# Pi-cromite
# User
# Manual

MMBasic Ver 5.4.06

For updates to this manual and more details on MMBasic
go to http://geoffg.net/micromite.html

or http://mmbasic.com

The Pi-cromite is a new addition to the Micromite family using the Raspberry Pi. The Pi-cromite firmware implements most of the features of the standard Micromite and the Micromite Plus as described in the *Micromite User Manual* and the *Micromite Plus Manual*. It has a number of differences and additional features and they are described in this document.

The focus of this manual is to describe just the features that are **unique** to the Pi-cromite. For general Micromite programming you should refer to the *Micromite User Manual* and the *Micromite Plus Manual* in addition to this manual.

# Contents

# Introduction

This section provides an introduction for users who are familiar with the Micromite and the Micromite Plus and need a summary of the extra features in the Pi-cromite.

The Pi-cromite is an extension of the standard Micromite and the Micromite Plus; most of the features of these two versions are also in the Pi-cromite. This includes features of the BASIC language, input/output, communications, etc. Some commands have changed slightly but for the main part Micromite programs will run unchanged on the Pi-cromite.

The following summarises features of the Pi-cromite as compared to the standard Micromite and the Micromite Plus:

### Raspberry Pi

The Pi-cromite is based on the Raspberry Pi. The Raspberry Pi is available a number of versions and has up to fifteen times the program space of the MX series used in the standard Micromite and is many times faster. The Pi3 is 5x faster than even a 252MHz PIC32MZ.

### High Speed Double Precision Floating Point

The Pi-cromite uses the built in hardware floating point capability of the Broadcom processor which is much faster than floating point on the standard Micromite and uses double precision floating point.

### I/O Pins

The Pi-cromite has 26 free I/O pins. The Broadcom chip does not support analogue input.

The Pi-cromite has one $I^2C$ port which can be implemented on any pair of pins, one SPI port, one high speed PWM channel (low speed PWM and servo control can be implemented on any/all I/O pins) and two serial COM ports.

### High Speed LCD Panels

The Pi-cromite supports eleven different sized LCD display panels from 1.44" to 8

### Socket I/O

The Pi-cromite can support opening a socket which allows network based applications to be programmed directly in Basic. The TRANSMIT command will automatically create valid HTTP headers including file lengths making web programming simple

# Micromite Family Summary

The Micromite Family consists of three major types, the standard Micromite, the Micromite Plus and the Pi-cromite. All use the same BASIC interpreter and have the same basic capabilities however they differ in the number of I/O pins, the amount of memory, the displays that they support and their intended use.

| | |
|---|---|
| Standard Micromite | Comes in a 28-pin or 44-pin package and is designed for small embedded controller applications and supports small LCD display panels. The 28-pin version is particularly easy to use as it is easy to solder and can be plugged into a standard 28-pin IC socket. |
| Micromite Plus | This uses a 64-pin and 100-pin TQFP surface mount package and supports a wide range of touch sensitive LCD display panels from 1.44" to 8" in addition to the standard features of the Micromite. It is intended as a sophisticated controller with easy to create on-screen controls such as buttons, switches, etc. |
| Micromite eXtreme | This comes in 64, 100-pin and 144-pin TQFP surface mount packages. The eXtreme version has all the features of the other two Micromites but is faster and has a larger memory capacity plus the ability to drive a VGA monitor for a large screen display. It works as a powerful, self contained computer with its own BASIC interpreter and instant start-up. |
| Pi-cromite | Runs on all versions of the Raspberry Pi with a 40-pin I/O connector. No analogue input capability but 5x faster than a Micromite eXtreme when running on a Pi 3. |

| | **Micromite** | | **Micromite Plus** | | **Micromite eXtreme** | | | **Pi-cromite** |
|---|---|---|---|---|---|---|---|---|
| | 28-pin DIP | 44-pin SMD | 64-pin SMD | 100-pin SMD | 100-pin SMD | 144-pin SMD | 64-pin SMD | Raspberry Pi |
| Maximum CPU Speed | 48 MHz | 48 MHz | 120 MHz | 120 MHz | 252MHz | 252 MHz | 252 MHz | 1200MHz |
| Maximum BASIC Program Size | 59 KB | 59 KB | 100 KB | 100 KB | 540 KB | 540 KB | 540 KB | 1024KB |
| RAM Memory Size | 52 KB | 52 KB | 108 KB | 108 KB | 460 KB | 460 KB | 460 KB | 1024KB |
| Clock Speed (MHz) | 5 to 48 | 5 to 48 | 5 to 120 | 5 to 120 | 200 to 252 | 200 to 252 | 200 to 252 | 700-1200 |
| Total Number of I/O pins | 19 | 33 | 45 | 77 | 75 | 115 | 46 | 26 |
| Number of Analog Inputs | 10 | 13 | 28 | 28 | 40 | 48 | 24 | 0 |
| Number of Serial I/O ports | 2 | 2 | 3 or 4 | 3 or 4 | 3 or 4 | 3 or 4 | 3 or 4 | 2 |
| Number of SPI Channels | 1 | 1 | 2 | 2 | 3 | 3 | 2 | 1 |
| Number of $I^2C$ Channels | 1 | 1 | 1 + RTC | 1 + RTC | 2 + RTC | 2 + RTC | 1 + RTC | 1 |
| Number of 1-Wire I/O pins | 19 | 33 | 45 | 77 | 75 | 115 | 46 | 26 |
| PWM or Servo Channels | 5 | 5 | 5 | 5 | 6 | 6 | 6 | 26 |
| Serial Console | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Native Linux |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| USB Console | | | ✓ | ✓ | ✓ | ✓ | ✓ | Native Linux |
| PS2 Keyboard and LCD Console | | | ✓ | ✓ | ✓ | ✓ | ✓ | Native Linux |
| SD Card Interface | | | ✓ | ✓ | ✓ | ✓ | ✓ | Native Linux |
| Supports ILI9341 LCD Displays | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Supports Ten LCD Panels from 1.44" to 8" (diameter) | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓+ ILI9481 |
| Supports VGA Displays | | | | | ✓ | ✓ | | |
| Sound Output (WAV/tones) | | | ✓ | ✓ | ✓ | ✓ | ✓ | Native Linux |
| Supports PS2 Mouse Input | | | | | ✓ | ✓ | ✓ | Native Linux |
| Floating Point Precision | Single | Single | Single | Single | Double | Double | Double | Double |
| Power Requirements | 3.3V 30 mA | 3.3V 30 mA | 3.3V 80 mA | 3.3V 80 mA | 3.3V 160 mA | 3.3V 160 mA | 3.3V 160 mA | 5V 100mA – 1.6A |

# Suitable Raspberry Pi and Installation Requirements

The Pi-cromite firmware supports any Raspberry Pi with the 40-pin I/O header. The code should work properly on A+, B+, Pi Zero (W), Pi2B, Pi3B but not currently on Model A, B, or B (revision 2).

It is recommended that Raspbian Lite is installed on single CPU versions of the Pi. Full Raspbian can be used on multi-CPU versions.

The code has been developed and tested on the Pi 3 Model B version 1.2 (full Raspbian) and the Pi Zero W V1.1 (Raspbian Lite) using Netbeans IDE V8.2 running on a W10 PC.

In order for the firmware to work pigpio software version 64 must be installed see: http://abyz.co.uk/rpi/pigpio/download.html for details of how to install this version.

Pins may be reserved for Linux use allowing devices like I2S audio DACs to be accessed. See the command ***OPTION PINS*** for more details.

All system devices and Linux commands can be accessed from MMBasic. See the ***SYSTEM*** command for more details.

The 3.3V rail on a Pi 3 is very noisy when the Pi is powered with the official Pi mains adapter. Devices such as IR receivers should be powered from the 5V rail using a 3.3V linear regulator to avoid issues.

There are known timing issues with the Pi 3 that may affect serial communications and Bitstream output – see the ***OPTION WAVETIME*** command for more details.

The Pi-cromite firmware should be copied to a suitable directory on the Pi before use. When first run the firmware creates a hidden file ***.options*** in the same directory. This file should be deleted before any new version is installed.

The firmware must be set as executable before running using ***chmod +x mmbasic***

The firmware requires privileged access to the Pi hardware and so must be run using the sudo command

***sudo ./mmbasic***

**NB** The Pi-cromite firmware sits in a tight loop polling for input and so will use 100% of one CPU. It is running at priority zero most of the time so the operating system is able to time-slice processor access. During time-sensitive I/O operations the priority will be raised and other processes locked out.

# 40-pin Pi-cromite Pinout

| Pin | | | | |
|---|---|---|---|---|
| 1 | 3V3 | | | |
| 2 | 5V | | | |
| 3 | DIGITAL_IN | DIGITAL_OUT | I2C-SDA* | |
| 4 | 5V | | | |
| 5 | DIGITAL_IN | DIGITAL_OUT | I2C-SCL* | |
| 6 | GND | | | |
| 7 | DIGITAL_IN | DIGITAL_OUT | COM2-TX | |
| 8 | DIGITAL_IN | DIGITAL_OUT | COM1-TX | |
| 9 | GND | | | |
| 10 | DIGITAL_IN | DIGITAL_OUT | COM1-RX | |
| 11 | DIGITAL_IN | DIGITAL_OUT | COM2-RX | |
| 12 | DIGITAL_IN | DIGITAL_OUT | COUNT | PWM-HS |
| 13 | DIGITAL_IN | DIGITAL_OUT | SSD1963-RS | ILI9341-CS* |
| 14 | GND | | | |
| 15 | DIGITAL_IN | DIGITAL_OUT | SSD1963-DB3 | |
| 16 | DIGITAL_IN | DIGITAL_OUT | SSD1963-DB4 | |
| 17 | 3V3 | | | |
| 18 | DIGITAL_IN | DIGITAL_OUT | SSD1963-DB5 | |
| 19 | DIGITAL_IN | DIGITAL_OUT | SPI-OUT | |
| 20 | GND | | | |
| 21 | DIGITAL_IN | DIGITAL_OUT | SPI-IN | |
| 22 | DIGITAL_IN | DIGITAL_OUT | SSD1963-DB6 | |
| 23 | DIGITAL_IN | DIGITAL_OUT | SPI-CLK | |
| 24 | DIGITAL_IN | DIGITAL_OUT | COUNT | TOUCH-IRQ* |
| 25 | GND | | | |
| 26 | DIGITAL_IN | DIGITAL_OUT | SSD1963-RS | TOUCH-CS* |
| 27 | ID_SD | | | |
| 28 | ID_SC | | | |
| 29 | DIGITAL_IN | DIGITAL_OUT | COUNT | ILI9341-DC* |
| 30 | GND | | | |
| 31 | DIGITAL_IN | DIGITAL_OUT | SSD1963-RESET | ILI9341-RESET* |
| 32 | DIGITAL_IN | DIGITAL_OUT | SSD1963-WR | |
| 33 | DIGITAL_IN | DIGITAL_OUT | COUNT | IR |
| 34 | GND | | | |
| 35 | DIGITAL_IN | DIGITAL_OUT | SSD1963-DB0 | |
| 36 | DIGITAL_IN | DIGITAL_OUT | SSD1963-RD | |
| 37 | DIGITAL_IN | DIGITAL_OUT | SSD1963-DB7 | |
| 38 | DIGITAL_IN | DIGITAL_OUT | SSD1963-DB1 | |
| 39 | GND | | | |
| 40 | DIGITAL_IN | DIGITAL_OUT | SSD1963-DB2 | |

* Recommended usage for compatibility between PCB designs

# Pi-cromite Features

### Double Precision Floating Point

The Pi-cromite uses the hardware floating point capability of the Broadcom chip and can therefore process floating point calculations faster than the Micromite and Micromite Plus. All floating point uses double precision calculations.

### Twenty Six PWM Channels

All pins can be used for PWM output up to 20KHz as well as for driving servos. In addition pin 12 can be used for PWM output up to 25MHz

### MM.DEVICE$

On the Pi-cromite the read only variable MM.DEVICE$ will return " Pi-cromite running on H/W version: hhhhhh ".

### CPU command

The Pi-cromite does not support dynamically changing the CPU speed or the sleep function. Accordingly the commands CPU speed and CPU SLEEP are not available. However the Pi-cromite does support “CPU SLEEP time” where time is specified in seconds.

### OPTION CONTROLS command

The Pi-cromite does not support the OPTION CONTROLS command instead the maximum number of GUI controls is set to 500.

### Longstring handling

The Pi-cromite supports a comprehensive set of commands and functions for handling long strings stored in integer arrays

### Mouse control in editor

While in the editor the left mouse button can be used to position the cursor. Just click on any character and the edit point will move to just before that character with the cursor on the character. Click just after the last character in a line to go to the end of that line. Click well past the end of a line to go to the start of the next line.
The mouse wheel can be used to scroll up and down the file.

### Delete and Backspace

Teraterm, Putty, and the Raspian desktop console window generate different codes for backspace and delete. The Pi-cromite defaults to support the console window and Putty. Use OPTION TERATERM ON to accept the default codes from Teraterm. Alternatively configure teraterm in the Keyboard setup window

### Auto, Ctrl-C and Ctrl-Z

Use AUTO (and not AUTOSAVE) to enter automatic program entry mode.

Use Ctrl-C to terminate a running program, to exit GUI TEST LCDPANEL, to exit GUIT TEST TOUCH or to exit from auto input mode.

Use Ctrl-Z at the command prompt to exit MMBasic and return to the Linux command prompt

### I2C

Before using I2C the pins to be used must be selected with OPTION I2C SDApinno, SCLpinno

Then you can use I2C exactly the same as the Micromite with the following limitations:
The implementation does not support 10-bit addressing (i.e. options 0 and 1 only)
The implementation is Master only.

### DATE$ and TIME$

Date$ and Time$ are derived from the Linux system clock. They are returned in standard MMBasic format when read but cannot be set by MMBasic (except by using the SYSTEM command with the requisite Linux sysntax)

# Commands (Pi-cromite Only)

| | |
|---|---|
| BITSTREAM pin, nbr, dur%() | Generates a stream of 'nbr' Hi/Lo (OR Lo/Hi) transitions on pin number 'pin'. The duration of each state after the transition is determined by the contents of array 'dur%()' which is specified in micro-seconds. The arrays can be INTEGER or FLOAT but the pulse length must be specified in microseconds.<br>The minimum pulse length and pulse increment is 1uS<br>The pin must previously have been set as a digital output.<br>The direction of the first pulse will be determined by the state of the pin when BITSTREAM is called. If it is high then the first pulse will be at zero, if it is low the first pulse will be at 3.3V.<br>The BITSTREAM command will automatically create an end transition to the last pulse.<br>**NB There is a known clock issue on the Pi 3. See OPTION WAVETIME for details** |
| BOX x1, y1, w, h [, lw] [,c] [,fill] | All parameters can now be expressed as arrays and the software will plot the number of boxes as determined by the dimensions of the smallest array. x1, y1, w, and h must all be arrays or all be single variables /constants otherwise an error will be generated. lw, c, and fill can be either arrays or single variables/constants. See the Micromite User manual for full details of parameter usage. |
| CIRCLE x, y, r [,lw] [, a] [, c] [, fill] | All parameters can now be expressed as arrays and the software will plot the number of boxes as determined by the dimensions of the smallest array. x, y and r must all be arrays or all be single variables /constants otherwise an error will be generated. lw, a, c, and fill can be either arrays or single variables/constants. See the Micromite User manual for full details of parameter usage. |
| GUI STARTLINE n | Sets the row in the graphics memory which will appear at the top of the screen (landscape or reverse landscape) or left of the screen (portrait or reverse portrait) for a 4.3" SSD1963 display initialised with OPTION LCDPANEL SSD1963_4P |
| LINE x1, y1, x2, y2 [, LW [, C]] | All parameters can now be expressed as arrays and the software will plot the number of boxes as determined by the dimensions of the smallest array. x1, y1, x2, and y2 must all be arrays or all be single variables /constants otherwise an error will be generated. lw and c can be either arrays or single variables/constants. See the Micromite User manual for full details of parameter usage. |

| | |
|---|---|
| LONGSTRING APPEND array%(), string$ | Append a normal MMBasic string to a long string variable. array%() is a long string variable while string$ is a normal MMBasic string expression. |
| LONGSTRING CLEAR array%() | Will clear the long string variable array%(). ie, it will be set to an empty string. |
| LONGSTRING COPY dest%(), src%() | Copy one long string to another. dest%() is the destination variable and src%() is the source variable. Whatever was in dest%() will be overwritten. |
| LONGSTRING CONCAT dest%(), src%() | Concatenate one long string to another. dest%() is the destination variable and src%() is the source variable. src%() will the added to the end of dest%() (the destination will not be overwritten). |
| LONGSTRING LCASE array%() | Will convert any uppercase characters in array%() to lowercase. array%() must be long string variable. |
| LONGSTRING LEFT dest%(), src%(), nbr | Will copy the left hand 'nbr' characters from src%() to dest%() overwriting whatever was in dest%(). ie, copy from the beginning of src%(). src%() and dest%() must be long string variables. 'nbr' must be an integer constant or expression. |
| LONGSTRING LOAD array%(), nbr, string$ | Will copy 'nbr' characters from string$ to the long string variable array%() overwriting whatever was in array%(). |
| LONGSTRING MID dest%(), src%(), start, nbr | Will copy 'nbr' characters from src%() to dest%() starting at character position 'start' overwriting whatever was in dest%(). ie, copy from the middle of src%(). 'nbr' is optional and if omitted the characters from 'start' to the end of the string will be copied src%() and dest%() must be long string variables. 'start' and 'nbr' must be an integer constants or expressions. |
| LONGSTRING REPLACE array%() , string$, start | Will substitute characters in the normal MMBasic string string$ into an existing long string array%() starting at position ‘start’ in the long string. |
| LONGSTRING RIGHT dest%(), src%(), nbr | Will copy the right hand 'nbr' characters from src%() to dest%() overwriting whatever was in dest%(). ie, copy from the end of src%(). src%() and dest%() must be long string variables. 'nbr' must be an integer constant or expression. |
| LONGSTRING TRIM array%(), nbr | Will trim ‘nbr’ characters from the left of a long string. array%() must be a long string variables. 'nbr' must be an integer constant or expression. |
| LONGSTRING UCASE array%() | Will convert any lowercase characters in array%() to uppercase. array%() must be long string variable. |
| NANO | Allows you to edit the current program using the Linux nano editor. MMBasic will automatically write the current program (if any) to a temporary file and open it in nano. To return to MMBasic save the file to the temporary file and exit nano. The edited file will then be restored into MMBasic memory |

| | |
|---|---|
| OPEN “SOCKET [,interrupt] ,[count]” as #n | Opens a socket for use in a MMBasic program. Once the socket is open then PRINT and INPUT commands and functions can be used exactly like any other file I/O.<br>Print output is buffered until a TRANSMIT command is sent which will create a valid HTTP header<br>The optional parameters specify a MMBasic interrupt routine ‘interrupt’ to be called when ‘count’ characters have been received on the socket.<br>See Appendix B for an example of socket programming. |
| OPTION CLOCK type | This command is used to specify which of the two main Pi clocks are used for MMBasic. The default allows hardware PWM to be used on pin 12 but will cause external commands using the PCM clock, such as output on the i2s port, to fail. Valid values are ‘PCM’ (default) and ‘PWM’. See Appendix D for an example of usage. |
| OPTION I2C SDApin, SCLpin | Specifies which pins are to be used for I2C. Any valid pin may be selected. They can be changed by using OPTION I2C DISABLE and then re-defining them. Must be run once before I2C can be used. The values are permanently stored in the .options file |
| OPTION LCDPANEL SSD1963_4P | Sets the 4.3” SSD1963 display up in 480 x 864 (landscape or reverse landscape) or 864x480 (portrait or reverse portrait) pixel mode. The screen viewport is 480x272 or 272x480 and the position of the viewport is controlled by GUI STARTLINE n. This mode of operation allows display updates to be done on a non-visible part of the graphics memory and then the viewport moved to see the updated image. The 4P display controller is fully compatible with TOUCH, MOUSE, CURSOR and GUI controls |
| OPTION LCDPANEL ILI9841, orientation, DCpin, RESETpin, CSpin | Initialises a TFT display using the ILI9841 controller. This supports 480 * 320 resolution. See the Micromite User manual, ILI9341 section, for full details of parameter usage. |
| OPTION AUTOREFRESH mode | If OPTION AUTOREFRESH ON is set drawing and GUI commands will update the screen immediately. If set to OFF then the screen will only be updated when a REFRESH command is called |
| OPTION AUTORUN mode | If OPTION AUTORUN ON is set the code will try and load and run the file "autorun.bas" on start up. The file must be in the same directory as MMBasic. If the file does not exist you will get a "file not found" error in which case just disable the option using OPTION AUTORUN OFF |
| OPTION PINS pinmask | Specify pins to be excluded from MMBasic e.g. ‘OPTION PINS &B10100’ will exclude pins 3 and 5 from use by MMBasic<br>OPTION LIST will report the ‘pinmask’ if non-zero<br>SETPIN reservedpin,DOUT will report a “reserved on boot” error<br>See Appendix D for an example of usage. |
| OPTION SOCKET sockno | Sets the socket number to use when opening a socket. This defaults to 80 but can be anything between 1 and 65535 |
| OPTION TERATERM mode | Sets the treatment of backspace and delete characters. Default mode is ‘OFF’. Setting mode to ‘ON’ will allow teraterm to be used properly in the editor. Default should suit Putty and the Linux console window |

| | |
|---|---|
| OPTION WAVETIME scale! | Sets a scaling factor to adjust the timing of serial output and bitstream output.. Values of 'scale' > 1.0 will increase the speed of the output. Defaults to 1.0. This parameter may need to be adjusted to correct for a kernel issue on the Pi 3. Timings can be affected by factors like the presence (or not) of a HDMI monitor, the presence (or not) of a USB mouse and/or keyboard, and the use of Raspbian Lite or Full.<br>The best way to determine if you have an issue is to open the serial port with a baudrate of 100 and transmit the character "U" repeatedly. Then measure the pulse length of the output which should be 10msec.<br>The option is only active when MMBasic determines it is running on a Pi 3 |
| PIXEL x, y [,c] | All parameters can now be expressed as arrays and the software will plot the number of boxes as determined by the dimensions of the smallest array. x and y must both be arrays or both be single variables /constants otherwise an error will be generated. c can be either an arrays or single variable/constant. See the Micromite User manual for full details of parameter usage. |
| PWM pin, freq, duty | Sets up a PWM output on 'pin' with frequency 'freq' Hz and duty cycle of 'duty'. There is no limit on the number of pins used in this way. Frequencies and duty cycles are individually set per pin within the following restrictions:<br>SPECIAL CASE PIN 12<br>This has true hardware PWM and the frequency can be set anywhere between 1Hz and 25MHz. Duty cycle is expressed as a percentage (like the Micromite). The underlying clock is 250MHz so with a 25MHz output the duty cycle will move in steps of 25/250 = 10%. At 250Hz the duty cycle will move in steps of 0.0001%<br><br>ALL OTHER PINS<br>These use a PWM based on a 1uS clock and the valid frequencies will be exact divisors of 1,000,000. The maximum frequency is 20KHz where the duty cycle will move in steps of 2%. If a frequency of say 433Hz is input the actual frequency will round to the nearest of:<br>1,2,4,5,8,10,16,20,25,32,40,50,80,100,125,160,200,250,400,500,625,800,1000,1250,2000,2500,4000,5000,10000,20000 |
| REFRESH | Forces a screen update when OPTION AUTOREFRESH is OFF |
| RBOX x1, y1, w, h [, r] [,c] [,fill] | All parameters can now be expressed as arrays and the software will plot the number of boxes as determined by the dimensions of the smallest array. x1, y1, w, and h must all be arrays or all be single variables /constants otherwise an error will be generated. r, c, and fill can be either arrays or single variables/constants. See the Micromite User manual for full details of parameter usage. |
| SERVO pin, period | Sets up a PWM output on 'pin' with a fixed frequency of 50Hz. Valid periods are specified in milli-seconds and are in the 'valid' range 1.0-2.0.<br>There is no limit on the number of pins used in this way. Periods are individually set per pin.<br>Of course you can use the PWM command to generate any non-standard servo output but be aware this can damage a servo. |

| | |
|---|---|
| SENSORFUSION type ax, ay, az, gx, gy, gz, mx, my, mz, pitch, roll, yaw [,p1] [,p2] | Calculates pitch, roll and yaw angles from accelerometer and magnetometer inputs. Valid fusion types are MAHONY and MADGWICK. Usage is described in Appendix A |
| SYSTEM string$ [,array%()] | Executes the Linux operating system command in 'string$'. If the optional parameter is specified, the output from the system command will be directed to the long string 'array%()' otherwise output will appear on the console (stdout). Output can also be directed to a file using standard Linux notation. See Appendix C for examples of usage. |
| TRANSMIT CODE nnn | Constructs and sends a numerical response to the open socket. Typical use would be "TRANSMIT CODE 404" to indicate page not found. |
| TRANSMIT FILE filename, content-type | This constructs an HTTP 1.1 header with the 'content-type' specified and the length of the file, sends it and then sends the contents of the file to the open socket |
| TRANSMIT PAGE filename [,content-type] | This constructs an HTTP 1.1 header with the 'content-type' specified and the length of the file, sends it and then sends the contents of the file to the open socket.<br>The content-type will default to text/html if not specified.<br>MMBasic will substitute current values for any MMBasic variables defined in the file inside curly brackets e.g. {myvar%}<br>Variables can be simple or array elements.<br>An opening curly bracket can be included in the output by using {{.<br><br>See Appendix C for examples of usage. |
| TRIANGLE X1, Y1, X2, Y2, X3, Y3 [, C [, FILL]] | All parameters can now be expressed as arrays and the software will plot the number of boxes as determined by the dimensions of the smallest array. x1, y1, x2, y2, x3,and y3 must all be arrays or all be single variables /constants otherwise an error will be generated c and fill can be either arrays or single variables/constants. See the Micromite Plus manual for full details of parameter usage. |

# Functions (Pi-cromite Only)

| | |
|---|---|
| MM.DEVICE$ | Returns "Pi-cromite running on H/W version:  hhhhhh" where hhhhhh is the  version ID of the Raspberry Pi |

# Appendix A
# Sensor Fusion

The Pi-cromite supports the calculation of pitch, roll and yaw angles from accelerometer and magnetometer inputs.

For information on this technology see https://github.com/kriswiner/MPU-6050/wiki/Affordable-9-DoF-Sensor-Fusion

The SENSORFUSION command supports both the MADGWICK and MAHONY fusion algorithms. The format of the command is:

**SENSORFUSION type ax, ay, az, gx, gy, gz, mx, my, mz, pitch, roll, yaw [,p1] [,p2]**

Type can be MAHONY or MADGWICK

Ax, ay, and az are the accelerations in the three directions and should be specified in units of standard gravitational acceleration.

Gx, gy, and gz are the instantaneous values of rotational speed which should be specified in radians per second.

Mx, my, and mz are the magnetic fields in the three directions and should be specified in nano-Tesla (nT)

Care must be taken to ensure that the x, y and z components are consistent between the three inputs. So , for example, using the MPU-9250 the correct input will be ax, ay,az, gx, gy, gz, **my, mx, -mz** based on the reading from the sensor.

Pitch, roll and yaw should be floating point variables and will contain the outputs from the sensor fusion.

The SENSORFUSION routine will automatically measure the time between consecutive calls and will use this in its internal calculations.

The Madwick algorithm takes an optional parameter p1. This is used as beta in the calculation. It defaults to 0.5 if not specified

The Mahony algorithm takes two optional parameters p1, and p2. These are used as Kp and Ki in the calculation. If not specified these default to 10.0 and 0.0 respectively.

A fully worked example of using the code is given on the BackShed forum at

http://www.thebackshed.com/forum/forum_posts.asp?TID=9321&PN=1&TPN=1

# Appendix B
# Socket Use

This Appendix contains a fully worked example of code to implement a simple interactive web site in MMBasic. The code is for a remote thermostat which requires a security code to be entered before the thermostat setting can be changed. The code buffers incoming HTTP requests in a long string and then uses a generic parsing routine to interpret the request. The HTML file shows how to include MMBasic variables which will have current values substituted into the HTML when the page is transmitted

See http://www.thebackshed.com/forum/forum_posts.asp?TID=9601 for more information and examples of the use of CSS files and embedded pictures

```
option explicit
option default none
Const starttemp = 18
Const maxargs = 32
Const relaypin=7
Const off=0
Const on=1
Dim a%(1000),i%
Dim integer sp(11)
Dim s$,pg$
Dim security$="123456"
Dim string cp(11)
Dim float tcurrent,tnew
Dim float tmax=-1000
Dim float tmin=1000
Dim arg$(1,maxargs-1)
Dim integer checked=0
Const check$="checked='checked'" 'string to set a readio button pressed
Dim string heating="off"
Const hon$="#ff0000" 'red
Const hoff$="#00ff00" 'green
Dim string hcol=hoff$
'
SetPin relaypin,dout
For i%=0 To 10
  sp(i%)=i%+starttemp-1 'set up the temperature values
  cp(i%)="" 'set up the radio buttons as not pressed
Next i%
tcurrent=TEMPR(11) 'get the current temperature
cp(checked)=check$
Open "socket,myint,100" As #2
Do
  TEMPR START 11
  Pause 2000
  tnew=TEMPR(11)
  If Abs(tnew-tcurrent)<10 Then tcurrent=tnew
  updateheater
  If tcurrent>tmax Then tmax=tcurrent
  If tcurrent<tmin Then tmin=tcurrent
Loop
'
Sub myint
  Local p%=0, t%=0
  Local g$
  Do While Not Eof(2)
    LongString append a%(),Input$(10,2)
  Loop
```

```
  p%=LInStr(a%(),"GET",1)
  t%=LInStr(a%(),"HTTP",1)
  If p%<>0 And t%<>0 Then 'full request received
    s$=LGetStr$(a%(),p%,t%-p%+4)
    LongString trim a%(),t%+4
    pg$= parserequest$(s$,i%)
    If i% Then
      If arg$(0,0)="Security" And arg$(1,0)=security$ Then 'valid update
        If arg$(0,1)="R" Or arg$(0,2)="R" Then
          cp(checked)=""
          If arg$(0,1)="R" Then checked=Asc(arg$(1,1))-Asc("A")
          If arg$(0,2)="R" Then checked=Asc(arg$(1,2))-Asc("A")
          cp(checked)=check$
          updateheater
        EndIf
        If arg$(0,1)="Check" Or arg$(0,2)="Check" Then
          tmin=tcurrent
          tmax=tcurrent
        EndIf
      EndIf
    EndIf
'
    If pg$="index.html" Then
      Transmit page "index.html"
    Else
      Transmit code 404
    EndIf
  EndIf
End Sub

'Function to parse an HTML GET request'
' Assumes that the request starts with "GET /"
' and ends with "HTTP"
'
Function parserequest$(req$, paramcount As integer)
  Local a$,b$
  Local integer inpos,startparam,processargs
  For inpos=0 To maxargs-1
    arg$(0,inpos)=""
    arg$(1,inpos)=""
  Next inpos
  paramcount=0
  a$=Mid$(req$,6,Len(req$)-10)
  inpos=Instr(a$,"?")
  If inpos<>0 Then 'parameters found
    processargs=1
    parserequest$=Left$(a$,inpos-1)
    a$=Mid$(a$,inpos+1)
    Do
      arg$(0,paramcount)=""
      arg$(1,paramcount)=""
      inpos=Instr(a$,"=")
      startparam=1
      arg$(0,paramcount)=Mid$(a$,startparam,inpos-startparam)
      startparam=inpos+1
      inpos=Instr(a$,"&")
      If inpos<>0 Then
            arg$(1,paramcount)=Mid$(a$,startparam,inpos-startparam)
            a$=Mid$(a$,inpos+1)
            paramcount=paramcount+1
      Else
        arg$(1,paramcount)=Mid$(a$,startparam)
        paramcount=paramcount+1
        processargs=0
      EndIf
    Loop While processargs
```

```
  Else
    parserequest$=a$
  EndIf
  If a$="" Then
    parserequest$="index"
  EndIf
  If Instr(parserequest$,".html")=0 And Instr(parserequest$,".HTML")=0 Then
  parserequest$=parserequest$+".html"
End Function

Sub updateheater
  Local float hcalc
  If checked=11 Then
    Pin(relaypin)=1
    heating="On"
    hcol=hon$
  EndIf
  If checked=0 Then
    Pin(relaypin)=0
    heating="Off"
    hcol=hoff$
  EndIf
  If checked>=1 And checked<=10 Then
    hcalc=checked+starttemp-1 'setpoint temperature
    If tcurrent>hcalc Then 'turn heating off
      Pin(relaypin)=0
      heating="Off"
      hcol=hoff$
    EndIf
    If tcurrent<hcalc Then 'turn heating on
      Pin(relaypin)=1
      heating="On"
      hcol=hon$
    EndIf
  EndIf
End Sub

<html>
<head>
    <title>Remote Thermostat</title>
</head>
<body>
    <form name='f1' method='get' action='index'>
        <h2 align='left'>Remote Thermostat V4.0</h2>
        Update Code: <input type='text' name='Security' size='6' value='000000'>
        <br>
        <p>
        <TABLE BORDER='1' CELLSPACING='0' CELLPADDING='5'>
            <TR>
                <TD>Heating</TD>
                <TD BGCOLOR='{HCOL}'>{heating}</TD>
            </TR>
        </TABLE>
        </p>
        <p>
        <TABLE BORDER='1' CELLSPACING='0' CELLPADDING='5'>
            <TR>
                <TD></TD>
                <TD>Temperature</TD>
            </TR>
            <TR>
                <TD>Current</TD>
                <TD>{TCURRENT}°C</TD>
            </TR>
            <TR>
                <TD>Max</TD>
```

```
            <TD>{TMAX}°C</TD>
          </TR>
          <TR>
            <TD>Min</TD>
            <TD>{TMIN}°C</TD>
          </TR>
        </TABLE>
        </p>
        <input name='Check' type='checkbox' value='Reset' onClick='this.form.submit()'>
        Reset Max/Min<p>
        <TABLE BORDER='1' CELLSPACING='0' CELLPADDING='5'>
          <TR>
            <TD>Thermostat</TD>
            <TD><input name='R' type='radio' {CP(0)} value='A' onClick='this.form.submit()'> Off<br></TD>
            <TD><input name='R' type='radio' {CP(1)} value='B' onClick='this.form.submit()'> {SP(1)}°C<br></
TD>
            <TD><input name='R' type='radio' {CP(2)} value='C' onClick='this.form.submit()'> {SP(2)}°C<br></
TD>
            <TD><input name='R' type='radio' {CP(3)} value='D' onClick='this.form.submit()'> {SP(3)}°C<br></
TD>
            <TD><input name='R' type='radio' {CP(4)} value='E' onClick='this.form.submit()'> {SP(4)}°C<br></
TD>
            <TD><input name='R' type='radio' {CP(5)} value='F' onClick='this.form.submit()'> {SP(5)}°C<br></
TD>
            <TD><input name='R' type='radio' {CP(6)} value='G' onClick='this.form.submit()'> {SP(6)}°C<br></
TD>
            <TD><input name='R' type='radio' {CP(7)} value='H' onClick='this.form.submit()'> {SP(7)}°C<br></
TD>
            <TD><input name='R' type='radio' {CP(8)} value='I' onClick='this.form.submit()'> {SP(8)}°C<br></T
D>
            <TD><input name='R' type='radio' {CP(9)} value='J' onClick='this.form.submit()'> {SP(9)}°C<br></T
D>
            <TD><input name='R' type='radio' {CP(10)} value='K' onClick='this.form.submit()'> {SP(10)}°C<br>
</TD>
            <TD><input name='R' type='radio' {CP(11)} value='L' onClick='this.form.submit()'> On<br></TD>
          </TR>
        </TABLE>
        </p>
    </form>
</body>
</html>
```

# Appendix C
# Code Examples

## Capturing system output to a file

```
System  "ls -al >> myfile"
Open "myfile"   For input As #1
Do
  Line Input #1,a$
  Print a$
Loop While Not Eof(#1)
Close #1
Kill "myfile"
```

NB  The >> will append to the file (and create it if required).
Instead you can use > and it will throw away any existing data then write the file.

## Capturing system output to a long string

```
DIM INTEGER a(1000)
SYSTEM "ls -al",a()
FOR i=1 to LLEN(a())
  print LGETSTR$(a(),i,1);
NEXT i
```

## Accessing a USB/UART from MMBasic

Thanks to TassyJim for the code snippet

```
 ' serial test program
 System "stty -F /dev/ttyUSB0 38400"
 OPEN "/dev/ttyUSB0" FOR random AS #5 ' open the serial port
 PRINT #5, "HELLO!!! Is anyone out there?"
 DO ' preferred way to receive serial data which might not be there!
   k$ = INPUT$(1,#5)
   IF k$="" THEN
     nodata=nodata+1
   ELSE
     result$=result$+k$
     PRINT ASC(k$),result$
     nodata=0
   ENDIF
   PAUSE 20
 LOOP UNTIL k$=CHR$(10)OR nodata=50 ' 50*20ms = 1 second timeout
 DO
   PRINT #5, "U";
 LOOP

 CLOSE #5
```

# Appendix D
# Using an I2S DAC with MMBasic

The pinout for I2S is:

Bit-clock : pin 12
LR-clock : pin 35
i2s-data-in : pin 38
i2s data-out : pin 40

I2S can be enabled with

```
sudo curl -sS https://get.pimoroni.com/phatdac | bash
```

Set the MMBasic to use the  PWM clock with

```
OPTION CLOCK PWM
```

Reserve the I2S pins so that MMBasic can't use them with

```
OPTION PINS &HA400000800
```

Then an audio file can be played over i2s either external to mmbasic (and unaffected by it), or from mmbasic using any appropriate Linux command

```
SYSTEM "aplay wavfile"
```

To play mp3 or flac files over i2s I found mpv worked,  omxplayer does not work

```
sudo apt-get install mpv
```