Micromite BackPack V2

with Touchscreen LCD and onboard programmer

By Geoff Graham

The Micromite LCD BackPack described in the May 2017 issue has been a very popular project. This revised version incorporates the Microbridge described in this issue. It adds a USB interface and the ability to program/reprogram the PIC32 chip while it's onboard. And the BackPack V2 also adds software control over the LCD backlight.

The Micromite *LCD BackPack* has been a huge hit since it was introduced in May last year. For those who missed it, the *BackPack* combines the Micromite, which is a low-cost, high-performance microcontroller programmed in BASIC, with an equally low-cost LCD touchscreen.

Together, the pair make a potent combination, allowing you to easily design a gadget with an advanced user interface. We have published quite a few examples of this, for example, the *DDS Signal Generator* in last month's issue of *EPE* – see April 2018.

While the original *Micromite LCD BackPack* was easy to build, it did require you to use an external USB/serial converter so that you could load and run programs.

You also needed a PIC32 programmer to load and update the MMBasic firmware in the Micromite, and many people felt that the cost of a genuine PICkit 3 programmer from Microchip was too expensive.

This new design includes both the USB/serial interface and PIC32 programming capability in a single additional chip, dubbed the *Microbridge* – see the separate article describing its operation in this issue.

Because the *Microbridge* is so cheap, it has been designed to be a permanent part of the *Micromite BackPack V2*. So now you can update the firmware in the Micromite and edit your BASIC program without any extra hardware.

We have also included the ability to control the LCD backlight brightness from within the BASIC program running on the Micromite.

This requires just four additional components plus the use of an extra I/O pin on the Micromite. These components are optional; you can either include them or use the original brightness control arrangement with a trimpot (keeping the PWM pin free for other uses).

Apart from the above additions, this new version of the *Micromite LCD BackPack* is exactly the same as the original. It is programmed in the same way, the I/O pins are the same and it will happily run programs written for the original version. It's the same basic formula but easier to use.

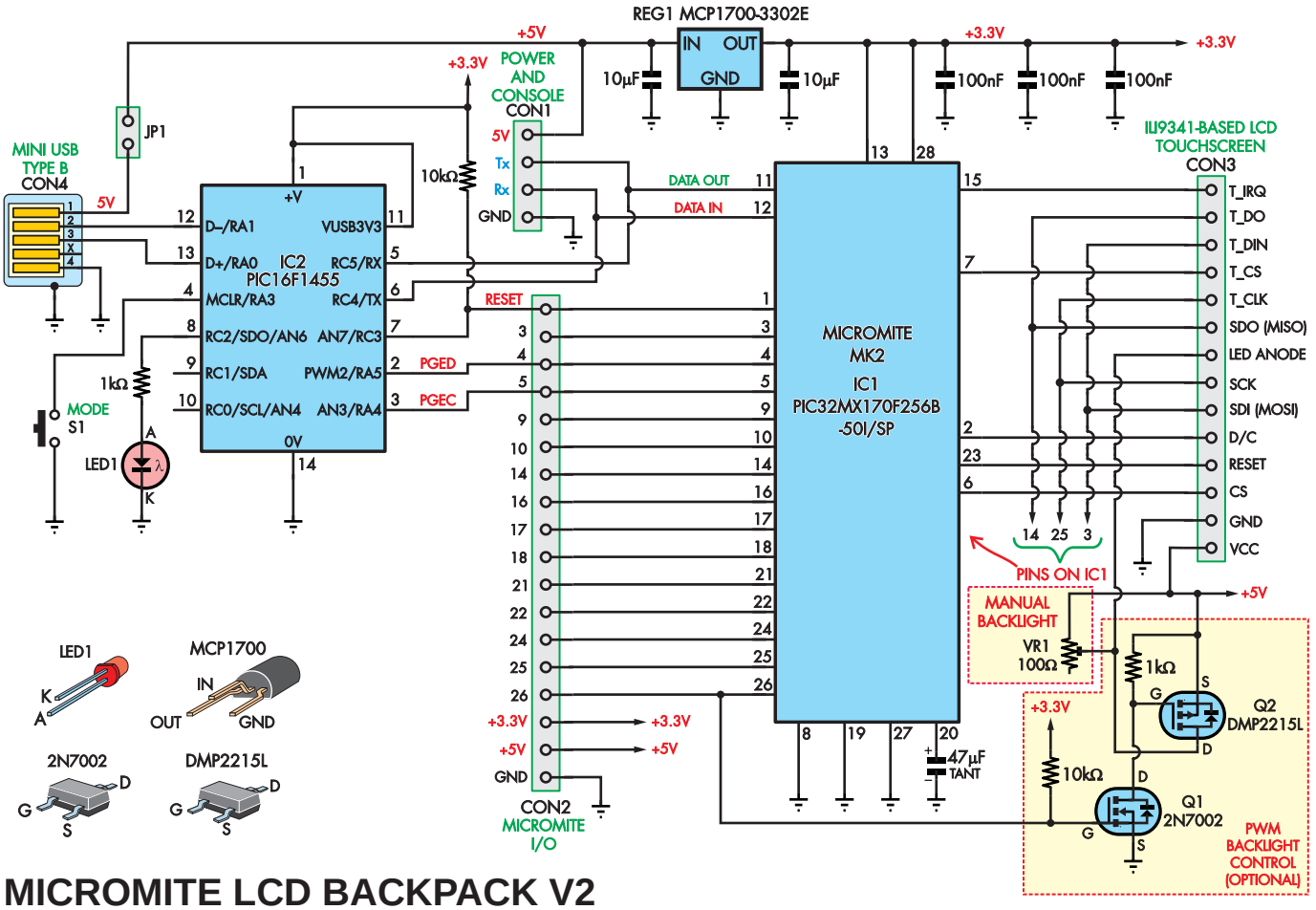
Circuit details

Fig.1 shows the complete circuit for the revised *Micromite LCD BackPack*, incorporating the *Microbridge*. IC2 is a Microchip PIC16F1455 microcontroller, which is both a USB/serial converter and a PIC32 programmer – the standalone *Microbridge* article (see page 28) describes its function in more detail.

When running as a USB/serial converter, pin 5 on the PIC16F1455 receives data (ie, data from the Micromite to the PC USB interface) and pin 6 transmits data (from the PC USB interface to the Micromite). These signals also run to the edge pins for the console connection (CON1) in case you build this PCB but for some reason do not plug the *Microbridge* IC (IC2) into its socket. In this case, you can use an external USB/serial converter.

The PIC32 programming interface from the Microbridge is on pins 7, 2 and 3 of IC2. These provide the reset function, program data and clock signals respectively. These connect to pins 1, 4 and 5 on the Micromite (IC1). The programming output on the *Microbridge* is only active when it is in programming mode, so the *Microbridge* does not interfere with the Micromite when it is using pins 4 and 5 as general purpose I/O pins.

As described in the *Microbridge* article, switch S1 is used to select programming mode and LED1 indicates the mode (lit solid when in programming mode).



MICROMITE LCD BACKPACK V2

Fig.1: complete circuit of the *BackPack V2*, incorporating the *Microbridge* (IC2) which acts as both a USB/serial converter and PIC32 programmer. Micromite chip IC1 runs the show, while REG1 supplies both ICs with a regulated 3.3V. IC1 has an internal 'core' regulator to provide itself with 1.8V which is filtered by the external 47µF tantalum or ceramic capacitor.

CON2 is the main I/O connector for the Micromite and is designed so that it can plug into a solderless breadboard for prototyping. The connector also makes it easy to add a third PCB to the *LCD BackPack* 'stack', which can carry circuitry specific to your application (such as amplifiers or relay drivers). This connector is wired identically to the original *BackPack*.

The Micromite communicates with the LCD panel using an SPI interface where pins 3 and 14 (on the Micromite) carry data to/from the LCD, while pin 25 provides the clock signal. When the Micromite pulls pin 6 low, it is communicating with the LCD panel, and when pin 7 is pulled low, the Micromite will be communicating with the touch controller on the display panel.

The 28-pin Micromite has only one SPI port and so pins 3, 14 and 25 (SPI data and clock) are also made available on CON2 so that you can also use this SPI serial channel to communicate with external devices.

Backlight control

For controlling the brightness of the LCD's backlight you have two choices. The first is to fit MOSFETs Q1 and Q2 to the PCB, along with their associated resistors (this area is marked with a box on the PCB). When you do this, PWM output 2A on the Micromite is used to control the backlight brightness from within your program. This is described in more detail later.

Alternatively, as with the original *BackPack* you can fit VR1, which is a 100Ω trimpot. This is in series with the power to the backlight LEDs so it limits the current drawn by them and therefore sets the brightness. Note that you should install one set of components or the other (not both).

In both cases, the LCD panel has a 3.9Ω resistor in series with the backlight so you will not burn out the backlight if you set the PWM output to 100% or wind VR1 all the way around to zero ohms.

The power supply is derived from either the 5V connector pin on CON1, or if JP1 is installed, from USB connector CON4. Powering the *Micromite LCD BackPack* from USB power is handy during program development, but for an embedded controller application, you would normally remove the jumper from JP1 and supply 5V power via CON1. Note that you should not try to power the *BackPack* from both CON1 and USB as you could cause damage to the USB interface on your computer.

The 3.3V power supply for both the Micromite and the *Microbridge* is provided by REG1, which is a fixed output regulator with a low dropout voltage suitable for use with USB power supplies. This supply is also made available on CON2 so you can use it for powering external circuits (to a maximum of 150mA).

Sourcing the LCD panel

The ILI9341-based LCD panel used in the *Micromite LCD BackPack* comes in three sizes: 2.2-inch, 2.4-inch or 2.8-inch diagonal.

The PCB for the *Micromite LCD BackPack V2* is designed to suit the mounting holes for the 2.8-inch version; however, compatible displays of any of these three sizes will plug into the PCB and will work perfectly. So your only issue with using a 2.2-inch or 2.4-inch display will be that you will need to use some other physical mounting arrangement.

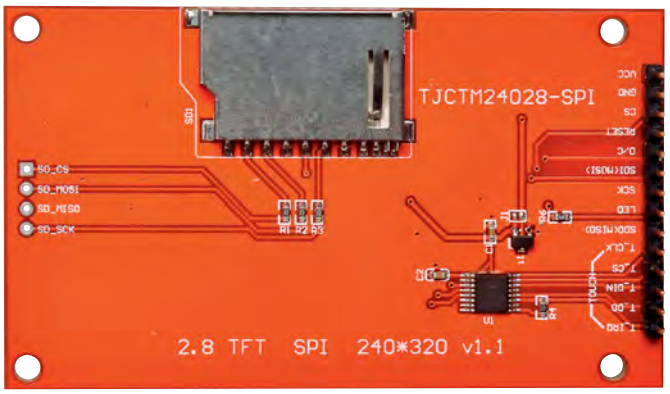
These displays also have an SD card socket, but it's not Micromite supported due to memory limitations.

The best place to find a suitable display is AliExpress or eBay, but

Firmware updates

For firmware updates and manual please check the author's website at: **geoffg.net/micromite.html**

You should also check out the Back Shed forum at: **www.theback-shed.com/forum/Microcontrollers** where there are many Micromite enthusiasts who are happy to help beginners.



The underside of the 2.8-inch ILI9341-based LCD panel we used in the *Micromite BackPack V2*. On the other side of the PCB to the top right of the LCD screen are the letters 2812C-SZ, which may prove useful when searching for this module.

other online markets also have them as well as some online retailers. There are many variations on offer, so make sure the display you purchase matches the photographs in this article. This is important; the Micromite has been extensively tested with the photographed display so you can be sure that it will work. Also ensure the touch controller is installed.

Other features to look out for in a compatible display are an orange PCB, a resolution of 320 × 240 pixels and an SPI interface. Often, the description will emphasise that the display is for use with the Arduino, but that is not relevant; it will work just as well with the Micromite. On eBay, the best way to find a suitable display is to search for the phrase 'ILI9341 LCD'. You should find many displays from US$7.00 upwards.

If you don't want to deal with any of that, then we recommend you purchase a kit from **micromite.org** – our preferred supplier for all things Micromite. The kit includes the LCD touchscreen, PCB, programmed microcontrollers and all the other bits you need to build the *BackPack V2* (apart from the acrylic lid)

Construction

Refer to the PCB overlay diagram, Fig.2. As usual, start construction with the low profile components such as resistors and work your way up to the bigger items such as the connectors.

Begin with the USB socket as this is the only required SMD component. Match the two small plastic pegs on the connector with the corresponding holes on the PCB and then solder the connector's mounting lugs using plenty of solder for strength.

Finally, using a fine point soldering iron tip, solder the signal pins. Examine the pin solder joints carefully under good light with magnification and clean up any bridges with solder wick and a little flux paste.

If you are installing the backlight PWM control components, you should mount Q1 and Q2 next as they are also surface mount types. They are not hard to solder as their pin spacing is quite wide. Don't get them mixed up as they look almost identical.

We recommend using a socket for both IC1 and IC2 as that will enable you to swap out the chips if you suspect that you have damaged one or both. The 14-pin female connector used for CON3 (the LCD panel) is difficult to source so unless you've purchased a kit, the best approach is to cut down a longer header to size and then use a file to smooth the rough edge so that it looks presentable.

The 10µF and 47µF tantalum capacitors are polarised (the longer lead is positive) so make sure that they are oriented according to the silk screen on the PCB. The 47µF capacitor is particularly critical and must be a tantalum or ceramic type, not electrolytic. Rather than using tantalum capacitors, we prefer to use SMD ceramic types with an X5R dielectric. In this case, you can use 10µF 6.3V capacitors in all three locations. They tend to be more reliable than tantalums, but are not as easy to obtain.

When soldering the pin headers for CON1 (power) and CON2 (input/output), remember that the headers should be mounted on the underside of the board, as illustrated in the photos. Don't mistakenly mount them on the top of the board because they will then be impossible to reach when an LCD panel is attached.

Before you plug the microcontrollers into their sockets, it is prudent to apply power and check that 3.3V is on the correct pins of IC1 and IC2, and 5V is on the correct pin of CON3. With that check made, remove power and plug in both microcontrollers and the LCD panel.

If you have a blank PIC16F1455 microcontroller, it should be programmed with the latest *Microbridge* firmware (**2410417A.HEX**), which can be downloaded from the *EPE* website. This can also be done using another Micromite and a 9V battery; see the *Microbridge* article for details on how to do this.

The *BackPack* PCB and the LCD panel can then be fastened together on all four corners with 12mm tapped spacers and M3 machine screws. Be careful when handling the LCD panel. The ILI9341 controller is sensitive to static electricity and can be easily destroyed with careless handling. Make sure that you are grounded when handling the display and avoid touching the connecting pins.

Programming the PIC32

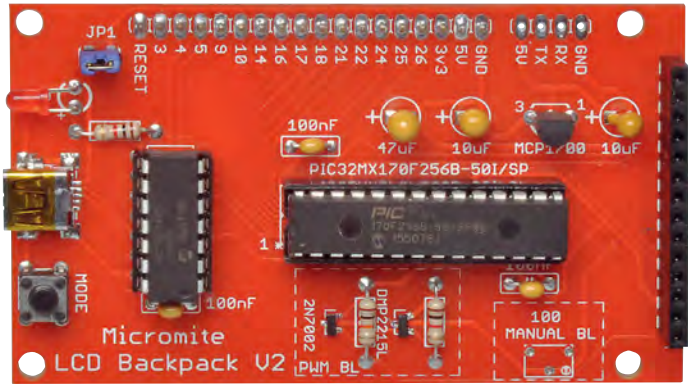
If you have a blank PIC32 chip, this needs to be programmed with the Micromite firmware via the *Microbridge*. This procedure is covered in detail in the *Microbridge* article so we will only provide an abbreviated description here.

The first step is to get the *Microbridge* working as a USB/serial bridge. This involves installing the correct drivers (available from **www.microchip.com/wwwproducts/en/MCP2200**) and launching a terminal emulator and connecting to the COM port created by the *Microbridge*. You can verify that everything is working correctly by typing characters into the terminal emulator and checking that LED1 on the *BackPack* flashes with each keystroke.

Now close the terminal emulator. This is important, as the programming operation will fail if it is still open. You need a Windows computer for the next step. Run the program **pic32prog** (available for download from the author's website) in a command prompt box with the command line:

```
pic32prog -d ascii:comxx yyyy.hex
```

Where xx is the COM port number created by Windows for the *Microbridge* and **yyyy.hex** is the file containing the latest Micromite firmware. For example, if your *Microbridge* was allocated the virtual serial port of COM6 and the file that you wanted to program was **Micromite_5.04.08.hex**, the command line that you should use would be:



The *Micromite LCD BackPack V2* includes the *Microbridge* (the 14-pin chip at left) which incorporates a USB/serial converter and a PIC32 programmer. You can also control the LCD backlight brightness via the BASIC program running on the Micromite. This uses four components that can be seen below IC1. Note, this is an early prototype and the final PCB differs slightly (it includes an extra 10kΩ resistor above IC2).

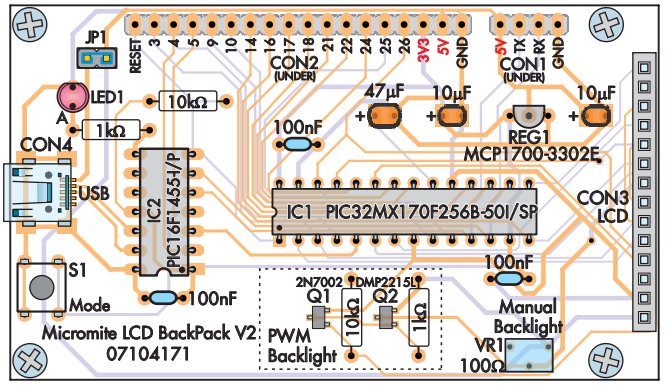


Fig.2: follow this overlay diagram to build the *Micromite LCD BackPack V2*. CON4 is the only required SMD component; SMD ceramic capacitors can optionally be used in place of the tantalum types for better reliability. If fitting Q1 and Q2, be sure to also install the two associated resistors and leave VR1 out. Note that CON1 and CON2 are fitted to the underside of the board.

```
pic32prog -d ascii:com6
  Micromite_5.04.08.hex
```

When you press Enter, **pic32prog** will automatically run through the programming sequence and then return to USB/serial mode. You can then launch your terminal emulator and when you press return you should see the Micromite command prompt (a greater than symbol '>').

Fault finding

Your *BackPack* should work first time, but if it does not, the first thing to do is check that the correct power voltages are on the IC1 and IC2 sockets and CON3 (the LCD connector). Then check the 5V current drain for the full module, including the LCD; it should range from 100mA to 200mA, depending on the setting of the backlight. If it is substantially lower than this, check that the PIC32 and the LCD are correctly seated in their sockets.

With the LCD removed, the current drain should be about 30mA. If it is a lot less than this, it indicates that the PIC32 processor has not started up and in that case, the 47µF capacitor is the most likely culprit. It must be a tantalum or multilayer ceramic type; not an electrolytic.

If the current drain is correct, check that the *Microbridge* is working correctly. Does your PC recognise it as a valid USB device? Do you have the correct driver installed? Do you have your terminal emulator configured correctly?

You can check the *Microbridge's* operation by typing characters into your terminal emulator and watching for the LED to flash as they are received by the *Microbridge*.

Configuring the Micromite

The next step is to configure the Micromite for the LCD panel. To do this, type the following line at the command prompt (via the USB/serial connection and your terminal emulator software) and hit the enter key:

```
OPTION LCDPANEL ILI9341, L, 2,
  23, 6
```

This tells the Micromite that the LCD panel is connected and which I/O pins are used for critical signals such as reset and device select. This option only needs to be entered once because the Micromite will store the setting in internal non-volatile memory and will automatically recall it whenever power is applied.

Following this command, the Micromite will initialise the display (which should go dark) and return to the command prompt. You can test the display by entering the following at the command prompt:

```
GUI TEST LCDPANEL
```

This will cause the Micromite to draw a series of rapidly overlapping coloured circles on the display as shown in the photo overleaf. This animated test will continue until you press a key on the console's keyboard and MMBasic will then return to the command prompt. To configure the touch feature, enter the following at the command prompt:

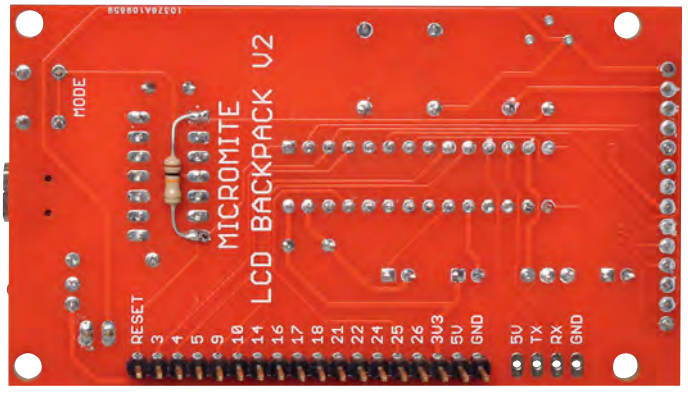
```
OPTION TOUCH 7, 15
```

This allocates the I/O pins for the touch controller and initialises it. This option is also stored in non-volatile memory and is automatically applied on power-up. Before you can use the touch facility, you need to calibrate it. This is done with the following command:

```
GUI CALIBRATE
```

This will cause MMBasic to draw a target in the upper left-hand corner of the screen. Using a pointy but blunt (ie, not too pointy) object, such as a toothpick, press on the exact centre of the target. After a second, the target will disappear and when you lift your implement another target will appear at upper right.

Continue pressing on the targets in this fashion until you have calibrated all four corners of the screen. The message



The underside of the prototype *LCD BackPack V2* contains the pin connections for the Micromite. Note that the 10kΩ resistor soldered between pins 1 and 7 of the PIC16F1455 is soldered through-hole on the top layer of the final PCB.

This is what the screen looks like when running 'GUI TEST LCDPANEL' as it draws a series of coloured circles on top of one another.

Win a BackPack v2

EPE is running a competition to win a fully-assembled Micromite BackPack v2 thanks to the generous sponsorship of Micromite online shop **micromite.org**

For entry details, please turn to page 27

'Done. No errors' should be displayed on the console. You also might get a message indicating that the calibration was inaccurate and in that case you should repeat it, taking more care to press steadily on the centre of each target.

As before, these calibration details are saved in non-volatile memory and will be re-applied at power up. You can now test the touch facility with the command:

```
GUI TEST TOUCH
```

This will clear the screen and when you touch it, pixels will be illuminated at the touch point. This enables you to test the accuracy of the calibration. Pressing any key in the console will terminate the test.

Using the Microbridge

Using the *Microbridge* interface is quite easy. If you have identified the COM number allocated by your operating system, you can enter this into the set-up of your terminal emulator (we recommend Tera Term for Windows). The *Microbridge* defaults to a speed of 38,400 baud, so your terminal emulator will need to be set to a value of 38,400 baud to match the default speed used by the Micromite's console.

You can change the interface to a higher speed, which makes program loading faster and more convenient. For example, at 230,400 baud the built in Micromite editor (the EDIT command) is blazingly fast. To make the change, you need to set the interface speed on the Micromite and then in your terminal emulator. First, change the speed of the Micromite by issuing the following command at the command prompt:

```
OPTION BAUDRATE 230400
```

The Micromite will immediately switch to this speed so you will see some junk characters in your terminal emulator window. You then need to re-configure your terminal emulator for 230,400 baud. Press Enter and you should see the MMBasic command prompt ('>'). Both the terminal emulator and the Micromite will remember this new speed so you do not need to set it again.

If you configure the Micromite to some other baud rate and forget what it is, you may be stuck with a Micromite that you cannot communicate with. If that happens, you can restore the Micromite to its original defaults using the *Microbridge*.

The reset can be performed by pressing the mode switch on the *Microbridge* for two or more seconds, while simultaneously sending a continuous stream of exclamation marks at 38,400 baud, via your terminal emulator. Then release the mode switch while still sending exclamation marks for another two or more seconds. This causes the LED to flash and the MCLR line is briefly driven low to cause the reset.

Parts list

1 double-sided PCB, available from the *EPE PCB Service*, coded 07104171, 86mm × 50mm
1 ILI9341-based touchscreen LCD panel, 320 × 240 pixels, 2.8-inch diagonal (2.2 or 2.4-inch displays need special mounting)
1 PCB-mount SPST momentary tactile pushbutton (S1)
1 100Ω 0.5W vertical side-adjust trimpot (only fit if Q1 and Q2 are omitted)
1 28-pin narrow low-profile DIL IC socket (for IC1)
1 14-pin low-profile DIL IC socket (for IC2)
1 2-pin male header, 2.54mm pitch and jumper shunt (JP1)
1 4-pin male header, 2.54mm pitch (CON1)
1 18-pin male header, 2.54mm pitch (CON2)
1 14-pin female header socket, 2.54mm pitch (CON3)
1 mini Type-B USB 2.0 socket, SMD mounting (CON4)
4 M3 × 12mm tapped spacers
4 M3 × 6mm pan-head machine screws
4 M3 × 8mm pan-head machine screws
4 nylon washers, 3mm ID, 6mm OD, 1mm thick
1 laser-cut lid (optional)

Semiconductors
1 PIC32MX170F256B-50I/SP microcontroller – a PIC32MX170F256B-I/SP can be used but will be limited to 40MHz
1 PIC16F1455-I/P microcontroller programmed with Microbridge firmware (IC2) – the PIC16LF1455-I/P and PIC16(L)F1454-I/P are also suitable
1 MCP1700-3302E/TO 3.3V linear regulator (REG1)
1 3mm red LED (LED1)
1 2N7002 N-channel MOSFET, SOT-23 package (Q1) (optional, for PWM-controlled LCD backlight)
1 DMP2215L P-channel MOSFET, SOT-23 package (Q2) (optional, for PWM-controlled LCD backlight)

Capacitors
3 100nF multi-layer ceramic
2 10µF 16V tantalum or SMD ceramic, X5R, 3216 (1206) size
1 47µF 16V tantalum or 10µF SMD ceramic, X5R, 3216 (1206) size

Resistors (all 0.25W, 5%)
2 10kΩ (1 optional, for PWM-controlled LCD backlight)
2 1kΩ (1 optional, for PWM-controlled LCD backlight)

This will completely restore the Micromite to its initial configuration of 38,400 baud and erase any program and options held in memory. As a result, you will need to re-configure the Micromite for the LCD panel as described earlier.

Backlight control

If you installed the 100Ω trimpot for manual backlight control, the brightness adjustment is as simple as tweaking VR1 to your preference.

If you installed the components for the PWM-controlled backlight (ie, Q1, Q2 and the two associated resistors), the brightness is controlled via the PWM command in MMBasic. By default, the backlight will be at full brightness but it can be controlled with the following command:

```
PWM 2, 250, xx
```

where 'xx' is the percentage of full brightness required. This can range from 0 to 100. For example, a brightness of 75% is a good compromise between visibility and power consumption and this can be set with the following command:

```
PWM 2, 250, 75
```

Within a program, you can get a nice fade from full brightness to black by using the following program fragment:

```
FOR i = 100 to 0 STEP -1
  PWM 2, 250, i
  PAUSE 4
NEXT i
```

The PWM output used for the backlight control appears on pin 26, so this pin is not available for general I/O if you installed the components for the programmed controlled backlight.

Interfacing with other circuitry

The *Micromite LCD Backpack* interfaces to the world using CON2, the main I/O connector. This is designed so that you can plug it into a solderless breadboard or connect to a third board mounted on the back of the *BackPack* (eg, see the *Touchscreen Voltage/Current Reference* project in the October and December 2017 issues). The silk screen on the PCB identifies each pin on the connector. The GND, 5V and 3.3V pins can be used to power your external interface circuitry.

The maximum current that can be drawn from the 3.3V pin is 150mA, while the maximum 5V load will depend on your 5V supply. The RESET pin is normally at 3.3V, pulled up by the onboard 10kΩ resistor, and if you pull it low the Micromite will reset.

The other I/O pins connect directly to the Micromite and are marked with the Micromite pin number. You should refer to the *Micromite User Manual* (available for download from the author's website **http://geoffg.net/micromite.html**) for details of what you can do with each pin.

Three of the pins on CON2 (pins 3, 14 and 25) are also connected to the LCD panel for communicating with the display using the SPI serial protocol. For this reason, they cannot be used as general-purpose I/O pins, however, they can still be used by you for SPI communications if needed – this is why they are included on this connector.

The *User Manual* describes how to use the SPI interface simultaneously with the LCD and it is not hard to do. However, for normal operation, you should make sure that you do not use pins 3, 14 and 25 for general I/O.

If you have any issues or questions then contact Phil Boyce via email (**phil.boyce@micromite.org**) and he will be able to assist you. We hope you enjoy using this new version of the *BackPack*.