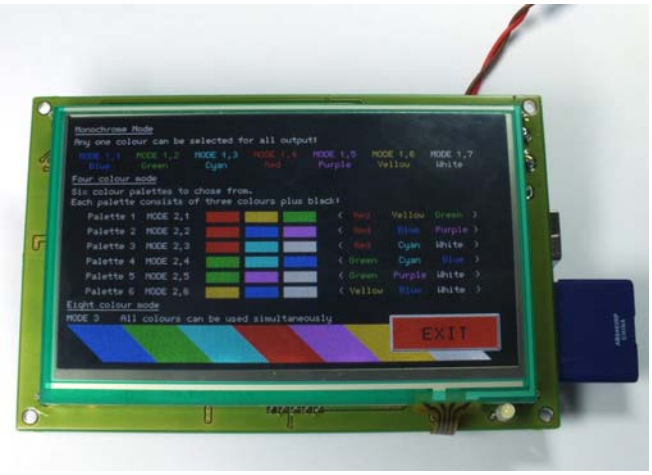
TFT Maximite Manual

Geoff Graham

For updates to this manual and more details on the Maximite
go to http://geoffg.net/tft-maximite.html

For updates, schematics, documentation to the TFT Maximite
go to https://github.com/heise/MAXIMITE

TFT Maximite



The TFT Maximite is a version of the popular Colour Maximite created by Carsten Meyer (cm@ct.de) for c't magazine.

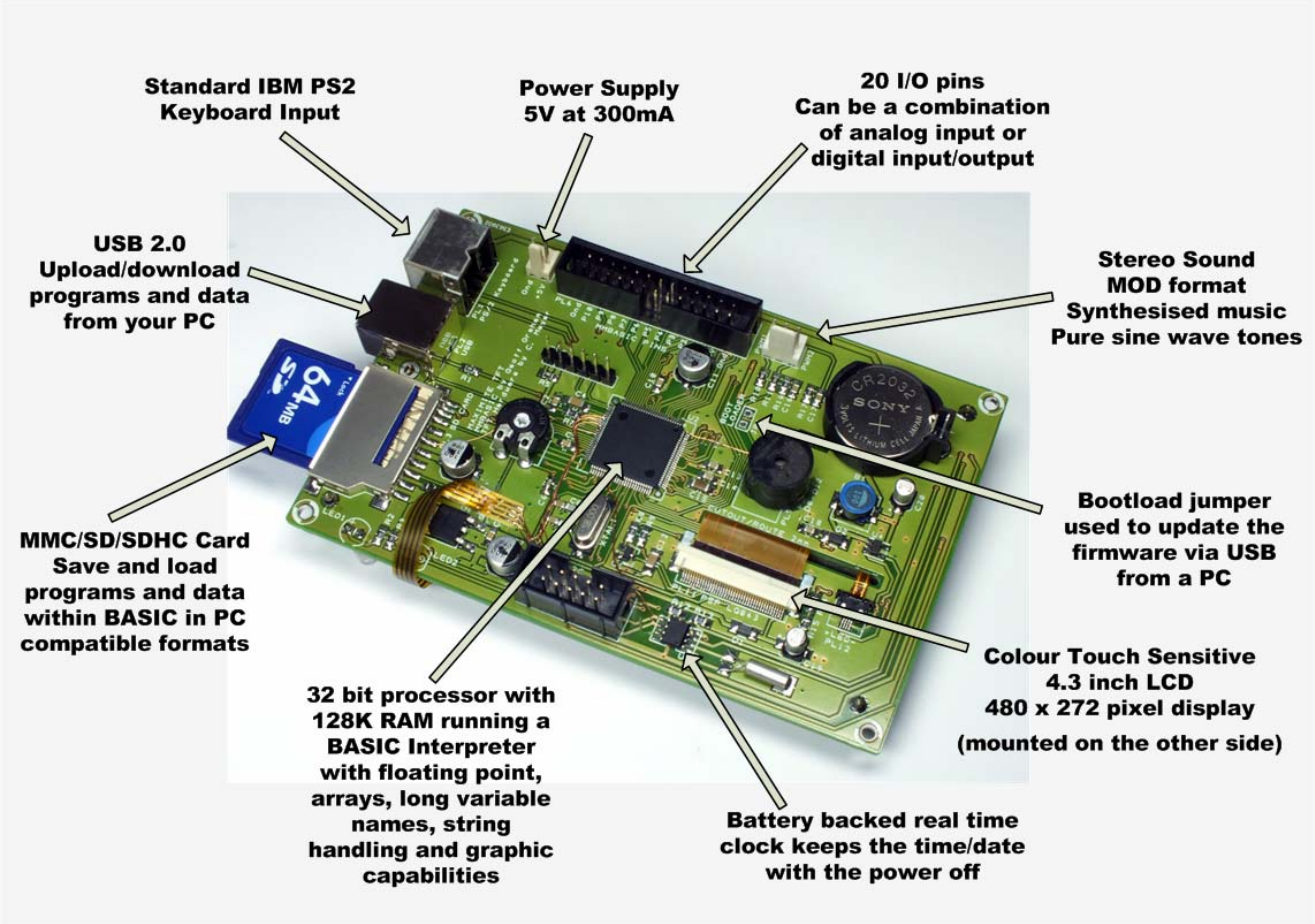
It includes a touch sensitive 4.3 inch LCD panel that provides a sharp 480 x 272 pixel display which is fully supported by MMBasic. This includes the ability to display and respond to touch sensitive controls (buttons, switches, etc) under control of your MMBasic program.

With the exception of the Arduino connector it has all the features of the Colour Maximite including colour, stereo music synthesiser, input from a standard PC compatible keyboard or USB, built in SD memory card, BASIC language and 20 input/output pins.

Refer to the MMBasic Language Manual for details of how to use MMBasic and the general features of the Maximite. This manual can be downloaded from http://geoffg.net/maximite.html.

For new firmware and other updates go to http://geoffg.net/tft-maximite.html

Features and Connectors



LCD Display

4.3 inch, 480 x 272 pixel, touch sensitive TFT LCD panel. MMBasic supports eight colours and 80 characters per line by 22 lines per screen using the standard font

USB

Implements the CDC (Communication Device Class) protocol over USB 2.0. This is a serial interface to the BASIC interpreter so, by using a terminal emulator on the host, programs can be entered, edited and run. Using this interface you can upload programs by streaming the text with a suitable terminal editor.

The Windows driver is available from http://geoffg.net/maximite.html. There is native support for the CDC protocol in Linux (the cdc-acm driver) and Apple OS/X.

Keyboard

Standard IBM compatible PS2 keyboard with mini-DIN connector or a USB/mini-DIN adapter.

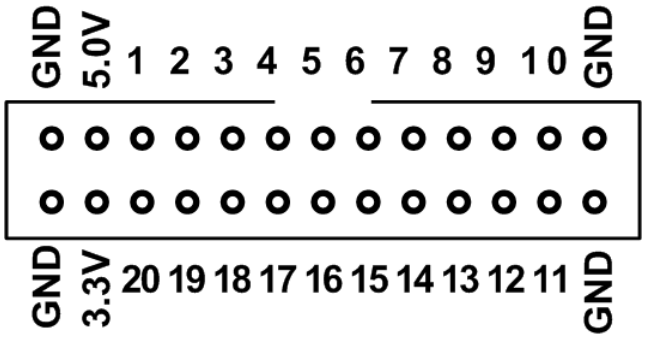
Non ASCII keys (such as the function keys) are mapped to special characters. See Appendix E of the MMBasic Language Manual for the details.

SD/MMC Card Interface

Will accept MMC, SD or SDHC memory cards formatted as FAT16 or FAT32 up to 32GB in capacity. Note that there is no advantage in using a fast SD card as the card is clocked at a fixed 20MHz, regardless of its speed rating.

I/O Connector

I/O pins are addressed in MMBasic as a number. This diagram lists the connections to this connector along with the I/O pin numbers used in MMBasic.



Serial I/O Ports

Serial port COM1: uses pin 15 for receive data (data in) and pin 16 for transmit data (data out). If flow control is specified pin 17 will be used for RTS (receive flow control – it is an output) and pin 18 will be CTS (transmit flow control – it is an input). Serial port COM2: uses pin 19 for receive data (data in) and pin 20 for transmit data (data out).

I^2C, SPI and 1-Wire Ports

I^2C uses pin 12 as the data line (SDA) and pin 13 the clock (SCL). Both lines need external pullup resistors. SPI can use any I/O pins while 1-Wire can use any pin in the range of 11 to 20.

Battery Backed Clock

To set the battery backed clock you use the standard commands in MMBasic for setting the time (TIME$ and DATE$). From then on MMBasic will automatically retrieve the current time and date on power up and display it under the Maximite logo– just to let you know that your battery backed clock is working correctly.

Electrical Characteristics

Power Supply

5V at 270mA typical (plus current draw from the I/O pins)

Digital Inputs

Logic Low:	0 to 0.65V
Logic High:	2.5V to 3.3V (I/O pins 1 to 10) 2.5V to 5.5V (I/O pins 11 to 20)
Input Impedance:	>1MΩ. All digital inputs are Schmitt Trigger buffered.
Frequency Response:	Up to 200KHz (pulse width 10nS or more) on the counting inputs (pins 11 to 14).

Analog Inputs (I/O pins 1 to 10)

Voltage Range:	0 to 3.3V
Accuracy:	Typically ±1%. This accuracy is dependent of the accuracy of the 3.3V supply voltage.
Input Impedance:	>1MΩ (for accurate readings the source impedance should be <10K)

Digital Outputs

Typical current draw or sink ability on any I/O pin:	10mA
Maximum current draw or sink on any I/O pin:	25mA
Maximum current draw or sink for all I/O pins combined:	150mA
Maximum open collector voltage (I/O pins 11 to 20):	5.5V

Audio Output

Audio Frequency Response:	<20Hz to 4KHz
Output Level:	0.5V pp (with components as specified) 0 to 3.3V when operated as a PWM output.

Battery Backed Clock (Colour Maximite)

Time keeping accuracy:	±50 ppm (typical at room temperatures)
Battery Life:	10 to 15 years (limited by the battery shelf life)

Special MMBasic Commands

MMBasic for the TFT Maximite has a number of special commands and functions to support the LCD display and touch interface.

CONFIG FONT 1 **or** **CONFIG FONT 2**	Set the default font to Font #1 which is the standard font of 10 x 5 pixel or Font #2 which is a larger font of 16 x 11 pixels. Which ever font is selected it will become the default on power up and will be reinstated whenever control returns to the input prompt. The power must be cycled for the new setting to take effect.
CONFIG LCD PSP **or** **CONFIG LCD HANN**	Set the video timing for the type of LCD display installed. The default is HANN and if it needs changing you may need to use the USB interface to access the command prompt. The power must be cycled for the new setting to take effect. Obsolete, since PSP Display not used anymore.

TOUCH SIZE sx, sy

or

TOUCH CREATE r, x, y, $caption, colour [, B | S | C | R | H | V| L] [, B | T | L | R | N | D]

or

TOUCH REMOVE r1 [, r2 [, …]]

or

TOUCH REMOVE ALL

or

TOUCH WAIT

or

TOUCH RELEASE

or

TOUCH INTERRUPT target

or

TOUCH BEEP ms

These commands are used to create and manage touch sensitive objects on the display. MMBasic will maintain these objects and automatically change their visible status in response to touch inputs on the screen when the program is running (and not waiting for input at the INPUT or LINE INPUT commands).

SIZE will set the size of any objects subsequently created using the CREATE option. 'sx' and 'sy' are the width and height of the object in pixels.

CREATE will create an object. 'r' is a reference number in the range of zero to 31 which will be used to subsequently refer to the object. 'x' and 'y' are the position of the object in pixels, upper left corner. '$caption' is the text to be applied to the object and 'colour' is the colour to be used.

' B | S | R | H | V | L | C' is a single character that will select the type of object to display:

- B = a momentary button, automatically releases when read
- S = a switch that toggles between OFF and ON
- P = a latching push button that toggles between OFF and ON
- C = a check box that toggles between true (display a tick mark) or false (no mark).
- R = a radio button. There can only be one set of radio buttons and when anyone is ticked any other radio button that is on will be turned off.
- H = a horizontal slider which tracks the touched knob position automatically. Object will return a value between 0 and horizontal size.
- V = a vertical slider which tracks the touched knob position automatically. Item will return a value between 0 and vertical size.
- L = a LED button that toggles between OFF and ON
- If no type is specified (ie, this argument is omitted) MMBasic will create an object with no visible image on the screen. This can be used to setup a sensitive area of the screen that will respond to touch input.

'B | T | L | R | N | D' is one ore more optional character(s) that will set some options for the object.

- B/T set bottom and/or top fill of vertical slider. Valid for sliders only
- L/R set left and/or right fill of horizontal slider. Valid for sliders only
- N no knob, slider will be drawn with a thin line instead of slider knob. Valid for sliders only
- D will disable the touch function, but object is drawn anyway. Object might be set or reset by TOUCHVAL(r) = value by program control. Beneficial for use of touch items as indicators.

REMOVE will remove one or more objects (reference number 'r1', 'r2', etc) from the screen. The shortcut REMOVE ALL will remove all active objects from the screen.

WAIT will pause MMBasic and wait for a touch to occur. Note that MMBasic will not respond to interrupts while waiting.

RELEASE will pause MMBasic and wait for a current touch to be removed from the screen. This will also prevent MMBasic from responding to an interrupt.

INTERRUPT will setup an interrupt which will call 'target' line number, label or user defined subroutine whenever a new touch occurs. Return from an interrupt is via the IRETURN statement except where a user defined subroutine is used (in that case END SUB or EXIT SUB is used). Note that subroutine parameters cannot be used.

To disable this interrupt, use numeric zero for the target, ie:
TOUCH INTERRUPT 0

BEEP will make a tick or beep sound through the TFT Maximite's beeper, length specified in milliseconds.

TOUCHVAL(r) = value	This will manually force the object with the reference number 'r' either off ('value' zero) or on ('value' is non zero). This can be used to set the state of a switch or checkbox immediately after it has been created. 'value' may also set the slider position.
value = TOUCHVAL (r)	This uses **TOUCHVAL** as a function to retrieve the status of a touched object. The function will return the status of the control will that reference number: Zero means off and 1 means selected or on. Retrieved 'value' may also represent the slider position (range 0 to the TOUCH SIZE used for the slider). If a touch object was not initialized with TOUCH CREATE, this function returns 0.
value = TOUCHED (r) **or** **xcoord = TOUCHED (#X)** **or** **ycoord = TOUCHED (#Y)** **or** **refnum = TOUCHED (#I)**	This will return 1 once if item 'r' was touched since last **TOUCHED**() call, otherwise 0. Subsequent calls will return '0' if no touch to referenced item occurred. Using #X or #Y as the argument will return the current x or y coordinate of the current touch point on the screen. If the screen is not being touched -1 will be returned. Using #I as the argument will return the reference number of the last control (i.e. object reference) that was touched.