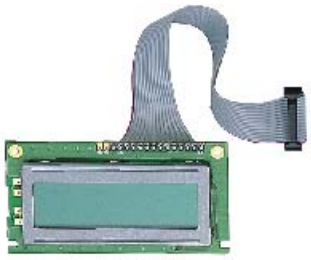
599 Menlo Drive, Suite 100
Rocklin, California 95765, USA
Office: (916) 624-8333
Fax: (916) 624-8003

General: info@parallax.com
Technical: support@parallax.com
Web Site: www.parallax.com
Educational:
http://www.parallax.com/html_pages/edu/index.asp

2 x16 Parallel LCD (#603-00006)

General Information

The 2 X16 Parallel LCD is a 8 bit or 4 bit parallel interfaced LCD. This unit allows the user to display text, numeral data and custom created characters.



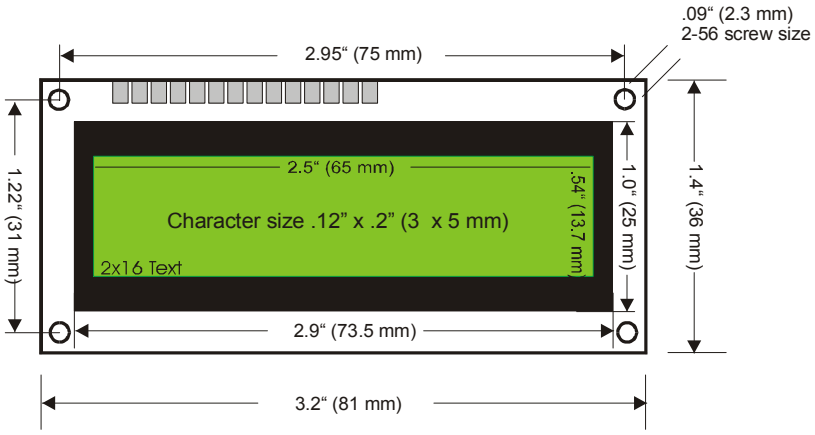
The LCD uses the HD44780 series LCD driver from Hitachi. The LCD is connected to a female 14-pin connector for easy interface with the BS2p24 Demo Board (#45183), BS2p40 Demo Board (#45186), and the NX-1000 Experiment Board (#28135).

Though the device has the ribbon cable and 14-pin connector it may also be hooked up manually using the diagram below.

Technical Specifications

Cable length: 6" (152 mm)
Power requirements: 5.0 +VDC

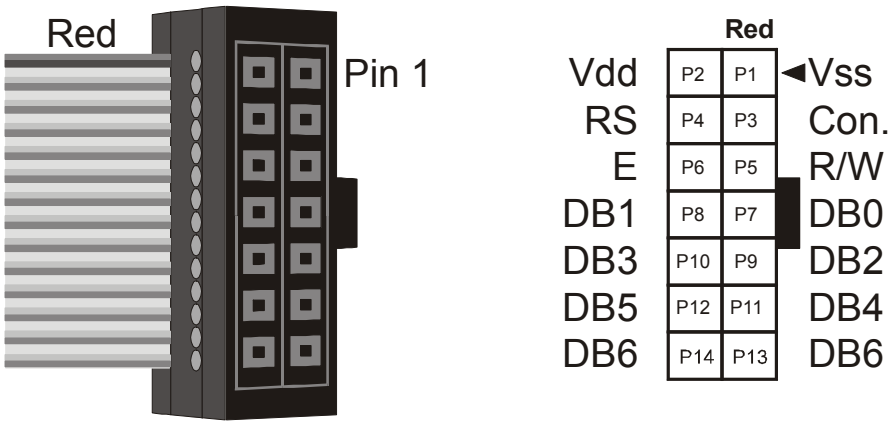
Dimensions may vary



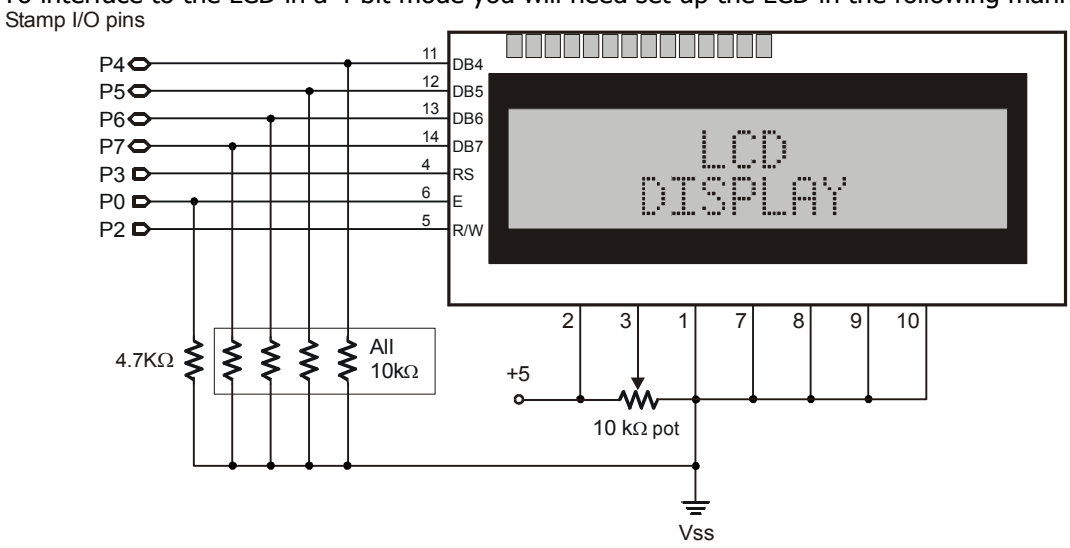
LCD Control from a BASIC Stamp

Parallax (www.parallax.com) publishes many circuits and examples to control LCDs. Most of these examples are available for download from our web site. These examples are featured in StampWorks, the Nuts and Volts of BASIC Stamps book, the free LCD Character Creator Software, and the BS2p Plus Pack.

Example codes are listed below for the BASIC Stamp 1 and 2 modules.



To interface to the LCD in a 4-bit mode you will need set up the LCD in the following manner.



BASIC Stamp 1 code

```
'Basic Stamp 1
'Title: Parallel_lcd.bas
pause 1000
Symbol    E         = 0       'Enable pin, 1 = enabled
Symbol    RS        = 3       'Register select pin, 0 = instruction
Symbol    RW        = 2   'Read / write control = 0 to write
Symbol    Char      = b3      'Character sent to LCD.
Symbol temp     = b4

' Set up the Stamp1 I/O
Begin:
        let pins = 0          'Clear the output lines
        let dirs = %11111000  'One input, 7 outputs.
        pause 1000                      'Wait 200 ms for LCD to reset.
        low RW                          'puts LCD in write mode
' This is the data that is stored in the stamp EEprom
          eeprom ("Hello, This is the LCD demo.")
```

```
i_LCD:
          let pins = %00110000          'wakes up LCD
'Send above data three times.
          pulsout E,1
          pause 10
          pulsout E,1
          pause 10
          pulsout E,1


          let pins = %00100000          'Set to 4-bit operation.
'Send above data three times.
          pulsout E,1
          pause 10
          pulsout E,1
          pause 10
          pulsout E,1
pause 10
          let char = %00101000          'Set to 4-bit operation.
          gosub wr_lcd
          let char = 1                  'Clears screen
          gosub wr_LCD

          let char = 6                  'set cursor direction
          gosub wr_LCD
          let char = 14                 'Sets cursor to underline
          gosub wr_LCD


          high      RS                  'Prepare to send characters.
'***********************Main program************************************
main:
          for b6 = 0 to 27    'Pulls in the data from the EEprom for display
          read b6,char
          if b6 = 15 then next_line
gosub wr_LCD

out:
next
end                             'End code
' Write the ASCII character in b3 to LCD.

Wr_LCD:
          temp = char & %11110000                ' logical AND data of high byte of I/O
pins
          pins = pins & %00001000 | temp          'logical Or the data leaving RS pin an
unchanged state

          pulsout e,1                            'Clocks out data
          pause 10
          temp =  char & %00001111 * 16 ' logical AND data to low byte of I/O pins 'and shifts to
the left
          pins = pins & %00001000 | temp          'logical or s the data leaving RS pin an
'unchanged state
          pulsout e,1                  'Clocks out data
          pause 100
          return

Next_line:

          low rs
          let char = 128+64            'Places cursor on line 2
          gosub wr_LCD
          high rs                               'Puts LCD in to display mode
          read b6,char
          gosub wr_LCD

goto out
```

BASIC Stamp 2,2e and 2sx code

```
'{$STAMP BS2}
'****************************************************************************
'* Title: 2X16Parallel_lcd.bs2
'* This code will work for the stamp2, stamp 2e and stamp 2sx
'****************************************************************************
'Setup for program symbols
E         con       0         'Enable pin for LCD
```

```
Rs       con      3        'LCD Register select pin, 0 = instruction, 1 = text
Char     var      byte     'Character to send to LCD
Inst     var      Char     'Induction to send to LCD. (Points to Char)
Index    var      word     'Character pointer
temp     var      byte
RW      con      2
'*****************************************************************************
'************************Main program*****************************************
'*****************************************************************************

'Setup stamp pins
Initialize:
         low rw
         Outs = %0000000000000000
         Dirs = %0000000011111111
         data "Hello, This is the LCD demo."
gosub Initlcd

Main:
         for temp = 0 to 27
         if temp = 15 then next_line
out:
         read temp,char
gosub Sendtext

next

stop

'Initialize the LCD

Initlcd:
         Pause 200
         Outs = %00110000  'Wakeup for the LCD
         Pulsout E,1       ' Send command three times with required delays
         Pause 10
         Pulsout E,1
         Pause 1
         Pulsout E,1
         pause 1
         Outs = %00100000  'Set to 4-bit operation
         pulsout E,1

         Inst = %00101000 'setup the LCD for two line display
gosub Sendinst

         Inst = %00001110 'Turns on cursor
gosub Sendinst

         Inst = %00000110 'Set to auto-increment cursor and on display shift
gosub Sendinst

         Inst = %00000001 'Clears LCD
gosub Sendinst

         Inst = 14          'Turns cursor to underline
gosub Sendinst
return

'Send an instruction to LCD
Sendinst:
         Low Rs                     'sets instruction mode
         Outb = Inst.highnib        'Send high nibble
         Pulsout E,1
         Outb = Inst.lownib         'Send low nibble
         Pulsout E,1
         High Rs                    ' Sets LCD back to text mode
Return

'Send text to LCD
Sendtext:
         Outb = Char.highnib        'Send high nibble
         Pulsout E,1
         Outb = char.lownib         'Send low nibble
         Pulsout E,1
```

```
         pause 100
return

Next_line:
         Inst = 128+64 'Send cursor to line 2
gosub Sendinst
goto out
```

BASIC Stamp2p24, p40 and 2pe code

```
'{$STAMP BS2p}
'{$PBASIC 2.5}

'Title: Parallel_lcd 2X16.bsp
'this code will work for the stamp2p and 2pe with the 2X16 parallel Lcd

 ' -----[ Declarations ]----------------------------------------------------
temp VAR Byte

Lcd_pin          CON     0
Non_op           CON     0
char             VAR     Byte
' -----[ Initialization ]------------------------------------------------

        DATA "Hello, This is the LCD demo."
Initlcd:
        PAUSE 1000
        FOR temp = 0 TO 2
        LCDCMD 0,48
        PAUSE 1
NEXT
        LCDCMD Lcd_pin,32 'sets lcd in 4 bit mode
        LCDCMD Lcd_pin,40 'sets lcd to 2 line mode with 5x8 font
        LCDCMD Lcd_pin,12 'turns on display with no cursor
        LCDCMD Lcd_pin,6  'set to auto-increment cursor
        LCDCMD Lcd_pin,1  'clears display
' -----[ Main Routine ]----------------------------------------------------
start:
        FOR temp = 0 TO 27
        READ temp,char
        IF temp = 15 THEN Next_line
out:
        LCDOUT 0,non_op,[char]
        PAUSE 100           ' This number adjust the rate of displaying the text
        NEXT
STOP
' -----[ Subroutines ]-----------------------------------------------------
Next_line:
        LCDCMD Lcd_pin,192 'places cursor to line 2
GOTO out
```