ACTIVITY #4: CUSTOM CHARACTERS AND LCD ANIMATION

While not every picture saves a thousand words, even the ones that only save a sentence or two are useful when you've only got 32 characters. One example of a useful picture is the hourglass picture the cursor on the computer screen uses to let you know that the program is busy. This simple animated icon works much better than a message somewhere on the screen that says, "please wait, the program is busy...". This activity uses the hourglass to introduce techniques for defining, storing, and displaying and animating custom characters.

Custom Characters in the Parallax LCD

The Parallax Serial LCD has room set aside for eight custom characters shown in Figure 1-13. To display custom character 0, just send the LCD the value 0 with the `SEROUT` command. Likewise, to display custom character 1, just send the value 1, to display custom character 2, send a 2, and so on. Note that custom characters 0 and 1 are pre-configured to be the backslash and tilde. Here is an example `SEROUT` command that displays both of them - `SEROUT 14, 84, [0, 1]`.

Example Program: PredefinedCustomCharacters.bs2

This example sends the serial LCD the two commands to make it display custom characters 0 and 1, the backslash "\" and tilde "~".

√ Enter and run the program, and verify that it displays the backslash and tilde

```
' Smart Sensors and Applications - PredefinedCustomCharacters.bs2
' {$STAMP BS2}
' {$PBASIC 2.5}
SEROUT 14, 84, [22, 12]                          ' Initialize LCD
PAUSE 5                                           ' 5 ms delay for clearing display
```
Defining (and Redefining) Custom Characters

The Parallax Serial LCD's custom characters are stored in its RAM. To define one of its eight custom characters, your SEROUT command has to tell the LCD which of the eight custom characters you are defining and then describe the on/off states of each pixel in the character.

Figure 1-14 shows the define commands you can send the LCD to tell it which custom character you are about to define. You can also think about it like this - to tell the LCD which custom character you are defining, send the value of the custom character plus 248. For example, if you want to define custom character 0, send 248, if you want to define custom character 1, send 249, and so on up to 255 for custom character 7.

Figure 1-14 Custom Character Define Commands

After sending the code that tells the LCD which custom character you are about to define, you have to then send eight bytes that describe the character. The LCD uses the lowest five bits of each byte it receives to describe each five pixel wide line in the character. Figure 1-15 shows an example of defining custom character 0 to be an hourglass that's just been turned upside down. Notice how each successive value in the SEROUT command corresponds to a row of pixels in the custom character. Notice also how the 1s correspond to black pixels, and the 0s correspond to white.
The SEROUT custom character definitions are not permanent. Each time the LCD's power is turned on and off the custom characters are erased. Since the BASIC Stamp and LCD share the same power supply, the BASIC Stamp's program also restarts when the power is reset. It's a good practice to define custom characters you plan to use at the beginning of a program, so that the BASIC Stamp can define the custom characters every time it's power is connected.

Here is another custom character definition of an hourglass with the sand drained into its bottom chamber. This definition uses 255 to tell the LCD to define it as custom character 7. It also uses a technique for drawing the characters with asterisks in the comments to the right of the SEROUT command. Start with a SEROUT command with all the binary values set to %00000, then draw the character with asterisks in the comment to the right. After it looks right, use the asterisks to dictate which zeros should be changed to ones.

SEROUT 14, 84, [255,                         ' Define custom character 7
%00000,                      '*   *   *   *'
%11111,                      '*     '*
%00001,                      '*   *   *'
%10001,                      '*   *   *'
%01010,                      '*   *   *'
%00100,                      '*     '*
%01110,                      '*     '*
%11111,                      '*     '*
%11111]                      '*     '*

Figure 1-16 shows how the two SEROUT commands just discussed will redefine the LCD's custom characters.
Figure 1-16 After Defining Custom Characters 0 and 7

Custom characters are sometimes defined with hexadecimal values. You will even see this in example programs available for download from the Parallax Serial LCD product pages at www.parallax.com. For information on how hexadecimal character definitions work, try the activity in Appendix B: Hexadecimal Character Definitions.

With these new custom character definitions, you can write a loop to make the hourglass toggle between empty and full, indicating that the user should wait. The DO...LOOP below does this by first placing the cursor on the character 5 in the LCD. Then it displays custom character 0, the hourglass that was just turned upside down. After a brief PAUSE, the program sends the backspace command (8) to get the cursor back to character 5. Then, it sends custom character 7, the hourglass with the sand drained into the base. By repeating this sequence, it looks as though the hourglass is turned upside-down, drained, turned again, drained again, and so on.

```
DO
  SEROUT 14, 84, [133] ' Cursor -> Line 1, char
  SEROUT 14, 84, [0] ' Display custom character 0
  PAUSE 1250 ' Delay for 1.25 seconds
  SEROUT 14, 84, [8] ' Backspace
  SEROUT 14, 84, [7] ' Display custom character 7
  PAUSE 1500 ' Delay for 1.50 seconds
LOOP
```

Example Program: Hourglass.bs2

This program defines and displays the hourglass custom characters just discussed.

√ Enter, save, and run the program.
√ Verify that it alternately displays the two hourglass characters at the sixth character in the LCD's top row.
' Example Program - Hourglass.bs2
' Define and display custom characters.

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

' -----[ Initialization ]-----------------------------------------------------

PAUSE 250                                    ' Debounce power supply

SEROUT 14, 84, [248,                      ' Define custom character 0
 %00000,
 %11111,
 %11111,
 %01110,
 %00100,
 %01010,
 %10001,
 %11111]

SEROUT 14, 84, [255,                      ' Define custom character 7
 %00000,
 %11111,
 %10001,
 %01010,
 %00100,
 %01110,
 %11111,
 %11111]

SEROUT 14, 84, [22, 12]                      ' Turn on display and clear
PAUSE 5                                      ' 5 ms delay for clearing display

' -----[ Main Routine ]-------------------------------------------------------

DO

SEROUT 14, 84, [133]                    ' Cursor -> Line 1, char
SEROUT 14, 84, [0]                      ' Display custom character 0
PAUSE 1250                              ' Delay for 1.25 seconds
SEROUT 14, 84, [8]                      ' Backspace
SEROUT 14, 84, [7]                      ' Display custom character 7
PAUSE 1500                              ' Delay for 1.50 seconds

LOOP
Your Turn

Figure 1-17 shows the custom characters depicting the grains of sand in the hourglass moving from the top to the bottom.

Figure 1-17 Figure with Drawing Canvas

√ Save Hourglass.bs2 as HourGlassYourTurn.bs2.
√ Expand the initialization routine so that it defines all eight custom characters as shown in Figure 1-17.
√ Modify that main routine so that it gives an animated hourglass effect as the grains of sand fall from top to bottom.

Here is a main routine you can also try for animating the eight custom characters once you have updated the initialization:

DO
  ' Place cursor at character 5, and display custom character 0.
  SEROUT 14, 84, 100, [133, 0]
  PAUSE 750                               ' 0.750 second delay
  ' Backspace, custom character 1, backspace, custom character 2, etc.
  ' optional pacing argument of 100 sends each value every 1/10 of a second.
  SEROUT 14, 84, 100, [8, 1, 8, 2, 8, 3, 8, 4, 8, 5, 8, 6, 8, 7, 8]
  PAUSE 750
LOOP

√ Try it!
Even though the LCD only stores 8 custom characters at a time, your program can store as many as you need. Remember, your program can redefine any of the custom characters at any time. This means, if your application needs twenty custom characters, your PBASIC program can store 20 custom characters and redefine custom characters as needed.

You can display the hourglass with just one custom character. The entire hourglass animation can be done with just one custom character. The trick is to redefine the custom character between each time the display is updated.