# Colour LCD Support

The Micromite supports two types of colour LCD display panels. A 2.4" 240x320 pixel TFT LCD with a ILI9341 controller and a 1.8" 128x160 pixel TFT LCD with a ST7735S controller. Both provide a bright colour display at a very reasonable cost (less than US$8 on eBay).

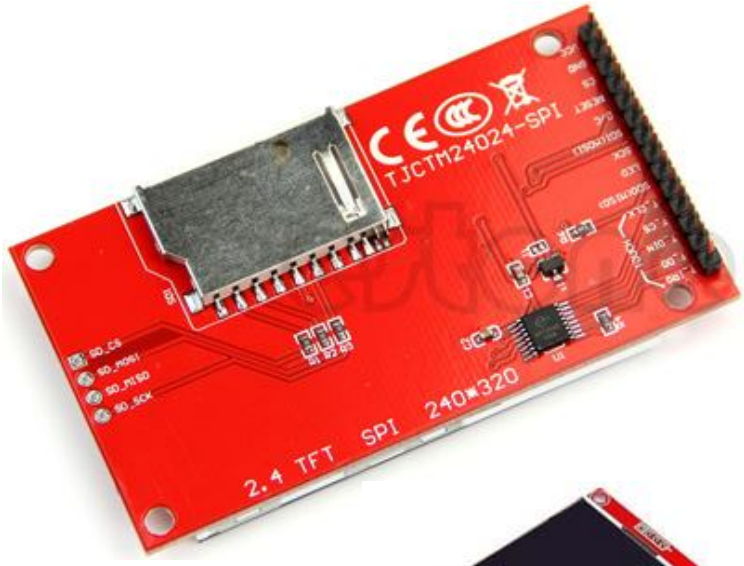
There are many similar displays on the market with subtle differences that could prevent them from working with the Micromite. MMBasic was tested with the displays illustrated below so, if you want to guarantee success make sure your display matches the photographs and the specifications listed below.

On eBay you can find suitable displays by searching for the controller name (ILI9341 or ST7735).

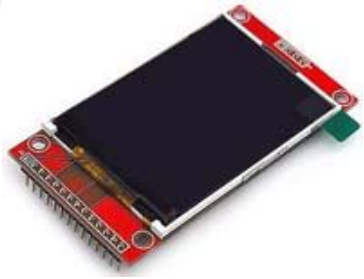
## ILI9341 Based Display

The basic specifications are:

- A 2.4 inch display
- Resolution of 240 x 320 pixels and a colour depth of 262K/65K
- A ILI9341 controller with a SPI serial interface




The display illustrated on the right also has a touch sensitive facility which is fully supported by MMBasic. There are versions of this display without the touch controller (the IC on the bottom right of the PCB) but there is not much point in purchasing these as the price difference is small.



Be aware that some eBay sellers show a photo of the board with the touch controller but further down the listing state that their product does not support touch.

There are also 2.2" and 2.8" versions of this display and the Micromite should work with them but at this time they have not been tested so this cannot be guaranteed.

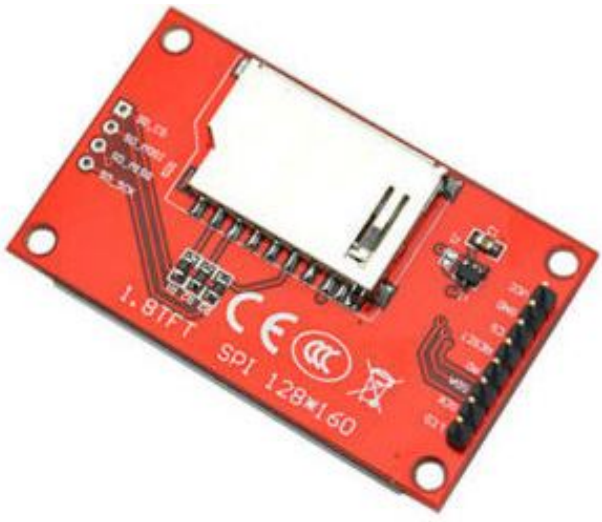
These displays also come with an SD card connector which is not supported by the 28-pin and 44-pin Micromites.

## ST7735 Based Display

These are smaller than the ILI9341 based displays and are handy when just a few lines of text are needed. They have:

- A 1.8 inch display
- Resolution of 128 x 160 pixels and a colour depth of 262K/65K
- A ST7735 controller with a SPI serial interface

These displays do not come with a touch controller.



## Micromite Connections

Both controllers use an SPI interface to communicate with the Micromite. The Micromite has just one SPI interface and therefore the SPI facility will not be available for general use if a colour LCD display is connected. Note that the touch controller and the display controller share the same SPI port on the Micromite.

The following table lists the connections required:

| ILI9341 Display | ST7735 Display | Description | 28-pin Micromite | 44-pin Micromite |
|---|---|---|---|---|
| T_IRQ | | Touch Interrupt | Configurable | |
| T_DO | | Touch Data Out (MISO) | Pin 14 | Pin 41 |
| T_DIN | | Touch Data In (MOSI) | Pin 3 | Pin 20 |
| T_CS | | Touch Chip Select | Configurable | |
| T_CLK | | Touch SPI Clock | Pin 25 | Pin 14 |
| SDO (MISO) | | Display Data Out (MISO) | Pin 14 | Pin 41 |
| LED | LED | Power supply for the backlight (see below) | | |
| SCK | SCK | Display SPI Clock | Pin 25 | Pin 14 |
| SDI (MOSI) | SDA | Display Data In (MOSI) | Pin 3 | Pin 20 |
| D/C | A0 | Display Data/Command Control | Configurable | |
| RESET | RESET | Display Reset (when pulled low) | Configurable | |
| CS | CS | Display Chip Select | Optional - Configurable | |
| GND | GND | Ground | | |
| VCC | VCC | 5V supply (the controller draws less than 10mA) | | |

Where a Micromite connection is listed as "configurable" the specific pin should be specified with the OPTION LCDPANEL or OPTION TOUCH commands (see below).

The backlight power (the LED connection) should be supplied from the main 5V supply via a current limiting resistor. Typical values for this resistor are:

ILI9341 based display: 18Ω for a current of about 63mA.

ST7735 based display: 39Ω for a current of about 30mA.

The value of this resistor can be varied to reduce the power consumption or to provide a brighter display.

## Configuring the LCD Panel

To use the display MMBasic must be configured using the OPTION LCDPANEL command which can be entered at the command prompt.

The syntax is:

```
OPTION LCDPANEL controller, orientation, D/C pin, reset pin [,CS pin]
```

Where:

'controller' can be either ILI9341 or ST7735.

'orientation' can be LANDSCAPE, PORTRAIT, RLANDSCAPE or RPORTRAIT. These can be abbreviated to L, P, RL or RP. The R prefix indicates the reverse or "upside down" orientation.

'C/D pin' and 'reset pin' are the Micromite I/O pins to be used for these functions. Any free pin can be used.

'CS pin' can also be any I/O pin but is optional. If a touch controller is not used this parameter can be left off the command and the CS pin on the LCD display wired permanently to ground. If the touch controller is used this pin must then be specified and connected to a Micromite I/O pin.

This command only needs to be run once as the parameters are stored in non volatile memory. Every time the Micromite is restarted MMBasic will automatically initialise the display ready for use. If the LCD panel is no longer required the command OPTION LCDPANEL DISABLE can be used which will disable the LCD panel feature and return the I/O pins for general use.

To test the display you can enter the command GUI TEST DISPLAY. You should see an animated display of colour circles being rapidly drawn on top of each other. Press any key on the console's keyboard to stop the test.

## Configuring Touch

To use the touch facility MMBasic must be configured using the OPTION TOUCH command. This should be done after the LCD panel has been configured (see above).

The syntax is:

```
OPTION TOUCH T_CS pin, T_IRQ pin
```

Where:

'T_CS pin' and 'T_IRQ pin' are the Micromite I/O pins to be used for these functions (any free pins can be used).

This command only needs to be run once as the parameters are stored in non volatile memory. Every time the Micromite is restarted MMBasic will automatically initialise the touch controller. If the touch facility is no longer required the command OPTION TOUCH DISABLE can be used to disable the touch feature and return the I/O pins for general use.

## Calibrating the Touch Screen

Before the touch facility can be used it must be calibrated. This is done using the GUI CALIBRATE command.


This command will present a target in the top left corner of the screen as illustrated. Using a pointed object such as a stylus press exactly on the intersection of the two lines and hold the stylus down for at least one second. MMBasic will record this location and then continue the calibration by sequentially displaying the target in the other three corners of the screen for touch and calibration.

Following calibration you can test the touch facility using the GUI TEST TOUCH command. This command will blank the screen and wait for a touch. When the screen is touched a white dot will be placed on the display marking the position on the screen. If the calibration was carried out successfully the dot should be displayed exactly under the location of the stylus on the screen.

To exit the test routine you can press any key on the console's keyboard.

## Graphic Functions and Commands

There are two functions and eight commands that you can use within your MMBasic program to interact with the LCD display.

All coordinates and measurements on the screen are done in terms of pixels with the X coordinate being the horizontal position and Y the vertical position. The top left corner of the screen has the coordinates X=0 and Y=0 and the values increase as you move down and to the right of the screen. The read only variables MM.HRES will return the width of the display in pixels and MM.HRES the height.

Colour is specified as a true colour 24 bit number where the top eight bits represent the intensity of the red colour, the middle eight bits the green intensity and the bottom eight bits the blue. The easiest way to generate this number is with the RGB() function which has the form:

```
RGB(red, green, blue)
```

A value of zero for a colour represents black and 255 represents full intensity. The RGB() function also supports a shortcut where you can specify common colours by naming them. For example, RGB(red) or RGB(cyan). The colours that can be named using the shortcut form are white, black, blue, green, cyan, red, magenta, yellow, brown and gray.

MMBasic will automatically translate all colours to the format required by the individual display controller which, in the case of the ILI9341 and ST7735 controllers, is 65K colours in the 565 format.

To detect if and where the screen is touched you can use the TOUCH(X) or TOUCH(Y) function. These will return the X and Y coordinates respectively of the currently touched location or -1 if the screen is not being touched.

Most commands have optional parameters. You can completely leave these optional parameters off the end of the command or you can use two commas in sequence to indicate a missing parameter. For example, the fifth parameter of the LINE command is optional so you can use this format:

```
LINE 0, 0, 100, 100, , rgb(red)
```

Optional parameters are indicated in this document by square brackets, for example: `[, font]`.

The defaults for many of these parameters can be set with the COLOUR and FONT commands. These commands are handy if your program uses a consistent font and colour scheme, you can then set the defaults and use the short version of the drawing commands throughout your program.

The COLOUR command takes the format: `COLOUR foreground-colour, background-colour`

The FONT command takes the format: `FONT font-number, scaling`

MMBasic for the 28-pin and 44-pin Micromites includes two fonts. Font 1 is eight pixels wide and 13 pixels high and contains the full set of upper and lower case letters and numbers. Font 2 is 32 pixels wide and 50 pixels high and consists of just the numbers 0 to 9 plus the colon (:) character.

The scaling parameter can range from 1 to 15 and will multiply the size of the pixels thereby making the displayed character correspondingly wider and higher. Eg, a scale of 2 will double the height and width.

In the following commands C is the drawing colour and defaults to the current foreground colour. FILL is the fill colour which defaults to -1 which indicates that no fill is to be used.

The drawing commands are:

- CLS [C]
  Clears the screen using the colour C. This defaults to the background colour set by the COLOUR command.

- PIXEL X, Y [,C]
  Illuminates a pixel.

- LINE X1, Y1, X2, Y2 [, LW] [,C]
  Draws a line starting at X1 and Y1 and ending at X2 and Y2.
  LW is the line's width and is only valid for horizontal or vertical lines. It defaults to 1 if not specified or if the line is a diagonal.

- BOX X1, Y1, W, H [,LW] [, C] [, FILL]
  Draws a box starting at X1 and Y1 which is W pixels wide and H pixels high.
  LW is the width of the sides of the box and can be zero. It defaults to 1.

- RBOX X1, Y1, W, H [,R] [, C] [, FILL]
  Draws a box with rounded corners starting at X1 and Y1 which is W pixels wide and H pixels high.
  R is the radius of the corners of the box. It defaults to 10.

- CIRCLE X, Y, R [,LW] [, A] [, C] [, FILL]
  Draws a circle with X and Y as the centre and a radius R. LW is the width of the line used for the circumference and can be zero (defaults to 1). A is the aspect ratio which is a floating point number and defaults to 1. For example, an aspect of 0.5 will draw an oval where the width is half the height.

- TEXT X, Y, STRING [,JUSTIFICATION] [, FONT] [, SCALE] [, C] [, BC]
  Displays a string starting at X and Y. JUSTIFICATION is one or two letters where the first letter is the horizontal justification around X and can be L, C or R for LEFT, CENTER, RIGHT and the second letter is the vertical placement around Y and can be T, M or B for TOP, MIDDLE, BOTTOM. The default justification is left/top. FONT and SCALE are optional and default to that set by the FONT command. C is the drawing colour and BC is the background colour. They are optional and default to that set by the COLOUR command.

- BITMAP X, Y, BITS [, WIDTH] [, HEIGHT] [, SCALE] [, C] [, BC]
  Displays the bits in a bitmap starting at X and Y. HEIGHT and WIDTH are the dimensions of the bitmap as displayed on the LCD panel and default to 8x8. SCALE, C and BC are the same as for the TEXT command. The bitmap can be an integer or a string variable or constant and is drawn using the first byte as the first bits of the top line (bit 7 first, then bit 6, etc) followed by the next byte, etc. When the top line has been filled the next line of the displayed bitmap will start with the next bit in the integer or string.

## Example

As an example the following program will draw a simple digital clock on an ILI9341 based LCD display. The program will terminate and return to the command prompt if the display screen is touched.

First the display and touch options must be configured by entering commands similar to these at the command prompt:

```
OPTION LCDPANEL ILI9341, L, 25, 26, 27
OPTION TOUCH 30, 32
```

These specify an ILI9341 based display in the landscape orientation and uses pins 25 to 27 for the LCD and pins 30 and 32 for the touch controller (all on the 44-pin Micromite). Your configuration will probably be different.

Next the touch feature should be calibrated by entering this command and following the calibration procedure.

```
GUI CALIBRATE
```

Finally you can enter and run the program:

```
OPTION EXPLICIT
CONST DBlue = RGB(0, 0, 128)            ' A dark blue colour

COLOUR RGB(GREEN), RGB(BLACK)
FONT 1, 3

BOX 0, 0, MM.HRes-1, MM.VRes/2, 3, RGB(RED), DBlue

DO
  TEXT MM.HRes/2, MM.VRes/4, TIME$, CM, 2, 1, RGB(CYAN), DBlue
  TEXT MM.HRes/2, MM.VRes*3/4, DATE$, CM
  IF TOUCH(X) <> -1 THEN END
LOOP
```

The program starts by defining a constant with a value corresponding to a dark blue colour and then sets the defaults for the colours and the font. It then draws a box with red walls and a dark blue interior.

Following this the program enters a continuous loop where it performs three functions:

1. Displays the current time inside the previously drawn box. The string is drawn centred both horizontally and vertically in the middle of the box. Note that the TEXT command overrides both the default font and colours to set its own parameters.
2. Draws the date centred in the lower half of the screen. In this case the TEXT command uses the default font and colours previously set.
3. Checks for a touch on the screen. This is indicated when the TOUCH(X) function returns something other than -1. In that case the program will terminate.

The screen display should look like this: