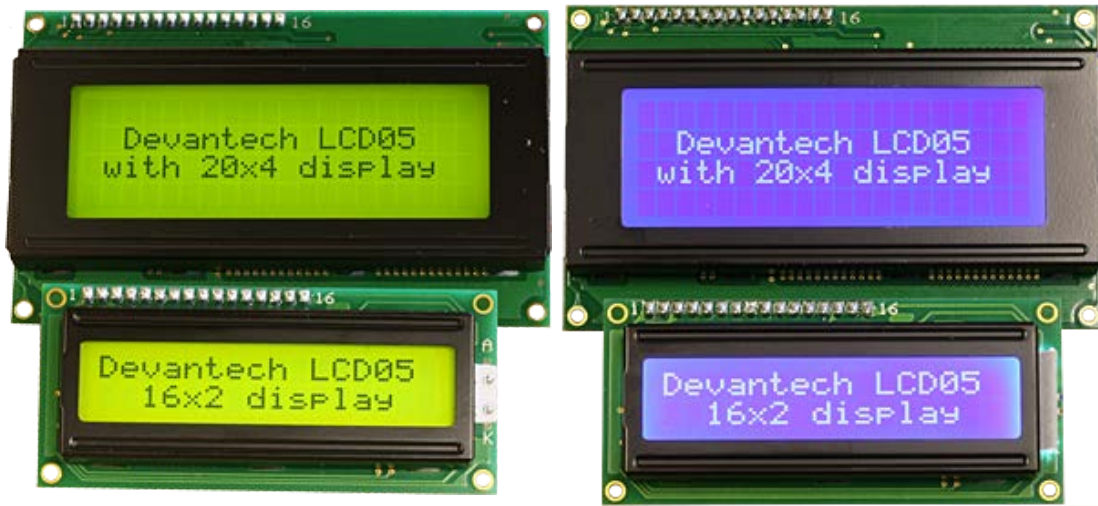


# LCD05 - I2C/Serial LCD

## Technical Documentation



### Overview

The I2C and serial display driver provides easy operation of a standard 20 x 4 or 16 x 2 LCD Text display. It requires only a 5v power supply and the two data connections for either mode, freeing up pins on your processor.

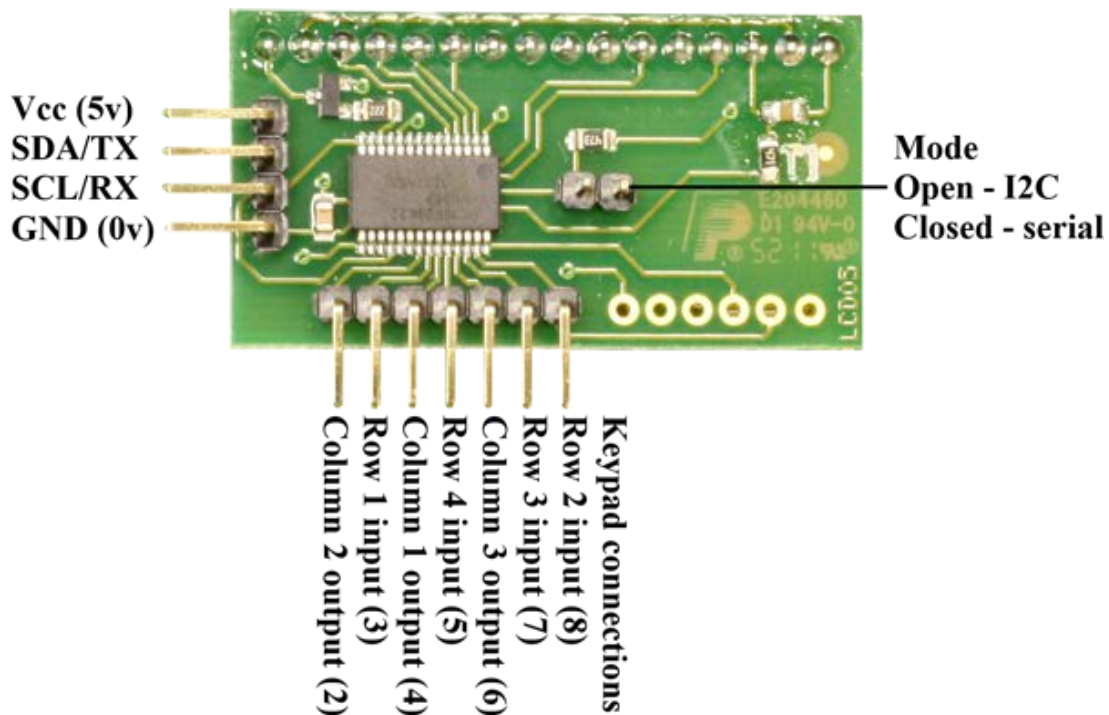
Many useful text formatting functions are provided and also the ability to put together custom characters into memory which can be called off as required.

There is a 100 byte FIFO buffer to ensure a minimum of delay in writing to the display.

The LCD05 controller is our new generation LCD display driver and adds additional features to our previous LCD03 driver while being fully compatible in both software and pin-out.

### Connections

Connection diagram of the LCD05.





Easy connections for 3 x 4 keypad, just wire pins 2-8 to corresponding numbers in brackets above.

Pins 1 and 9 are unconnected - only pins 2 - 8 are used.

## Serial/I2C Mode selection

A single jumper link is used to select Serial or I2C mode. When the jumper is present (factory default) the module is in serial mode. When the jumper is removed the module is in I2C mode. The mode jumper is only checked as part of the power-up sequence, so make sure the display is off before changing modes. If you change the link position while the module is powered, nothing will change until next time you switch on.

## FIFO

Because the I2C communications operate faster than the display can accept data, all commands and text you send to the display are placed in the FIFO (First In, First Out) buffer. This data is sent to the display as fast as it will accept it. The FIFO is 100 bytes in length and the number of free bytes can be returned in either mode. When sending lots of text to the display, you should check the number of free bytes so you don't overflow the buffer. If the buffer does overflow, the excess bytes are ignored. The FIFO buffer exists in Serial mode too, but at 9600 the display is fast enough and you are unlikely to need to check it.

## I2C Operation

### I2C Bus

The I2C display is located on the I2C bus at an address of 0XC6. The SCL and SDA lines should have pull-up resistors on them somewhere on the bus. You only require 1 pair of resistors for the whole I2C bus, not specifically for the LCD05. They are normally on the master controller and you may already have them. If not, anything between 1k8 and 10k should work. I recommend 1k8 for best noise immunity. I2C mode is compatible with SMBus voltage levels.

### I2C Registers

The LCD05 has four registers, three being read only information registers. Register 0 is a dual purpose register, when written, it is the command register where all of the instructions from the commands section should be sent. When read it returns the number of free bytes in the FIFO buffer.

| register | Read                                | Write            |
|----------|-------------------------------------|------------------|
| 0        | Number of free bytes in FIFO buffer | Command register |
| 1        | Keypad state Low byte               | x                |
| 2        | Keypad state High byte              | x                |
| 3        | Version                             | x                |

## I2C Commands

| decimal | command                 | description  |
|---------|-------------------------|--|
| 0       | null (ignored)          | Ignored as a no operation  |
| 1       | Cursor Home             | Sets the cursor to the home position (top left)  |
| 2       | Set cursor (1-80 or 32) | Cursor to a position specified by the next byte, where 1 is the top left and 80/32 is the bottom |

|        |                                | right   |
|--------|--------------------------------|---|
| 3      | set cursor (line, column)      | Sets cursor using two bytes, where first byte is the line and the second byte is the column |
| 4      | Hide cursor                    | stops the position cursor from appearing on the display                                     |
| 5      | Show underline cursor          | Changes the cursor to the underline type  |
| 6      | Show blinking cursor           | Changes the cursor to the blinking type   |
| 8      | Backspace                      | deletes the preceding character from the current position on the display                    |
| 9      | Horizontal tab (by tab set)    | Moves the current position across by the tab space set by command 18 (default tab space 4)  |
| 10     | Smart line feed                | Moves the cursor down one line to the position beneath in the same column                   |
| 11     | Vertical tab                   | Moves the cursor up one line to the position above in the same column                       |
| 12     | Clear screen                   | Clears the screen and sets cursor to the home position                                      |
| 13     | Carriage Return                | Moves the cursor to the start of the next line  |
| 17     | Clear Column                   | Clears the contents of the current column and moves cursor right by one column              |
| 18     | Tab set                        | Sets the required tab size, the following byte can be a size of between 1 and 10            |
| 19     | Backlight on                   | Turns the backlight of the LCD05 on   |
| 20     | Backlight off (default)        | Turns the backlight of the LCD05 off  |
| 21     | Disable startup message        | Disables the display of setup information at power up                                       |
| 22     | Enable startup message         | Enables the display of setup information at power up  |
| 23     | Save as start-up screen        | Saves current screen and displays it on power up  |
| 24     | Set display type               | Sets which display is connected to the LCD05 (see changing display type below)              |
| 25     | <a href="#">Change Address</a> | First byte of sequence to change LCD05 address (see changing address)                       |
| 27     | Custom char generator          | allows 8 custom chars to be built. See custom char generator below                          |
| 28     | Double keypad scan rate        | Increases the frequency of the keypad scan to 20hz  |
| 29     | Normal keypad scan rate        | Returns to the default keypad scan frequency of 10hz  |
| 30     | Contrast Set                   | Byte following this command will be contrast level (0-255)                                  |
| 31     | Brightness Set                 | Byte following this command will be brightness level (0-255)                                |
| 32-255 | ASCII chars                    | Writes ASCII chars straight to the display  |

## Changing address

The LCD05 software has the facility to easily change the modules address on the I2C bus system in the range of 0xC6 - 0xCE ( even numbers only ). This is achieved by sending the following sequence:

- 0x19 ( 1st byte of change address command )
- 0xA0 ( 2nd byte of change address command )
- 0xAA ( 3rd byte of change address command )
- 0xA5 ( 4th byte of change address command )
- 0xC6, 0xC8, 0xCA, 0xCC or 0xCE ( new address to map LCD05 at )

The Current address is displayed every time the module is first powered as a default, this may be changed to no display or custom display by user command.

## Serial operation

The Serial mode operates over a link with a baud rate of 9600 bps (no parity, 2 stop bits) and 5v signals, Do NOT connect RS232 directly to the module - you will destroy it. Use a MAX232 or equivalent to convert the RS232 levels to 5v. Operation is with the same command set as the I2C mode with an additional set of commands to request data to be sent i.e. the software version.

## Serial commands

| decimal | command | description |
|---------|---------|-------------|
|         |         |             |

|        |                                |  |
|--------|--------------------------------|--|
| 0      | null (ignored)                 | Ignored as a no operation  |
| 1      | Cursor Home                    | Sets the cursor to the home position (top left)  |
| 2      | Set cursor (1-80 or 32)        | Cursor to a position specified by the next byte, where 1 is the top left and 80/32 is the bottom right |
| 3      | set cursor (line, column)      | Sets cursor using two bytes, where first byte is the line and the second byte is the column            |
| 4      | Hide cursor                    | stops the position cursor from appearing on the display  |
| 5      | Show underline cursor          | Changes the cursor to the underline type   |
| 6      | Show blinking cursor           | Changes the cursor to the blinking type  |
| 8      | Backspace                      | deletes the preceding character from the current position on the display                               |
| 9      | Horizontal tab (by tab set)    | Moves the current position across by the tab space set by command 18 (default tab space 4)             |
| 10     | Smart line feed                | Moves the cursor down one line to the position beneath in the same column                              |
| 11     | Vertical tab                   | Moves the cursor up one line to the position above in the same column                                  |
| 12     | Clear screen                   | Clears the screen and sets cursor to the home position   |
| 13     | Carriage Return                | Moves the cursor to the start of the next line   |
| 14     | FIFO status                    | Results in the module returning its FIFO status, the number of free bytes from 0 to 100                |
| 15     | Software version               | Module returns a single byte software version  |
| 16     | <a href="#">Keypad status</a>  | Returns keypad input status, two bytes returned with the low byte first (see keypad section)           |
| 17     | Clear Column                   | Clears the contents of the current column and moves cursor right by one column                         |
| 18     | Tab set                        | Sets the required tab size, the following byte can be a size of between 1 and 10                       |
| 19     | Backlight on                   | Turns the backlight of the LCD05 on  |
| 20     | Backlight off (default)        | Turns the backlight of the LCD05 off   |
| 21     | Disable startup message        | Disables the display of setup information at power up  |
| 22     | Enable startup message         | Enables the display of setup information at power up   |
| 23     | Save as start-up screen        | Saves current screen and displays it on power up   |
| 24     | Set display type               | Sets which display is connected to the LCD05 (see changing display type below)                         |
| 25     | <a href="#">Change Address</a> | First byte of sequence to change LCD05 address (see changing address)                                  |
| 27     | Custom char generator          | allows 8 custom chars to be built. See custom char generator below                                     |
| 28     | Double keypad scan rate        | Increases the frequency of the keypad scan to 20hz   |
| 29     | Normal keypad scan rate        | Returns to the default keypad scan frequency of 10hz   |
| 30     | Contrast Set                   | Byte following this command will be contrast level (0-255)   |
| 31     | Brightness Set                 | Byte following this command will be brightness level (0-255)   |
| 32-255 | ASCII chars                    | Writes ASCII chars straight to the display   |

### Contrast and brightness set

Contrast and brightness may be adjusted using software commands to vary the level between 0 and 255. Following a power cycle the LCD05 will automatically revert to a predetermined level for the display that is attached.

### Changing display type

Because there are four variations of the LCD display that the LCD05 connects to a software command can be used to switch between them.

Firstly a command of 24 must be issued, this is then followed by the display type attached:

| LCD style    | Byte |
|--------------|------|
| 20 x 4 Green | 3    |

|              |   |
|--------------|---|
| 20 x 4 Blue  | 4 |
| 16 x 2 Green | 5 |
| 16 x 2 Blue  | 6 |

## Custom character generator

Custom characters can be generated by sending an 8 byte map. The first thing that must be done is to send a command of 27 to indicate that you intend to make a custom char. Next you have to specify the position in ram of one of the 8 available chars you intend to build, the 8 chars are mapped at positions 128-135. Then the pattern should be sent as below.

How to calculate an 8 byte character pattern:

Eight bytes are sent with the highest bit masked on, the next two bits are ignored (x) and the final 5 bits show the line pattern.

|               | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | sent byte      |
|---------------|-------|-------|-------|-------|-------|----------------|
| <b>Byte 0</b> |       |       |       |       |       | 1xx00000 (128) |
| <b>Byte 1</b> |       |       | ■     |       |       | 1xx00100 (132) |
| <b>Byte 2</b> |       | ■     | ■     | ■     |       | 1xx01110 (142) |
| <b>Byte 3</b> | ■     |       | ■     |       | ■     | 1xx10101 (149) |
| <b>Byte 4</b> |       |       | ■     |       |       | 1xx00100 (132) |
| <b>Byte 5</b> |       |       | ■     |       |       | 1xx00100 (132) |
| <b>Byte 6</b> |       |       | ■     |       |       | 1xx00100 (132) |
| <b>Byte 7</b> |       |       |       |       |       | 1xx00000 (128) |

So to build the above arrow into location 128 you would send this sequence to the command register:

- 27 (char generate command)
- 128 (location to be filled)
- 128 (byte 0)
- 132 (byte1)
- 142 (byte2)
- 149 (byte3)
- 132 (byte 4)
- 132 (byte 5)
- 132 (byte 6)
- 128 (byte 7)

Now the char is built into a memory location it can be called at any time as long as the module remains powered by simply sending the address between 128 and 135.

## Keypad input

An added feature of the module is the ability to connect a 3 x 4 Keypad, the module will automatically scan the status of the keys at regular intervals. The result is then provided in two bytes for easy use, shown by the diagram below. Each bit represents the row and column of each button on the keypad. The second row is the keypad characters when using our keypad.

|   |   |   | High | byte |     |     |     |  |     |     | Low | byte |     |     |     |     |
|---|---|---|------|------|-----|-----|-----|--|-----|-----|-----|------|-----|-----|-----|-----|
| 0 | 0 | 0 | 0    | 4/3  | 4/2 | 4/1 | 3/3 |  | 3/2 | 3/1 | 2/3 | 2/2  | 2/1 | 1/3 | 1/2 | 1/1 |
| 0 | 0 | 0 | 0    | #    | 0   | *   | 9   |  | 8   | 7   | 6   | 5    | 4   | 3   | 2   | 1   |

Any key press will result in the corresponding bit in the byte being driven high. In I2C mode these bytes are available for reading from registers one and two.

In serial mode, just send 16 to the LCD05 and it will reply with the two bytes as above with the low byte first.

## Character Map

Remember you can also add up to 8 custom characters.

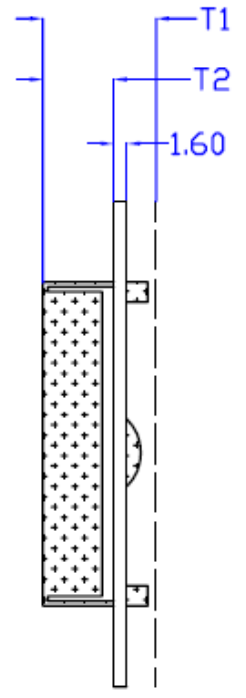
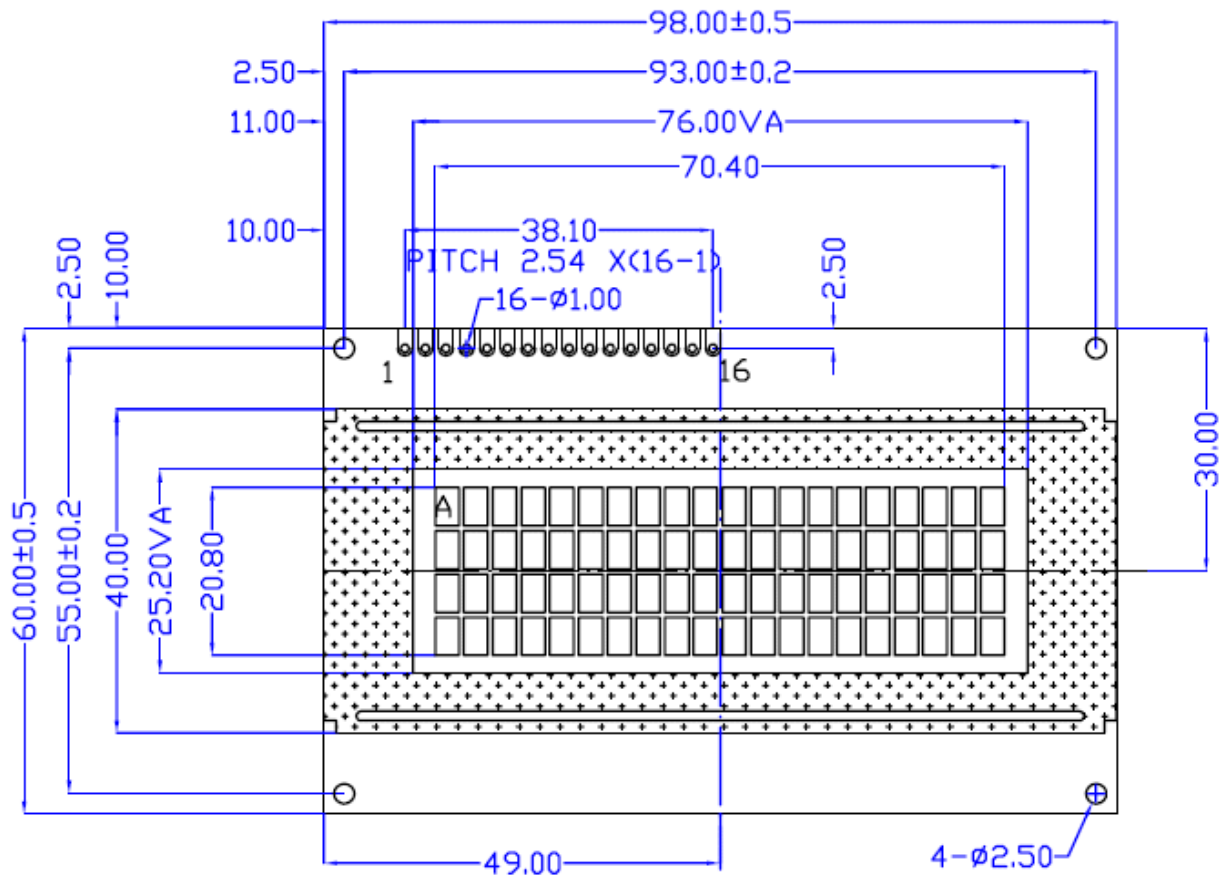
| Lower 4 Bits \ Upper 4 Bits | 0000       | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
|-----------------------------|------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx0000                    | CG RAM (1) |      |      | 0    | @    | P    | `    | P    |      |      |      | -    | 9    | 3    | 0    | P    |
| xxxx0001                    | (2)        |      | !    | 1    | A    | Q    | a    | 9    |      |      | o    | 7    | 4    |      | ä    | q    |
| xxxx0010                    | (3)        |      | "    | 2    | B    | R    | b    | r    |      |      | 「    | イ    | ツ    | ×    | ß    | θ    |
| xxxx0011                    | (4)        |      | #    | 3    | C    | S    | c    | s    |      |      | 」    | ウ    | テ    | ε    | ε    | ∞    |
| xxxx0100                    | (5)        |      | \$   | 4    | D    | T    | d    | t    |      |      | 、    | イ    | ト    | ト    | μ    | Ω    |
| xxxx0101                    | (6)        |      | %    | 5    | E    | U    | e    | u    |      |      | ・    | オ    | ナ    | 1    | 0    | ü    |
| xxxx0110                    | (7)        |      | &    | 6    | F    | V    | f    | v    |      |      | ヲ    | カ    | ニ    | ヨ    | ρ    | Σ    |
| xxxx0111                    | (8)        |      | '    | 7    | G    | W    | g    | w    |      |      | ア    | キ    | ヌ    | ラ    | q    | π    |
| xxxx1000                    | (1)        |      | (    | 8    | H    | X    | h    | x    |      |      | ィ    | ク    | ネ    | リ    | γ    | ×    |
| xxxx1001                    | (2)        |      | )    | 9    | I    | Y    | i    | y    |      |      | ウ    | ケ    | ル    |      | γ    | γ    |
| xxxx1010                    | (3)        |      | *    | :    | J    | Z    | j    | z    |      |      | エ    | コ    | ン    | レ    | j    | ≠    |
| xxxx1011                    | (4)        |      | +    | ;    | K    | C    | k    | c    |      |      | オ    | サ    | ヒ    | ロ    | *    | π    |
| xxxx1100                    | (5)        |      | ,    | <    | L    | ¥    | l    | l    |      |      | カ    | シ    | フ    | ワ    | φ    | π    |
| xxxx1101                    | (6)        |      | -    | =    | M    | J    | m    | }    |      |      | ユ    | ズ    | ン    |      | ε    | ÷    |
| xxxx1110                    | (7)        |      | .    | >    | N    | ^    | n    | →    |      |      | ヨ    | セ    | ホ    | °    | ñ    |      |
| xxxx1111                    | (8)        |      | /    | ?    | O    | _    | o    | +    |      |      | ッ    | ソ    | マ    | °    | ö    | ■    |

## Current consumption (typical)

| Display      | Backlight off | Backlight on |
|--------------|---------------|--------------|
| 20 x 4 green | 5 mA          | 135 mA       |
| 16 x 2 green | 5 mA          | 80 mA        |
| 20 x 4 blue  | 5 mA          | 55 mA        |
| 16 x 2 blue  | 5 mA          | 40 mA        |

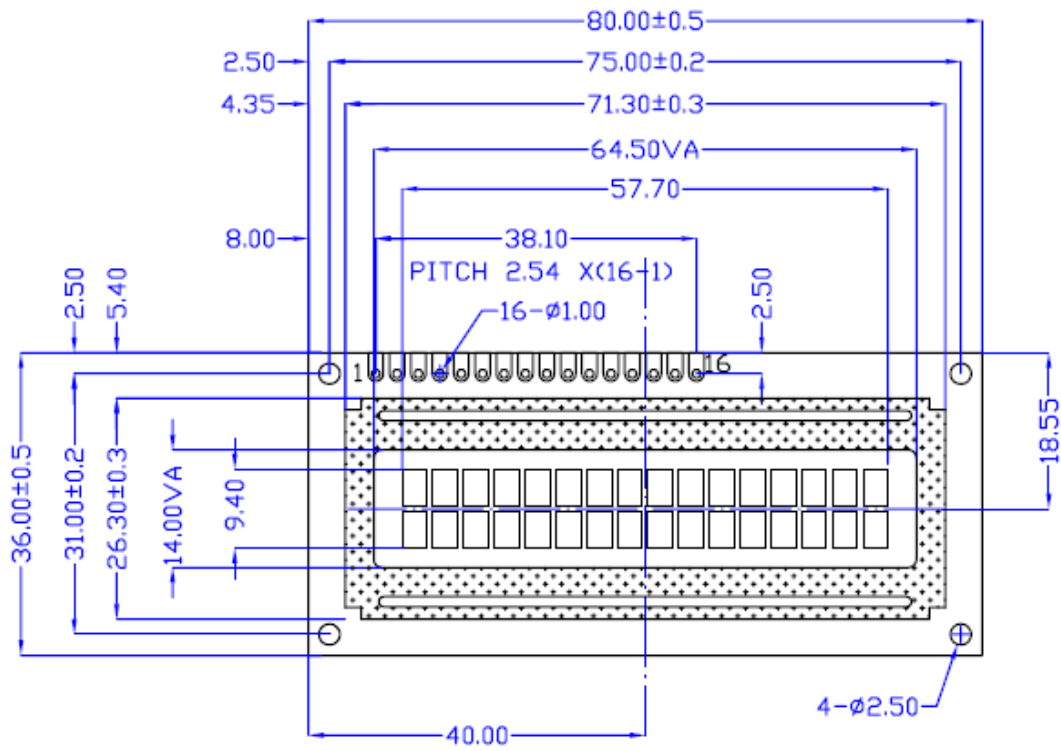
# LCD Module Dimensions

### 20 x 4 LCD dimensions



\* FOR LED BACKLIGHT  
 $T1=14.0 \text{MAX}$   $T2=9.0 \pm 0.3$

### 16 x 2 LCD dimensions



**Keypad Dimensions**

