

LCD6402B Evaluation Kit

User Guide

LCD6402-EVAL



Kit Contents:

- LCD6402-DEV Development/Evaluation Board
- LCD6402B-PA Intelligent 128x64 Graphic LCD Module
- Serial extension lead
- LCDLAB Software
- 12V AC-DC adaptor
- User guide and data sheets

Summit Electronics Ltd

www.summitelectronics.co.uk

1 INTRODUCTION

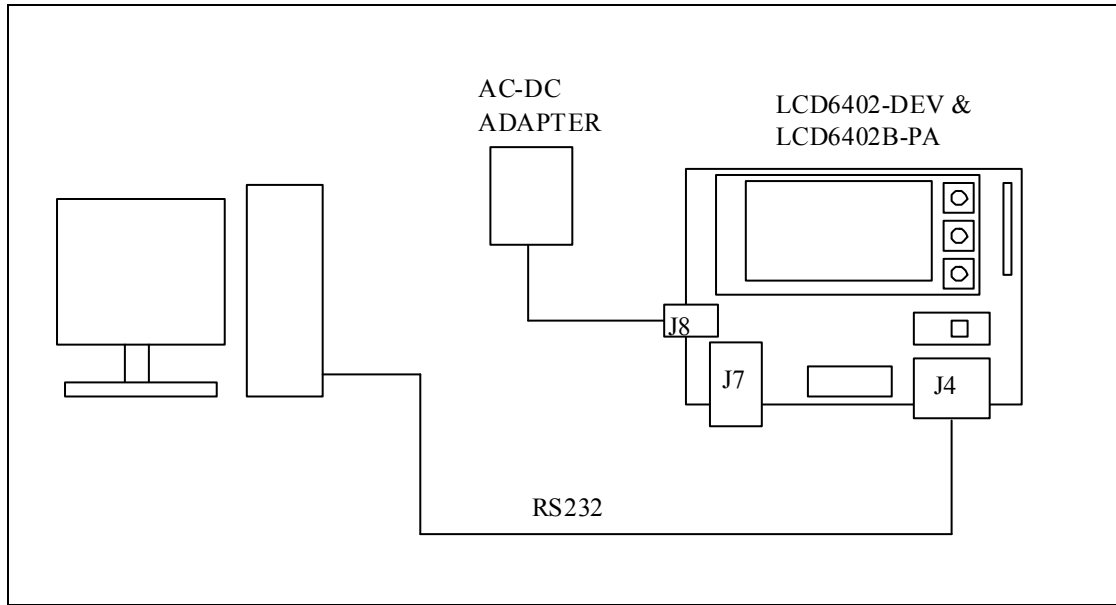
The LCD6402-EVAL evaluation kit offers a fast and convenient way to evaluate the LCD6402B intelligent 128x64 graphic LCD module. In the following tutorials you will be able to add bitmaps, create your own fonts, draw lines and circles and print inverted text. Also included with this guide is a collection of data sheets and application notes relevant to the LCD6402B.

1.1 GETTING STARTED

Before we can get going you need to set up the evaluation kit as follows (also see also the figure below):

1. Remove the LCD6402-DEV and LCD6402B from the packaging*.
2. Carefully plug the LCD6402B into the LCD6402-DEV making sure the pins line up correctly with the LCD6402-DEV. At this stage you may remove the protective plastic film from the LCD screen.
3. Remove the plug-in power supply (PSU) from its packaging and plug it into a wall outlet.
4. Plug the other end of the PSU into J8 on the LCD6402-DEV. The module will light up and display the text 'Console V1.000'.
5. Plug one end of the serial lead into J4 on the LCD6402-DEV and the other end into a serial port on a PC running Microsoft Windows.
6. Start Hyper Terminal or other similar software on the PC and set up the serial port to the following:
 - Baud rate : 9600
 - Data bits : 8
 - Stop bits : 1
 - Flow control : None
7. Press the reset button (SW2) on the LCD6402-DEV and the following message will be displayed on the PC: 'Summit Electronics Ltd. LCD6402-DEV Console V1.000'. If this message does not appear check the correct serial port has been select in the terminal program and try again.
8. Place the CD in your CD drive and install LCDLAB.

* *Observe anti static precautions when handling these components.*



LCD6402B-EVAL General arrangement



LCD6402-DEV & LCD6402B assembly

2 TUTORIALS

These tutorials will help familiarise you with the LCD6402B but are by no means exhaustive of the module's full potential. Example screen shots are shown where possible.

2.1 TUTORIAL #1: Using the console

Every LCD6402-DEV is ready programmed with the console program. All you have to do is enter commands like 'PS "Hello World"' and the console program checks it for syntax before converting it into the appropriate graphic instructions and sending them over the I2C bus to the LCD6402B. Both the 'C' source code and hex file are available as an example interface to the LCD6402B.

Before doing anything, ensure that SW1 on the LCD6402B-DEV is switched to the 'HOST' position.

2.1.1 Getting help

To get help, type '?' on the PC and press return. A list of available commands will be shown as follows:

```
BL n          : Backlight on/off
CC r          : Draw circle with radius r
CF r          : Draw filled circle
CS           : Clear screen
CU x y       : Set cursor
FG n         : Set foreground colour. n=0 (black), n=255 (white)
BG n         : Set background colour. n=0 (black), n=255 (white)
MD n         : Set mode. Eg: OR=0, XOR=1...
RT x0 y0 x1 y1: Draw rectangle
RF x0 y0 x1 y1: Draw filled rectangle
FT n         : Set font
LT x y       : Draw line to
OR x y       : Set origin
PS "c..."   : Print string
PX x y n     : Set pixel. n=1 (FG colour), n=0 (BG colour)
RC n         : Recall object n=0..127
RS           : Reset
HR n         : Print horizontal raster byte n
VR n         : Print vertical raster byte n
$W r n       : Write value n to LCD64XX register r
$R r         : Read and print contents of LCD64XX register r
```

This list serves as a quick reference. For a complete description of the console commands see the LCD6402-DEV data sheet.

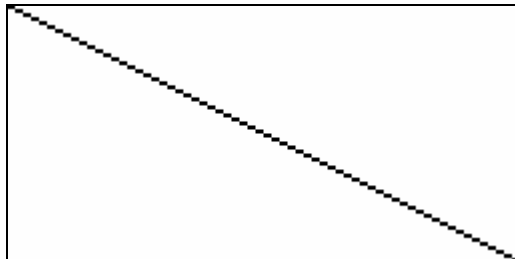
2.1.2 Graphic functions

Just about every graphic function is available with the console program except bitmaps and fonts which will be dealt with later.

In general the graphics cursor needs moving to a point on the display before carrying out the desired function. Below are a few examples for you to try out. If things go wrong just press reset on the LCD6402-DEV to start again.

Example 1: Draw line from 0,0 to 127,63.

```
>CU 0 0  
>LT 127 63
```



Example 2: Clear screen

```
>CS
```

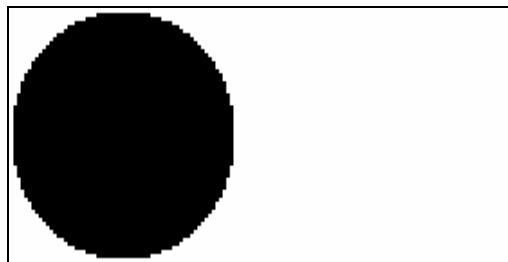
Example 3: Print "Hello World" at 0 32 using built-in font type 1.

```
>FT 1  
>CU 0 32  
>PS "Hello World"
```



Example 4: Draw a filled circle at 32 32 with radius 30.

```
>CU 32 32  
>CF 30
```



Example 5: Continuing on from example 4, print exclusively or (XOR) text at 32, 32.

```
>MD 0x21  
>PS "Hello World"
```



Note the MD command has set the mode to: background colour off and foreground colour XOR. For more details on this see MODE register in the LCD6402B data sheet.

Example 6: Continuing on from example 5, erase the text "Hello World".

```
>CU 32 32
>PS "Hello World"
```

Example 7: Reverse foreground and back ground colours.

```
>FG 255
>BG 0
>CS
```

The above examples illustrate a selection of the graphic functions available.

It is also worth noting at this stage that the console program is unsuitable for being directly driven from a computer program since it doesn't employ flow control.

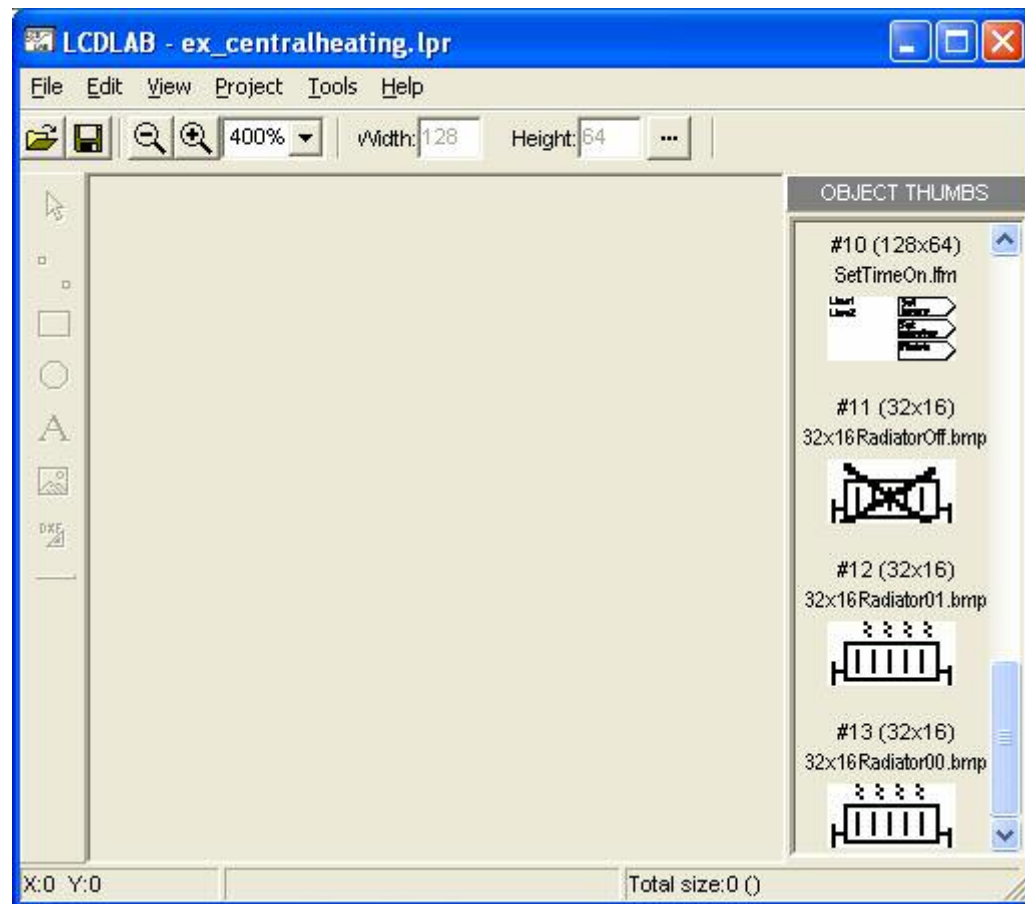
2.2 TUTORIAL #2: Programming and recalling bitmaps

Bitmaps can be created either by entering them into the host code programmatically or by recalling them from the LCD6402B on-board EEPROM. The first method is rarely used and is beyond our discussion here but if you are interested, see the example projects detailed in the LCD6402-DEV data sheet.

The second method uses LCDLAB. Start LCDLAB and from the main menu 'File | Open Project' select the following project:

```
'LCD6402-DEV\ex_centralheating\LCDLAB\ex_centralheating.lpr'
```

LCDLAB will look similar to the following screen shot:



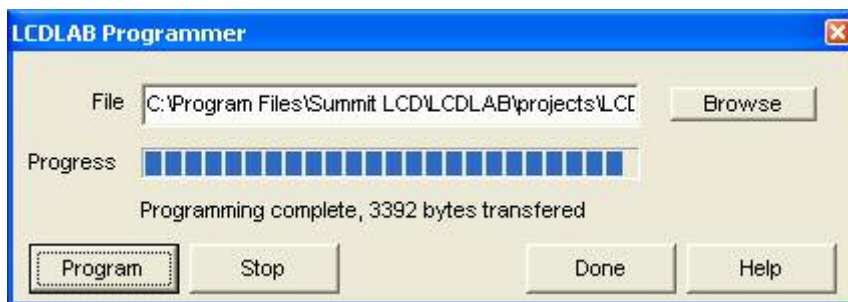
On the right you will see a number of object thumbs. Scroll down to the bottom and you will see a number of images ending in '.bmp' numbered #11~#13. These are bitmaps that were added to the project from 'View | Project Manager...'. See Tutorial #3 for more details.

We are now going to program the LCD6402B with these images (and all the other graphics) but before starting we need to terminate (or disconnect) the terminal program that you were using in tutorial #1 because LCDLAB now wants to use the serial port.

If this is the first time you are using LCDLAB you need to configure the serial port by selecting 'Tools | Communication Setup...' from the main menu. Select the desired port from the 'Port' dialog'.

Now do the following:

1. Ensure a serial cable is connected between PC and LCD6402-DEV.
2. Set SW1 to 'LCD' on the LCD6402-DEV.
3. From LCDLAB main menu select 'Tools | Programmer...' and click 'Program'. If you get the message 'Target not responding' check that the serial port is correctly chosen, that SW1 is in the 'LCD' position and power is applied to the LCD6402-DEV. The programmer window will appear similar to the following:



Click 'Done' when the programming is complete.

4. Set SW1 to 'HOST' on the LCD6402-DEV.
5. Restart your terminal program (or reconnect the port). LCDLAB won't use the serial port as long as the 'LCD Programmer' window is closed.
6. Press LCD6402-DEV reset button.

The bitmaps are now programmed to the LCD. We can recall them to any desired position on the LCD. The bitmap origin is the top left corner so first position the cursor and then recall the bitmap.

Using the console program try out the following examples:

Example 1: Clear screen and draw bitmap (object #11) at 0 0.

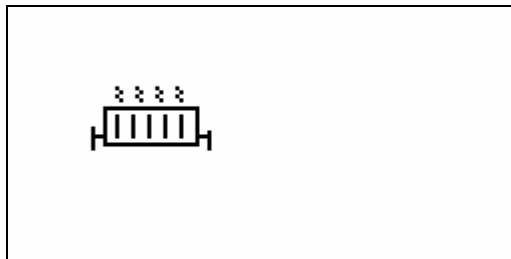
```
>CS  
>RC 11
```



Note: Clear screen command (CS) resets the cursor to 0 0.

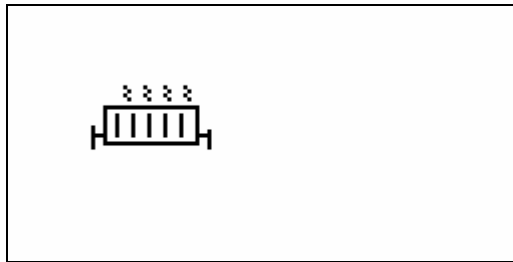
Example 2: Draw bitmap #12 at 20, 32.

```
>CS  
>CU 20 32  
>RC 12
```



Example 3: Carrying on from example 2, draw bitmap (#13) at 20 32.

```
>CU 20 32  
>RC 13
```

Observe that a simple two image animation has taken place; try examples 2 and 3 again.

Example 4: Clear screen, draw bitmap #11 at 20 32 and erase it.

Draw bitmap

```
>CU 20 32
```

```
>RC 11
```

Erase bitmap

```
>MD 0x21
```

```
>CU 20 32
```

```
>RC 11
```

This is a method of producing flashing graphics by the host and is further enhanced with the use of the save/restore context commands. See LCD6402B data sheet COMMAND DESCRIPTIONS for more information.

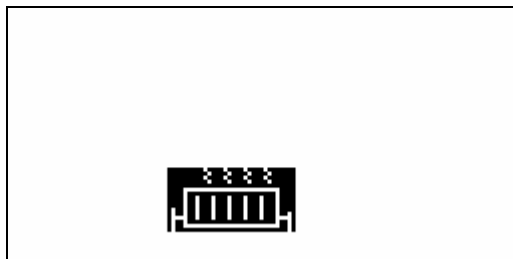
Example 5: Clear screen and draw negative of bitmap #12 at 40 40.

```
>CU 40 40
```

```
>FG 255
```

```
>BG 0
```

```
>RC 12
```



Now you have a feel for how to draw bitmaps at different locations on the LCD. Next we will look at adding more bitmaps.

2.3 Tutorial #3: Adding bitmaps to LCDLAB

With LCDLAB up and running from our previous tutorial let's add a new bitmap as follows:

1. Open the project manager from 'View | Project Manager...', and click the 'Objects' tab.
2. Click 'Add' and select '48x32time.bmp' (This is in the LCD6402-DEV\ex_centralheating\LCDLAB\ folder) and click 'Open'.

3. The bit map has now been added to the project and will appear as the last object (#14) in the 'OBJECT THUMBS' scroll.
4. Carry out the procedure in Tutorial #2 to program the modified project to the LCD6402B.

Let's test it out with the console program remembering to put SW1 back to HOST and start your terminal program etc.

From the terminal type:

```
>CS  
>RC 12
```

...to get:



Up to now we have dealt with bitmaps as objects in their own right, but what if we want a collage of bitmaps and other graphics, all in fixed locations on the display? It would be time consuming to position these programmatically with many cursor commands and then recalling each individual image. The answer to this is in the next tutorial - 'Working with compound objects'.

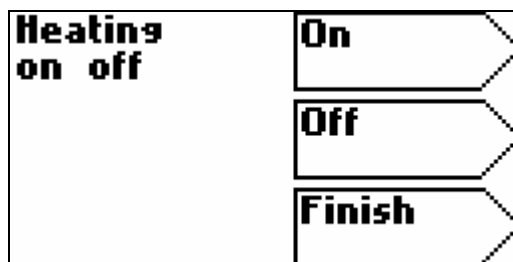
2.4 Tutorial #4: Working with compound objects

A compound object consists of one or more differing types of graphic, for example lines and bitmaps. In a lot of cases we want graphics at fixed locations on the display and this is where compound objects are used. Assuming you have done the above tutorials we can experiment with this new object type.

Press reset on the LCD6402-DEV and, using the console program, try the following examples:

Example 1: Draw compound object #5

```
>CS  
>RC 5
```



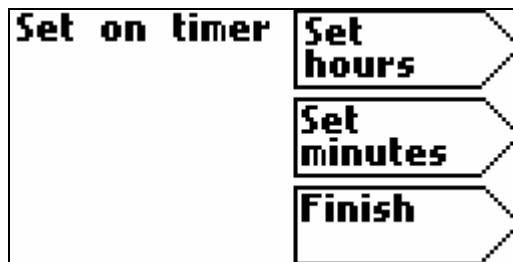
Example 2: Draw compound object #6

```
>CS  
>RC 6
```



Example 3: Draw compound object #7

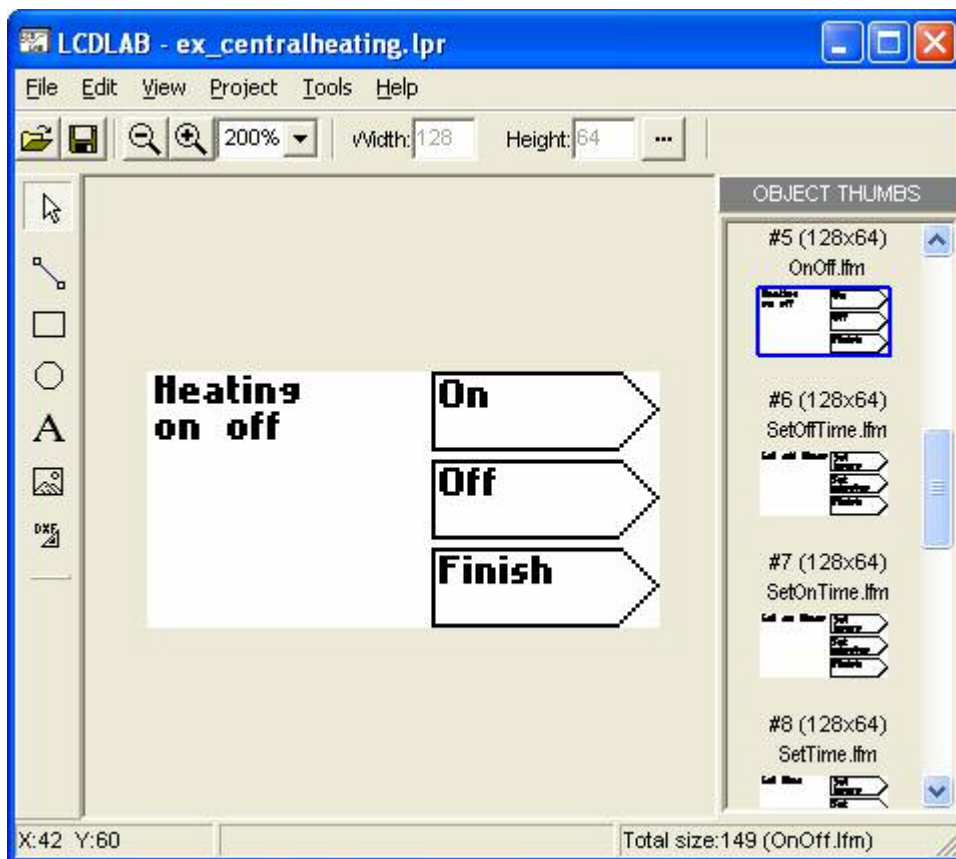
```
>CS  
>RC 7
```



So just by clearing the screen and recalling the compound object by its index is all that's required to get a screen full of lines and text. What this means is that the host controller needs very little coding to produce detailed graphics.

Let's now suppose you want to modify the text on object #5 to read 'Boiler' in place of 'Heating'. Do the following:

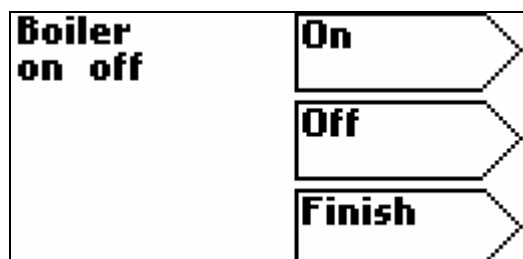
1. In LCDLAB scroll the OBJECT THUMBS pane until object #5 is in view, click the mouse on the image and LCDLAB will appear similar to the screen shot below.



2. Position the mouse over the text 'Heating' and double left click.
3. Change the Text dialogue box to read 'Boiler' and click OK.
4. Go back and click on the OBJECT THUMB image as in step 1. Click Yes on the pop-up dialogue to keep the changes.
5. Program the LCD6402B with the updated project in the usual way. By now you will be used to moving SW1 to the LCD position etc.

Let's test it out from the console with:

```
>CS  
>RC 5
```

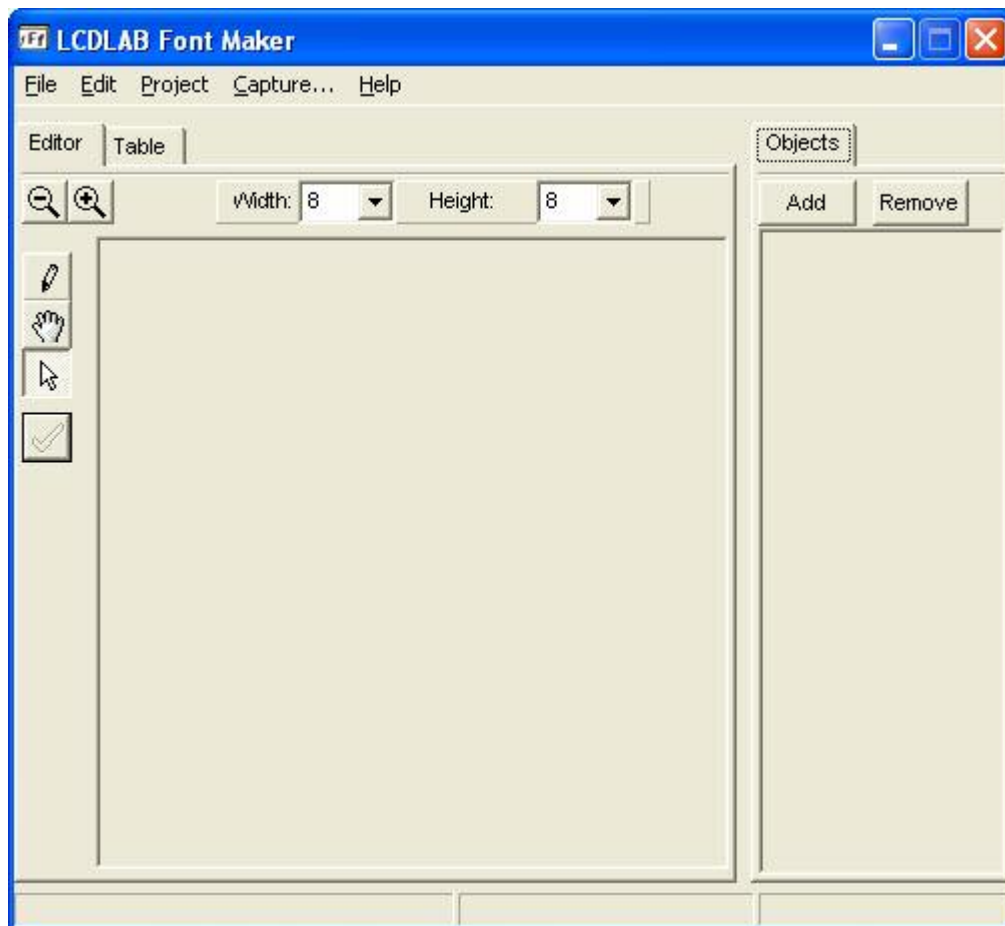


Not only can text be modified in this way, but you can also add lines and circles, insert bitmaps, cut and paste, group and ungroup objects and more. See LCDLAB help for more details.

2.5 Tutorial #5 Creating and using fonts

With LCDLAB comes Font Maker, a powerful tool for creating fonts for your LCD application. Font Maker can capture almost any font from the operating system and program it to the LCD6402B.

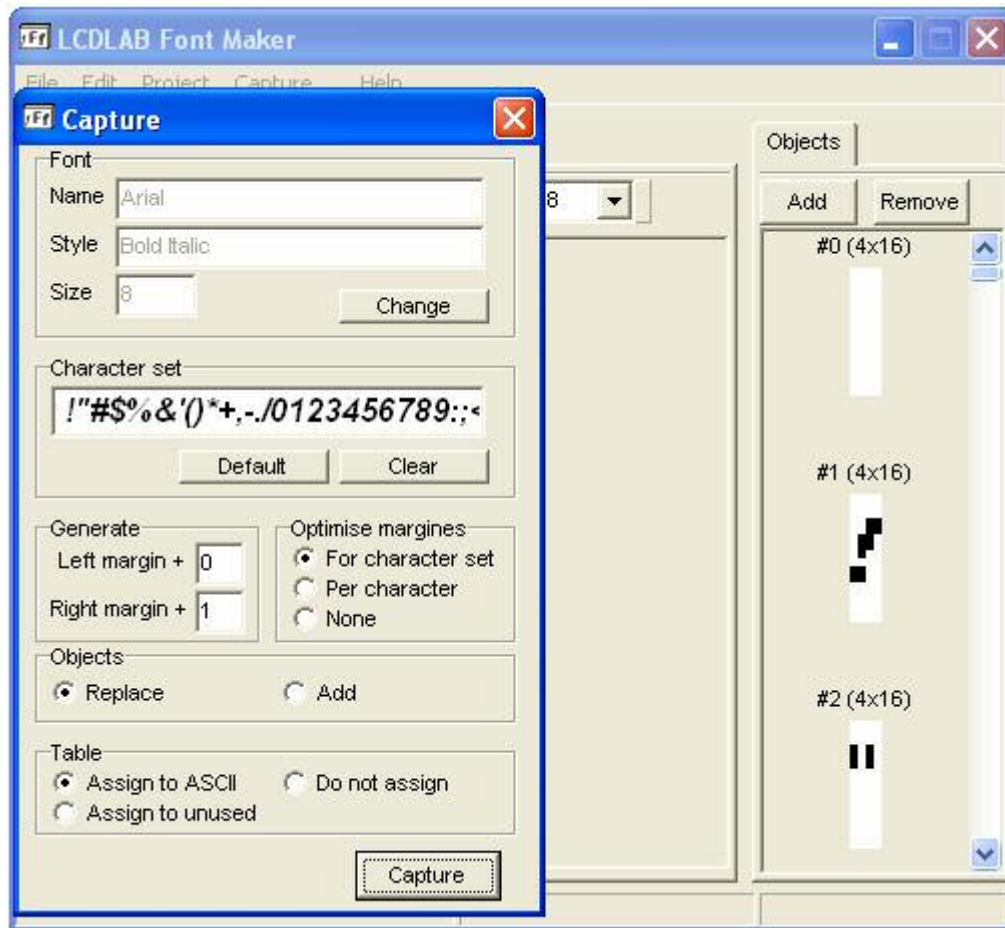
Start Font Maker by selecting 'Tools | Font Maker...' from LCDLAB's main menu and you will be presented with the following:



To create a font, do the following:


1. Click 'Capture...' from the main menu
2. Click Change and set Font: to Aerial, Font style: to Bold Italic and Size: to 8.
3. Click 'Capture'.

Font Maker will look similar to the screen shot below.



That completes the operation. If you want a different font just click 'Change' and repeat the above.

Once the font has been captured, the next thing is to save and add it to LCDLAB as follows:

1. Close the capture window by clicking .
2. Save the font to our project directory in tutorial #4 by selecting 'File | Save' and name it 'ArialBoldItalic16'.
3. Close Font Maker.
4. From the main menu in LCDLAB open the project manager with 'View | Project Manager...' and click the Fonts tab.
5. Click Add and browse for the file saved in step 2.

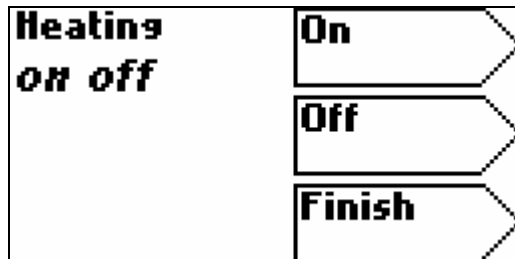
Now we are going to put our font to use by modifying some text in object #5. So, scroll to and select object #5 then do the following:

1. Double left click on the text 'On Off'.
2. Select 'ArialBoldItalic16.fnt (Type 9)' from the Font combo-box and click OK.
3. Click object #5 on the OBJECT THUMBS and then click Yes to save the changes.

4. Program the LCD6402B in the normal way.

Try it out from the console by typing:

```
>CS  
>RC 5
```



There are a couple things to note about fonts:

- i) The larger the font the more memory it will use up in the LCD6402B. This can be managed by reducing the character set. See Font Maker help for more information.
- ii) Font height is in multiples of 8. Fonts that don't fit the height criteria will have their bottom margin padded out until they do. This uses up unnecessary space and can be reduced by adjusting the font size accordingly during capture.

This completes the tutorials for the LCD6402B and you will now have an appreciation of the module's capability. For further reading see the enclosed application notes and data sheets.

LCD6402-EVAL Evaluation Kit User Guide

Disclaimer

The information in this document is offered in good faith as representing the characteristic of its product. Summit Electronics Ltd will not accept any responsibility whatsoever for any damage or loss sustained as a result of any inclusion, omission or error in documentation, verbal advice or other data. Use of Summit Electronics Ltd products in medical applications is not approved. Such use includes, but is not limited to life support systems.