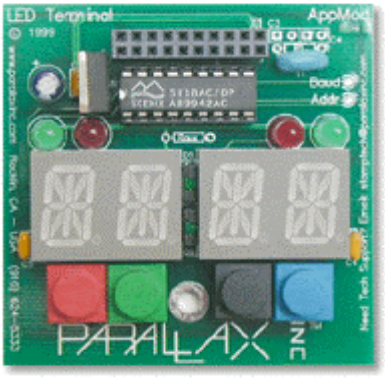
599 Menlo Drive, Suite 100
Rocklin, California 95765, USA
Office/Tech Support: (916) 624-8333
Fax: (916) 624-8003

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 the LED Terminal AppMod

The *LED Terminal AppMod* provides an interface for direct interaction between a user and the host. It offers a four digit alphanumeric LED display, six individual LEDs, four pushbuttons, and a real-time clock. Communication with the AppMod is performed, serially, via the Javelin's I/O pin P6, using the UART class. The LED Terminal display features are available to you through methods within the LedTerminal class. If you would like to experiment with the AppMod's other features, please review the "commands" section.

Downloads, Parts, and Equipment for the LED Terminal AppMod

This application note, the library file, the javadoc file, the test program (Program Listing 1.1), and the demonstration program (which will demonstrate the methods available to you in the LedTerminal.java class), are all available to you for free download from:

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

You can use the AppNote003-LedTerminal.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

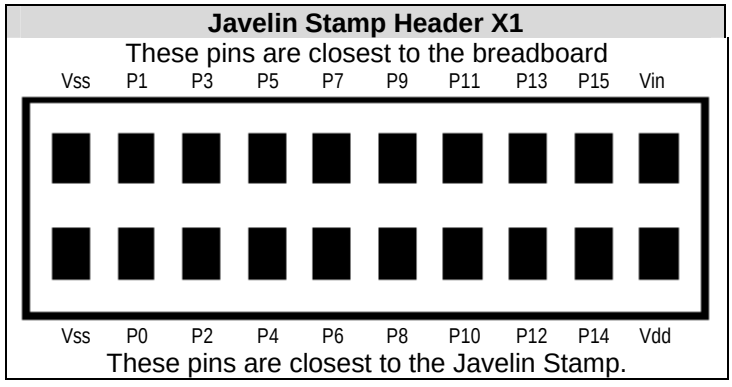
<root path>\doc\AppNote004-LedTerminal.pdf
<root path>\doc\LedTerminal.pdf
<root path>\lib\stamp\peripheral\AppMod\LedTerminal.java
<root path>\Projects\examples\peripheral\AppMod \LedTerminalDisplay.java
<root path>\Projects\examples\peripheral\AppMod\LedTerminalDemo .java

The *LedTerminal* class is for use with the Parallax LED Terminal AppMod (part #29112). The equipment used to test this example includes a Javelin Stamp, Javelin Stamp Demo Board, serial cable, 7.5 V, 1000 mA DC power supply, serial cable, and PC with the Javelin Stamp IDE v2.01.

LED Terminal AppMod Installation

The LED Terminal AppMod requires 6 – 9 Volts DC and will draw between 40-150 mA depending on how many LED segments are energized. Please use caution when touching or handling the LED Terminal Display while it is energized (or even if it was recently energized) as the on-board voltage regulator can get warm during normal operation.

As with all AppMods, the LED Terminal AppMod has a special stack-through 2x10 header. This header allows you to connect the LED Terminal directly to the Javelin Demo board, or stack them on other AppMods which might be on the Javelin Demo Board. We've provided the pinout of the AppMod header below, as viewed from the top.



The AppMod has this header located on one side of the board. On the opposite side is a hole where you attach the standoff with the supplied screw. The Javelin Demo board will also have a hole in it for the other end of the standoff to be inserted into. You can then attach the supplied nut to the bottom of the Javelin Demo board.

Warning	Be careful not to plug the AppMod into the Javelin board backwards! Doing so could damage the AppMod and the Javelin. Verify that the power supply is not connected before you attempt to plug in the AppMod.

Testing the LED Terminal AppMod Class

You will first need to instantiate a LedTerminal object to access the library's methods. This can be done with the following line of code which will create a *new* LedTerminal object called *msg*:

```
LedTerminal msg  = new LedTerminal();
```

Now that you have an object to play with we will flash two messages within a loop. Each message will remain on the screen for 1/10th of a second as dictated by setting the delay to 1000. Notice we are displaying 4 characters. If you were to try to display more than 4, the extra characters will be truncated. If you try to display less than 4 characters, the characters will appear right-justified.

```
for(int i=0;i<10;i++){                  // Loop for 10 times
  msg.display("XOOX",1000);             // Display the message "xoox"
  msg.display("OXXO",1000);             // Display the message "oxxo"
}//end for
```

Next we will display a scrolling message, or a banner, here we use a delay of 2/10th of a second, or 2000, to be the pause before another character is displayed.

```
msg.banner("The time will be set to 10am",2000);   // msg, delay
```

Another useful feature of the AppMod is a clock. Here's the line of code to set the clock to 10:00am with 0 seconds, and a zero day counter.

```
msg.setTime(0,10,0,0);                   // day, hour, min, sec
```

To display the time you simply pass a boolean true to the **`displayTime`** method as shown here:

```
msg.displayTime(true);                   // Display current Time
```

Tip	Use the LedTerminal javadoc file (LedTerminal.pdf in your <"Javelin Stamp IDE/doc"> folder) as a reference to find out more about using the methods available to you.

Program Listing 1.1 is the complete code for the examples above.

Program Listing 1.1 – The LedTerminal Display

```
// Author: Parallax Inc. (javelintech@parallaxinc.com)
// Version: 1.0 Sept 23, 2002

import stamp.core.*;
import stamp.appmod.*;

// This class is for the Parallax Inc "LED Terminal AppMod"

public class LedTerminalDisplay {
  public static void main() {
    LedTerminal msg  = new LedTerminal();   // Create the Led Message Object

    for(int i=0;i<10;i++){                  // Loop for 10 times
      msg.display("XOOX",1000);             // Display the message "xoox"
      msg.display("OXXO",1000);             // Display the message "oxxo"
    }//end for

    // Use the banner routine to scroll messages, you set the delay.
    msg.banner("The time will be set to 10am",2000);   // msg, delay

    msg.setTime(0,10,0,0);                  // day, hour, min, sec
    msg.displayTime(true);                  // Display current Time

  }// end main
}//end class
```

Additional Features of the LedTerminal Library

In addition to the display and banner methods that were explained above this demonstration program uses the following fields to control the LED's:

- **`led1()`**, **`led2()`**, **`led3()`**, **`led4()`** – These fields will accept a boolean TRUE or FALSE. Setting these will turn on/off the green and red discrete LED's above the display when used with the **`display()`** or **`banner()`** methods.
- **`dec1()`**, **`dec2()`**, **`dec3()`**, **`dec4()`** – These fields will accept a boolean TRUE or FALSE. Setting these will turn on/off the segmented-LED's decimal when used with the **`display()`** or **`banner()`** methods.
- **`colonTop()`**, **`conolBot()`** – These fields will accept a boolean TRUE or FALSE. Setting these will turn on/off the discrete LED's in the center of the AppMod, used for the clock's colon, when used with the **`display()`** or **`banner()`** methods.

To control the LED's you will need two bytes: highbyte & lowbyte

```
highbyte lowbyte
XXXXXX00 00000000
      || |||||||`-> Decimal Point #1 (Right-most, Least Significant digit)
      || ||||||`--> Decimal Point #2                 .
      || |||||`---> Decimal Point #3                 .
      || ||||`----> Decimal Point #4 (Left-most, Most Significant digit)
      || |||`-----> LED #1 (Right-most)
      || ||`------> LED #2      .
      || |`-------> LED #3      .
      || `--------> LED #4 (Left-most)
      |'----------> Colon mini-LED (top)
      `-----------> Colon mini-LED (bottom)
```

Direct Command Programming Formats

The following command formats can be passed directly to the LED Terminal with the **`direct()`** method.

ASC - Send ASCII (text) data as well as LED control to the AppMod

4 char	3 char	4 char	1 char	1 char
[init]	[directive]	[ASCII Data]	[LED highbyte]	[LED lowbyte]
!LT0	ASC	data	0x00	0x00

BCD - Show Binary Coded Decimal data as received via serial interface.
Note: data will be displayed as Hex values 0-9, A-F

4 char	3 char	4 char	1 char	1 char
[init]	[directive]	[binary Data]	[LED highbyte]	[LED lowbyte]
!LT0	BCD	data	0x00	0x00

BIN - Show decimal value of HEX input.
Note: data consists of 2 bytes (0-270F) will be converted to decimal (0-9999)

4 char	3 char	2 char	1 char	1 char
[init]	[directive]	[binary Data]	[LED highbyte]	[LED lowbyte]
!LT0	BIN	byte, byte	0x00	0x00

RTI - Real-Time clock Initialization. Set the real-time clock.

4 char	3 char	4 char	1 char	1 char
[init]	[directive]	[binary Data]	[unused]	[unused]
!LT0	RTI	day, hour, min, sec	0x00	0x00

RTH - Real-Time clock Hide. Do not display real-time clock.

4 char	3 char
[init]	[directive]
!LT0	RTH

RTS - Real-Time clock Show. Display the real-time clock.

4 char	3 char
[init]	[directive]
!LT0	RTS

LedTerminal Demonstration

Program Listing 1.2 is a detailed step-by-step fully comprehensive demonstration of all the features of the LedTerminal library methods. In addition to displaying a message Program Listing 1.2 will show you how to display messages with the discrete LEDs. The code is highly commented for your use, but more information is available to you from the library's java doc HTML file.

*The Javelin's current UART class **version 1.0 (IDEv2.01)** is designed for single-direction communication on a given I/O pin. With this limitation, the LedTerminal.java class can not receive serial data from the Led Terminal AppMod (such as the state of the buttons)*

Program Listing 1.2 – The LedTerminal Demonstration

```
// Author: Parallax Inc. (javelintech@parallaxinc.com)
// Version: 1.0 Aug 1, 2002

import stamp.core.*;
import stamp.appmod.*;

// This class is for the Parallax Inc "LED Terminal AppMod"
// It will demonstrate the methods available to you with the LedMsg class.
// The LED AppMod is not case-sensitive; lower case letters will appear as
// uppercase.  Not all symbols can be displayed.

public class LedTerminalDemo {
  public static void main() {
    LedTerminal msg  = new LedTerminal();   // Create the Led Message Object
    final int w1     = 1000;                // Wait delay: 1/10 second
    final int w10    = 10000;               // Wait delay: 1 seconds
    final int w20    = 20000;               // Wait delay: 2 seconds

    msg.display("****",w20);                // Display the contents of String 's'

    // This loop will 'flash' two messages to the display
    for(int i=0;i<10;i++){                  // Loop for 10 times
      msg.display("XOOX",w1);               // Display the message "xoox"
      msg.display("OXXO",w1);               // Display the message "oxxo"
    }//end for

    msg.display("123",w20);     // Less than 4 characters will be right justified
    msg.display("ABCDEFG",w20); // Greater than 4 characters will be truncated
    msg.display("",w20);        // Will blank the display
```

```
// There are boolean variables that you can set that will control discrete LEDs
// These states will be reflected in every message displayed.
// For complete information check the LedMsg class
msg.led1=true;                          // Set led1 'ON'
msg.display("LED1",w10);                // Display message & LED1
msg.led1=false;                         // Set led1 'OFF'
msg.display("",w10);                    // Clear screen, clear LED1
msg.dec1=true;                          // Set dec1 'ON'
msg.display("",w10);                    // Add decimal #1 to display
msg.dec4=true;                          // Set dec4 'ON'
msg.display("",w10);                    // Add decimal #4 to display
msg.dec1=false;                         // Set dec1 'OFF'
msg.dec4=false;                         // Set dec4 'OFF'
msg.display("",w20);                    // Remove from display

// Use the banner routine to scroll messages, you set the delay.
msg.banner("01234.abcd",w10);           // Scroll message, delay 10000
msg.banner("01234.abcd",w1);            // Scroll message, delay 1000
msg.banner("01234.abcd",0);             // Scroll message, delay 0
CPU.delay(5000);                        // Pause

String s="Parallax Inc";                // Place the message in a string
msg.banner(s,4000);                     // Then pass the string with a delay

// This next section will demonstrate how you can access the display
// directly.  This will allow you more control over the discrete LEDs etc.
// You can also access other programming formats not used within this example.
StringBuffer sb = new StringBuffer(13); // Create a StringBuffer
sb.append("!LT0ASC----");               // Set ASCII message directive+message
sb.append((char)0x00);                  // Modify for LEDs & decimals
sb.append((char)0x00);                  // Modify for colon
msg.direct(sb);                         // Display message

// This loop will perform binary shifting on the character #12.
// Which is in numerical location #11
char c = 1;                             // Start off with a binary 1
for(int i=0;i<8;i++){                   // Loop thru all 8 binary digits
  sb.insert(11,c);                      // Insert new number in StringBuffer
  sb.delete(12,13);                     // Delete old number in StringBuffer
  msg.direct(sb);                       // Display Message (watch the LED's)
  CPU.delay(10000);                     // Pause
  c=(char)(c<<1);                       // Shift to next binary digit
}

msg.setTime(0,3,17,0);                  // Set & dsiplay clock for 3:17am
CPU.delay(32000);                       // Wait
CPU.delay(32000);                       // Wait some more...

// Notice the time will no longer display after a message is displayed
```

```
    msg.display("OOPS",w10);
    msg.display("TIME",w10);
    msg.display("GONE",w20);
    CPU.delay(32000);                       // Wait

    msg.displayTime(true);                  // Display current Time

  }// end main
}//end class
```

Published Resources – for More Information

This class was developed to allow the Javelin to interact with the Parallax LED Terminal AppMod. Much care has been taken in creating this class. Below you will find useful information that was used in creating this document.

"Using the LED Terminal AppMod", Application Manual, Parallax, Inc., 06/07/00
This document will give you detailed information about programming the AppMod with the BASIC stamp. It also has helpful information on how to construct the command strings.

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. Other Javelin Stamp users who monitor the list will see your questions and reply with helpful tips, part numbers, pointers to useful web pages, etc.