Chipsmall

Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Parallax Serial LCD

2 rows x 16 characters, Non-backlit, with Piezospeaker (#27976)
2 rows x 16 characters, Backlit, with Piezospeaker (#27977)
4 rows x 20 characters, Backlit, with Piezospeaker (#27979)

The Parallax Serial LCDs are very functional, low-cost liquid crystal displays that can be easily interfaced to and controlled by a microcontroller using a I/O pin. The LCD displays provide basic text wrapping so that your text looks correct on the display. Full control over all of their advanced LCD features allows you to move the cursor anywhere on the display with a single instruction and turn the display on and off in any configuration. They support visible ASCII characters Dec 32-127, and in addition you may define up to eight of your own custom characters to display anywhere on the LCD.

NOTE: If your Serial LCD Display does not have a speaker on the back, use the specifications and information in the Product Change Notice: Revision E and Earlier section on page 11.

Features

- Clear 40-pixel characters (8 H x 5 W)

- Supports ASCII DEC characters 32-127

- Define up to eight custom characters

- Automatic text wrapping

- Single command cursor placement

- Single command clears the display

- Define up to eight custom characters

- Select 2400, 9600, or 19,200 baud with switches on the back of the device

- Display type: STN, YG, positive transflective LCD

- Adjustable contrast knob on the back of the device

- YG LED for backlit model displays

Key Specifications

- Power requirements:
  - o Non-backlit: +5 VDC, 20 mA
  - o Backlit: +5 VDC, 20 mA (light off), ~ 80 mA typical (light on)

- Communication: Selectable asynchronous serial baud rates: 2400, 9600, 19200

- Operating temperature: -4 to +158 °F (-20 to +70°C)

- Dimensions: NOTE - Board and LCD size and style may vary
  - o 2x16: Approx. 1.42 x 3.15 in (36 x 80 mm)
  - o 4x20: Approx. 2.37 x 3.86 in (60.2 x 98.1 mm)

Quick-Start Circuit

The Serial LCDs should be powered from an external regulated 5 V power supply. Make sure the power supply has an adequate current rating to power the Serial LCD and the BASIC Stamp, Propeller chip, or whichever microcontroller and other devices you are using.

CAUTION

DO NOT PROVIDE A SIGNAL TO THE 'RX' PIN BEFORE APPLYING 5 VDC TO THE '5V' PIN.
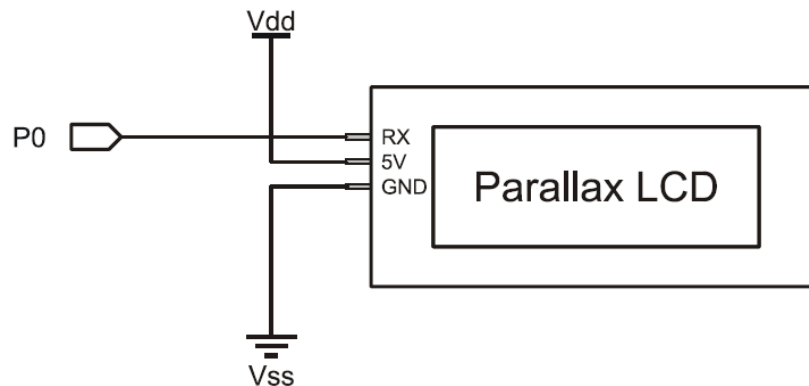
Baud Rate Setup

After connecting the Serial LCD, you will need to select the baud rate at which you are going to send it data. You have three choices: 2400, 9600, and 19,200 baud. To set the baud rate, move the dip switches on the back of the LCD into the correct positions according to the table next to the switches, which is also repeated below:

MODE	SW1	SW2
Test	OFF	OFF
2,400	ON	OFF
9,600	OFF	ON
19,200	ON	ON
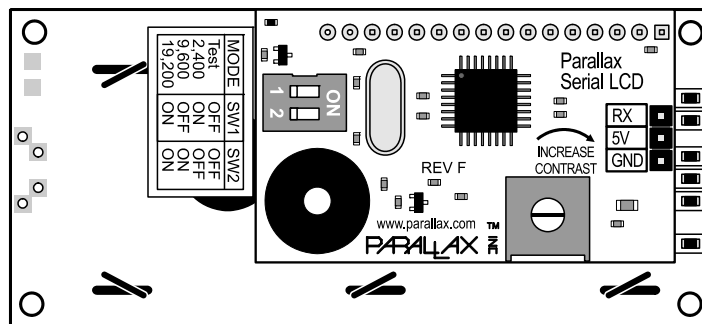
As you can see from the table, there is also a fourth choice called Test. Use this Test mode to confirm that the power and ground to the LCD are hooked up correctly before you send it any data. Move the dip switches to the Test setting and turn on the power. The LCD display should turn on with the backlight on (models 27977, 27979) and display the following text:

```
Parallax, Inc.
www.parallax.com
```

If you don't see the text at first, try adjusting the LCD contrast by turning the pot labeled "Increase Contrast" with a screwdriver. Turn it in the direction of the arrow to make the characters show up more clearly. If you still don't see the characters, go back and check your electrical connections and try again. Once you've successfully completed test mode, move the dip switches to the correct positions to select the baud rate you want to use for your application.

Displaying Text

Now that the LCD is set up, it's time to start sending text to the display. To display a character of text on the Serial LCD, simply send the ASCII code of that character to the Serial LCD over the serial port at the correct baud rate.

When a character is received, the Serial LCD displays that character at the current cursor position and then moves the cursor one position to the right. When you first turn on the LCD, the cursor is in the leftmost position on the top line, as you might expect. The short bar on the bottom of the character shows where the cursor is positioned currently.

Once you've sent a full line characters to the LCD, you will notice that the cursor automatically wraps around to the leftmost position of the second line, just like the text in a book. The text will wrap like this at the end of every line, with the end of the bottom line wrapping back around to the top line of the LCD. The text will never "run off" the display; you'll always see all of the characters you send.

Example code is provided below. You may download the example code files from the 27976, 27977, or 27979 product pages at www.parallax.com; just enter any of these product numbers in the "search" field on the home page

BASIC Stamp® 2 Example Code

You may download the example code from the 27976, 27977, or 27979 product pages at www.parallax.com; just enter any of these product numbers in the "search" field on the home page. Try the following code on your BASIC Stamp 2 to send a text string to the LCD display. First, set the baud rate on your Serial LCD to 19,200. Then, load the code below into your BASIC Stamp 2 and run it.

You will see the text string show up and wrap to the second line of the display.

In all of your Serial LCD code, you should pause for 100 ms at start-up to give time for the Serial LCD to initialize. You should also set the serial port pin on the BASIC Stamp to HIGH before the 100 ms start-up delay, as this is the normal state of a serial port when it isn't sending any data.

```
' {$STAMP BS2}
' {$PBASIC 2.5}

TxPin        CON    0
Baud19200    CON    32

  HIGH TxPin                  ' Set pin high to be a serial port
  PAUSE 100                   ' Pause for Serial LCD to initialize

  SEROUT TxPin, Baud19200, ["Hello, this text will wrap."]
```

Propeller P8X32A Example Code

Try the following code on your Propeller to send a text string to the LCD display. First, set the baud rate on your Serial LCD to 19,200. Then, load the code below into your Propeller and load RAM or EEPROM.

You will see the text string show up and wrap to the second line of the display.

In all of your Serial LCD code, you should wait for 100 ms after starting the FullDuplexSerial.spin object, to give the object and the Serial LCD time to initialize. The FullDuplexSerial.spin object is included with the Propeller Tool.

```
{{
Serial_LCD_Demo.spin
For Parallax Serial LCDs 27976, 27977, 27979
}}

CON
  _clkmode = xtal1 + pll16x
  _xinfreq = 5_000_000

  TX_PIN = 0
  BAUD = 19_200

OBJ

  LCD : "FullDuplexSerial.spin"

PUB Main

  LCD.start(TX_PIN, TX_PIN, %1000, 19_200)
  waitcnt(clkfreq / 100 + cnt)                ' Pause for FullDuplexSerial.spin to initialize
  LCD.str(string("Hello, this text will wrap."))
```

Moving the Cursor

When you send a character to the Serial LCD, it always displays at the current cursor position. There are a few different ways to move the cursor on the Serial LCD display. After each character you send, the cursor automatically moves over one position. Along with this, there is a standard set of cursor move commands including Backspace, Carriage Return, and Line Feed.

The Backspace/Left command (Dec 8) moves the cursor one place to the left and the Right command (Dec 9) moves the cursor one place to the right. These can be useful for moving the cursor around to overwrite existing text. These commands wrap to the next line of the display, if necessary. The Line Feed command (Dec 10) moves the cursor to the next line of the display without changing the horizontal position of the cursor. The Carriage Return command (Dec 13) also moves the cursor to the next line, but it moves the cursor to the leftmost position on that line as well. The Form Feed command (Dec 12) clears the entire display and moves the cursor to the leftmost position on line 0, just like when you first turn on the display. You will need to pause for 5mS in your code after sending the Form Feed command, to give the Serial LCD time to clear the display. Except for Form Feed, none of these move commands affects the characters on the display.

There are also direct move commands that you can use to move the cursor to any position on the display with a single command. The commands in the range Dec 128 to 143 and Dec 148 to 163 move the cursor to the 16 different positions on each of the two lines of the model 27976 and 27977 LCDs. The commands in the range Dec 128 to 207 move the cursor to the 20 different positions on each of the four lines of the model 27979 LCD.

Controlling the Display

You also have control over the various display modes of the Serial LCD. The display-off command (Dec 21) turns off the display so that all of the characters disappear. The characters aren't erased from the display, though, and you can even keep writing new characters to the display when it is turned off. A trick to make a lot of text show up all at once, even at a slow baud rate, is to turn off the display and then send all of your text. Then, when you turn the display on again, all of the text appears instantly.

The display-on commands (Dec 22 to 25) turn the display back on and also control whether you want to display the cursor and/or make the cursor character blink. The cursor is the short bar that shows up below the character at the current cursor position. The blink option makes that character blink on and off repeatedly. You can turn the cursor and blink options on or off, in any combination, as listed in the command set table. You can change the cursor and blink mode even if the display is already on; you don't need to turn it off and then back on again.

With models 27977 and 27979, you can also control the backlight of the display. The backlight lights up the display so that it is easier to see in the dark. There are commands to turn the backlight on (Dec 17) and off (Dec 18).

Custom Characters

The Serial LCD has the capability to store up to eight user-defined custom characters. The custom characters are stored in RAM and so they need to be redefined if you turn off the power. You can display the custom characters by sending the commands Dec 0 to 7, as shown in the command set table. The custom character will display at the current cursor position.

The custom characters are five pixels wide by eight pixels high. Each of the characters is stored as a series of eight data bytes where the low five bits of each byte represent a row of pixels in the character. The high three bits of each byte are ignored. A bit value of one turns that pixel on (i.e. makes it black). The bottom row of pixels is often left blank (all zeros) to make it easier to see the cursor.

To define a custom character, you will send a total of 9 bytes to the Serial LCD. The first byte needs to be a valid define-custom-character command (Dec 248 to 255) and must be followed by eight data bytes that define the pixels of the character. The Serial LCD will always use the next eight bytes it receives to set the pixels of the character. The data bytes define the character starting at the topmost row of pixels, as shown in the example code.

BASIC Stamp 2 Custom Character Example

Define a custom character using the code example below. First, set the baud rate on your Serial LCD to 19,200. Then, load the code below into your BASIC Stamp 2 and run it. You will see a diamond character appear on the screen.

```
' {$STAMP BS2}
' {$PBASIC 2.5}

TxPin           CON    0
Baud19200       CON    32

  HIGH TxPin                          ' Set pin high to be a serial port
  PAUSE 100                           ' Pause for Serial LCD to initialize

  SEROUT TxPin, Baud19200, [250]      ' Define custom character 2
                                      ' Now send the eight data bytes
  SEROUT TxPin, Baud19200, [%00000]   ' %00000 =
  SEROUT TxPin, Baud19200, [%00100]   ' %00100 = *
  SEROUT TxPin, Baud19200, [%01110]   ' %01110 = * * *
  SEROUT TxPin, Baud19200, [%11111]   ' %11111 = * * * * *
  SEROUT TxPin, Baud19200, [%01110]   ' %01110 = * * *
  SEROUT TxPin, Baud19200, [%00100]   ' %00100 = *
  SEROUT TxPin, Baud19200, [%00000]   ' %00000 =
  SEROUT TxPin, Baud19200, [%00000]   ' %00000 =
  SEROUT TxPin, Baud19200, [2]        ' Display the new custom character 2
```

Propeller™ P8X32A Example Code

Define a custom character using the code example below. First, set the baud rate on your Serial LCD to 19,200. Then, load the code below into your Propeller and load RAM or EEPROM. You will see a diamond character appear on the screen. Note: the FullDuplexSerial.spin object is included with the Propeller Tool software.

```
{{
Serial_LCD_Custom_Character.spin
For Parallax Serial LCDs 27976, 27977, 27979
}}

CON
  _clkmode = xtal1 + pll16x
  _xinfreq = 5_000_000

  TX_PIN = 0
  BAUD = 19_200

OBJ

  LCD : "FullDuplexSerial.spin"


PUB Main
  LCD.start(TX_PIN, TX_PIN, %1000, 19_200)
  waitcnt(clkfreq / 100 + cnt)                  ' Pause for FullDuplexSerial.spin to initialize


  LCD.tx(250)                                   ' Define custom character 2
                                                ' Now send the eight data bytes
  LCD.tx(%00000)                                ' %00000 =
  LCD.tx(%00100)                                ' %00100 = *
  LCD.tx(%01110)                                ' %01110 = * * *
  LCD.tx(%11111)                                ' %11111 = * * * * *
  LCD.tx(%01110)                                ' %01110 = * * *
  LCD.tx(%00100)                                ' %00100 = *
  LCD.tx(%00000)                                ' %00000 =
  LCD.tx(%00000)                                ' %00000 =
  LCD.tx(2)                                     ' Display the new custom character 2
```

Playing Music

The Serial LCD has a built-in piezoelectric speaker which can play musical notes. The LCD includes a sophisticated music player enabling users to program songs and tunes. You can play musical notes by sending three types of note commands (Dec 214 to 232), as shown in the command set table.

Set Length

The "set length" commands (Dec 208 to 214) determines the length of time each note will play for. This value remains the same until another "set length" command is received. A note's length can range from a 1/64th note to a whole note. A whole note is 2 seconds long.

Set Scale

The "set scale" commands (Dec 215 to 219) determines the octave each note will play in. This value remains the same until another "set scale" command is received. The current octave can be set to a value between 3 and 7, and each octave consists of 12 notes. The frequency for each note and scale combination is shown in the table below.

Scale	A	A#	B	C	C#	D	D#	E	F	F#	G	G#
3	220	233	247	262	277	294	311	330	349	370	392	415
4	440	466	494	523	554	587	622	659	698	740	784	831
5	880	932	988	1047	1109	1175	1245	1319	1397	1480	1568	1661
6	1760	1865	1976	2093	2217	2349	2489	2637	2794	2960	3136	3322
7	3520	3729	3951	4186	4435	4699	4978	5274	5588	5920	6272	6645

Play Note

The "play note" commands (Dec 220 to 232) play a single note for the currently set time and in the currently set octave.

Command Set

The tables on the following pages list all of the valid Serial LCD commands. Commands marked as N/A are invalid and are ignored. The lines of the LCD display are numbered starting from 0, with line 0 being the top line. The character positions on each line are numbered starting from 0, with position 0 being the leftmost position on the line.

Dec	Hex	Action
0	00	Display custom character 0
1	01	Display custom character 1
2	02	Display custom character 2
3	03	Display custom character 3
4	04	Display custom character 4
5	05	Display custom character 5
6	06	Display custom character 6
7	07	Display custom character 7
8	08	Backspace / Left - The cursor is moved one position to the left. The command doesn't erase the character.
9	09	Right - The cursor is moved one position to the right. The command doesn't erase the character.
10	0A	Line Feed - The cursor is moved down one line. For the two line LCD model, if on line 0 it goes to line 1. If on line 1, it wraps around to line 0. The horizontal position remains the same.
Dec	Hex	Action
11	0B	N/A
12	0C	Form Feed - The cursor is moved to position 0 on line 0 and the entire display is cleared. Users must pause 5mS after this command.
13	0D	Carriage Return – For the two line LCD model, if on line 0 the cursor is moved to position 0 on line 1. If on line 1, it wraps around to position 0 on line 0.
14 - 16	0E - 10	N/A
17	11	Turn backlight on (only on models 27977, 27979)

Dec	Hex	Action
18	12	Turn backlight off (Default)
19 - 20	13 - 14	N/A
21	15	Turn the display off
22	16	Turn the display on, with cursor off and no blink
23	17	Turn the display on, with cursor off and character blink
24	18	Turn the display on, with cursor on and no blink (Default)
25	19	Turn the display on, with cursor on and character blink
26 - 31	1A - 1F	N/A
32 - 127	20 - 7F	Display ASCII characters. See the ASCII character set table.
128	80	Move cursor to line 0, position 0
129	81	Move cursor to line 0, position 1
130	82	Move cursor to line 0, position 2
131	83	Move cursor to line 0, position 3
132	84	Move cursor to line 0, position 4
133	85	Move cursor to line 0, position 5
134	86	Move cursor to line 0, position 6
135	87	Move cursor to line 0, position 7
136	88	Move cursor to line 0, position 8
137	89	Move cursor to line 0, position 9
138	8A	Move cursor to line 0, position 10
139	8B	Move cursor to line 0, position 11
140	8C	Move cursor to line 0, position 12
141	8D	Move cursor to line 0, position 13
142	8E	Move cursor to line 0, position 14
143	8F	Move cursor to line 0, position 15
144	90	Move cursor to line 0, position 16 (only on model 27979)
145	91	Move cursor to line 0, position 17 (only on model 27979)
146	92	Move cursor to line 0, position 18 (only on model 27979)
147	93	Move cursor to line 0, position 19 (only on model 27979)
148	94	Move cursor to line 1, position 0
149	95	Move cursor to line 1, position 1
150	96	Move cursor to line 1, position 2
151	97	Move cursor to line 1, position 3
152	98	Move cursor to line 1, position 4
153	99	Move cursor to line 1, position 5
154	9A	Move cursor to line 1, position 6
155	9B	Move cursor to line 1, position 7
156	9C	Move cursor to line 1, position 8
Dec	Hex	Action
157	9D	Move cursor to line 1, position 9
158	9E	Move cursor to line 1, position 10
159	9F	Move cursor to line 1, position 11
160	A0	Move cursor to line 1, position 12
161	A1	Move cursor to line 1, position 13
162	A2	Move cursor to line 1, position 14
163	A3	Move cursor to line 1, position 15

Dec	Hex	Action
164	A4	Move cursor to line 1, position 16 (only on model 27979)
165	A5	Move cursor to line 1, position 17 (only on model 27979)
166	A6	Move cursor to line 1, position 18 (only on model 27979)
167	A7	Move cursor to line 1, position 19 (only on model 27979)
168	A8	Move cursor to line 2, position 0 (only on model 27979)
169	A9	Move cursor to line 2, position 1 (only on model 27979)
170	AA	Move cursor to line 2, position 2 (only on model 27979)
171	AB	Move cursor to line 2, position 3 (only on model 27979)
172	AC	Move cursor to line 2, position 4 (only on model 27979)
173	AD	Move cursor to line 2, position 5 (only on model 27979)
174	AE	Move cursor to line 2, position 6 (only on model 27979)
175	AF	Move cursor to line 2, position 7 (only on model 27979)
176	B0	Move cursor to line 2, position 8 (only on model 27979)
177	B1	Move cursor to line 2, position 9 (only on model 27979)
178	B2	Move cursor to line 2, position 10 (only on model 27979)
179	B3	Move cursor to line 2, position 11 (only on model 27979)
180	B4	Move cursor to line 2, position 12 (only on model 27979)
181	B5	Move cursor to line 2, position 13 (only on model 27979)
182	B6	Move cursor to line 2, position 14 (only on model 27979)
183	B7	Move cursor to line 2, position 15 (only on model 27979)
184	B8	Move cursor to line 2, position 16 (only on model 27979)
185	B9	Move cursor to line 2, position 17 (only on model 27979)
186	BA	Move cursor to line 2, position 18 (only on model 27979)
187	BB	Move cursor to line 2, position 19 (only on model 27979)
188	BC	Move cursor to line 3, position 0 (only on model 27979)
189	BD	Move cursor to line 3, position 1 (only on model 27979)
190	BE	Move cursor to line 3, position 2 (only on model 27979)
191	BF	Move cursor to line 3, position 3 (only on model 27979)
192	C0	Move cursor to line 3, position 4 (only on model 27979)
193	C1	Move cursor to line 3, position 5 (only on model 27979)
194	C2	Move cursor to line 3, position 6 (only on model 27979)
195	C3	Move cursor to line 3, position 7 (only on model 27979)
196	C4	Move cursor to line 3, position 8 (only on model 27979)
197	C5	Move cursor to line 3, position 9 (only on model 27979)
198	C6	Move cursor to line 3, position 10 (only on model 27979)
199	C7	Move cursor to line 3, position 11 (only on model 27979)
200	C8	Move cursor to line 3, position 12 (only on model 27979)
201	C9	Move cursor to line 3, position 13 (only on model 27979)
202	CA	Move cursor to line 3, position 14 (only on model 27979)
Dec	Hex	Action
203	CB	Move cursor to line 3, position 15 (only on model 27979)
204	CC	Move cursor to line 3, position 16 (only on model 27979)
205	CD	Move cursor to line 3, position 17 (only on model 27979)
206	CE	Move cursor to line 3, position 18 (only on model 27979)
207	CF	Move cursor to line 3, position 19 (only on model 27979)
208	D0	Set note length to 1/64 note

Dec	Hex	Action
209	D1	Set note length to 1/32 note
210	D2	Set note length to 1/16 note
211	D3	Set note length to 1/8 note
212	D4	Set note length to 1/4 note
213	D5	Set note length to 1/2 note
214	D6	Set note length to whole note (2 seconds)
215	D7	Select the 3rd scale (A = 220 Hz)
216	D8	Select the 4th scale (A = 440 Hz)
217	D9	Select the 5th scale (A = 880 Hz)
218	DA	Select the 6th scale (A = 1760 Hz)
219	DB	Select the 7th scale (A = 3520 Hz)
220	DC	Play A note
221	DD	Play A# note
222	DE	Play B
223	DF	Play C
224	E0	Play C#
225	E1	Play D
226	E2	Play D#
227	E3	Play E
228	E4	Play F
229	E5	Play F#
230	E6	Play G
231	E7	Play G#
232	E8	Pause for current note length (no sound)
233 - 247	E9 – F7	N/A
248	F8	Define custom character 0. This command must be followed by eight data bytes.
249	F9	Define custom character 1. This command must be followed by eight data bytes.
250	FA	Define custom character 2. This command must be followed by eight data bytes.
251	FB	Define custom character 3. This command must be followed by eight data bytes.
252	FC	Define custom character 4. This command must be followed by eight data bytes.
253	FD	Define custom character 5. This command must be followed by eight data bytes.
254	FE	Define custom character 6. This command must be followed by eight data bytes.
255	FF	Define custom character 7. This command must be followed by eight data bytes.

# Product Change Notice: Revision E and Earlier
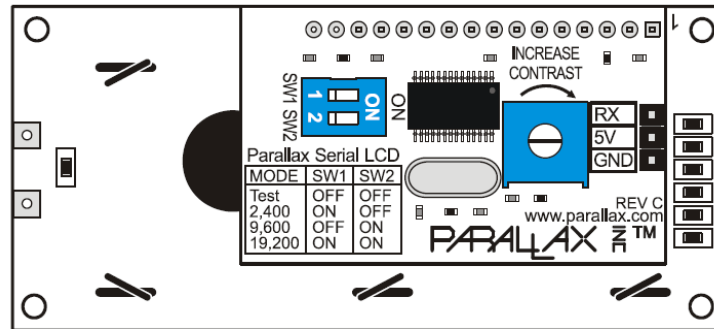
If your Serial LCD Display does not have a piezospeaker on the back and looks similar to the image above, please use the following information and specifications when using this product. All connections, test code and commands remain the same, except for commands controlling the piezospeaker.

Key Specifications

- Power requirements:
    - Non-backlit: +5 VDC, 20 mA
    - Backlit: +5 VDC, 20 mA (light off), 80 mA (light on)
- Communication: Selectable asynchronous serial baud rates: 2400, 9600, 19200
- Operating temperature: -4 to +158 °F (-20 to +70°C)
- Dimensions: **Board and LCD size and style may vary**
    - 2x16: Approx. 1.5 x 3.15 in (38 x 80 mm)
    - 4x20: Approx. 2.4 x 3.9 in (60 x 100 mm)

REV E CAUTION: DO NOT PROVIDE A SIGNAL TO THE 'RX' PIN BEFORE APPLYING 5 VDC TO THE '5V' PIN.
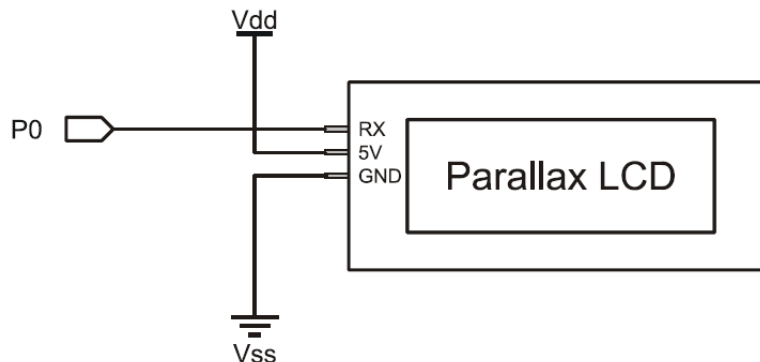
Revision History

Version 3.1: corrected current draw specification on Page 1 from 50 mA to 80 mA.