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

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Smart Sensors and Applications

Student Guide

VERSION 1.0

PARALLAX INC

WARRANTY

Parallax Inc. warrants its products against defects in materials and workmanship for a period of 90 days from receipt of product. If you discover a defect, Parallax Inc. will, at its option, repair or replace the merchandise, or refund the purchase price. Before returning the product to Parallax, call for a Return Merchandise Authorization (RMA) number. Write the RMA number on the outside of the box used to return the merchandise to Parallax. Please enclose the following along with the returned merchandise: your name, telephone number, shipping address, and a description of the problem. Parallax will return your product or its replacement using the same shipping method used to ship the product to Parallax.

14-DAY MONEY BACK GUARANTEE

If, within 14 days of having received your product, you find that it does not suit your needs, you may return it for a full refund. Parallax Inc. will refund the purchase price of the product, excluding shipping/handling costs. This guarantee is void if the product has been altered or damaged. See the Warranty section above for instructions on returning a product to Parallax.

COPYRIGHTS AND TRADEMARKS

This text is available in printed format from Parallax Inc. Because we print the text in volume, the consumer price is often less than typical retail duplication charges.

ISBN 1-928982-39-5

DISCLAIMER OF LIABILITY

Parallax Inc. is not responsible for special, incidental, or consequential damages resulting from any breach of warranty, or under any legal theory, including lost profits, downtime, goodwill, damage to or replacement of equipment or property, or any costs of recovering, reprogramming, or reproducing any data stored in or used with Parallax products. Parallax Inc. is also not responsible for any personal damage, including that to life and health, resulting from use of any of our products. You take full responsibility for your BASIC Stamp application, no matter how life-threatening it may be.

3RD PRINTING

We maintain active web-based discussion forums for people interested in Parallax products. These lists are accessible from www.parallax.com:

- Propeller chip – This list is specifically for our customers using Propeller chips and products.
- BASIC Stamp – This list is widely utilized by engineers, hobbyists and students who share their BASIC Stamp projects and ask questions.
- Stamps in Class® – Created for educators and students, subscribers discuss the use of the Stamps in Class curriculum in their courses. The list provides an opportunity for both students and educators to ask questions and get answers.
- Parallax Educators – A private forum exclusively for educators and those who contribute to the development of Stamps in Class. Parallax created this group to obtain feedback on our curricula and to provide a place for educators to develop and obtain Teacher's Guides.
- Robotics – Designed for Parallax robots, this forum is intended to be an open dialogue for robotics enthusiasts. Topics include assembly, source code, expansion, and manual updates. The Boe-Bot®, Toddler®, SumoBot®, HexCrawler and QuadCrawler robots are discussed here.
- SX Microcontrollers and SX-Key – Discussion of programming the SX microcontroller with Parallax assembly language SX – Key® tools and 3rd party BASIC and C compilers.
- Javelin Stamp – Discussion of application and design using the Javelin Stamp, a Parallax module that is programmed using a subset of Sun Microsystems' Java® programming language.

ERRATA

While great effort is made to assure the accuracy of our texts, errors may still exist. If you find an error, please let us know by sending an email to editor@parallax.com. We continually strive to improve all of our educational materials and documentation, and frequently revise our texts. Occasionally, an errata sheet with a list of known errors and corrections for a given text will be posted to our web site, www.parallax.com. Please check the individual product page's free downloads for an errata file.

Table of Contents

Preface

INTRODUCTION AND AUTHOR'S NOTE

The first time I saw the term "smart sensor" was in Tracy Allen's *Applied Sensors* text (then known as *Earth Measurements*). Tracy aptly applied this term to the DS1620 digital thermometer, which has built-in electronics that simplify microcontroller temperature measurements. In addition, it can remember settings it receives from a microcontroller and even function on its own as a thermostat controller.

In contrast to smart sensors, primitive sensors are devices or materials that have some electrical property that changes with some physical phenomenon. An example of a primitive sensor from *What's a Microcontroller?* is the cadmium sulfide photoresistor. Its resistance changes with light intensity. With the right circuit and program, microcontroller light measurements are possible. Other examples of common primitive sensors are current/voltage output temperature sensors, microphone transducers, and even the potentiometer, which is a rotational position sensor.

Inside every smart sensor is one or more primitive sensors and support circuitry. The thing that makes a smart sensor "smart" is the additional, built-in electronics. The electronics make these sensors able to do one or more of the following:

- Pre-process their measured values into meaningful quantities
- Communicate their measurements with digital signals and communication protocols
- Orchestrate the actions of primitive circuits and sensors to "take" measurements
- Make decisions and initiate action based on sensed conditions, independent of a microcontroller
- Remember calibration or configuration settings

During my first encounter with a smart sensor, I thought to myself, "Wow, an entire kit full of these smart sensors with a book could be REALLY interesting! I sure hope somebody does a kit and book like that soon..." Little did I know that "soon" would end up being almost six years later, and that "somebody" would turn out to be me. And if one of my bosses was to have told me back then that the kit would contain an accelerometer, ultrasonic rangefinder, digital compass, and a serial LCD for mobile measurements, I might just have come completely unglued. Since it was only recently possible for us to

put together such an awesome group of components into a single kit, I'd have to say it was worth the wait.

In keeping with the rest of the Stamps in Class tutorials, this book is a collection of activities, some of which cover basics, some more advanced, and some demonstrate applications or building blocks for various products and/or inventions. The first half of the book introduces each sensor, along with some mobile LCD displayed measurements. Then, the second half of the book has lots of applications for you to try, such as tilt video games, custom measurement tools, and diagnostic devices for hobby and sports pursuits. The page limit to keep these books inside our packaging is 350, and it was kind of difficult to stop there. Additional Smart Sensor activities for the Boe-Bot robot can be found in the Stamps in Class forum at www.parallax.com.

While this book covers the basics and demonstrates some example applications, it really only scratches the surface of what you can do with these devices. The main purpose of this book is to provide some building blocks and ideas for future class projects and inventions. For example, after finishing chapter 3, our book reviewer Kris Magri put her Board of Education with the accelerometer and LCD on her dashboard, and now her car has a flatland acceleration meter along with the speedometer. With a few modifications to the code, it could be made into a rollover warning system for 4-wheel drive. After looking at the mechanical sighting device used to predict avalanche conditions in mountainous areas based on hill incline, Ken Gracey whipped up the digital version in one night with the same parts that went onto Kris's dashboard.

The dashboard acceleration and avalanche risk meters are just two novel examples of the many applications, projects, and inventions that the Smart Sensors kit and text can inspire. We'd like to see what you do with your kit on the Stamps in Class forum. It doesn't matter whether you think your project turned out to be cool, unique, corny, or whatever. Just take a few minutes to post things you've made with theses smart sensors to http://forums.parallax.com/forums/ → Stamps in Class. Make sure to include a few snapshots, a brief description, and preferably the schematic and PBASIC program.

So, have fun with this kit and book, and we'll look forward to seeing your inventions on the Stamps in Class forum.

OVERVIEW

The smart sensors kit contains four devices that, when used with the BASIC Stamp and Board of Education or HomeWork Board, can be building blocks for a variety of inventions and student projects. Here is a list of the devices:

- Parallax 2x16 Serial LCD
- Ping))) Ultrasonic Rangefinder
- Memsic 2125 2-Axis Accelerometer
- Hitachi HM55B Compass Module

Aside from providing both the equipment and how-to information for student projects, this text has two major emphases that provide theory, examples, and required calculations, which can be used to reinforce a variety of measurement, physics/engineering, and trigonometric concepts. These emphases are:

- Math techniques for scaling raw sensor values into measurements that are meaningful because they are expressed in common unit systems.
- Interpreting the projection of gravity and magnetic vector fields onto Cartesian sensing axes.

BEFORE YOU START

To perform the experiments in this text, you will need to have your Board of Education or HomeWork Board connected to your computer, the BASIC Stamp Editor software installed, and to have verified the communication between your computer and your BASIC Stamp. For detailed instructions, see *What's a Microcontroller?* which is available as a free download from www.parallax.com. You will also need the parts contained in the Smart Sensors Parts Kit. For a full listing of system, software, and hardware requirements, see Appendix D.

THE STAMPS IN CLASS EDUCATIONAL SERIES

The Stamps in Class series of texts and kits provides affordable resources for electronics and engineering education. All of the books listed are available for free download from www.parallax.com. The versions cited below were current at the time of this printing. Please check our web sites www.parallax.com or www.stampsinclass.com for the latest revisions; we continually strive to improve our educational program.

Stamps in Class Student Guides:

What's a Microcontroller? is the recommended entry level text to the Stamps In Class educational series. Some students instead start with *Robotics with the Boe-Bot*, also designed for beginners.

"*What's a Microcontroller?*", Student Guide, Version 2.2, Parallax Inc., 2004
"*Robotics with the Boe-Bot*", Student Guide, Version 2.2, Parallax Inc., 2004

You may continue on with other Educational Project topics, or you may wish to explore our other Robotics Kits.

Educational Project Kits:

The following texts and kits provides a variety of activities that are useful to hobbyists, inventors and product designers interested in trying a wide range of projects.

"*Smart Sensors and Applications*", Student Guide, Version 1.0, Parallax Inc., 2006
"*Process Control*", Student Guide, Version 1.0, Parallax Inc., 2006
"*Applied Sensors*", Student Guide, Version 1.3, Parallax Inc., 2003
"*Basic Analog and Digital*", Student Guide, Version 1.3, Parallax Inc., 2004
"*Understanding Signals*", Student Guide, Version 1.0, Parallax Inc., 2003

Robotics Kits:

To gain experience with robotics, consider continuing with the following Stamps in Class student guides, each of which has a corresponding robot kit:

"*IR Remote for the Boe-Bot*", Student Guide, Version 1.1, Parallax Inc., 2004
"*Applied Robotics with the SumoBot*", Student Guide, Version 1.0, Parallax Inc., 2005
"*Advanced Robotics: with the Toddler*", Student Guide, Version 1.2, Parallax Inc., 2003

Reference

This book is an essential reference for all Stamps in Class Student Guides. It is packed with information on the BASIC Stamp series of microcontroller modules, our BASIC Stamp Editor, and our PBASIC programming languages.

"*BASIC Stamp Manual*", Version 2.2, Parallax Inc., 2005

FOREIGN TRANSLATIONS

Parallax educational texts may be translated to other languages with our permission (e-mail translations@parallax.com). If you plan on doing any translations please contact us so we can provide the correctly-formatted MS Word documents, images, etc. We also maintain a private discussion group for Parallax translators which you may join. This will ensure that you are kept current on our frequent text revisions.

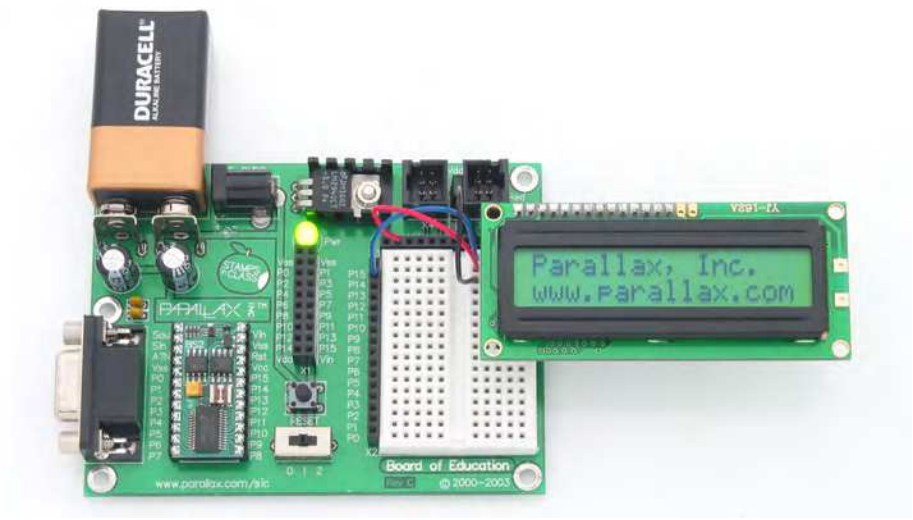
SPECIAL CONTRIBUTORS

Parallax Inc. would like to recognize their Education Team members: Project Manager Aristides Alvarez, Technical Illustrator Rich Allred, Graphic Designer Larissa Crittenden, Technical Reviewer Kris Magri, and Technical Editor Stephanie Lindsay. In addition, thanks go to customer Steve Nicholson for test-driving most of the activities. As always, special thanks go to Ken Gracey, the founder of Parallax Inc.'s Stamps in Class educational program.

Chapter 1: The Parallax Serial LCD Display

Displaying the information a sensor sends in a readable format has many uses, and in some applications, it's all that matters. The digital thermometer is a common example that can be found in many households. Inside each digital thermometer, there is a temperature probe, a microcontroller, and a liquid crystal display (LCD) for displaying the measurements. The BASIC Stamp microcontroller and Parallax Serial LCD shown in Figure 1-1 can provide the microcontroller and display elements for this type of product. This setup is also great for displaying mobile measurements, making it possible to disconnect your board from the PC and Debug Terminal and field-test your smart sensors.

Figure 1-1: BASIC Stamp, Board of Education, and Parallax Serial LCD



The activities in this chapter introduce some Parallax LCD basics, like connecting the LCD to the BASIC Stamp, turning it on and off, placing its cursor, and displaying text and digits. Later chapters will introduce creating and animating custom characters and displaying scrolling messages.

LCDS IN PRODUCTS

The products shown in Figure 1-2 all have liquid crystal displays. They are easy to read, and the smaller ones consume very little power. Think about how many products you own with liquid crystal displays. Think also as you go through these activities about the various BASIC Stamp projects, prototypes and inventions you've got in the works, and how a serial LCD might enhance or help them to completion.

Figure 1-2: Product Examples with LCD Displays

Clockwise from top-left: cell phone, portable GPS unit, calculator, digital multimeter, office clock, laptop computer, oscilloscope, office phone.

THE PARALLAX SERIAL LCD - YOUR MOBILE DEBUG TERMINAL

If you've worked through any of the other Stamps in Class texts, you're probably familiar with what a valuable tool the Debug Terminal can be. The Debug Terminal is a window that you can use to make your computer display messages it receives from the BASIC Stamp. It's especially useful for displaying diagnostic messages and variable values,

making it easier to isolate program bugs. It's also a handy tool for testing circuits, sensors, and more.

The Debug Terminal has one drawback, and that's the serial cable connection. Consider how many times it wasn't convenient to have your board connected to the computer to test a sensor, or find out what the Boe-Bot robot was "seeing" with its infrared object detectors in another room. These are all situations that can be remedied with the Parallax Serial LCD shown in Figure 1-3. Once you've built up a sensor circuit on the Board of Education, you can use a battery and the Parallax Serial LCD to take the setup as far away from your programming station as you want, all the while displaying sensor measurements and other diagnostic information.

Figure 1-3
Parallax (2×16)
Serial LCD

The Parallax 2×16 Serial LCD has two sixteen-character-wide rows for displaying messages. The display is controlled by serial messages from the BASIC Stamp. The BASIC Stamp sends these messages from a single I/O pin that is connected to the LCD's serial input. There are two versions, standard and backlit:

Version	Parallax Part #
Standard	27976
Backlit	27977

Serial Vs Parallel LCDs

The Parallel LCD is probably the most common type of LCD. It typically requires a minimum of 6 I/O pins for the BASIC Stamp to control. Also, unless you are using a BASIC Stamp 2p, 2pe, or 2px, the code for controlling them tends to be more complex than serial LCD code.

The serial LCD is actually just a parallel LCD with an extra microcontroller. The extra microcontroller on the serial LCD converts the serial messages from the BASIC Stamp to the parallel messages that control the parallel LCD.

ACTIVITY #1: CONNECTING AND TESTING THE LCD

Along with the electrical connections and some simple PBASIC test programs for the Parallax Serial LCD, this activity introduces the **SEROUT** command. It also demonstrates how **DEBUG** is just a special case of **SEROUT**. This is especially useful for working with your serial LCD because you can take many of the **DEBUG** command arguments and use them with the **SEROUT** command to control and format the information your LCD displays.

Parts Required

(1) Parallax 2×16 Serial LCD
(3) Jumper wires

In addition to the Parallax Serial LCD and three wires, it will be especially important to have the Parallax Serial LCD Documentation (included as Appendix B in this text). Although it's only a few pages, it has a long list of values you can send to your LCD to make it perform functions similar to those you've used with the Debug Terminal. Cursor control, carriage returns, clear screen and so on, all have their own special codes. In some cases, these codes are identical to the ones for the Debug Terminal; in other cases, they are quite different.

Building the Serial LCD Circuit

Connecting the Parallax Serial LCD to the BASIC Stamp is amazingly simple, as shown in Figure 1-4. You only need to make three connections: one for power, one for ground, and one for signal. The LCD's RX pin is for the signal and should be connected to a BASIC Stamp I/O pin. In this activity, we will use P14. The LCD's GND pin should be connected to Vss on the Board of Education, and the LCD's 5 V pin should be connected to Vdd.

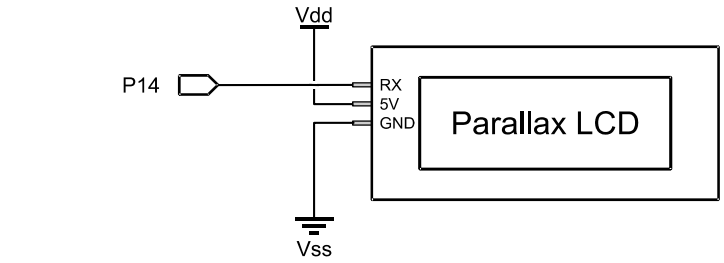
CAUTIONS: Wiring mistakes can damage this LCD.

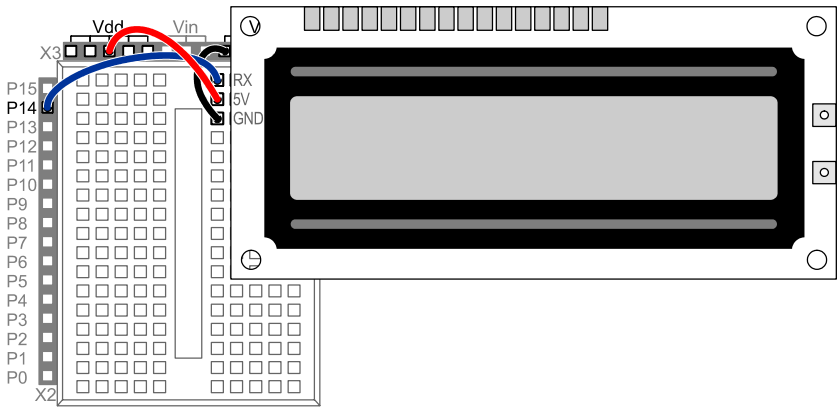
Rev D and earlier models if this LCD had five pins. If you have a 5-pin model, please see Figure B-1 on page 306 to verify the correct pins to use in the circuits in this book.

The five-pin version is NOT pin compatible with Scott Edwards or Matrix Orbital models. If you have used other brands of serial LCDs before, be aware that this LCD's pinout is different. Don't make the mistake of using the same wiring that you used for other models.

- √ Disconnect power from your Board of Education.
- √ Connect the Board of Education's Vss socket to the LCD's GND pin.
- √ Connect the Board of Education's P14 socket to the LCD's RX pin, as shown in Figure 1-4.
- √ Connect the Board of Education's Vdd socket to the LCD's 5V pin
- √ Do not turn the power back on yet.

Figure 1-4: Schematic and Wiring Diagram





Testing the Serial LCD

The Parallax Serial LCD has a self-test mode you can use to make sure it's in working order and that the contrast is properly set. Figure 1-5 shows the back of the LCD module. The switches labeled (SW1 and SW2) are for self-test mode and baud rate adjustment, and there is a contrast adjustment potentiometer labeled "INCREASE CONTRAST."

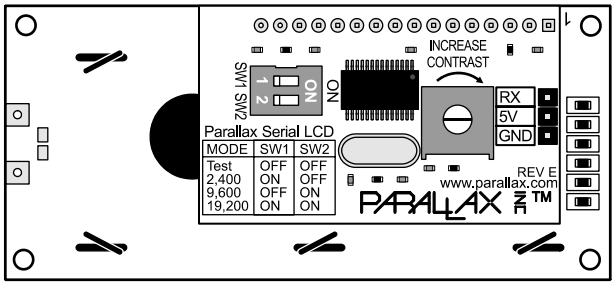


Figure 1-5
LCD Module - Back View

- √ The power to your board should still be off.
- √ Find SW1 and SW2 on the underside of the LCD module shown in Figure 1-6 .
- √ Set SW1 off.
- √ Set SW2 off.
- √ Turn the power back on now.

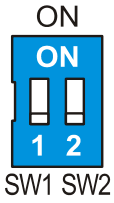


Figure 1-6
Setting Baud Rate Switches to Self-test Mode

- √ When you turn the power back on, the LCD should display the text "Parallax, Inc." on the top line (Line 0) and "www.parallax.com" on the bottom line (Line 1), as you can also see in Figure 1-3. If you leave the LCD in this mode for a while, a custom character reminiscent of 1980's video games will appear and eat all the text.

√ If the display seems dim or looks blank, there is a contrast adjustment potentiometer shown in Figure 1-7 that you can turn with a screwdriver. If the display is already clear and all the characters look good, you probably don't need to adjust it. If the characters are too dim, or they appear in gray boxes, adjusting the potentiometer should help.

√ Adjust the contrast potentiometer if needed.

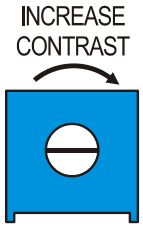


Figure 1-7
Contrast
Adjustment
Potentiometer

Adjusting the LCD to Receive Messages from the BASIC Stamp

Serial communication involves a baud rate. That's the number of bits per second (bps) the sender transmits, and the receiver has to be ready to receive data at the same baud rate. In this chapter's activities, the BASIC Stamp will be programmed to send messages to the LCD at 9600 bps. You can adjust the same switches you used for the LCD self-test to set this baud rate.

√ Turn the Board of Education power off.
√ Leave SW1 in the OFF position.
√ Set SW2 to ON as shown in Figure 1-8.
√ Turn the power back on now.

The screen will remain blank until you program the BASIC Stamp 2 to control the display.

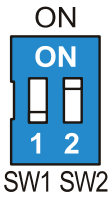


Figure 1-8
Baud Rate Switches to 9600 bps

Figure 1-9 shows the mode table printed on the back of the Parallax Serial LCD. If you want to send messages at other baud rates, (2400 or 19,200 bps), use this table and adjust SW1 and SW2 accordingly.

Parallax Serial LCD

MODE	SW1	SW2
Test	OFF	OFF
2,400	ON	OFF
9,600	OFF	ON
19,200	ON	ON

Figure 1-9
Baud Rate Switch Settings

ACTIVITY #2: DISPLAYING SIMPLE MESSAGES

As mentioned earlier, the commands that send text, numbers, formatters and control codes (control characters) to a serial LCD are related to the **DEBUG** command. In fact, the **DEBUG** command is just a special version of a more general command called **SEROUT**. The **SEROUT** command has many uses, some of which include sending messages to serial LCDs, other BASIC Stamp modules, and computers.

In this activity, you will program the BASIC Stamp to make the LCD display text messages and numeric values. As a first step into animation, you will also modify the programs to flash the text and numbers on and off. The **SEROUT** command will be your tool for accomplishing these tasks. You will use the **SEROUT** command to send text, numbers, control codes, and formatters to the Parallax Serial LCD. As you will soon see, the text, numbers, and formatters are identical to ones you use with the **DEBUG** command. The control codes will be a little different, but with some practice, they'll be just as easy to use as **CR**, **CLS**, **HOME**, and **CRSRXY**. (If you are not familiar with **CRSRXY**, you can learn more about it in Chapter 5, Activity #1.)

The minimal version of the **SEROUT** command's syntax looks like this:

SEROUT ***Pin, BaudMode,*** **[** ***DataItem,*** **{*****DataItem, ...*****}** **]**

In our programs, the ***Pin*** argument has to be 14 since the LCD's RX (receive data) pin is connected to BASIC Stamp I/O pin P14.

The ***BaudMode*** argument is a value that tells the BASIC Stamp how fast to send the serial data, and it determines some of the serial signal characteristics as well. The BASIC Stamp Editor's Help program has tables that give the ***BaudMode*** values for common baud rates and signals. It turns out that 84 is the ***BaudMode*** argument for 9600 bits per second (bps), 8 data bits, no parity, true signal. This is exactly what the Parallax Serial LCD is designed to receive.

DataItem arguments can be the text between quotes like "Hello". They can also be control characters like **CR**, **CLS**, or values, with or without the formatters like **DEC**, **BIN**, and **?**. If they are sent with formatters, they are sent as the characters that represent the value. If they are sent without formatters, they will be sent as values, like 22, 12, and 13. We can send unformatted values like these to the LCD, which will interpret them as control codes.

More about SEROUT

If you want to try using the Debug Terminal with **SEROUT** instead of **DEBUG**, first open it from the toolbar via Run → Debug → New. Next, select Run → Identify to see which port your BASIC Stamp is using. Then, in the Debug Terminal, set the Com Port to match. Note that you can also change the Debug Terminal's Baud Rate and other communication parameters.

There's lots more to learn about the **SEROUT**. Both the BASIC Stamp Manual and the BASIC Stamp Editor's PBASIC Syntax Guide give the **SEROUT** command extensive coverage. The BASIC Stamp Manual is available for free download from www.parallax.com → Downloads → Documentation. If your BASIC Stamp Editor supports PBASIC 2.5, you probably already have the PBASIC Syntax guide. To access it, simply select Index from the BASIC Stamp Editor's Help menu.

Simple Text Messages and Control Codes

Unlike the Debug Terminal, the serial LCD needs to be turned on with a command from the BASIC Stamp. The LCD has to receive the value 22 from the BASIC Stamp to

activate its display. Here's the PBASIC command for sending the serial LCD the value 22:

```
SEROUT 14, 84, [22]
```

Used in this way, **22** is an example of an LCD control code. Here's a list of some more basic control codes:

- **12** clears the display. Note: always follow with **PAUSE 5** to give the LCD time to clear the display.
- **13** is a carriage return; it sends the cursor to the next line.
- **21** turns the LCD off.
- **22** turns the LCD on.

Commands to turn the backlighting on and off (for the backlit LCD only):

Some LCDs have backlighting so that you can read them when it's dark. If you have the backlit version of the Parallax Serial LCD (part number 27977), you can control the backlighting with these values:

- **17** turns the backlighting on.
- **18** turns the backlighting off.

In PBASIC, CR is a predefined constant for the value 13. Whenever you use the constant **CR** in a **DEBUG** command, it sends the value 13 to the Debug Terminal. The Debug Terminal moves the cursor to the beginning of the next line whenever it receives the value 13. In this case, the two commands below are the equivalent:

```
SEROUT 14, 84, ["See this?", CR, "The LCD works!"]

SEROUT 14, 84, ["See this?", 13, "The LCD works!"]
```

While this works with CR, it does not work for other predefined PBASIC constants. For example, **CLS**, which is a predefined constant for the number 0, does not clear the LCD display. The Parallax Serial LCD equivalent of **CLS** is **12**. Likewise, **HOME**, which is a predefined constant for the value 1, does not send the cursor to the top left "home" character in the LCD display. The control code **128** does that for the Parallax Serial LCD.

Example Program - LcdTestMessage.bs2

√ Enter, save, and run LcdTestMessage.bs2. Verify that it displayed the message "See this?" on Line 0 and "The LCD works!" on Line 1, as in Figure 1-10.

```
' Smart Sensors and Applications - LcdTestMessage.bs2
' Display a test message on the Parallax Serial LCD.

' {$STAMP BS2}                              ' Target device = BASIC Stamp 2
' {$PBASIC 2.5}                             ' Language      = PBASIC 2.5

SEROUT 14, 84, [22, 12]                     ' Initialize LCD
PAUSE 5

SEROUT 14, 84, ["See this?", 13,            ' Text message, carriage return
                "The LCD works!"]           ' more text on Line 1.
END                                         ' Program end
```

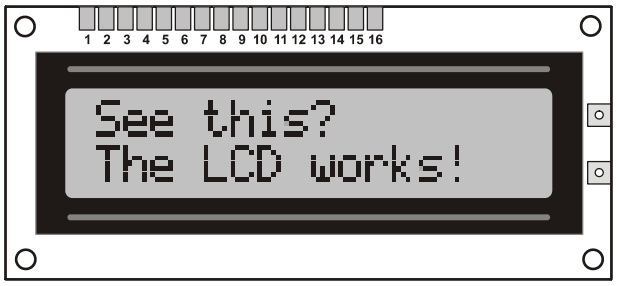


Figure 1-10
Text Display

If the LCD didn't display properly: Double-check your wiring, your program, and the SW settings on the back of the LCD. Also try disconnecting and reconnecting power to your Board of Education. If needed, Go through the check-marked instructions leading up to this program, and verify that each one was completed correctly.

Your Turn - Control Codes to Make the Display Flash On/Off

Remember that **22** turns the display on, and **21** turns it off? You can use these control codes to make the text flash on and off.

√ Replace the **END** command in LcdTestMessage.bs2 with this code block.

```
DO                                  ' Start DO...LOOP code block
  PAUSE 600                         ' 6/10 second delay
  SEROUT 14, 84, [22]               ' Turn display on
  PAUSE 400                         ' 4/10 second delay
  SEROUT 14, 84, [21]               ' Turn display off
LOOP                                ' Repeat DO...LOOP code block
```

√ Run the modified program and note the effect.

Display Numbers with Formatters

Most of the formatters that worked for displaying numbers with the Debug Terminal can also be used with the Parallax Serial LCD. The **DEC** formatter is probably the most useful, but you can also use **DIG**, **REP**, **ASC**, **BIN**, **HEX**, **SDEC**, and most of the others. For example, if you want to display the decimal value of a variable named **counter**, you can use commands like this:

```
SEROUT 14, 84, [DEC counter]
```

Example Program - LcdTestNumbers.bs2

Aside from demonstrating that you can display variable values on the serial LCD, this program also shows what happens if the program sends more than 16 printable characters to Line 0. It wraps to Line 1. Also, after printing sixteen more characters and filling Line 1, the text will wrap again, to Line 0.

√ Enter, save, and run LcdTestNumbers.bs2

```
' Smart Sensors and Applications – LcdTestNumbers.bs2
' Display number values with the Parallax Serial LCD.

' {$STAMP BS2}                              ' Target device = BASIC Stamp 2
' {$PBASIC 2.5}                             ' Language      = PBASIC 2.5

counter        VAR     Byte                 ' FOR...NEXT loop index

SEROUT 14, 84, [22, 12]                     ' Initialize LCD
PAUSE 5                                     ' 5 ms delay for clearing display

FOR counter = 0 TO 12                       ' Count to 12; increment at 1/2 s

  SEROUT 14, 84, [DEC counter, " "]
  PAUSE 500

NEXT

END                                         ' Program end
```

√ Verify that the display resembles Figure 1-11.

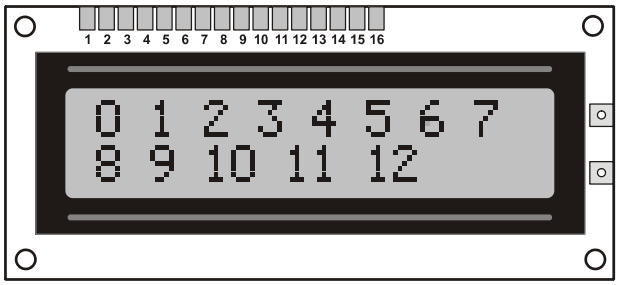


Figure 1-11
Display Numbers

Your Turn - Other Formatters

- √ Try replacing **DEC** with **DEC2** and observe what happens.
- √ Repeat with the **?** formatter.
- √ If necessary, look these commands up in either the BASIC Stamp Manual or the BASIC Stamp Editor's Help. Also try them in the Debug Terminal.
- √ What are the similarities and differences between using these formatters in the Debug Terminal and using them in the Parallax Serial LCD?

Control Codes for Cursor Positioning

The LCD's control codes are different from the **DEBUG** command's control characters. For example, **HOME**, and **CRSRXY** just don't have the same effect they do with the Debug Terminal. However, there are cursor commands for the Parallax Serial LCD that you can use to control the cursor's X and Y coordinates. You can also send the cursor to the top-left "home position". Take a look at the LCD documentation's Command Set section beginning on page 312. It lists all the valid control commands for the LCD; below are a few examples from the list that control cursor position.

- **8** Cursor left
- **9** Cursor right
- **10** Cursor down (bottom line will wrap to top line)
- **128** to **143** Position cursor on Line 0, character 0 to 15
- **148** to **163** Position cursor on Line 1, character 0 to 15