DATA SHEET

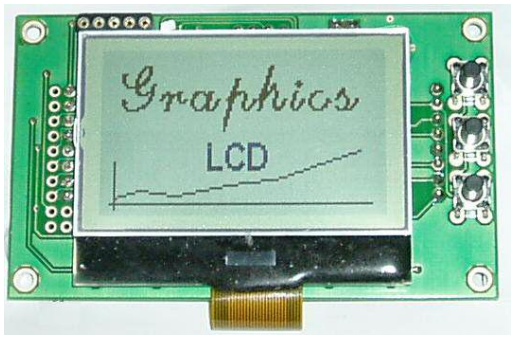
LCD6402B Intelligent 128x64 Monochrome Graphics LCD Module

128x64 INTELLIGENT MONOCHROME GRAPHIC LCD

FEATURES

- 128 x 64 Graphics LCD
- Colour: Black on white
- White back-light
- Built-in graphics functions:
  - Line
  - Circle / filled circle
  - Rectangle / filled rectangle
  - Text printing
  - Bit map

- Customisable fonts
- I2C interface
- Operates from single 5V supply
- Digital contrast control
- 32768 bytes of user EEPROM
- Back-light control
- PIN & IDC connector versions
- Optional 3 x onboard user keys
- Keypad interface
- Free LCD graphics programming tool



Overall size:

80W x 50H x 14D (mm)

Viewing area: 50W X 31H (mm)

APPLICATIONS

- Access control
- Alarm panels
- Portable instruments
- Security
- Information panels
- Vending machines

INTRODUCTION

The LCD6402B is a low cost *intelligent* monochrome liquid crystal graphic display featuring I2C interface, 32K-bytes of non-volatile memory and keypad interface. Operation is from a single 5V supply.

The LCD6402B is designed for easy implementation into embedded applications.

Table of Contents

Table of Figures

Table of Tables

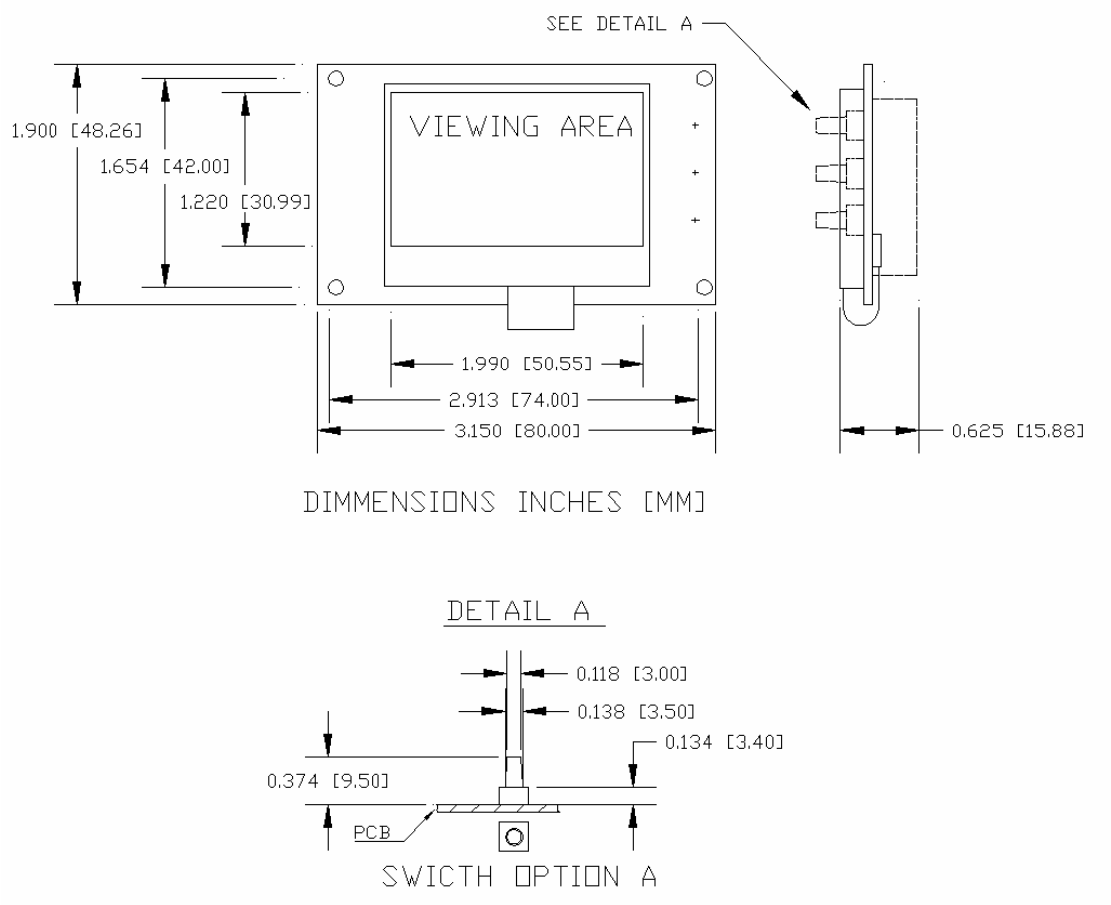


Figure 1 Overall dimensions

For more detailed dimensions see section 11.

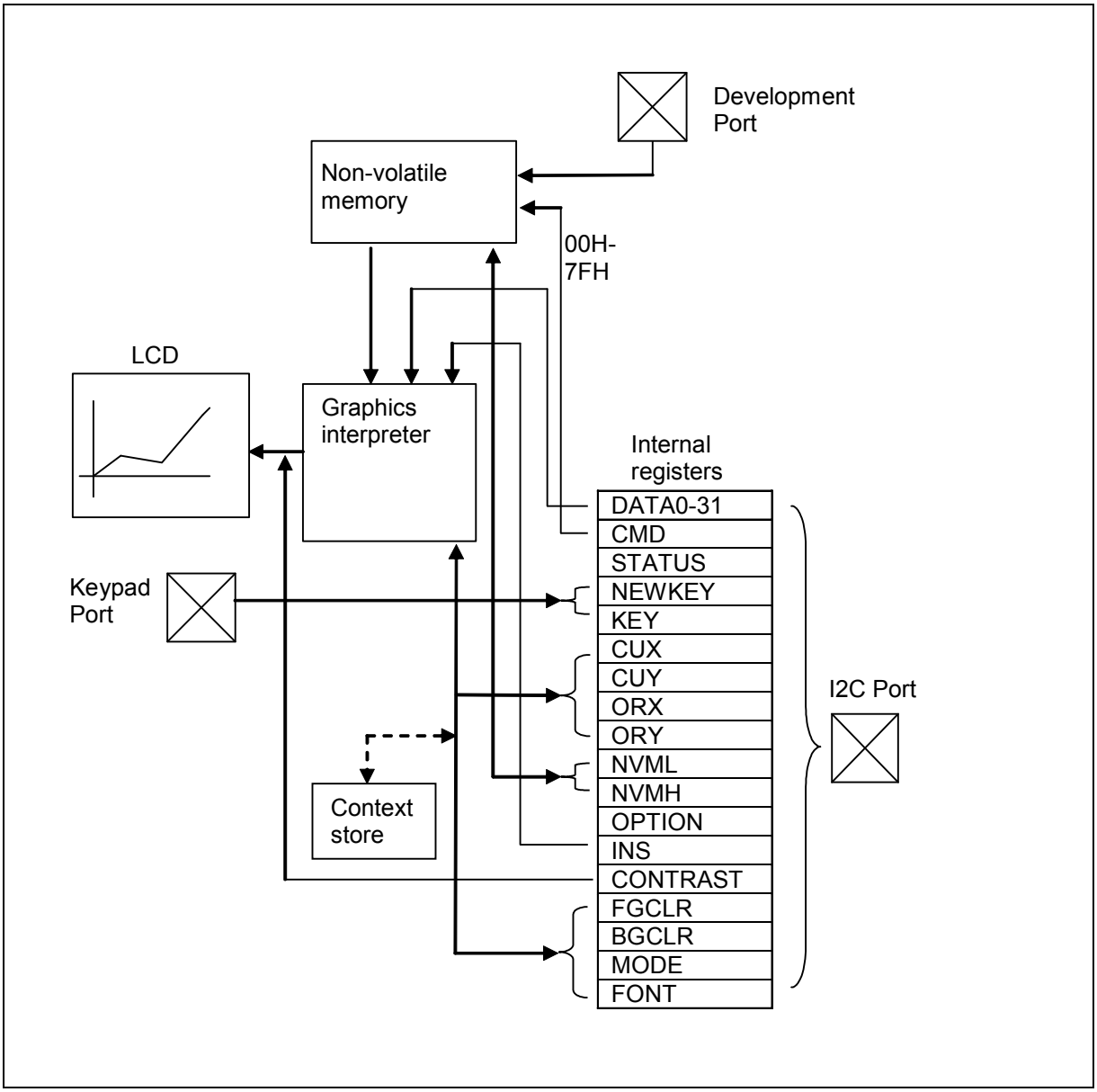


Figure 2 LCD6402B block diagram

1 GENERAL DESCRIPTION

The LCD6402B is a low cost intelligent graphic liquid crystal display with 2-wire I2C bus interface, 32768 bytes of on board non-volatile (EEPROM) memory and keypad interface.

Unlike many LCD graphic modules, the LCD6402B has a number of in-built graphic functions such as line, circle and font generation.

The LCD6402B is capable of storing and recalling graphic images and fonts programmed to its EEPROM memory saving host processor CPU time and storage requirements.

A 16-button 4x4-matrix keypad can be connected to the LCD6402B keypad connector allowing user input to be collected via the I2C bus.

1.1 APPLICATIONS

The LCD6402B has numerous applications and is ideal for portable instruments, heating and ventilation control panels and vehicle instrumentation. The LCD6402B offers the potential of an intuitive, easy to use, operator friendly interface from the simplest of host controllers such as Parallax BASIC stamps to high end microcontrollers.

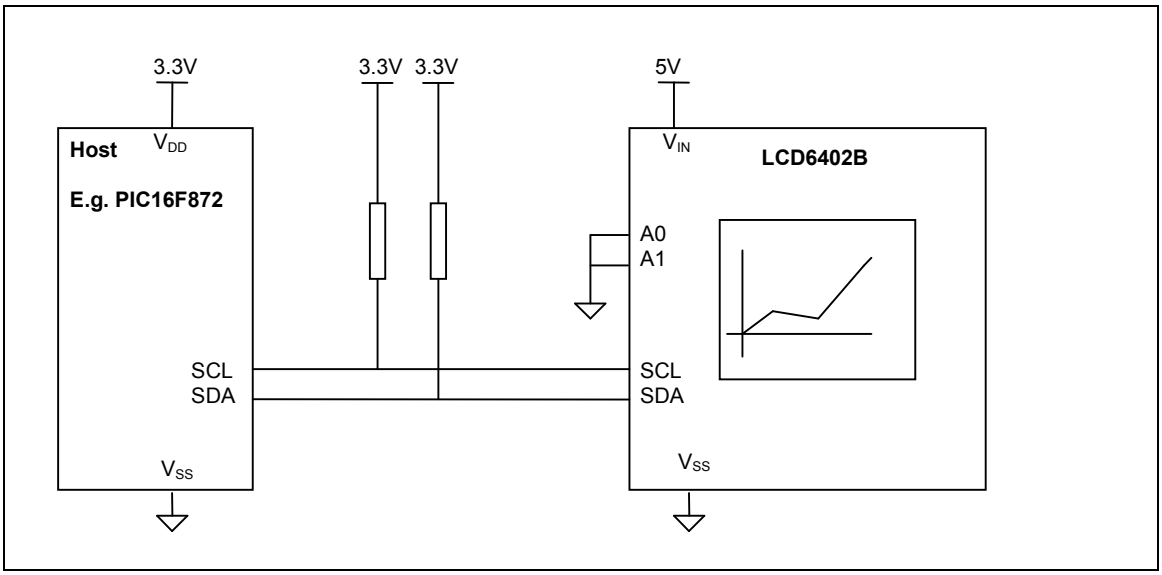


Figure 3 Application with minimal interface requirements (3.3V)

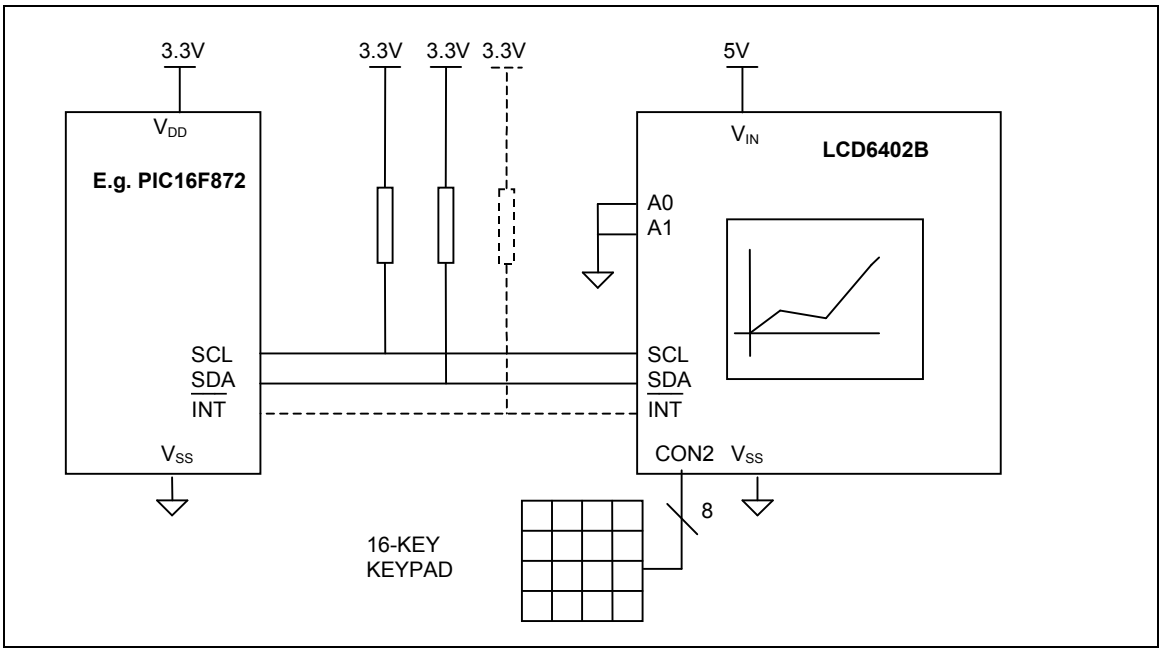


Figure 4 Application with keypad and optional INT interface (3.3V)

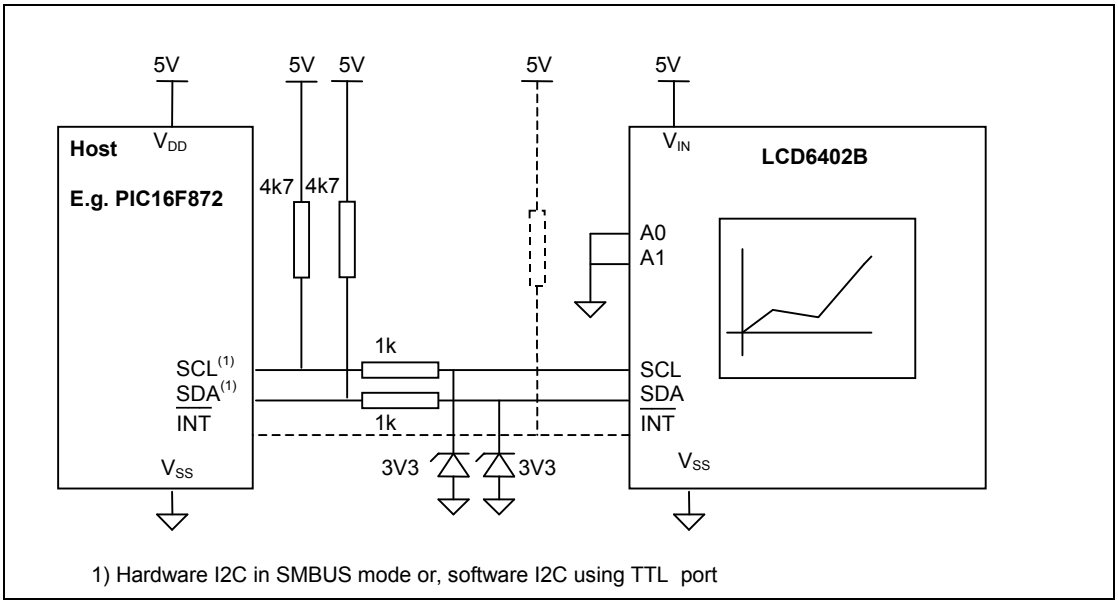


Figure 5 Application using 5V logic

2 CONNECTOR PIN OUT DESCRIPTION

The LCD6402B is available in two connector versions; PIN and IDC. The PIN version has three single row pin headers and the IDC version has one 2-row box style plug suitable for use with an IDC ribbon cable socket.

2.1 CONNECTOR PIN OUT - PIN VERSION

See section 2.3 for pin descriptions.

Table 1 J1A: Power and communications pin out - PIN VERSION

Pin Number	Pin Name	Pin number	Pin Name
1	A0	5	SDA
2	A1	6	SCL
3	$\overline{\text{RES}}$	7	$\overline{\text{INT}}$
4	VSS	8	VIN

Table 2 J2: Keypad interface pin out - PIN VERSION

Pin Number	Pin Name	Pin number	Pin Name
1	R0	5	C0
2	R1	6	C1
3	R2	7	C2
4	R3	8	C3

Table 3 J3: Development port pin out - PIN VERSION

Pin Number	Pin Name	Pin number	Pin Name
1	TX	3	ATN
2	RX	4	GND

2.2 CONNECTOR PIN OUT - IDC VERSION

See section 2.3 for pin descriptions.

Table 4 J1: Pin out - IDC VERSION

Pin Number	Pin Name	Pin number	
1	A0	2	R0
3	A1	4	R1
5	$\overline{RES}$	6	R2
7	VSS	8	R3
9	SDA	10	C0
11	SCL	12	C1
13	$\overline{INT}$	14	C2
15	VIN	16	C3
17	ATN	18	TX
19	GND	20	RX

2.3 PIN DESCRIPTIONS

Table 5 Pin Descriptions

Pin name	Type	Description
A0(1), A1(1)	I	The A0 and A1 set the group2 code. See Figure 18.
ATN	I	RS232 DTR in. For programming purposes only.
C0	O	Keypad column output 0
C1	O	Keypad column output 1
C2	O	Keypad column output 2
C3	O	Keypad column output 3
GND(3)	I	Input. RS232 common (0V)
$\overline{INT}$(2)	O	The $\overline{INT}$ pin is an active low open-drain configurable port may be used to indicate an event in the LCD6402B or be used as a general purpose port. This pin requires an external pull-up resistor See OPTION register for configuration details.
R0	I	Keypad row select 0
R1	I	Keypad row select 1

Pin name	Type	Description
R2	I	Keypad row select 2
R3	I	Keypad row select 3
$\overline{RES}$	I/O	Active low hardware reset input. This input is internally tied to V_{DD} by a 10k resister and can be left unconnected. Note that this input is driven low when ATN >1.2V
RX	I	RS232 receive in. For programming purposes only.
SDA(1)	I/O	I2C serial data. SDA is a bi-directional pin used to transfer data into and out of the display driver. It is an open drain output and may be wire-ORed with any number of open drain or open collector outputs. An open drain output requires the use of a pull-up resistor.
SCL(1)	I/O	I2C serial clock. The SCL input is used to clock all data into and out of the display driver.
TX	O	RS232 transmit out. For programming purposes only.
VIN	I	Positive supply (+5V)
VSS	I	Negative supply (0V)

1) 3.3V logic levels
2) Refer to section 10 CHARACTERISTICS
3) Internally tied to V_{SS}

3 REGISTER ORGANISATION

The LCD6402B is controlled by reading and writing a number of internal registers. See Table 6 Register map.

Each register can be read and written via the I2C bus. See section 9 for more information.

Table 6 Register map

Register	Name
00H	DATA0
01H..	DATA1..
	32 data registers
1FH	DATA31
20H	CMD
21H	STATUS
22H	NEWKEY
23H	KEY
24H	CUX
25H	CUY

Register	Name
26H	ORX
27H	ORY
28H	NVML
29H	NVMH
2AH	OPTION
2BH	INS
2CH	CONTRAST
2DH	FGCLR
2EH	BGCLR
2FH	MODE
30H	FONT

3.1 DATA Registers (00H-1FH)

There are 32 read/write data registers. The data registers are used for passing parameters for the command in the CMD register (20H).

3.2 CMD Register (20H)

This read/write register is used to initiate a command by writing a valid command value to it. During the execution of the command reading, this register will return the command value. When execution is complete, CMD is automatically cleared to zero. See section 4 for details of valid commands.

3.3 STATUS Register (21H)

The status register shows the state of the LCD6402B.

Bit	7	6	5	4	3	2	1	0
	BSY	0	0	0	0	0	0	KP
Init.	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R/W

KP		**BSY**	
0	No keypad event	0	Not busy
1	Keypad key pressed*	1	Busy

* *Cleared by writing 0 to the STATUS register*

3.4 NEWKEY Register (22H)

The NEWKEY register contains the scan value of the most recently pressed keypad key, if fitted. Once NEWKEY has been read, it is cleared automatically to zero.

Reading the NEWKEY register also clears the KP event bit in the STATUS register.

3.5 KEY Register (23H)

The KEY register contains the scan value of the currently pressed key. If no key is pressed, the value is set to zero.

3.6 CUX Register (24H)

Read/write register containing the position of the graphics cursor for the x-axis.

3.7 CUY Register (25H)

Read/write register containing the position of the graphics cursor for the y-axis.

3.8 ORX Register (26H)

Read/write register containing the position of the graphics origin for the x-axis.

3.9 ORY Register (27H)

Read/write register containing the position of the graphics origin for the y-axis.

3.10 NVML Register (28H)

The NVL is used with the NVH register to form the lower 8-bits of the non-volatile memory pointer.

3.11 NVMH Register (29H)

The NVH is used with the NVL register to form the upper 8-bits of the non-volatile memory pointer.

3.12 OPTION Register (2AH)

The OPTION register is a read and write register, containing various control bits to configure the $\overline{INT}$ pin.

Bit	7	6	5	4	3	2	1	0
	IC3	IC2	IC1	IC0	-	-	-	GPP
Init.	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

GPP General purpose output port

When IC0:IC3 = 0000 the GPP bit is written to the $\overline{INT}$ pin as follows:

0 $\overline{INT}$ pin high Z

1 $\overline{INT}$ pin driven low

IC3:IC0 $\overline{INT}$ output configuration bits

0000 Use the $\overline{INT}$ pin to output the GPP bit of the OPTION register

0010 Use the $\overline{INT}$ pin to output the BSY bit of the STATUS register

1000 Use the $\overline{INT}$ pin to output the KP bit of the STATUS register

1001 The $\overline{INT}$ pin is driven low* for any change in the STATUS register

* *Reset by writing zero to the STATUS register*

3.13 INS Register (2BH)

The INS (graphic instruction) register executes graphic instruction written to it. See Section 5 for more information on graphic instructions.

3.14 CONTRAST Register (2CH)

The CONTRAST register controls the LCD contrast and is effectively non-volatile. CONTRAST can have any value in the range 0 to 63. The default value is the midrange value, 32. Setting the contrast has immediate effect and

is saved to an internal non-volatile register. The same value may be written to the CONTRAST register an indefinite number of times. If a different value is used on each subsequent write the number of writes is limited to 1,000,000 cycles.

3.15 FGCLR Register (2DH)

The value in the FGCLR register determines the foreground colour as follows:

FFH White

00H Black (Init. value)

FGCLR is used for the 'pen' colour for all graphic functions.

3.16 BGCLR Register (2EH)

The value in the FGCLR register determines the background colour as follows:

FFH White (Init. value)

00H Black

BGCLR is used for the 'paper' colour for raster based functions.

3.17 MODE Register (2FH)

The colour mode register.

Read/write register containing the colour mode flags.

Bit	7	6	5	4	3	2	1	0
	-	-	BG1	BG0	-	-	FG1	FG0
Init.	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

FG0		**BG0**	
0	OR foreground colour	0	OR background colour
1	XOR foreground colour	1	XOR background colour
FG1		**BG1**	
0	Foreground colour on	0	Background colour on
1	Foreground colour off	1	Background colour off

3.18 FONT Register (2EH)

Read/write register containing the current font type. See 5.4.

4 COMMANDS

Commands are written to register CMD (20H). Commands may only be written to this register when the BSY flag is clear in the STATUS register with the exception of command 91H.

4.1 COMMAND SUMMARY

Table 7 Command summary

Command	Description
00H-7FH	Recall graphic object number from NVM
90H	Reserved
91H	Reset graphic interpreter
92H	Back light on
93H	Back light off
94H	Clear screen
A0H	Reserved
A1H	Write NVM 32-byte page from data register
A2H	Reserved
A3H	Read NVM 32-byte page to data registers
A8H	Save context
A9H	Restore context
C0H	Power down
C1H	Power up
C2H	Standby
FFH	Reset

4.2 COMMAND DESCRIPTIONS

4.2.1 Commands 00H-7FH: Recall graphic object number from NVM

Commands in the range 00H to 7FH directly recall graphic objects 0 to 127 from the on board non-volatile memory (NVMEM) respectfully.

For example writing 08H to the command register will retrieve the graphic instructions for object 8 from the NVMEM and pass them to the graphic interpreter.

4.2.2 Command 91H: Reset graphic interpreter

Writing 91H to the command register resets the graphics interpreter. This command has immediate affect regardless of the BSY flag state.

4.2.3 Command 92H: Back light on

Switch the LCD backlight on.

4.2.4 Command 93H: Back light off

Switch the LCD backlight off

4.2.5 Command 94H: Clear screen

Clears the LCD screen. Resets CUX, CUY, ORX, ORY, & MODE registers to zero.

4.2.6 Command A1H: Write NVM 32-byte page from data register

Writes 32-bytes of data* held in DATA0-31 register (00H-1FH) to non-volatile memory location specified by the NVM address registers, NVML and NVMH. This command auto increments the NVM address registers by 32.

* *The NVM can only be written with J7 unlinked.*

4.2.7 Command A3H: Read NVM 32-byte page to data registers

Reads 32-bytes from the non-volatile memory location specified by the NVM address registers, NVML and NVMH and returns it in data registers DATA0-31 (00H-1FH). This command auto increments the NVM address registers by 32.

4.2.8 Command A8H: Save Context

Copies the contents of the following registers to the context store:

CUX, CUY, ORX, ORY, FGCLR, BGCLR, MODE, FONT

4.2.9 Command A9H: Restore Context

Copies the contents of the context store to the following registers:

CUX, CUY, ORX, ORY, FGCLR, BGCLR, MODE, FONT

4.2.10 Command C0H: Power down

Powers down the LCD and places the LCD6402B in standby mode. The LCD screen remains blank in power down mode. Use the power up command C1H to return from power down mode. Power down mode draws the lowest current I_{PD} of all operating modes.

4.2.11 Command C1H: Power up

Powers up the LCD and restores the LCD screen. Use this command to return from power down mode.

4.2.12 Command C2H: Standby

Places the LCD6402B in standby mode. Standby mode is returned from by any of the following:

i) Keypad activity

ii) I2C bus activity

Standby mode draws less current (I_{SB}) than during normal operation but more than power down mode. The graphics remain visible in standby mode.

4.2.13 Command FFH: Reset

Resets the LCD6402B by performing a power on reset.

5 GRAPHICS

Graphic functions are performed by writing directly to the INS(2BH)register. Before a graphics instruction can be executed, a busy check must be made (see section 8).

Graphic instructions are executed serially by the graphics interpreter. If the interpreter executes an invalid instruction, an illegal state may occur where the execution process never completes. If this occurs when recalling graphics from the non-volatile memory, the interpreter is automatically reset by the graphics watchdog. The interpreter may also be reset by writing the reset interpreter command (A1H) to the command register.

Example: Draw line from co-ordinate 12,6 to 56,33

The instruction codes shown in Table 8 below are written to the INS(2BH) register. In general a busy check is required before commencing a new instruction (see section 8 for more details on busy check).

Table 8 Example: Draw line

Instructions	Description
-	Perform busy check
3AH	Move cursor to:
0CH	X=12
06H	Y=6
-	Perform busy check
52H	Draw line to:
38H	X=56
21H	Y=33

5.1 GRAPHIC ORIGIN

All graphic entities (lines, circles fonts etc) are drawn relative to the graphics origin. By default, the graphics origin is set to the top left of the display area. See Figure 6.

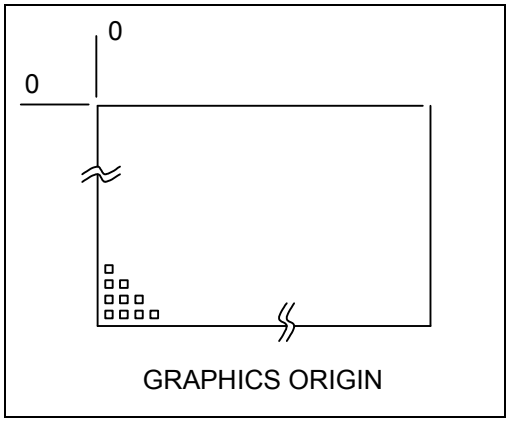


Figure 6 Graphics origin

The graphics origin may be moved to a different location by writing to the ORY and ORX registers.

5.2 GRAPHICS INSTRUCTION SET SUMMARY

Instruction	Name	Operation
08H	gRESET	Reset display
21H, 01H	gCLS	Clear screen
2AH, x, y	gORIGIN	Set display origin
3AH, x, y	gCURSOR	Move cursor to x, y
41H, clr	gFGCLR	Set fore ground colour to clr
42H, clr	gBGCLR	Set background colour to clr
43H, n	gMODE	Set colour mode.
4AH, x, y, n	gPIX	Set pixel at x, y to FGCLR (n=1) or BGCLR (n=0)
52H, x, y	gLINETO	Draw line to x, y
5CH, x0, y0, x1, y1	gRECT	Draw rectangle with corners x0, y0 and x1, y1
64H, x0, y0, x1, y1	gRECTF	Draw filled rectangle with corners x0, y0 and x1, y1
69H, f	gFONT	Select font type f
71H, c	gPUTC(1)	Print character c
79H, r	gHRAST(1)	Print horizontal raster byte r
80H, c0..cn, 0	gPUTS(1)	Print null terminated string c0 to cn
8AH, w/8, h, rdat…	gHBMP(1)	Print bit map using horizontal raster data (rdat) of w pixels wide by h pixels high
99H, r	gCIRCLE	Draw circle with radius r
9AH, r	gCIRCLEF	Draw filled circle with radius r
A1H, r	gVRAST(1)	Print vertical raster byte r
AAH, w, h/8, rdat…	gVBMP(1)	Print bit map using vertical raster data (rdat) of w pixels wide by h pixels high
F1H, FDH, n	gCALLI(2)	Recall object by index

1) Raster based function.

2) Instruction may use raster based functions.

5.3 GRAPHICS INSTRUCTION SET DESCRIPTION

5.3.1 gRESET

Reset display.

08H

5.3.2 gCLS

Clear graphics screen and reset CUX, CUY, ORX, ORY & MODE to zero.

21H	01H

5.3.3 gCURSOR

Move graphics cursor. Loads the CUX & CUY registers with x & y respectively.

3AH	x	y

x : x cursor position

y : y cursor position

5.3.4 gFGCLR

Set foreground colour.

Loads the FGCLR register with clr. See section 3.15

41H	clr

5.3.5 gBGCLR

Set background colour.

Loads the BGCLR register with clr. See section 3.15

41H	clr

5.3.6 gMODE

Set colour mode.

Loads the MODE register with n. See section 3.15

41H	N

5.3.7 gPIX

Set pixel to current colour at location x, y.

4AH	x	y	n

x : x co-ordinate

y : y co-ordinate

n : 0=background colour,

: 1=foreground colour

5.3.8 gLINETO

Draw line from current cursor to location x, y.

52H	X	y

x : x co-ordinate

y : y co-ordinate

5.3.9 gRECT

Draw rectangle from corner x0, y0 to corner x1, y1.

5CH	x0	y0	x1	y1

x0 : left

y0 : bottom

x1 : top

y1 : right

5.3.10 gRECTF

Draw filled rectangle from corner x0, y1 to corner x1, y2.

64H	x0	y0	x1	y2

x0 : left

y0 : bottom

x1 : top

y1 : right

5.3.11 gFONT

Select font type. Font types 0,1 and 2 select built-in fonts. Font types 8 to 15 select custom fonts stored in non-volatile memory.

69H	f

f=0 : 3x5 (WxH)

f=1 : 5x8 (WxH)

f=2 : 5x16 (WxH)

f=8 - 15 : Custom fonts

5.3.12 gPUTC

Print character on display. Increments CUX by the font width.

71H	C

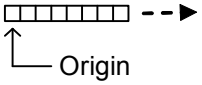
c : ASCII character

5.3.13 gHRAST8

Print 8-bit horizontal raster pattern at current graphics cursor.

79H	r

r : 8-bit raster. MSB is left most pixel and origin. CUX is auto incremented by 8.

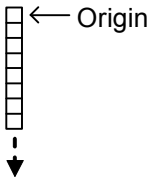


5.3.14 gVRAST8

Print 8-bit horizontal raster pattern at current graphics cursor.

A1H	r

r : 8-bit raster. MSB is top most pixel and origin. CUY is auto incremented by 8.



5.3.15 gPUTS

Print ASCII null terminated string starting at current cursor position. Increments CUX by the line width.

80H	c0	…	cn	00H

5.3.16 gHBMP

Print bit map using horizontal raster data with top left origin at current

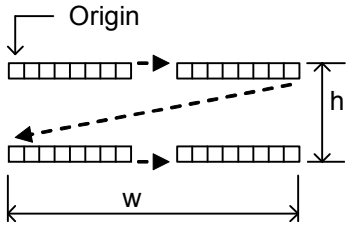
cursor. CUX is auto incremented by the value of w.

88H	w/8	h	rdat	…

w/8 : width of bitmap in pixels/8

h : height of bit map in pixels

rdat : (w . h) rdat bytes



5.3.17 gCIRCLE

Draw circle with centre at current graphics cursor using radius r.

99H	r

5.3.18 gORIGIN

Move graphics origin. ORX & ORY registers with x & y respectively.

2AH	x	y

x : x origin position

y : y origin position

5.3.19 gVBMP

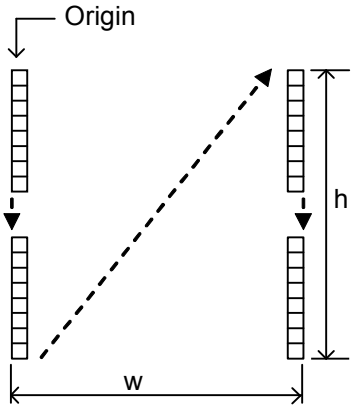
Print bit map using vertical raster data with bottom left origin at current. CUX is auto incremented by the value of w.

AAH	w	h/8	rdat	…

w : width of bitmap in pixels/8

h/8 : height of bit map in pixels

rdat : (w * h/8) rasta bytes



5.3.20 gCALLI

Recall graphic object by index from NVM. Calls may not be nested or re-cursed.

F0H	FDH	i

i : 00H ~ 7FH

5.4 GRAPHIC CHARACTER FONTS

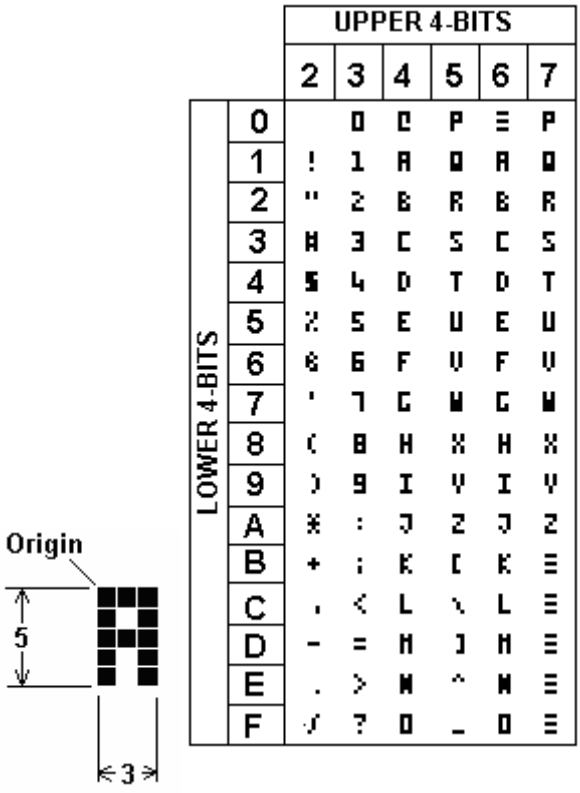
There are three built-in font types:

- type 0 (3x5),
- type 1 (5x8)
- type 2 (5x16)*

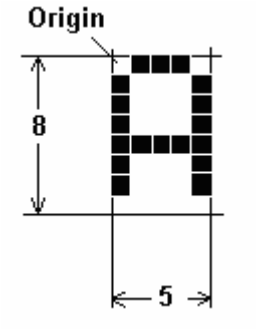
** Font type 2 is a scaled version of font type 1.*

In addition to these three font types there may also be up to eight customised fonts. See section 13 for more details on custom fonts. Fonts are selected by writing the appropriate value to the FONT register or by using the gFONT graphic instruction.

5.4.1 Font 0 (3x5) Character set

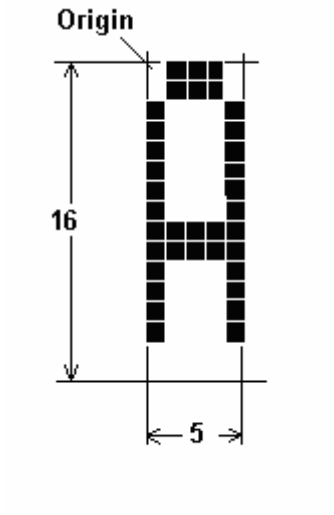


5.4.2 Font 1 (5x8) Character set



LOWER 4-BITS	UPPER 4-BITS					
	2	3	4	5	6	7
0		0	@	P	`	p
1	!	1	A	Q	a	q
2	"	2	B	R	b	r
3	#	3	C	S	c	s
4	$	4	D	T	d	t
5	%	5	E	U	e	u
6	&	6	F	V	f	v
7	'	7	G	W	g	w
8	(	8	H	X	h	x
9	)	9	I	Y	i	y
A	*	:	J	Z	j	z
B	+	;	K	[	k	{
C	,	<	L	\	l	|
D	-	=	M	]	m	}
E	.	>	N	^	n	~
F	/	?	O	_	o	⌂

5.4.3 Font 2 (5x16) Character set



LOWER 4-BITS	UPPER 4-BITS 2	3	4	5	6	7
0		0	@	P	`	p
1	!	1	A	Q	a	q
2	"	2	B	R	b	r
3	#	3	C	S	c	s
4	$	4	D	T	d	t
5	%	5	E	U	e	u
6	&	6	F	V	f	v
7	'	7	G	W	g	w
8	(	8	H	X	h	x
9	)	9	I	Y	i	y
A	*	:	J	Z	j	z
B	+	;	K	[	k	{
C	,	<	L	\	l	|
D	-	=	M	]	m	}
E	.	>	N	^	n	~
F	/	?	O	_	o	⌂

5.5 GRAPHIC PERFORMANCE

The LCD6402B offers good performance since graphics can be rendered in a way that only updates the required areas of the display. The refresh time therefore depends on the complexity of the graphics being rendered.

The actual performance can be quantified by the pixel render rate (PR) expressed in units of pixels per second. Table 9 below shows PR values measured for various graphic operations.

For example, it will take approximately 0.2 seconds to render a bitmap image 64x64 pixels in size (i.e. 64x64/20000).

Table 9 Typical pixel render rates

Operation	Pixels/second
Line, Circle, Text, Bitmap(1), Rectangle	20000
Clear screen	500000

Notes

1. Typical figure when recalling from EEPROM

6 NON-VOLATILE MEMORY

The non-volatile memory (NVM) consists of a 32768-byte serial EEPROM. The NVM is modified by reading and writing the data and control registers or by using LCDLAB (see section 13 for more details) and serial programming lead. EEPROM data is read or written 32-bytes at a time.

6.1 WRITE PROTECT

The NVM is write protected by making a link across J7.

6.2 PROGRAMMING

The NVM may be programmed via the I2C bus for general data storage. Allowance must be made if the NVM is also programmed by LCDLAB. In general LCDLAB uses the NVM from address 0000H upwards, and programs by overwriting data without performing a full erase. If the NVM is not fully used by LCDLAB there will be free space available for general data storage.

6.3 WRITE 32-BYTE PAGE

To write a 32-byte page of data from the data registers (DATA0 to DATA31) to the NVM requires the following steps:

1. Set the NVM address by writing the low and high address to the NVML and NVMH registers respectively.
2. Write the desired data to registers DATA0 to DATA31
3. Write command A1H* to register CMD. The BSY flag in the STATUS register will remain active during the write.

* *The NVM can only be written with J7 unlinked.*

6.4 READ 32-BYTE PAGE

To read a 32-byte page of data from the NVM to the data registers (DATA0-DATA31) requires the following steps:

1. Set the NVM address by writing the low and high address to the NVML and NVMH registers respectively.
2. Write command A3H to register CMD. The BSY flag in the STATUS register will remain active during the read.

Note: The data cannot be read reliable until the busy flag has cleared.

7 KEYPAD INTERFACE

The keypad interface (see section 2 for connector details) allows a 16-key (4x4-matrix) keypad to be connected to the LCD6402B. There may be any number of keys from 1 to 16 (see Figure 7 & Figure 8). Each key has a scan number as follows: SW1 = 1, SW2 = 2 etc. The scan number zero represents no key pressed. Switches SW13, SW14, SW15 share the optional on board user keys.

Figure 7 Example 16-key arrangement

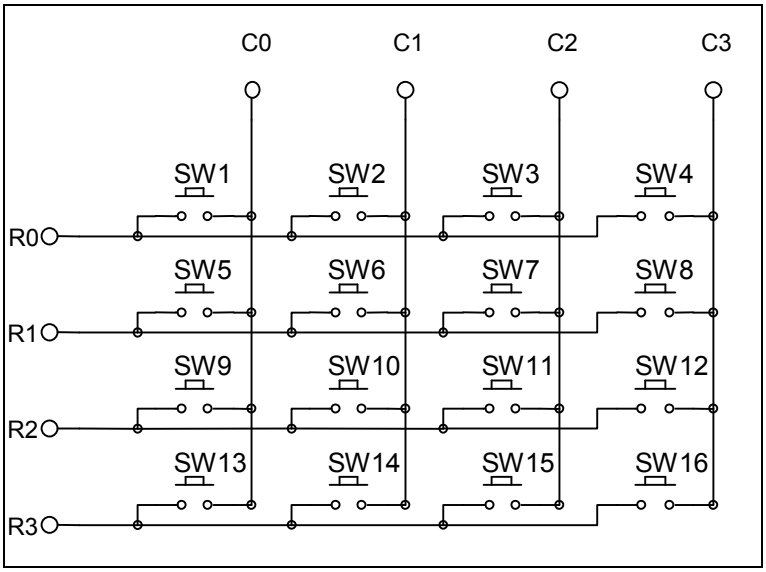
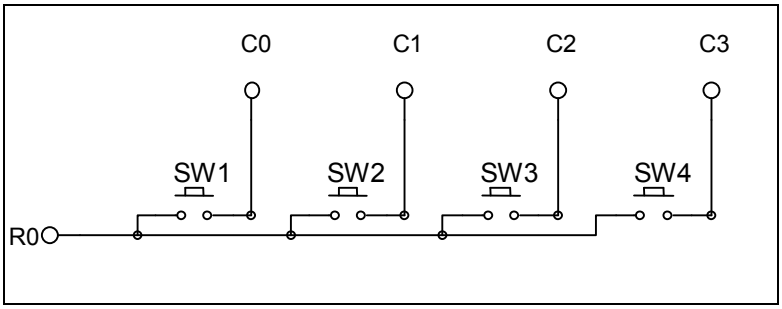


Figure 8 Example 4-key arrangement



There are three methods for reading the keypad

i) Host reads the NEWKEY or KEY registers at regular intervals until a non-zero value is read (i.e. the scan code).

ii) Host controller polls the $\overline{INT}$ pin* and when low, reads the NEWKEY or KEY register.

iii) Host controller is interrupted by $\overline{INT}$ pin* on its falling edge and reads NEWKEY or KEY register.

** See OPTION register for more details.*

Methods ii) and iii) require writing zero to the STATUS register to reset the $\overline{\text{INT}}$ pin to its inactive high state.

8 BUSY CHECK

Determining the busy state of the LCD6402B is required before executing a command or graphic instruction. In general, the busy flag only needs monitoring before the execution of a command or before the start of a new instruction*. Modifying the data or control registers while the LCD6402B is busy can result in unpredictable behaviour. The busy state is determined by either reading the STATUS register or using the $\overline{\text{INT}}$ pin.

* *With the exception of gPUTS*

8.1 STATUS REGISTER POLLING

The STATUS register is polled prior to executing a command until BSY goes low. This is the simplest method of determining the busy state but may have the disadvantage of using significant host CPU and I2C bus time. See Figure 9.

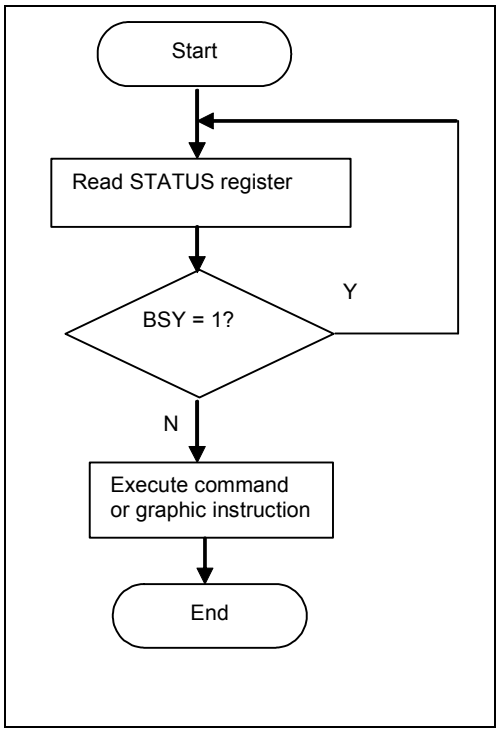


Figure 9 Status register polling

8.2 $\overline{\text{INT}}$ PIN POLLING

The $\overline{\text{INT}}$ pin can be configured to output the BSY flag of the STATUS register by setting IC0:IC3 to '2' in the OPTION register. Once configured the host controller reads the $\overline{\text{INT}}$ pin to determine the busy state of the LCD6402B. See Figure 10

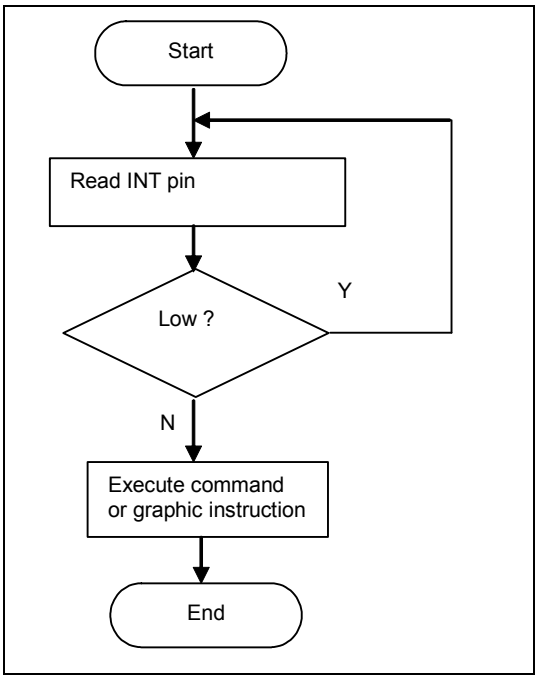


Figure 10 $\overline{\text{INT}}$ pin monitoring

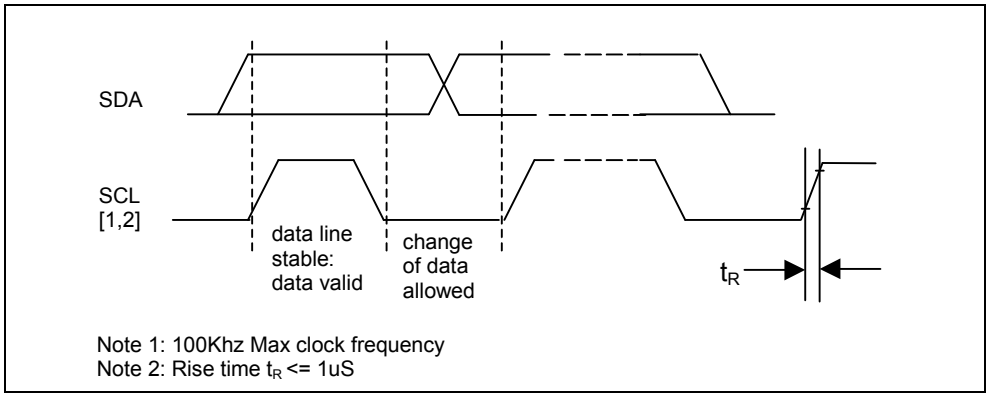
9 CHARACTERISTICS OF I2C BUS

The I2C-bus is for bi-directional, two-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor. Data transfer may be initiated only when the bus is not busy.

9.1 BIT TRANSFER

One data bit is transferred during each clock pulse (see Figure 11). The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as a control signal.

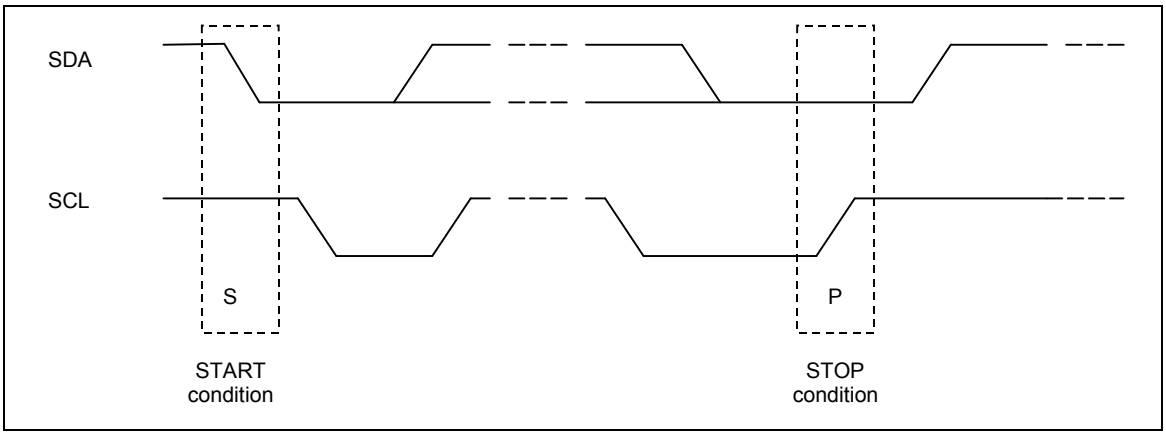
Figure 11 Bit transfer and clock timing



9.2 START AND STOP CONDITIONS

Both data and clock lines remain HIGH when the bus is not busy (see Figure 12). A HIGH-to-LOW transition of the data line while the clock is HIGH is defined as the start condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the stop condition (P).

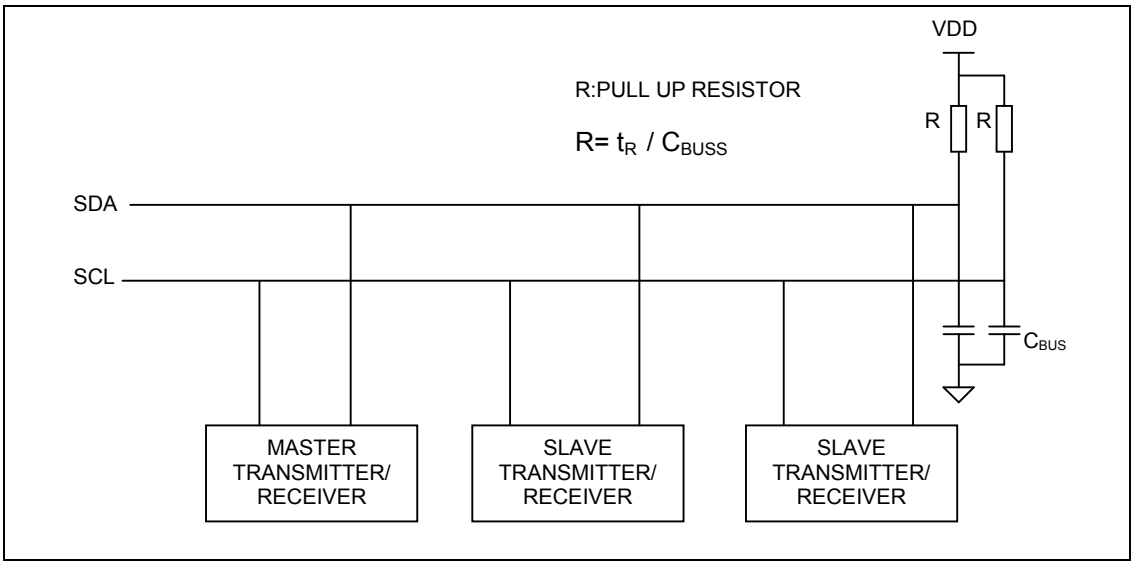
Figure 12 Definition of start and stop conditions



9.3 SYSTEM CONFIGURATION

A device generating a message is known as the 'transmitter'; a device receiving a message is known as the 'receiver'. The device that controls the message is the 'master' and the devices which are controlled by the master are the 'slaves'.

Figure 13 System configuration

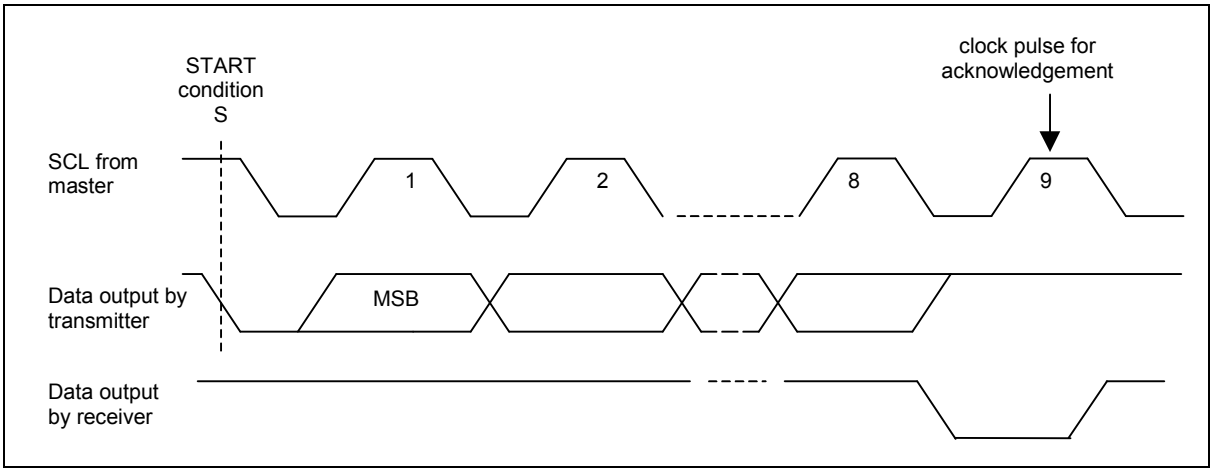


9.4 ACKNOWLEDGE

The number of data bytes transferred between the start and stop conditions from transmitter to receiver is unlimited. Each byte of eight bits is followed by an acknowledge bit (see Figure 14). The acknowledge bit is a HIGH level signal put on the bus by the transmitter during which time the master generates an extra acknowledge-related clock pulse. A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also, a master receiver must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter.

The device that acknowledges must pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge-related clock pulse (set-up and hold times must be taken into consideration). A master receiver must signal an end of data to the transmitter by not generating an acknowledge on the last byte that has been clocked out of the slave. In this event the transmitter must leave the data line HIGH to enable the master to generate a stop condition.

Figure 14 Acknowledge on the I2C-bus



9.5 I2C-BUS PROTOCOL

Before any data is transmitted on the I2C-bus, the device which should respond is addressed first. The addressing is always carried out with the first byte transmitted after the start procedure. The I2C-bus configuration for the WRITE and READ cycles are shown in Figure 15, Figure 16 & Figure 17.

Figure 15 Master transmits to slave receiver (WRITE) mode

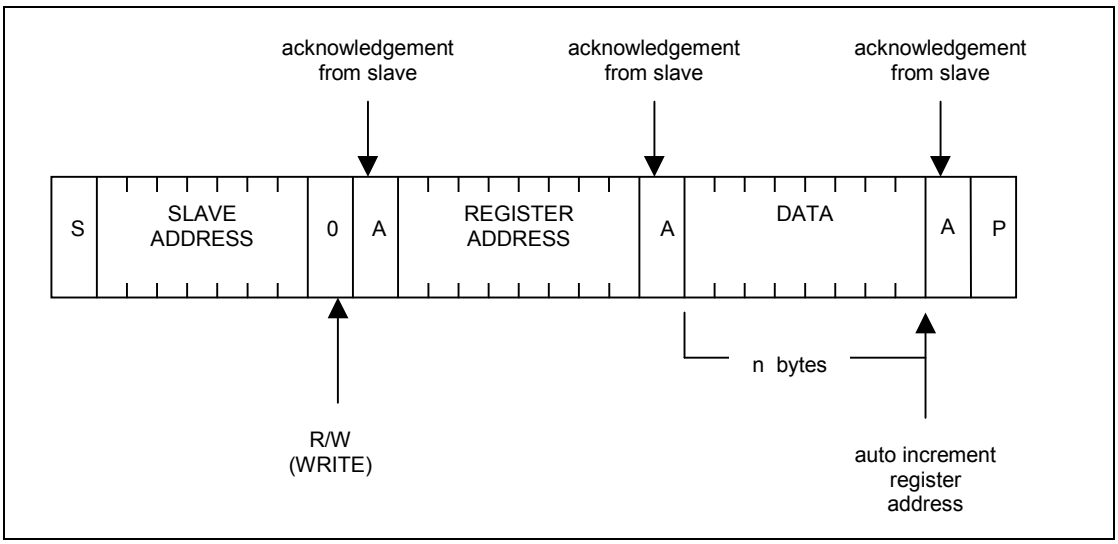


Figure 16 Master reads after setting register address (WRITE register address; READ data)

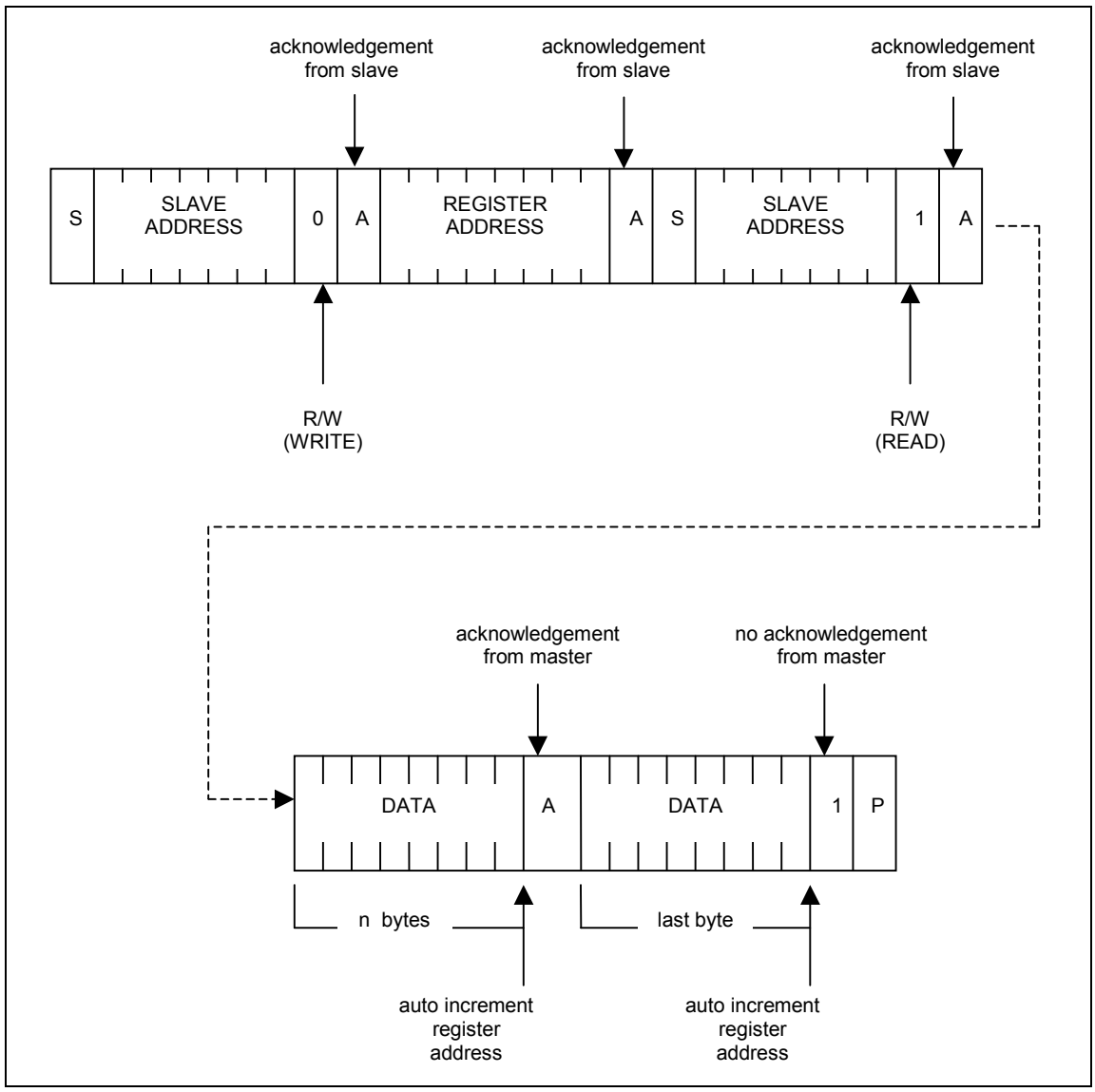
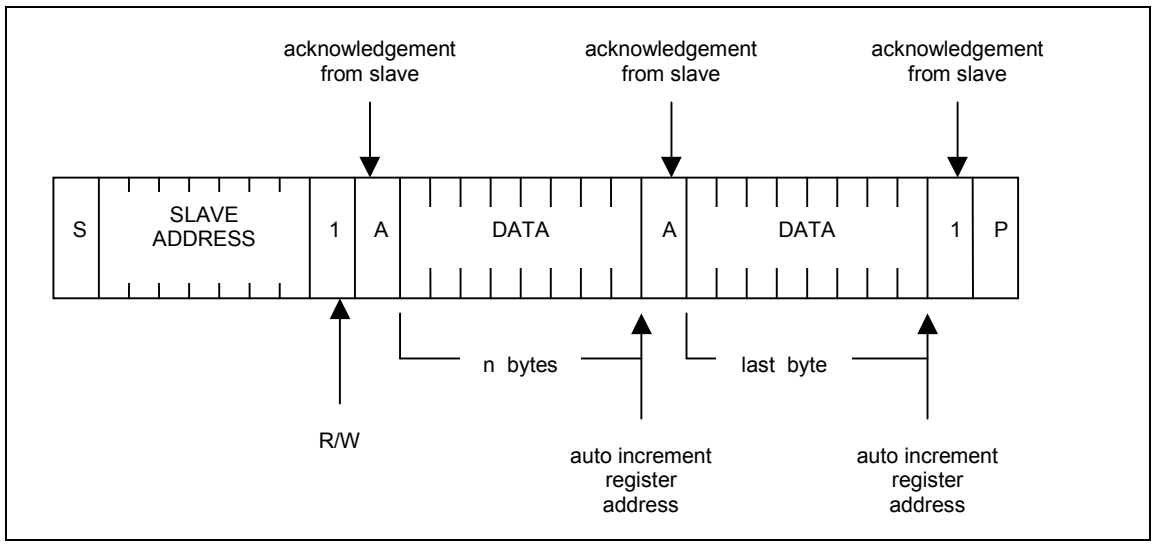


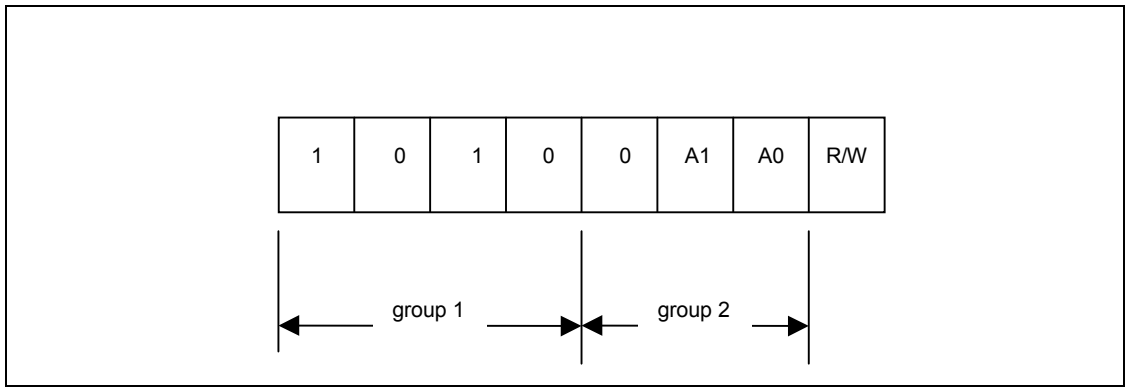
Figure 17 Master reads slave immediately after first byte (READ mode)



9.6 I2C SLAVE ADDRESS

The LCD6402B has a fixed combination 1 0 1 0 as group 1, while group 2 is programmable according to A0, and A1 (J1) hardware inputs (see Figure 18).

Figure 18 Slave address



10 CHARACTERISTICS

10.1 ELECTRICAL CHARACTERISTICS

Table 10 DC Characteristics

V_{IN} = 5V (unless stated otherwise), Internal V_{DD}=3.3V, V_{SS} = 0 V

Sym.	Parameter	Conditions	Min.	Typ.	Max.	Unit
V_{IN}	Supply voltage		3.3(1)	5	5.5	V
V_{DD}	Internal logic voltage			3.3		V
I_{DD}	Supply current	Backlight off		3.25		mA
I_{SB}	Standby current	Standby enabled, no activity on I2C bus.		455		uA
I_{PD}	Power down current	Power down enabled no activity on I2C bus. Display blank.		105		uA
I_{BL}	Backlight current	Backlight on		34		mA
SDA, SCL						
V_{IL}	Low level input voltage		V_{SS}		$0.3V_{DD}$	V
V_{IH}	High level input voltage		$0.7V_{DD}$	-	V_{DD}	V
I_{OL}	Low level output current					mA
$\overline{INT}$						
V_{OH}	High level output	Open drain	-	-	8.5	V
V_{OL}	Low level ouput	I_{OL}=8.5mA	V_{SS}	-	0.6	V
A0, A1						
V_{IL}	Low level input voltage		V_{SS}		$0.3V_{DD}$	V
V_{IH}	High level input voltage		$0.7V_{DD}$	-	V_{DD}	V

Sym.	Parameter	Conditions	Min.	Typ.	Max.	Unit
$\overline{RES}$						
V_{IL}	Low level input voltage	Internal 10K pull up(2)	V_{SS}		$0.3V_{DD}$	V
V_{IH}	High level input voltage	Internal 10K pull up(2)	$0.7V_{DD}$	-	V_{DD}	V

Notes:

1) Internal V_{DD} reduces to 3V. Backlight becomes inoperative.
2) This input is shared with the ATN input of the development port.

Table 11 AC Characteristics

V_{IN} = 5V (unless stated otherwise), Internal V_{DD}=3.3V, V_{SS} = 0 V

Sym.	Description	Min.	Typ.	Max.	Unit
F_{CLK}	I2C serial clock input (SCL)	-	-	100	kHz

10.2 LIQUID CRYSTAL DISPLAY CHARACTERISTICS

Table 12 LCD Mechanical Characteristics

Parameter	Specification	Unit
Viewing area	50.60(W) x 31.0(H)	MM
Display format	128(W) x 64(H)	Dots
Dot size	0.349(W) x 0.418(H)	MM
Dot pitch	0.364(W) x 0.433(H)	MM

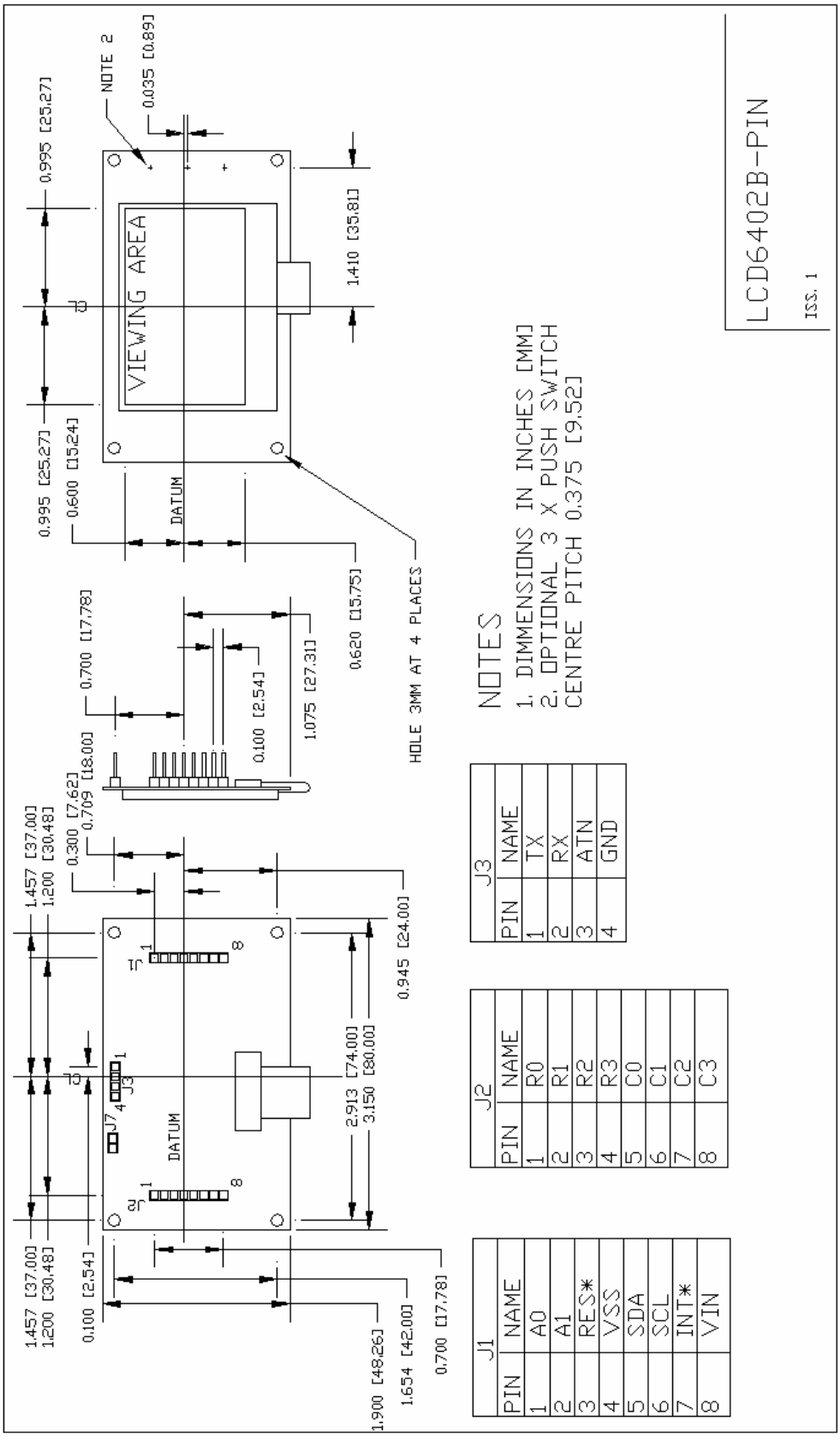
Table 13 LCD Environmental Condition

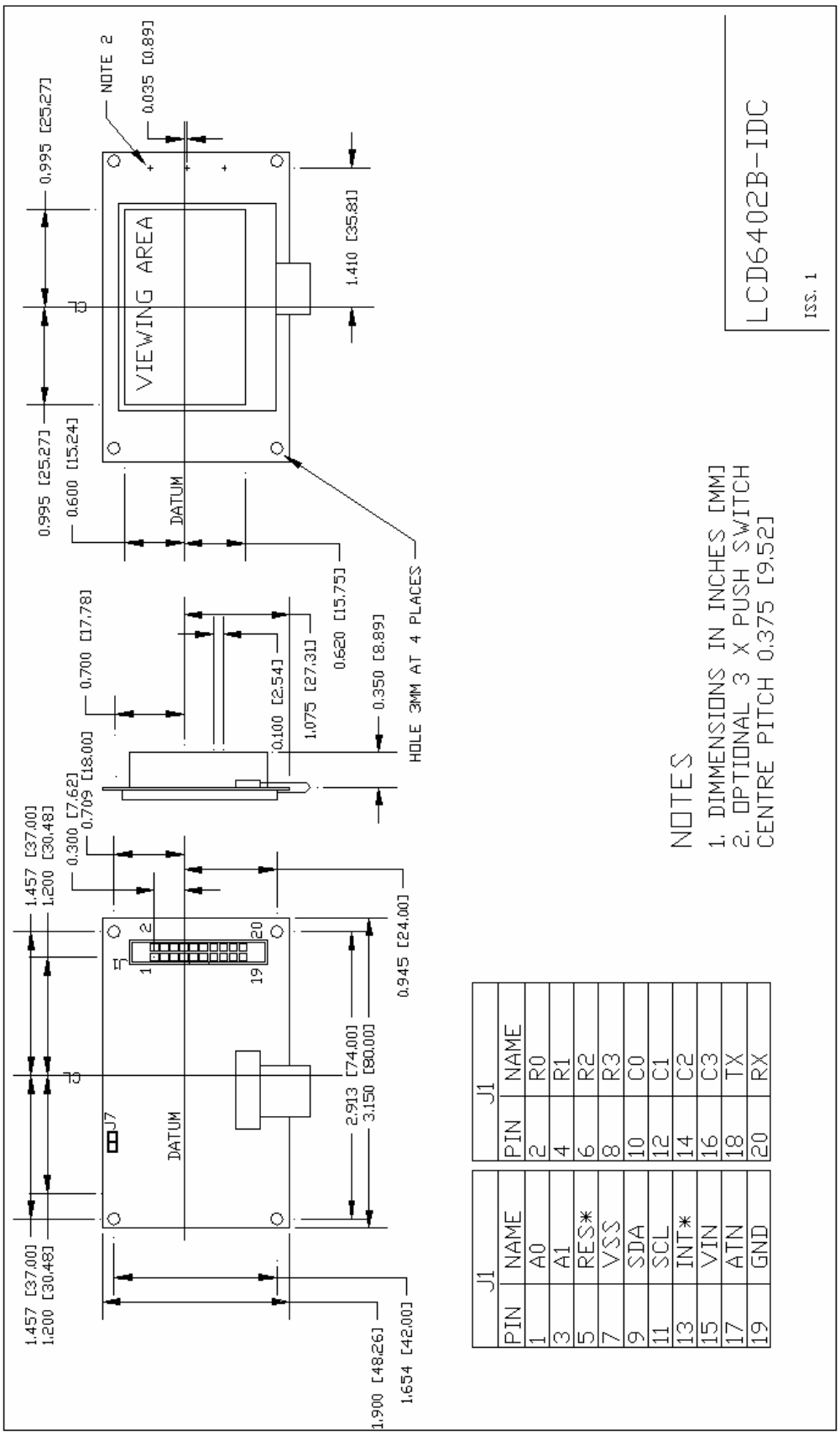
Item	Operating temp. (^{O}C)		Storage temp. (^{O}C)		Remark
	Min	Max	Min	Max	
Ambient temperature	0	+50	-20	+65	Dry
Humidity	95% max RH for Ta ≤ 40 ^{O}C <95% RH for Ta >40 ^{O}C				No condensation
Vibration (IEC 68-2-6)	Freq. 10 ~ 55Hz Amp. 0.75mm Dura. 20 cycles each direction				3 directions
Shock (IEC 68-2-27) Half-sine pulse shape	Pulse duration : 11 ms Peak acceleration: 981 m/s$_2$ = 100g No. shocks : 3 shocks in 3 mutually perpendicular axes.				3 directions

Table 14 LCD Optical Characteristics

Parameter	Specification
Colour	FSTN Positive Black & White transflective
Backlight colour	White LED
Backlight luminance	200 cd/m^2 (typical)
Viewing angle	12 o'clock direction.
Internal LCD bias temperature gradient	-0.05% / ^{O}C

11 MECHANICAL DETAILS

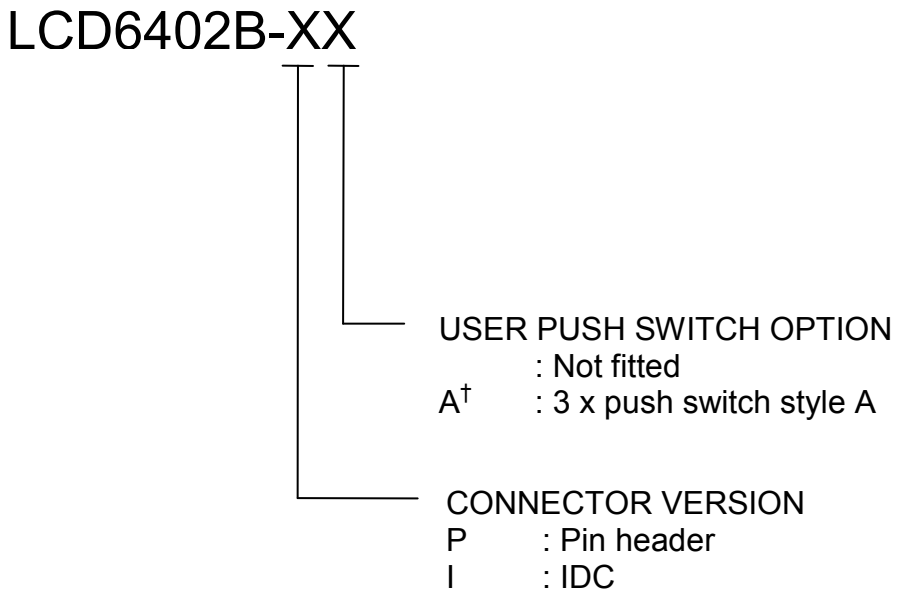




J1	
PIN	NAME
1	A0
3	A1
5	RES*
7	VSS
9	SDA
11	SCL
13	INT*
15	VIN
17	ATN
19	GND

J1	
PIN	NAME
2	R0
4	R1
6	R2
8	R3
10	C0
12	C1
14	C2
16	C3
18	TX
20	RX

12 PRODUCT CODE INFORMATION



† See Figure 1 for switch option A.

13 DEVELOPMENT TOOLS

LCDLAB is a free software tool for use with the LCD6402B that allows ease of development for small graphic LCD operator panels.

When creating an operator interface, one of the problems has always been the arduous task of manually entering the co-ordinates of every single graphics item such as lines, text and bitmaps (bitmaps being almost impossible). This can take a great deal of time and subsequent changes are equally tedious.

LCDLAB goes a long way in solving this problem by taking exported output from drawing packages such as AutoCAD and Windows Paint, and converting them automatically to LCD6402B graphic instructions. The LCD6402B can then be programmed with the graphic instruction by LCDLAB with a simple RS232 connection between the LCD6402B and PC.

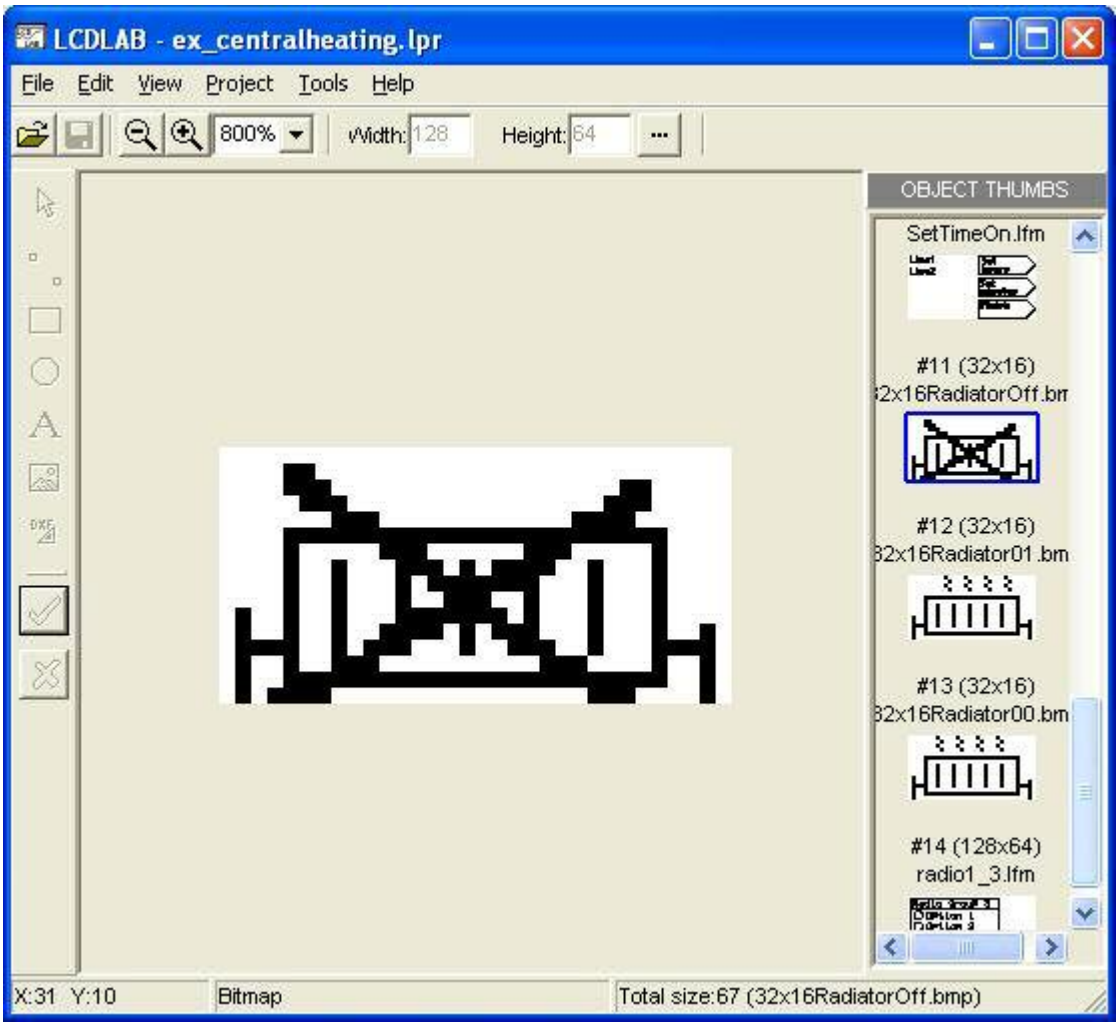


Figure 19 LCDLAB Screen shot

13.1 LCDLAB Font Maker

Font Maker is a powerful tool combined with LCDLAB for capturing, creating and editing fonts for LCD applications from any of those installed in the operating system.

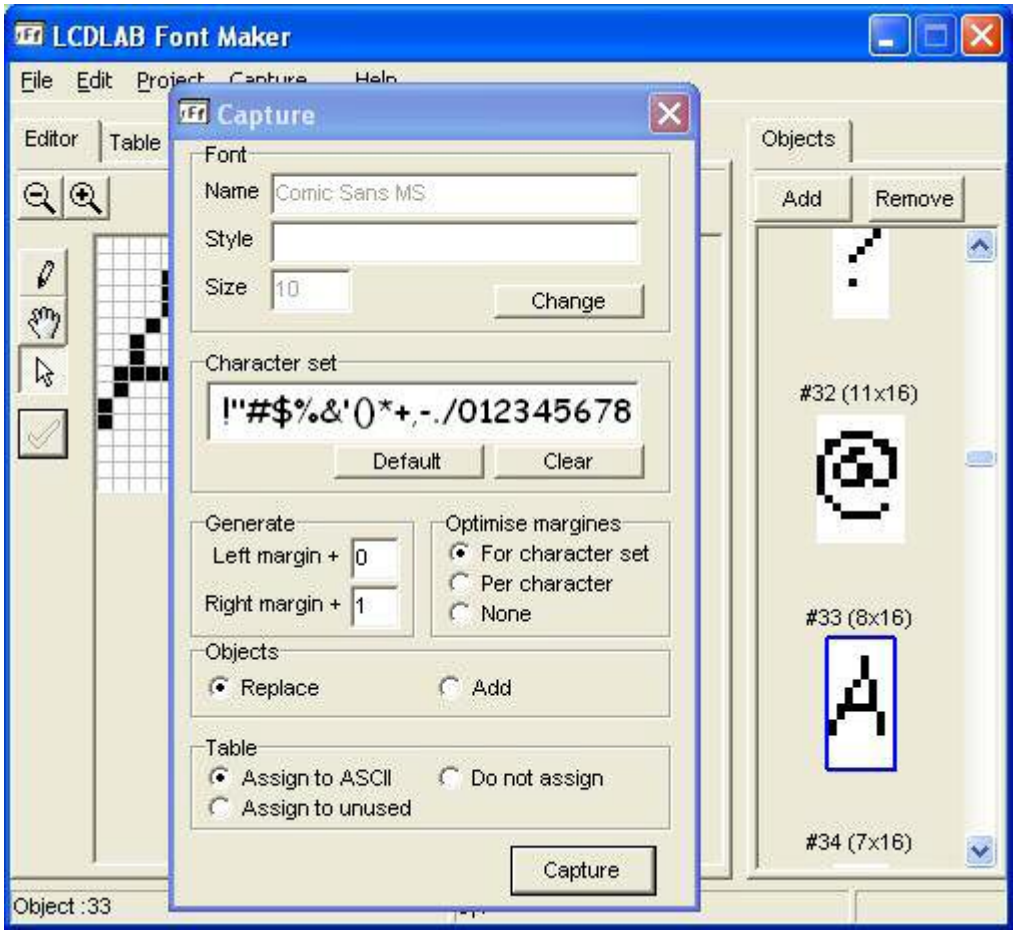


Figure 20 Font Maker screen shot

13.2 Development set up

LCDLAB runs under Microsoft Win32 and requires a serial lead connecting the PC com-port to the development port on the LCD6402B (see Figure 21).

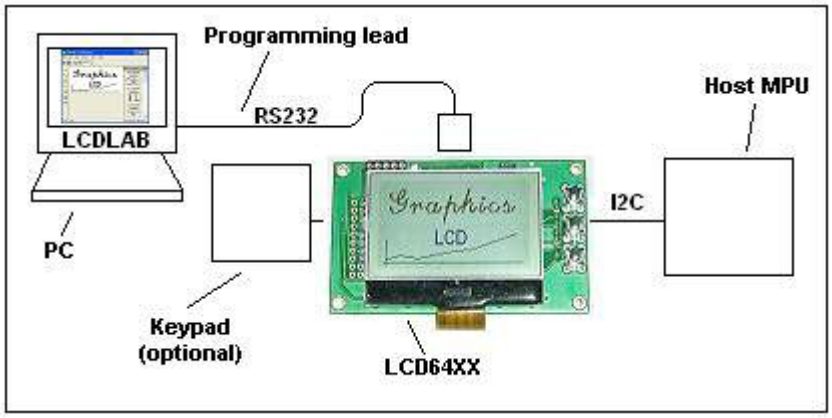


Figure 21 Development arrangement of LCDLAB and LCD6402B

See Table 15 and

Table 16 below for connection details.

Table 15 PC 9-way D connections

PC 9-way 'D'	LCD6402B Pin Name
2 (RX)	TX
3 (TX)	RX
4 (DTR)	ATN
5 (GND)	GND

Table 16 PC 25-way D connections

PC 25-way 'D'	LCD6402B Pin Name
3 (RX)	TX
2 (TX)	RX
20 (DTR)	ATN
7 (GND)	GND

13.3 LCD6402-DEV Development/Evaluation Board

The LCD6402-DEV is a chassis board for evaluating the LCD6402B 128x64 intelligent monochrome graphic LCD module and can also be used for developing simple host applications*. For more details refer to data sheet LCD6402-DEV.PDF

** Requires the addition of a suitable compiler and ICD.*

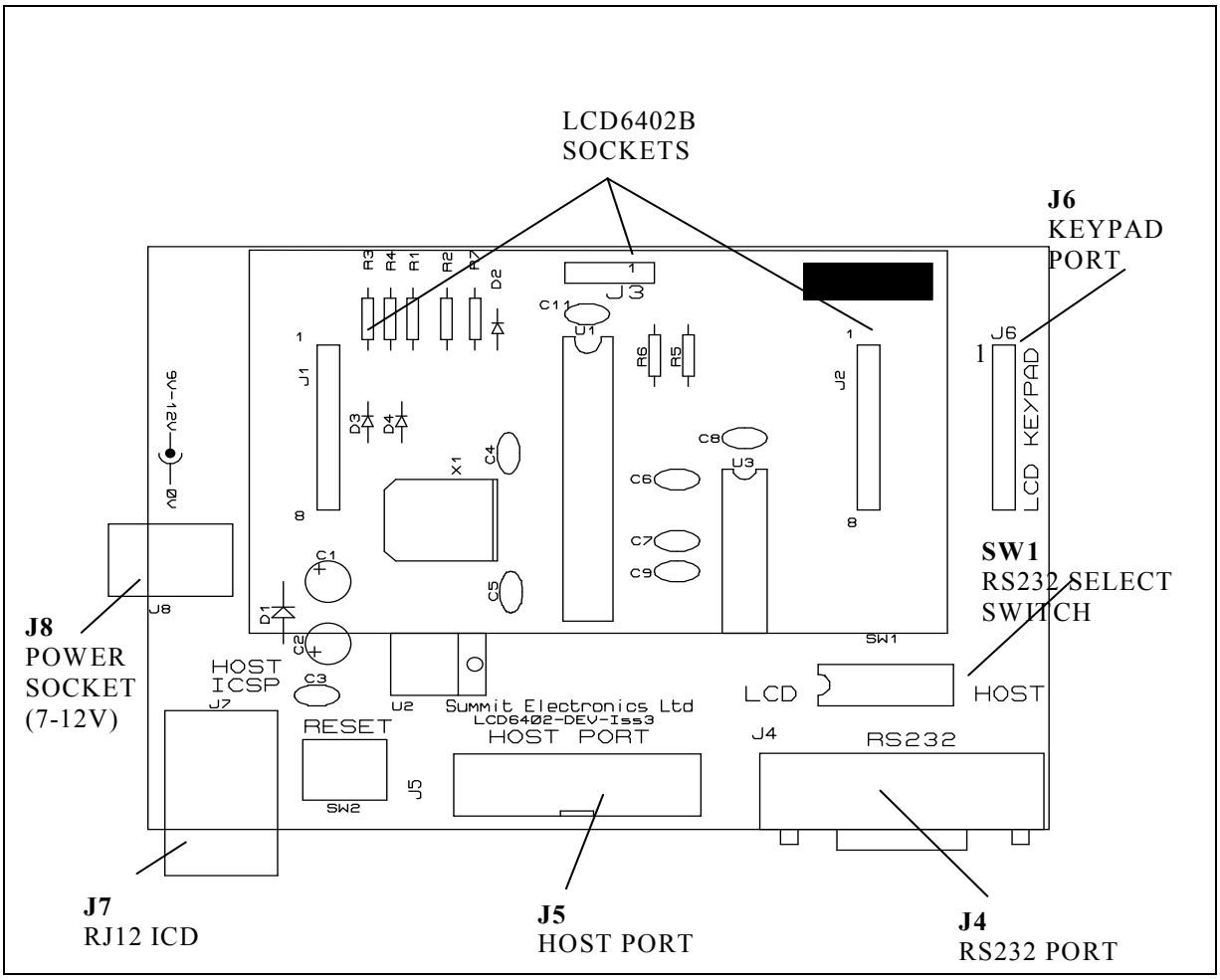


Figure 22 LCD6402B Development /Evaluation board

14 COMMENT FORM

We are always looking to improve our documentation. Please help us by taking a minute to complete this form. Thank you.

1. How do you use this document? (Please tick the corresponding box)

   I read it from the beginning to end.

   I only read the sections relevant to my immediate need.

2. When you need to find information in this guide, where is the first place you usually look? (Please tick the corresponding box)

   Contents page

   I flick through the document

3. How easy did you find it to locate information in this document? (Please circle your answer)

1	2	3	4	5
Very easy				Not at all easy

4. How clear do you find the information in this document? (Please circle your answer)

1	2	3	4	5
Very clear				Not clear at all

5. How well did you understand the product **before** reading this document? (Please circle your answer)

1	2	3	4	5
Very well				Not at all

6. How well do you understand the product **after** reading this document? (Please circle your answer)

1	2	3	4	5
Very well				Not at all

7. What changes/additions do you feel would improve this document?
   Any other comments?

_____

_____

_____

Please forward this comment form to Summit Electronics Ltd Technical Documentation at:

THE OAKS
17 ELNOR LANE
WHALEY BRIDGE
HIGH PEAK
SK23 7EX

or FAX: +44 (0)1663 719570
or email: support@summitelectronics.co.uk

Thank you for your assistance.

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.

Notes

