on the Analog Power pin this measurement can be used to infer the supply voltage to the Micromite which in turn can be used to adjust readings made on the standard analog inputs to compensate for voltage variations (eg, when running on battery power). See the section "Analog Inputs" in the manual for a full description.
- The echoing of characters received at the console can be turned off and on. This is useful when the console is used as a third general purpose serial port. See the command OPTION CONSOLE ECHO.
- The polarity of the console transmit and receive lines can be inverted with the OPTION CONSOLE INVERT command. The OPTION CONSOLE AUTO command will specify that the console serial input/output is to be automatically inverted if the console Rx pin is low at power up. These commands allows the console to be used with RS232 signals without a converter and allows the use of a PICAXE style programming cable.
- CFUNCTION, I2C, PEEK and POKE will accept arrays specified by a pair of empty of brackets. eg: `arg()`. For backwards compatibility the old standard of specifying the first element (ie, `arg(0)`) is still supported.
- Extended the POKE command and PEEK() function to use 64-bit integer addresses. Also, the PEEK() function can now return the address of a variable in memory in addition to the contents of that address.
- PEEK and POKE now validate the memory address to reduce the chance of the operation causing a MIPS memory exception.

- You can now pass arrays to subroutines or functions. In the argument supplied and in the parameter list the array is represented by an empty pair of brackets. eg: `arg()`. The array will be passed by reference so any changes to it in the subroutine or function will change the array passed. See the section "Passing Arrays" in the User Manual.
- `END IF` and `ELSE IF` can be used as alternatives to `ENDIF` and `ELSEIF` respectively
- The `TIMER` function will now count up for over 200 million years before rolling over to zero (it is now a 64-bit counter).
- Extended the `HEX$()`, `OCT$()` and `BIN$()` functions to take an optional second argument which specifies the minimum number of characters in the returned string. Numbers with less than this number of digits will be padded with zero as the leading character(s).
- The `SPI` function can now send/receive data in 32 bit blocks (in addition to 8 and 16 bits).
- Added the `SPI WRITE` and `SPI READ` commands for high speed transfer of bulk data over a `SPI` communications channel.
- The `DS18B20()` function will now return a temperature of 1000 (obviously invalid) if the sensor does not respond or shows an error. Previously the program would abort with an error message.
- Using the break key (normally `Ctrl-C`) in the editor will now prompt the user and ask if they want to abandon their edits (ie, will act the same as the escape key).
- Changed `OPTION PIN` to use eight digits and also introduced a three second delay after a PIN has been entered. This makes it much harder for a brute force attack on the PIN to succeed. Also, the PIN can be set to 99999999 which will permanently lock the console and cannot be removed.
- The `MEMORY` command will now only show details of the memory used by saved variables and `CFunctions` if these features have been used. This is to reduce screen clutter.
- Increased the maximum allowable number of sub/functions (combined) from 32 to 100.
- The type of a user defined function can be specified by adding `AS <type>` to the end of the function's definition. For example:

```
FUNCTION FUNNAME(arg1, arg2) AS INTEGER
```
- The type of an argument to a user defined function or subroutine can be specified by adding `AS <type>` to the end of the argument's definition. For example:

```
SUB FUNNAME arg1 AS INTEGER, arg2 AS STRING
```
- To prevent a common type of programming error variables with the same name but different types are no longer supported. ie, `var = 123 : var$ = "abc"` will now cause an error.
- The space allocated to the `VAR SAVE` command is now 2KB.
- Added `CPU RESTART` command to force a restart of the processor.
- Changed the string returned by the automatic variable `MM.DEVICE$` to "Micromite MkII". This is because V4.6 and later will only run on the Micromite MkII (ie, PIC32MX170F256).
- A suitable message will be displayed (rather than the normal power on banner) if the watchdog timer caused a restart.
- When turning a pin off using the `SETPIN` command you must use either the command `SETPIN x, OFF` -or- `SETPIN x, 0`. Previously you could use a variable set to zero but this facility caused difficulty to locate errors when the mode for a pin was accidentally misspelt.
- Before starting `I2C` communications the `I2C OPEN` command will first check for a stuck slave and will try to clear it by pulsing the clock line.

- The I²C commands will now accept integers for send data, receive data and receive data length.
- SAVE VAR without any arguments will erase the saved variable space.
- The NEW command will now also erase any saved variables.
- Fixed an issue with the I²C slave receive buffer not being cleared when the I2C slave was restarted after previous activity.

See: http://www.thebackshed.com/forum/forum_posts.asp?TID=6599&PN=1

and http://www.thebackshed.com/forum/forum_posts.asp?TID=6527&PN=1&TPN=2

Note that CTRL-C does not close a communications channel because CONTINUE might be used to resume running the program. When RUN or NEW is used all communications and I²C activity will be stopped.

- Fixed a bug in the KEYPAD command which prevented the editor from being run after the KEYPAD command reported an invalid pin error.
- Fixed a bug which caused the PWM command to not correctly configure the relevant PWM pins as outputs.
- Fixed two bugs which caused strange results when functions were nested in the command line of a function. Functions can now be used in the arguments provided to a subroutine or function and can be nested within each argument to an unlimited level.
- Fixed a bug which prevented the use of expressions in the arguments to PEEK().
- Updated the manual to describe the new features. The manual now includes a full bookmark listing which can be displayed in most PDF viewers and used to jump to a particular topic.

Change log for Micromite MMBasic Ver 4.5D

- Added the PULSEIN function to measure the width of an input pulse.
- The DS18B20() function now always generates an error if the sensor is disconnected (previously it could return a reading of -127.938 in some cases).
- The XMODEM receive function will now wait for the sender to stop sending data before reporting an out of memory error. This means that the error message will be visible in the terminal emulator window.
- Fixed a bug in the DS18B20() function that caused erroneous negative readings.
- Fixed a bug which caused the processor to reboot when the EOF() function was used.
- Corrected a bug which could cause a strange behaviour of MMBasic when SETPIN was used on pin 11 or greater. The effect was intermittent and rare.
- Fixed an error where the PORT() function returned the input pins in reverse order.
- Updated the manual to describe the PULESIN function.

Change log for Micromite MMBasic Ver 4.5C

- Corrected a bug which allowed the RND() function to return the number 1 (the specifications state that it must return a number from zero to less than 1).
- Fixed a bug that caused an error when the LENGTH keyword was used when defining a string array using the LOCAL command.
- Fixed a bug that kept resetting the variable used in the KEYPAD command to -1.

- Corrected a rare issue caused by some low cost USB-serial adapters that could cause V4.5 of the firmware to partially erase itself when there was a glitch in the power during start up. In that case the only solution was to re program the chip using a PIC32 programmer.

Note: V4.5 was the first release of MMBasic specifically for the Micromite