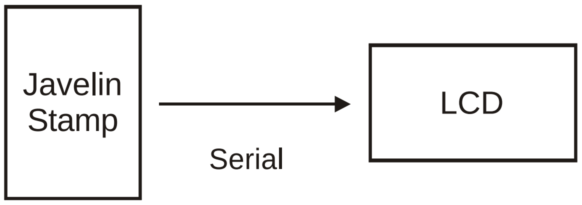
599 Menlo Drive, Suite 100
Rocklin, California 95765, USA
Office/Tech Support: (916) 624-8333
Fax: (916) 624-8003

Web Site: www.javelinstamp.com
Home Page: www.parallaxinc.com

General: info@parallaxinc.com
Sales: sales@parallaxinc.com
Technical: javelintech@parallaxinc.com



Contents

Introduction to BPI-216 Serial LCD Module

The BPI-216 module consists of a 2 line by 16 character (2x16) LCD with a built in serial interface. The serial interface allows the BPI-216 to be easily programmed.

With this class you can send text messages to the LCD display. Methods are included for you to create your own custom graphical characters which you can store in character-generator (CG) RAM. Once these characters are created, displaying them is easy. You also have control over the cursor, turning the display on or off, and scrolling messages.

This application note includes:

1) The BPI216 library, which contain all methods needed to program your BPI-216 serial LCD.
2) A test program that will help you verify the unit is working properly.
3) An animation application, which will show you how to create custom graphical characters and animate them.
4) A demonstration program, which will give you a step-by-step guide on how to access each method within the BPI216 library.

Downloads, Parts, and Equipment for the BPI-216

This application note (AppNote009b-BPI216.doc), the BPI216 library file (BPI216.java), the library's javadoc file (BPI216.pdf), the test program (BPI216Test.java), the demonstration program (BPI216Demo.java) which will demonstrate the methods available to you in the BPI216 library, and an application example (BPI216Animation.java) are all available to you for free download from:

http://www.javelinstamp.com/Applications.htm

You can use the AppNote009b-BPI216.exe, to install the files listed below. These files must be located in specific paths within the Javelin Stamp IDE directory. Although the path to this directory can be different, the default root path is: C:\Program Files\Parallax Inc\Javelin Stamp IDE

The file list below is organized by directory, then by filename, please verify that your file list is organized in the same way.

```
<root path>\doc\AppNote009b-BPI216.pdf
<root path>\doc\BPI216.pdf
<root path>\Projects\examples\peripheral\display\lcd\serial\BPI216Test.java
<root path>\Projects\examples\peripheral\display\lcd\serial\BPI216Demo.java
<root path>\Projects\examples\peripheral\display\lcd\serial\BPI216Animation.java
<root path>\lib\stamp\peripheral\display\lcd\serial\BPI216.java
```

Table 9.1 lists the parts you will need for this application note.

Table 9.1: Parts List

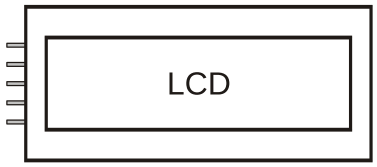
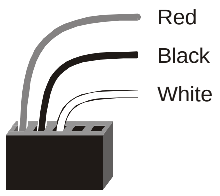
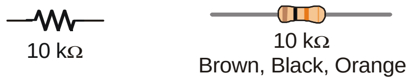
Quantity	Part Ordering Info and Part Description	Schematic Symbol/Pin Map
1	BPI-216 - 2x16 Serial LCD Parallax Part #27910 *(non-backlit)* - or - Parallax Part #27923 *(backlit)*	
1	LCD Serial cable Parallax Part #27946	

Table 9.2 lists the *optional* parts you will need to run the BPI-216Demo.java program. For more information please see the section *BPI216 Demonstration Program*.

Table 9.2: Optional Parts List

Quantity	Part Ordering Info and Part Description	Schematic Symbol/Pin Map
1	Resistor 10 kΩ ¼ W-5% Parallax Part # 150-01030	
1	Tact switch Parallax Part #400-00002	

The equipment used to test this example includes a Javelin Stamp, Javelin Stamp Demo Board, 7.5 V, 1000 mA DC power supply, serial cable, and PC with the Javelin Stamp IDE v2.01.

BPI-216 Example Circuit

The BPI-216 needs only three connections to successfully connect to the Javelin and utilize the BPI216 library; see Figure 9.1. Simply connect your LCD cable to the back of the BPI-216 header, and make the following connections to the Javelin.

- ✓ The red wire from the LCD cable (+5 V) connects to Vdd on the Javelin demo board.
- ✓ The black wire from the LCD cable (ground) connects to Vss on the Javelin demo board.
- ✓ The white wire from the LCD cable (data) connects to P10 of the Javelin demo board.

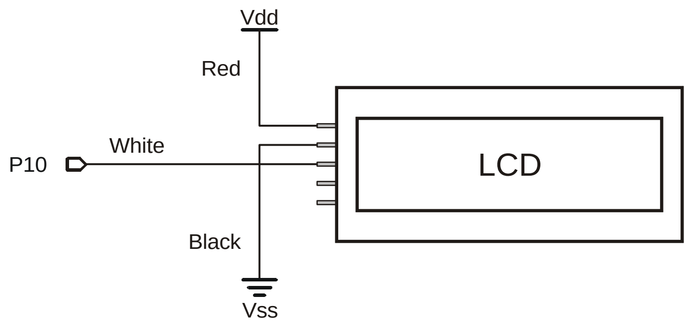


Figure 9.1: Wiring diagram for the BPI-216

Tip	The BPI-216 has a header with five pin connections *(+5, GND, SER, GND, +5)*, the LCD cable has three wires on a five pin header *(empty, empty, white, black, red)*. The LCD cable needs to connect the red wire to either +5 pin, the black wire to either GND pin and the white to the SER pin on the BPI-216.

Testing the BPI-216 LCD

Program Listing 9.1 is a short program that will verify that the circuit in Figure 9.1 is working properly. This program will display “Hello World” on the BPI-216 LCD display.

The first line of code will create a constant LCD_PIN to connect pin 15 (I/O pin P10) on the Javelin Stamp to the serial data pin on the BPI-216 (SER).

```
final static int LCD_PIN = CPU.pin10;
```

Next we need to create a UART so we can communicate to the BPI-216. This UART will be for transmitting data to the BPI-216 with inverted logic, using a 9600 baud rate, with one stop bit.

```
static Uart txOut = new Uart( Uart.dirTransmit, LCD_PIN,
                    Uart.invert, Uart.speed9600, Uart.stop1);
```

Now that we have the correct Javelin pin and UART defined, we can create a BPI216 object. Here we create a new BPI216 object called **myLCD**.

```
static BPI216 myLCD = new BPI216(txOut);   // create BPI216 object
```

Since the object has been created we can enter the main program. The first thing we will do is clear the screen of the BPI-216. To do this we call the **CLS()** method.

```
myLCD.CLS();                                // clear the screen
```

Now that the screen is clear, we can send it a message. Here we will call the **write()** method and send the message "Hello World!" to be displayed.

```
myLCD.write("Hello World!");                // display message
```

The test program (Program Listing 9.1) will display output on the LCD similar to Figure 9.2.

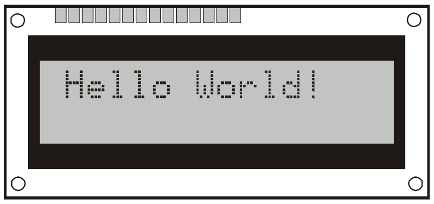


Figure 9.2
BPI-216 display from
Program Listing 9.1

If the output on your display is blank, verify that the leads from the BPI-216 are connected to the Javelin correctly see Figure 9.1.

If the output on your display is does not look like that of Figure 9.2, but does display characters of some sort, the BPI-216 might be set at the wrong baud rate. Verify that the BPI-216 is set at 9600 baud by checking the switch on the back of the unit. The switch marked 24 and 96 should be in the 96 position. Check the users manual of your BPI-216 for more information. You will need to power the unit off and back on again for it to enter the new baud rate mode.

Creating Custom Characters for the BPI-216

The BPI-216 allows for you to create custom characters. The 128 built-in characters are stored in ROM and will automatically be displayed when you print the corresponding character's memory location. There are 8 character generated (CG) RAM locations (0-7) that you can use to store your own custom characters.

Each of these programmable characters is five pixels wide and eight pixels high (5x8). It takes eight bytes to define each character, one byte per row (the three left-most bits are ignored). Here are the steps needed to create you own custom character (please refer to Figure 9.3).

1) Using a 5x8 block grid (see Figure 9.3) draw out the image that you would like to create.
2) Next create the binary values for each row where a solid block would represent the 1 and an empty block would represent a zero. For example, row 1 (byte 1) would have two 1's representing the 2 solid blocks and three 0's, representing the 3 empty block (in the following sequence: 01010). Now using a calculator such as the one that is built into Microsoft's Operating Systems (use advanced mode) enter the binary number, ignoring the preceding zeros, and convert this number to decimal. Try a converting a few rows from Figure 9.3 yourself.
3) Next you will need to store these numbers into a character array. The array location zero will contain byte zero, array location 1 will contain byte 1, etc. When you're finished your array should look like this:

```
final static char[] SMILEY = {0,10,0,4,17,14,0,0};
```

4) Now you will need to store your new character in 1 of 8 CGRAM locations (0-7). Here's how to store the array **`SMILEY`** into CGRAM location #3:

```
myLCD.createChar5x8(3, SMILEY);
```

5) To display this character on the BPI-216 simply pass the CGRAM location into the **`write()`** method like this:

```
myLCD.write(3);
```

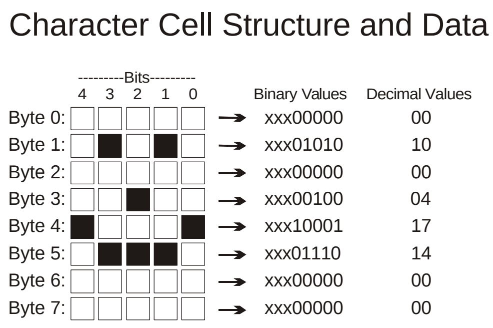


Figure 9.3
LCD Character Structure

Off Screen Character RAM

The BPI-216 has 2 rows and 16 columns (2x16). But, there is extra RAM that effectively allows you to store characters for a 2x40 display (see Figure 9.4). You can only physically view a 2x16 segment at any given time. This allows you to scroll messages off the screen, effectively moving the 2x16 display window within the 2x40 display space. For a complete demonstration of this see the Demonstration section of this application note for information on the BPI216Demo.java class.

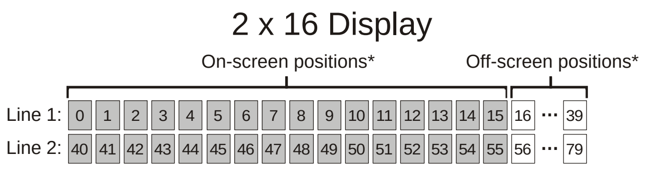

2 x 16 Display
On-screen positions*
Off-screen positions*
Line 1: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ··· 39
Line 2: 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 ··· 79

Figure 9.4
LCD
Character
Structure

Program Listing 9.1 – The BPI-216 Test

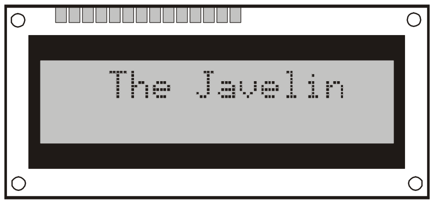
```
import stamp.core.*;
import stamp.peripheral.display.lcd.serial.BPI216;

/* Test program to verify the BPI-216 serial LCD is properly configured.
 *
 * For more information see AppNote009 from Parallax Inc.
 * Version 1.0 Dec. 24th, 2002
 */
public class BPI216Test {

  final static int LCD_PIN = CPU.pin10;
  static Uart txOut   = new Uart( Uart.dirTransmit, LCD_PIN, Uart.invert,
                                  Uart.speed9600, Uart.stop1);
  static BPI216 myLCD = new BPI216(txOut);  // create BPI216 object

  public static void main() {
    myLCD.CLS();                            // clear the screen
    myLCD.write("Hello World!");            // display message
  }// end main
}// end class
```

The BPI-216 Animation Application

Program Listing 9.2 uses the same connections shown in Figure 9.1 to create an animation sequence with custom generated characters. These custom characters are created by passing a character array and a character generated (CG) RAM location into the **`createChar5x8()`** method. Then a simple message is displayed on the screen, and an animated character based upon 3 custom CGRAM locations will move across the display. As this "character" moves across the display it will replace the existing message with a new one.

The output shown in Figure 9.5 should appear in the Messages from the Javelin Window/Debugger. If your display is blank or garbage characters appear, please go to the section titled *Testing the PBI-216 LCD.*

Figure 9.5
BPI-216 display for
Program Listing 9.2



BPI216Animation.java uses the following methods from the BPI216 library:

- **`CLS()`**
  Clears LCD and returns cursor to row 1, column 0
- **`write()`**
  - Writes specified CGRAM location on LCD at cursor position
  - Writes string on LCD at cursor position
- **`cursorMove()`**
  Moves LCD cursor to specified row and column position
- **`displayOff()`**
  Turns display off without changing contents of RAM
- **`displayOn()`**
  Turns display on, no cursor

Program Listing 9.2 – BPI216Animation

```
import stamp.core.*;
import stamp.peripheral.display.lcd.serial.BPI216;

/* Will display an animated sequence with custom generated characters for the
 * BPI-216 serial LCD.
 *
 * For more information see AppNote009 from Parallax Inc.
 * Version 1.0 Dec. 24th, 2002
 */
public class BPI216Animation {

  final static int LCD_PIN = CPU.pin10;              // define Javelin I/O pin
  static Uart txOut   = new Uart( Uart.dirTransmit, LCD_PIN, Uart.invert,
                                  Uart.speed9600, Uart.stop1 );  // create Uart
  static BPI216 myLCD = new BPI216(txOut);           // create LCD object

  // Define Graphic bitmaps
  final static char[] MOUTH0 = {0x0E,0x1F,0x1F,0x1F,0x1F,0x1F,0x0E,0x00};
  final static char[] MOUTH1 = {0x0E,0x1F,0x1F,0x18,0x1F,0x1F,0x0E,0x00};
  final static char[] MOUTH2 = {0x0E,0x1F,0x1C,0x18,0x1C,0x1F,0x0E,0x00};
  final static char[] SMILEY = {0x00,0x0A,0x0A,0x00,0x11,0x0E,0x06,0x00};
  final static int[]  CELLS  = {2,1,0,1};            // animation sequence

  static String msg = new String(" IS VERY COOL!  ");

  public static void main() {

    // Store custom characters in CGRAM.
    myLCD.createChar5x8(0, MOUTH0);
    myLCD.createChar5x8(1, MOUTH1);
    myLCD.createChar5x8(2, MOUTH2);
    myLCD.createChar5x8(3, SMILEY);

    while (true) {

      myLCD.CLS();
      CPU.delay(5000);
      myLCD.write("  The Javelin");
      CPU.delay(10000);

      // Animation cycle.
      for (int addr = 0; addr < 16; addr++) {        // loop length of screen
        for (int cycle = 0; cycle <= 4; cycle++) {   // cycle CG graphics
          myLCD.cursorMove(1, addr);
          if (cycle < 4) {
           myLCD.write(CELLS[cycle]);                // write CG image # of CELLS
          }// if
          else {
           myLCD.write(msg.charAt(addr));            // write msg up to cursor
          }// else
          CPU.delay(750);
        }// for cycle
      }// for addr

      // Flash display on/off.
      for (int cycle = 0; cycle <= 4; cycle++) {
        myLCD.displayOff();
        CPU.delay(2500);
        myLCD.displayOn();
        CPU.delay(2500);
      }// for cycle
      CPU.delay(5000);
    }// while
  }// main
}// class
```

BPI216 Demonstration Program

(BPI216Demo.java) is a detailed, step-by-step, fully comprehensive demonstration of the BPI216 library class. The demonstration program is interactive by using a pushbutton to step you through each method of the library.

Here are the steps needed to build this pushbutton circuit (refer to Figure 9.6).

1) Connect I/O pin P1 of the Javelin Demo Board to the lower right pin of the push button.
2) Connect a 10 KΩ resistor from ground (Vss) to the lower right pin of the push button. This is the same connection point that you used for step 1.
3) Connect the top left pin of the push button to +5 V (Vdd)
4) Connect the top lead of the BPI216 (+5 V) to + 5 V (Vdd) on the Javelin Demo Board.
5) Connect the 2nd to the top lead of the BPI216 (GND) to GND (Vss) on the Javelin Demo Board.
6) Connect the middle lead of the BPI216 (SER) to Pin 10 (P10) of the Javelin Stamp

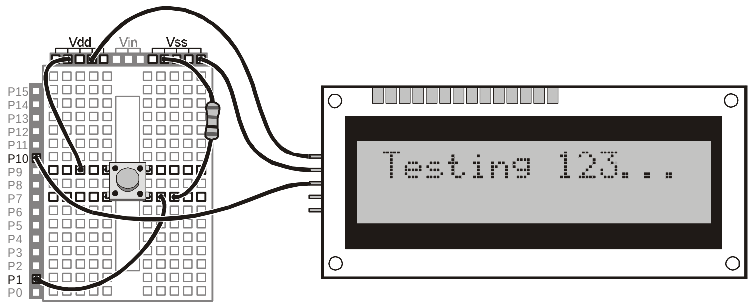


Figure 9.6: Pushbutton circuit for the BPI216Demo.java class

Published Resources – for More Information

This class was developed to allow the Javelin to interact with the BPI-216 serial LCD module. Much care has been taken in creating and documenting this application note. Below you will find useful information that was used in creating this document.

"BPI-216 Serial LCD Modules", User Manual, Scott Edwards Electronics, Inc., 07/00

This is the users manual for the BPI-216, it contains information on how to change the BPI-216 settings, the differences of the back-lit and no back-lit models, and much more.

"Basic Stamp Manual", Users Manual, Parallax Inc., 2000

The sections referenced from this book are for the `LCDCMD` command.

Javelin Stamp Discussion Forum – Questions and Answers

The Parallax, Inc. Javelin Stamp Discussion Forum is a searchable repository of design questions and answers for the Javelin Stamp. To view the Javelin Stamp Forum, go to www.javelinstamp.com and follow the Discussion link. You can also join this forum and post your own questions. The Parallax technical staff, and Javelin Stamp users who monitor the list, will see your questions and reply with helpful tips, part numbers, pointers to useful web pages, etc.