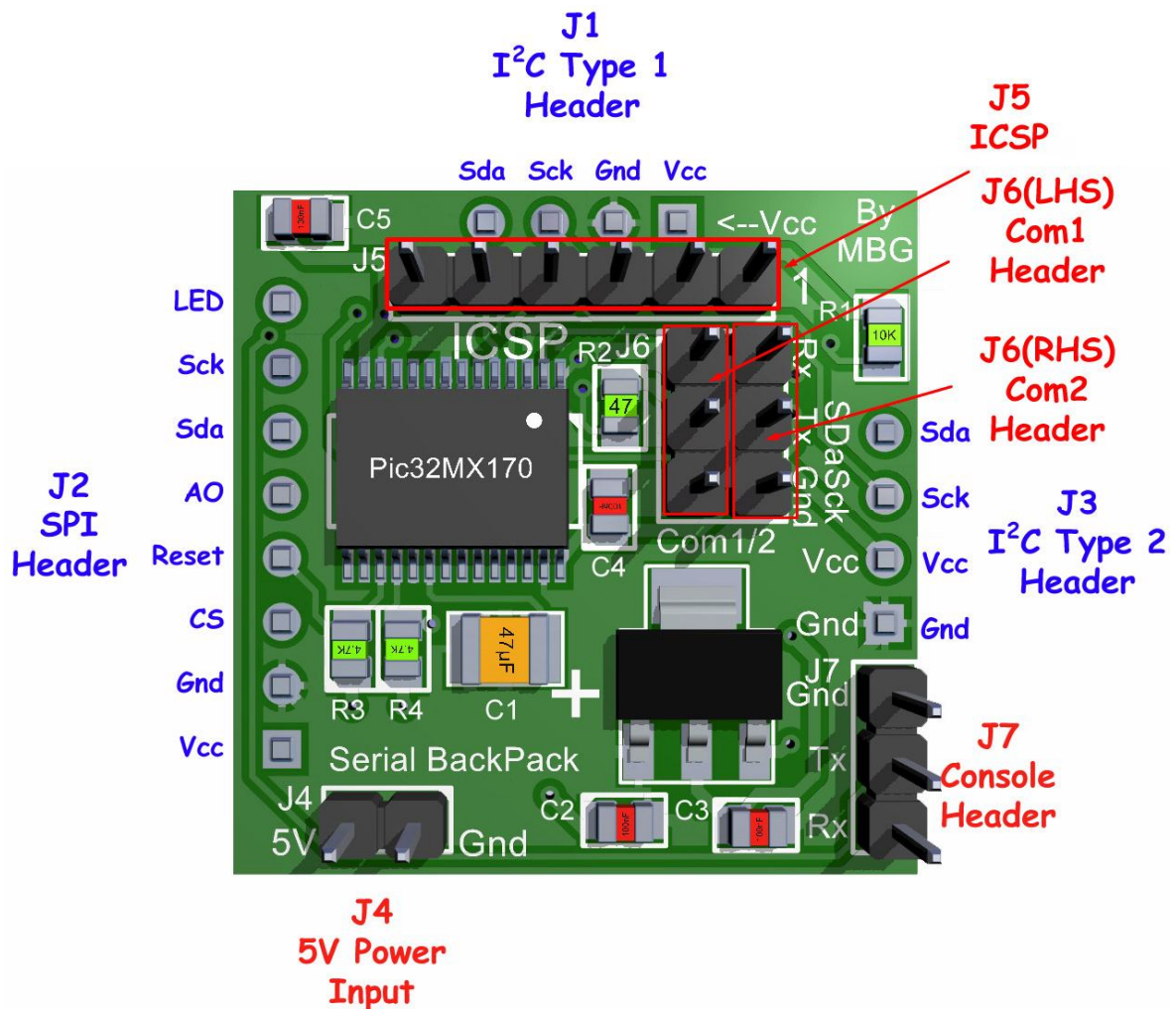
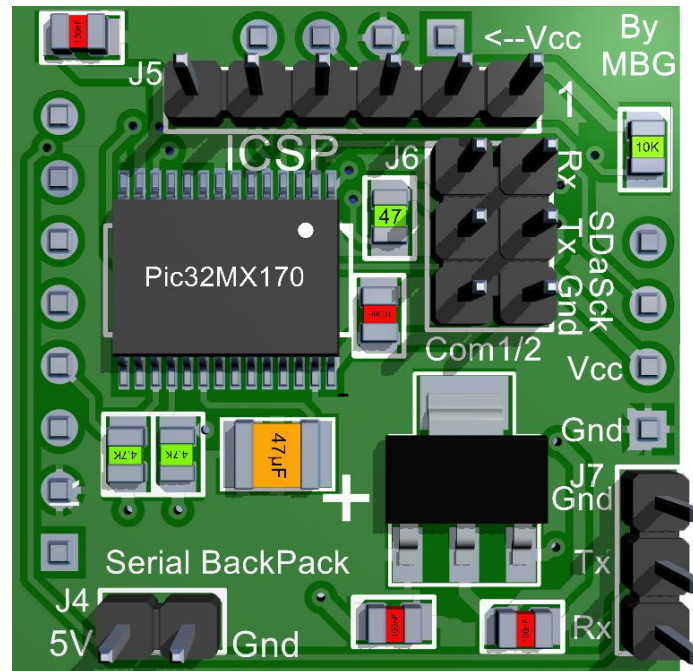


Serial Backpack (ERRATA See end of Doc.)



Board Concept.

The Backpack170 and SMDBackPack170 were both designed to fit onto the back of the readily available 2.4" and 2.8" TFT screens. I was approached by JMan from The Back Shed Forum to see if we couldn't come up with a similar board that could interface to the 0.96" OLED I²C displays and the small (8pin) SPI modules (also the popular Nokia 5110 modules). The main design criteria was to be approximately the same size as the 0.96" OLED displays which is 27.3mm x 27.3mm.

As there are 2 common pinouts for the OLED modules we initially were going to have jumper blocks to configure the header to suit but after thinking about this, I decided that if I provide two I²C headers, each configured, for one OLED pinout then the other header 'could' be used to drive other devices (LED panel or Keypad etc.). The SPI header was straight forward as it seems that all the common small SPI display modules keep to a common pinout. Then I had to provide some communication options and an ICSP header to actually flash the Pic32MX170, whilst it was quite a challenge, I actually managed to get COM1, COM2 and Console headers onto this postage stamp sized board.

Serial Backpack design.

Serial Backpack is primarily designed to be a 'slave' board that mounts onto the back of a 0.96" OLED display module, as such, I expect that data will be sent to it via one of the COM ports, the console, I²C or SPI. It is of course possible to run Serial Backpack stand-alone but, due to its size, I was not able to provide ready access to many IO lines. The only IO lines catered for are the ones allocated for the various communication ports, if you chose not to use a particular port then the following IO lines may be accessible to do other tasks.

NOTE! Pin(17) and Pin(18) may not be useable in all cases as they are shared with both I²C and SPI interfaces.

I'Face	Header(s)	MicroMite Pin No.s	Repective J'x' Labelling
I2C	J1, J3	Pin(17) & Pin(18)	SCK , Sda
SPI	J2	Pin(3), Pin(5), Pin(17), Pin(18) & Pin(25)	SDa, A0, Reset, CS & SCK
Com1	J6	Pin(21) & Pin(22)	Com1 Tx & Com1 Rx
Com2	J6	Pin(9) & Pin(10)	Com2 Tx & Com1 Rx

Power Source

J1 is used to Power the Serial BackPack.

Serial BackPack has its own on-board 3v3 regulator so that it won't put extra current drain on a master Microprocessor's 3v3 regulator. Just supply 5Vdc (regulated) into J4 (watch out for the polarity). I have allowed enough space to fit a 2 position 2.54mm screw terminal jack if you prefer to fit that instead of the standard male pin header.

Connecting to a screen

As mentioned earlier Serial BackPack is designed to mount onto the back of one of several different small display modules. Some examples are shown below but there may be many others that Serial BackPack could interface to just as easily.

OLED 0.96" Display: (Type 1) header J1 OLED 0.96" Display: (Type 2) header J2



Note the Gnd and Vcc pins on these modules are reversed to each other.

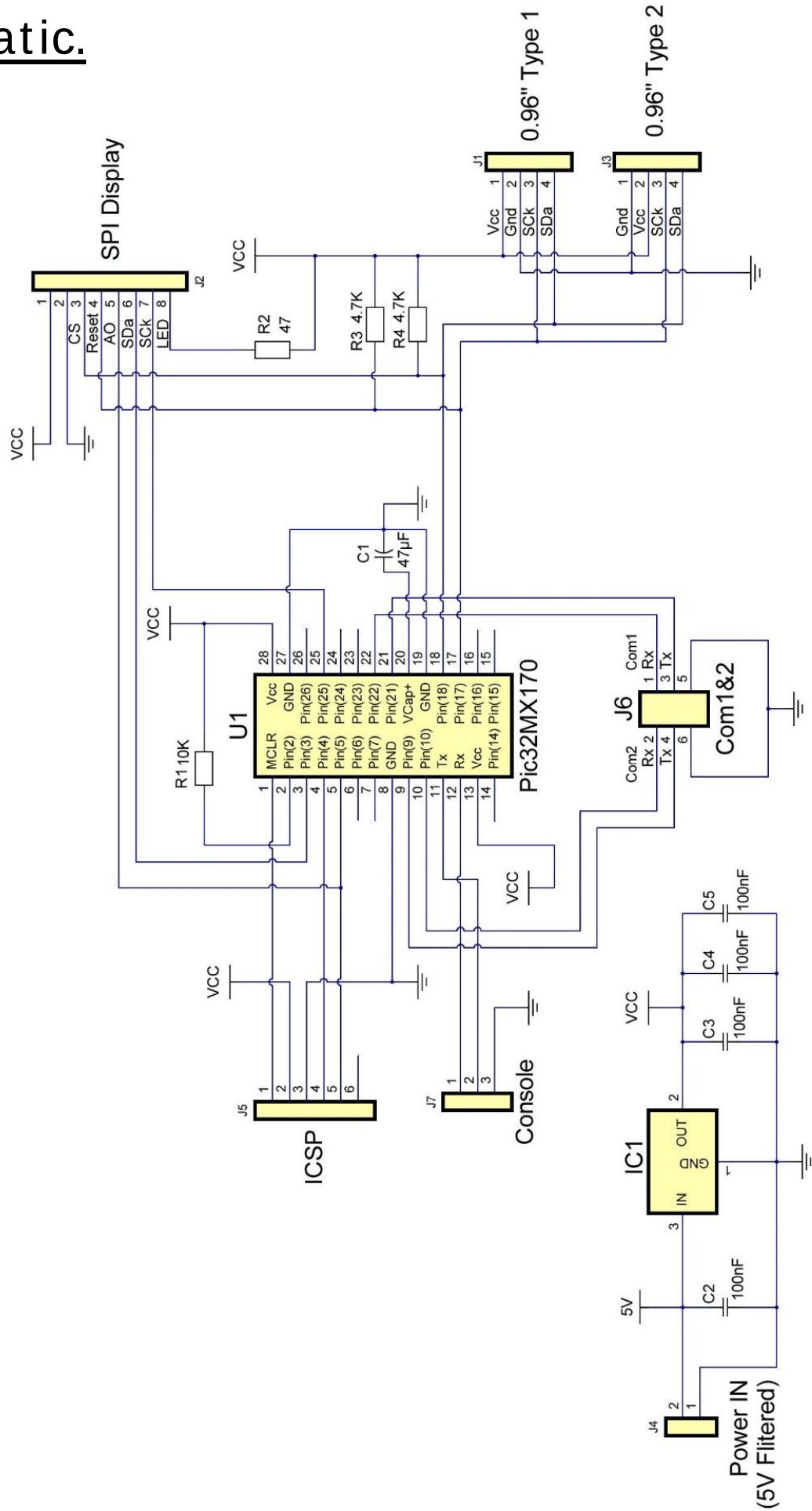


A typical Small SPI based TFT module.

These typically come in 128x128 and 128x160 formats. There are many types Available including those that have Nokia 5110 screens fitted to a suitable Base PCB. **(check pinout as there are 2 types)**



Schematic.



Bill of Materials

Ref.	Description
J1	4pin SIL Female Header
J2	8pin SIL Female Header
J3	4pin SIL Female Header
J4	2pin SIL Male Header
J5	6pin SIL Male Header
J6	6pin DIL Male Header
J7	3pin SIL Male Header
U1	Pic32MX170 SSOP
IC1	AMS1117-3v3 Vreg
C1	47μF 6v3 3225 SMD
C2	100nF
C3	100nF
C4	100nF
C5	100nF
R1	10K Ohm SMD 0805
R2	47 Ohm SMD 0805
R3	4.7K Ohm SMD 0805
R4	4.7K Ohm SMD 0806

NOTES!

J1, J2 & J3

Typically only one of these ports will connect to a display module by either directly soldering to the display module (Display on the underside of the PCB) or alternatively via a female header mounted on the underside of the PCB. The unused ports may have male headers fitted to the top side for access to be used as general IO if desired.

J4

J4 is the main power input to supply power to Serial Backpack, it should be a regulated dc source of 5-6Vdc. (5V suggested). If you prefer the 2 pin male header may be substituted for a 2pin 2.54mm screw terminal jack.

J5

J5 is the ICSP header for flashing MMBasic into the Pic32MX170.

NOTE! If additional General IO is required, Micromite Pin(4) and Pin(5) is also available for use on J5 (appears at J5 pins 4 and 5 respectively) as ICSP is only used when flashing the Pic32MX170 with new firmware.

J6

J6 is the header to provide access to COM1 and COM2 (or as use as general IO)

J7

J7 is the Console header 3v3 TTL levels only. For connection to a PC via a USB-TTL adapter.

Backlight Control

Serial Backpack has no facility to control the backlight, the backlight is simply fed from 3v3 via a 47 Ohm resistor.

Programming the Serial Backpack

As most of you know, I am pretty lousy at programing in MMBasic and totally ignorant when it comes to programming in `C', luckily we are blessed, especially in `The Back Shed' Community with many people who are not handicapped like me in that area. Peter Mather has written some C libraries and published them on TBS forum. User John Gerrard, who instigated the design of the Serial Backpack has graciously allowed me to distribute some Demo code (based on the Peter Mather libraries) with the Serial Backpack PCB.

You will find these in the Serial Backpack directory in my `site' listing as follows:

<http://www.dontronics.com/micks-mite/files/>

On my `site' you will find 3 ZIP files that each contain 3 programs

LCD_BackPack-SSD1306	(Software for the 1306 OLED 0.96" I ² C Display)
LCD_BackPack-5110	(Software for the NOKIA 5110 SPI Display)
LCD_BackPack-ILI9163	(Software for the ILI9163 TFT SPI Display)

Each contains 3 files (I will use as an example the ones for the OLED 1306 displays)

SSD1306-i2c-Driver.bas	(This contains the C driver and configuration settings)
SD1306-BackPack.bas	(This is the code written for the Serial Backpack itself)
SSD1306-Demo.bas	(This is a demo program that will run on a MicroMite to Send commands to the Serial Backpack)

Once you have flashed MMBasic into your Serial BackPack (using a PicKit3 or similar)

1. Decide on the module type (I will use the 1306 in this example)
2. Load the SD1306-i2c-Driver.bas (C based OLED Driver)
3. Edit the startup parameters if necessary to set the Size and Orientation (SD1306 only) the TFT and 5110 are Square(128x128 and 84x84)
Therefore these settings are not required
4. At the MMBasic prompt type "library save" ... without the ""
5. Power cycle the board you should see a message on the console that the driver is loaded (Not for the TFT)
6. Load the correct BackPack for your LCD type (ie. SD1306-BackPack.bas)
7. At the prompt type "option Autorun on" ... without the ""
8. Power cycle the board again.
9. Connect Tx and Gnd from the source (Sending MicroMite) to com1 Rx and Gnd.
10. Load the correct demo for your LCD (SSD1306-Demo.bas) to the source MicroMite and run it. This will send out a continuous stream of DEMO commands via serial to the Serial BackPack.
11. Happy demo :)

Comments From John Gerrard

Reference to the MMBasic manual page 21 (as of Ver 5.1)

The serial BackPack uses the same commands and syntax as a directly connected LCD except they need to be sent via the serial port as string ie "Text 64,15,Serial LCD,CM,1,1,RGB(Yellow)" and NO spaces after each parameter.

The needs it needs to be terminated with a Chr\$(13)

To make it easier to send the LCD command via the serial port use the simple Sub below:

```
LCDOut ("Text 64,15,Serial LCD,CM,1,1,RGB(Yellow)")
```

```
Sub LCDOut(SO1$)
```

```
Print #1, SO1$+Chr$(13)
```

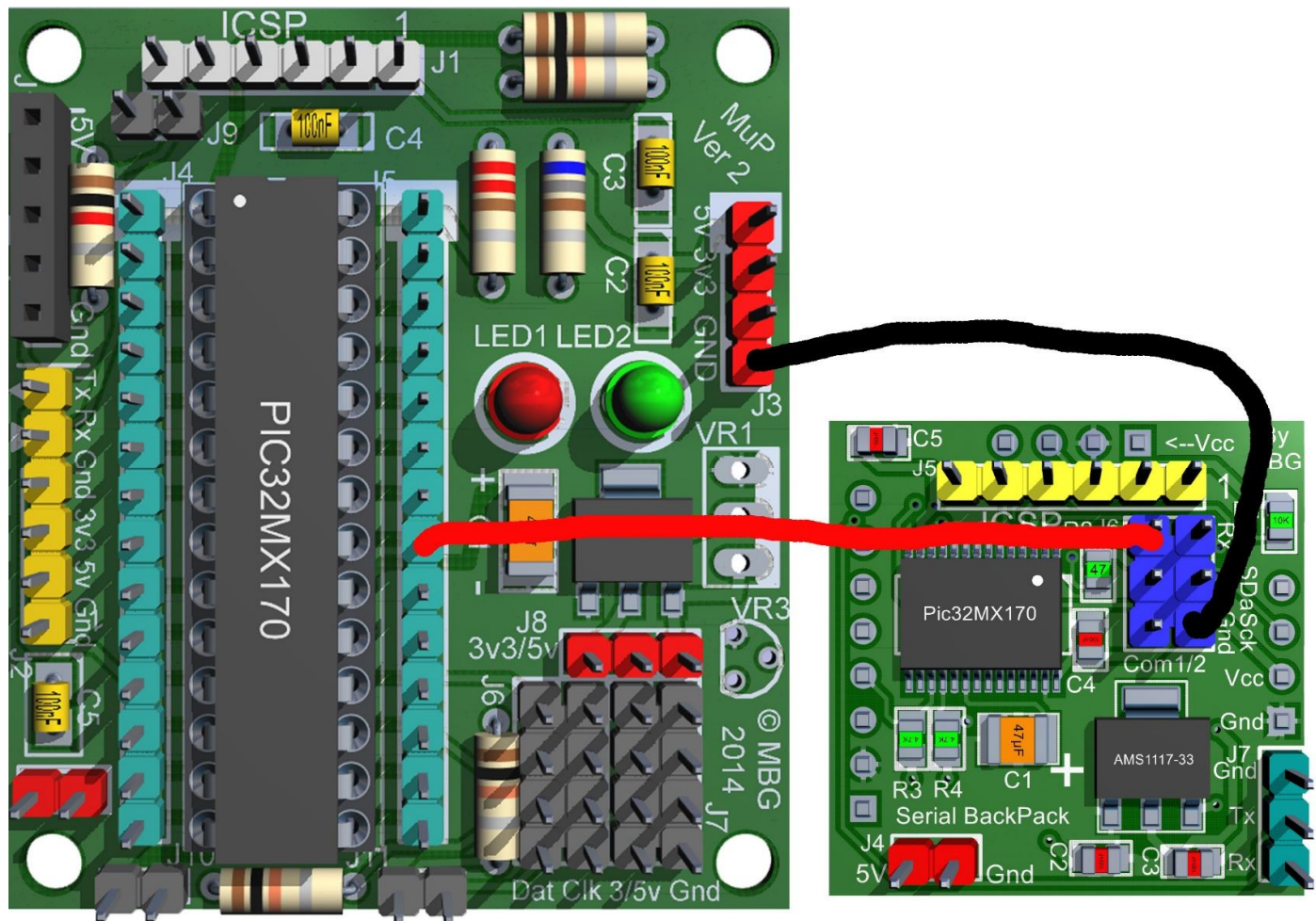
```
End Sub
```

Currently the Bitmap command is not supported, hopefully I can have this done soon. I will ask Peter to include the driver load message for the ILI9163 LCD Driver as the message is helpful when starting out.

Driving the Serial Backpack via a MuP (Mick's Umite PCB ... details on my site)

The image below, (please excuse the poor attempt at drawing wires), shows a MuP on the left (loaded with the DEMO sending program) wired to a Serial Backpack (loaded with the relevant Serial Backpack program and C-Driver library).

As can be seen, only 2 wires are required, The RED wire is from MuP Pin(21) which is Tx to the Serial Backpack COM1 Rx input and the BLACK wire is a GND connection.



28 Pin Micromite Connections

(reprinted, with kind permission by Geoff Graham)

The following diagram shows the possible functions of each I/O pin on the Micromite.

Note that the physical pins on the chip and the pin numbers used in MMBasic are the same. This means that nine pins are not available in MMBasic as they are dedicated to functions such as power and ground. These pins are highlighted in grey in the diagram.

RESET Wired to +V directly or via 10K resist	1	28	ANALOG POWER (+2.3 to +3.6V)
DIGITAL INT ANALOG	2	27	ANALOG GROUND
SPI OUT DIGITAL INT ANALOG	3	26	ANALOG DIGITAL PWM 2A
PWM 1A DIGITAL INT ANALOG	4	25	ANALOG DIGITAL SPI CLOCK
PWM 1B DIGITAL INT ANALOG	5	24	ANALOG DIGITAL PWM 2B
PWM 1C DIGITAL INT ANALOG	6	23	ANALOG DIGITAL
COM1: ENABLE DIGITAL INT ANALOG	7	22	DIGITAL 5V COM1: RECEIVE
GROUND	8	21	DIGITAL 5V COM1: TRANSMIT
COM2: TRANSMIT INT DIGITAL	9	20	47µF TANT CAPACITOR (+)
COM2: RECEIVE INT DIGITAL	10	19	GROUND
CONSOLE Tx (DATA OUT)	11	18	DIGITAL 5V COUNT I ² C DATA
CONSOLE Rx (DATA IN)	12	17	DIGITAL 5V COUNT I ² C CLOCK
POWER (+2.3 to +3.6V)	13	16	DIGITAL 5V COUNT WAKEUP IR
SPI IN 5V DIGITAL	14	15	DIGITAL 5V COUNT

The notation is as follows (the mnemonic in brackets is the mode used in the SETPIN command):

ANALOG	These pins can be used to measure voltage (AIN).
DIGITAL	Can be used for digital I/O such as digital input (DIN), digital output (DOUT) and open collector output (OOUT).
INT	Can be used to generate an interrupt (INTH, INTL and INTB).
COUNT	Can be used to measure frequency (FIN), period (PIN) or counting (CIN).
5V	These pins can be connected to 5V circuits. All other I/O pins are strictly 3.3V maximum.
COM xxx	These are used for serial communications (see Appendix A)
I ² C xxx	These are used for I ² C communications (see Appendix B)
SPI xxx	If SPI is enabled these pins will be used for SPI I/O (see Appendix D)
PWM xxx	PWM or SERVO output (see the PWM and SERVO commands)
IR	This can be used to receive signals from an infrared remote control (see the IR command)
WAKEUP	This pin can be used to wake the CPU from a sleep (see the CPU SLEEP command).

Useful Links:

Further information on the software and hardware implementation can be found in the forums located here:

2.4" SPI Colour Screen

http://www.thebackshed.com/forum/forum_posts.asp?TID=7355&KW=2.4%22

Full hardware design for the MicroMite (uMite) project can be found here:

<http://geoffg.net/micromite.html>

The Back Shed Microcontroller Forum

http://www.thebackshed.com/forum/forum_topics.asp?FID=16

[MicroMite.org](http://www.micromite.org) (They stock a wide range of other uMite goodies including my boards)

Dontronics, A supplier of other hobbyist items and microprocessor boards

<http://www.shop-dontronics.com/>

My recommended TTL serial cable for BP170

<https://www.shop-dontronics.com/ftdi-usb-to-serial-ttl-level-3.3v-converter-cable>

A shameless plug of other Products and offerings I have to offer

http://www.thebackshed.com/forum/forum_posts.asp?TID=6992&KW=***

Or

<http://www.dontronics.com/micks-mites/files/>

ERRATA

Well Huge embarrassment to me.



It transpires that I have inadvertently connected the MCLR, 10k Ohm, Pull-up resistor onto Pin 2 instead of Pin 1 (see the schematic).

Strangely enough Serial Backpack still works flawlessly so I assume that there must be an internal pullup in the PIC32MX170F256B chip..

In either case where does that leave us?

Well, I am glad you asked.

These are your options..

1. Ignore it as it seems to work flawlessly
2. Solder short Pin 1 to Pin 2. As Pin(2) is undefined it would be safe to do that.
3. Cut the track going from R1 to Pin 2 and solder a short wire to Pin 1 of the PIC chip.
4. Contact me if you have bought a Version 1 and I will provide a Version 2 for \$1.00Aus plus postage.
5. Relegate it to a non-critical task and buy some Version 2.

Version 2? I hear you asking, Yes, I have done a Version 2 and it will be released soon.

The good news is that, after input from Peter Mather Version 2 is a lot more versatile, with more IO lines available, making it quite a useful board even in a stand-alone mode.

As a consequence, I have pulled Version 1 from sale and it will be replaced by Version 2.

Read the Version 2 manual for more details located on my site at:

<http://www.dontronics.com/micks-mite/files/>