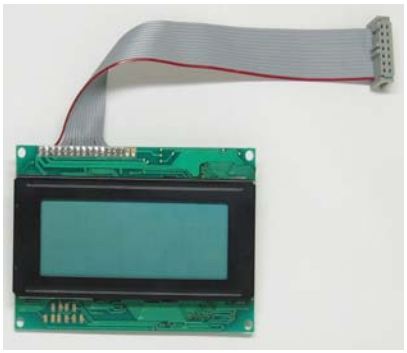
599 Menlo Drive, Suite 100
Rocklin, California 95765, USA
Office: (916) 624-8333
Fax: (916) 624-8003

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

4 x 20 Parallel LCD (#603-00004)

General Information



The 4 X20 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.

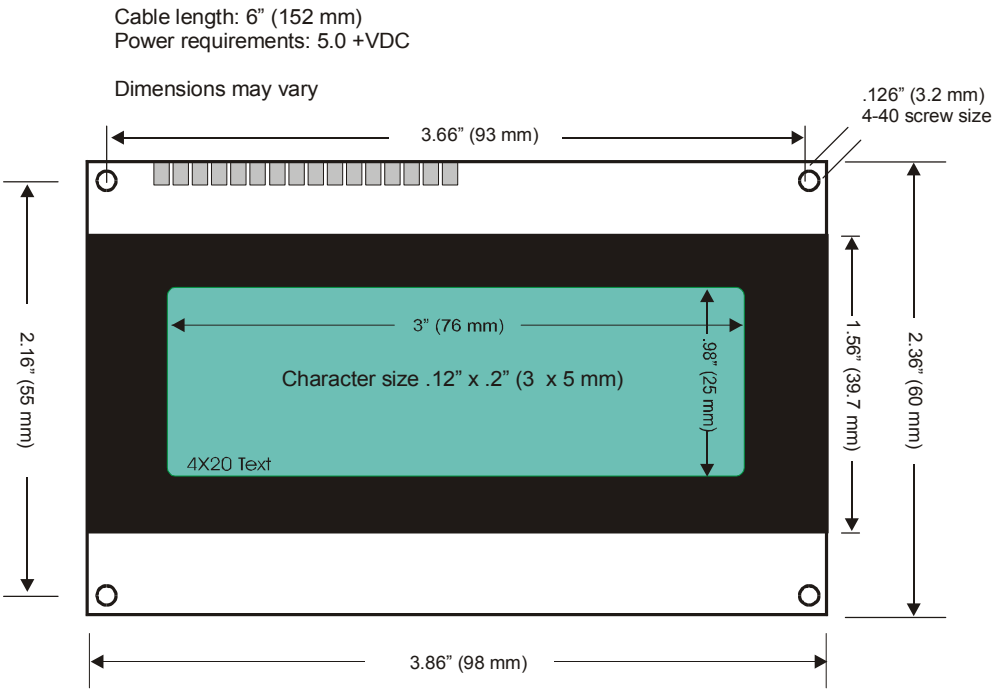
The LCD uses the HD44780 series LCD driver from Hitachi. The LCD is connected to a female 14-pin connector for easy interface with the BS2p24 Demo Board (#45183), BS2p40 Demo Board (#45186), and the NX-1000 Experiment Board (#28135).

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

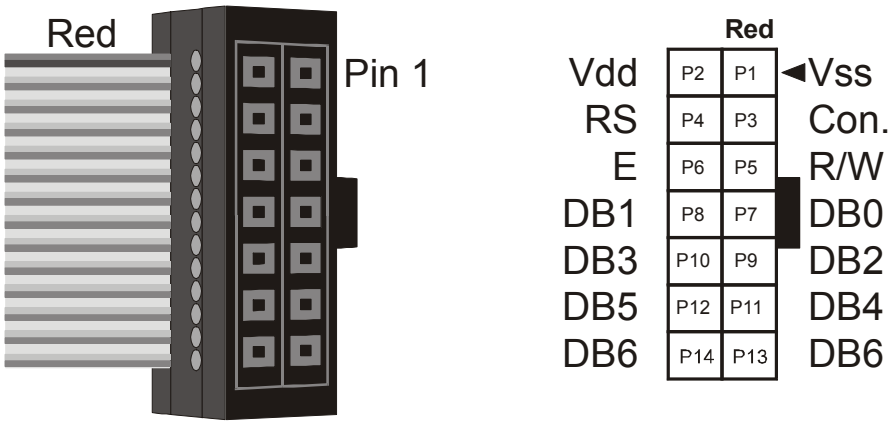
Dimensions may vary



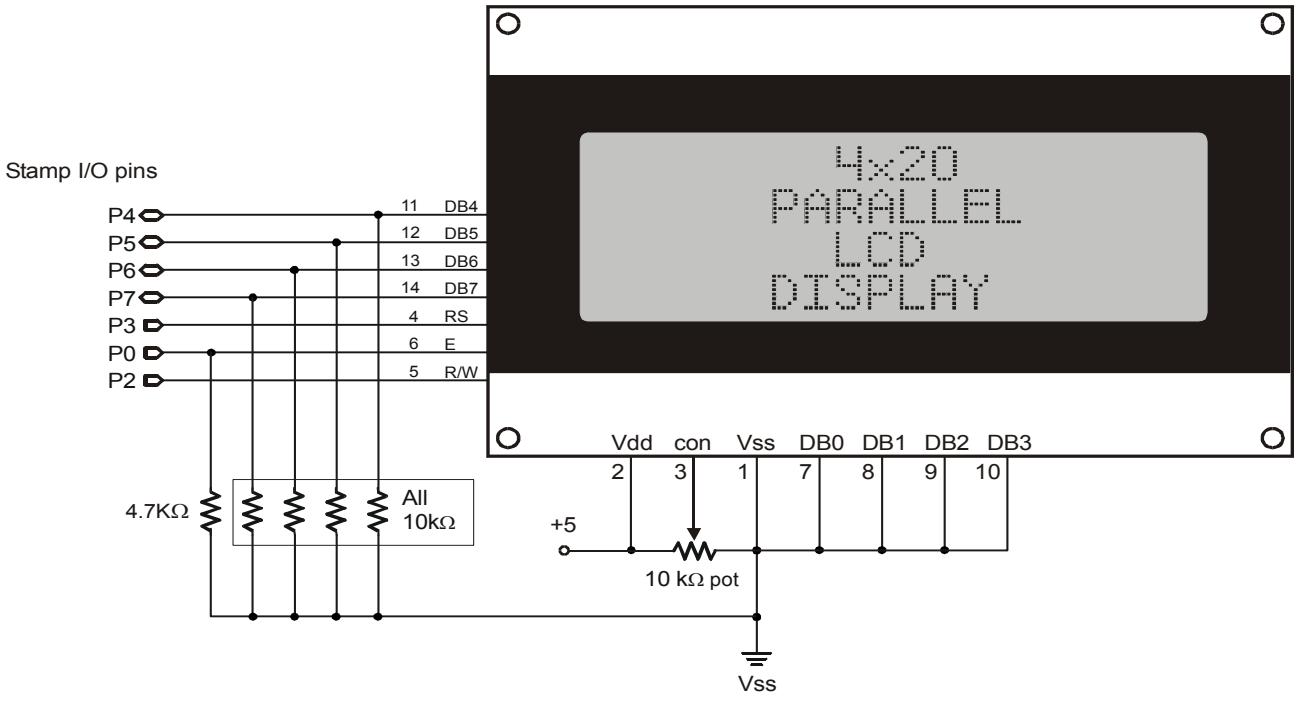
LCD Control from a BASIC Stamp

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 for the 4X20 parallel lcd in a 4 bit mode.)


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 69    'Pulls in the data from the EEprom for display
          read b6,char
          if b6 = 19 then line_2
          if b6 = 36 then line_3
          if b6 = 54 then line_4

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

```
          pause 100
          return

line_2:

          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
line_3:

          low rs
          let char = 128+20             'Places cursor on line 3
          gosub wr_LCD
          high rs                       'Puts LCD in to display mode
          read b6,char
          gosub wr_LCD
line_4:

          low rs
          let char = 128+84             'Places cursor on line 4
          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

```
'{$STAMP BS2}
'{$PBASIC 2.5}
'
'Title: Parallel_lcd 4X20.bs2
'this code will work for the stamp2, stamp 2e and stamp 2sx and the 4X20 Lcd
' -----[ Declarations ]----------------------------------------------------

'LCD control pins
E                   CON                 0               'Enable PIN FOR LCD
RW                  CON                 2               'read WRITE PIN
Rs                  CON                 3               'LCD Register SELECT PIN, 0 = instruction,
1 = text
'LCD commands for more information on these functions please see the data sheet.
Cl_screen           CON                 %1              'clears LCD screen
Return_home         CON                 %10             'returns to home positions
Shift_to_right      CON                 $110            'shifts to the right
blinking_cur_on     CON                 %1101           'Blinking on display on
blinking_cur_off    CON                 %1100           'Blinking off display on
display_off         CON                 %1000           'display off
display_on          CON                 %1100           'display on
inter_4_bit         CON                 %100000         '4 bit interface
two_line            CON                 %101000         '2 line mode
one_line            CON                 %100000         '1 line mode
underline_on        CON                 %1110           'underline_on
Underline_off       CON                 %1100           'underline off

'Lcd variables
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            'working variable


' -----[ Initialization ]--------------------------------------------------
'Setup stamp pins
Initialize:  'label
LOW rw                                                  'sets LCD TO WRITE mode
OUTS = %0000000000000000                                'sets I/O pins TO OUTPUT
DIRS = %0000000011111111                                'sets first 8 HIGH AND last 8 LOW
'stores text to be read in to EEprom on stamp
DATA "Hello, This is the lcd demo for the 4X20 parallel lcd in a 4 bit mode."
GOSUB Initlcd                                           'jumps to specified label
' -----[ Main Routine ]----------------------------------------------------
Main:                                                   'Label
 FOR temp = 0 TO 69                                     'Stays in loop until temp = 69
         SELECT  temp                                   'watches the temp variable
      CASE 19                                           'If temp = 19 then do commands below
              Inst = 128+64                             'sets Inst variable to 192 which is line 2
```

```
of LCD
              GOSUB Sendinst                                'jumps to specified label
      CASE 36                                               'If temp = 36then do commands below
              Inst = 128+20                                 'sets Inst variable to 148 which is line 3
of LCD
              GOSUB Sendinst                                'jumps to specified label
      CASE 54                                               'If Temp = 54 then do commands below
              Inst = 128+84                                 'sets Inst variable to 212 which is line 4
of LCD
              GOSUB Sendinst                                'jumps to specified label
         ENDSELECT                                          'ends case select
out:                                                        'Label
    READ temp,char                                          'pulls data from internal EEprom
                GOSUB Sendtext                              'jumps to specified label

 NEXT                                                       'exits loop when temp = 69
 inst = blinking_cur_on                                     'Turns on blinking cursor
 GOSUB Sendinst                                             'jumps to specified label
STOP                                                        'Stops the stamp
' -----[ Subroutines ]------------------------------------------------------

'Initialize the LCD
Initlcd:                                                    'Label
PAUSE 200                                                   'Pause for 200 ms
OUTS = %00110000                                            'Wakeup for the LCD
'Send command three times with required delays
PULSOUT E,1                                                 'Pulses out on Enable for 2us
PAUSE 10                                                    'pauses for 10ms
PULSOUT E,1                                                 'Pulses out on Enable for 2us
PAUSE 1                                                     'pauses for 1ms
PULSOUT E,1                                                 'Pulses out on Enable for 2us
PAUSE 1                                                     'pauses for 1ms

OUTS =  inter_4_bit                                         'set to 4-bit operation
PULSOUT E,1                                                 'pulses enable pin
Inst = two_line                                             'setup the LCD for two line display
GOSUB Sendinst                                              'jumps to specified label
Inst =  display_on                                          'Turns on cursor
GOSUB Sendinst                                              'jumps to specified label
Inst = Shift_to_right                                       'SET TO AUTO-Increment Cursor and on
display shift
GOSUB Sendinst                                              'jumps to specified label
Inst = Cl_screen                                            'Clears LCD
GOSUB Sendinst                                              'jumps to specified label
Inst = underline_off                                        'Turns cursor to underline
GOSUB Sendinst                                              'jumps to specified label
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
```

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

'Title: Parallel_lcd 4X20.bsp
'this code will work for the stamp2p and 2pe with the 4 X20 parallel Lcd

 ' -----[ Declarations ]----------------------------------------------------
'LCD control pins
Lcd_pin          CON     0                              'Pin for LCD
Non_op           CON     0                              'Null command
'LCD commands for more information on these functions please see the data sheet.
Cl_screen           CON                  %1             'clears LCD screen
Return_home         CON                  %10            'returns to home positions
Shift_to_right      CON                  $110           'shifts to the right
blinking_cur_on     CON                  %1101          'blinking on display on
blinking_cur_off    CON                  %1100          'blinking off display on
display_off         CON                  %1000          'display off
display_on          CON                  %1100          'display on
inter_4_bit         CON                  %100000        '4 bit interface
two_line            CON                  %101000        '2 line mode
one_line            CON                  %100000        '1 line mode
underline_on        CON                  %1110          'underline_on
Underline_off       CON                  %1100          'underline off

'Lcd variables
char                VAR                  Byte           'Character to send to LCD
inst                VAR                  Byte           'Inductions To send TO LCD. (Points to
Char)
index               VAR                  Word           'Character pointer
temp                VAR                  Byte           'working variable
' -----[ Initialization ]------------------------------------------------
'Setup stamp pins
Initialize:                                             'label
'stores text to be read in to EEprom on stamp
         DATA "Hello, This is the lcd demo for the 4X20 parallel lcd in a 4 bit mode."
Initlcd:                                               'label
         PAUSE 1000                                       'pauses for wait 1000ms
         FOR temp = 0 TO 2                                'loops until temp = 2
         LCDCMD 0,48                                      'Wake ups LCD
         PAUSE 1                                          'Pause for 1 ms
NEXT                                                    'when temp = 2 code will continue
 LCDCMD Lcd_pin,inter_4_bit                               'Sets Lcd In 4 bit mode
 LCDCMD Lcd_pin,two_line                                  'sets LCD to 2 line mode with 5x8 font
 LCDCMD Lcd_pin,display_on                                'turns on display with no cursor
 LCDCMD Lcd_pin,Shift_to_right                            'set to auto-increment cursor
 LCDCMD Lcd_pin,Cl_screen                                 'clears display
' -----[ Main Routine ]---------------------------------------------------
start:                                                  'Label
 FOR temp = 0 TO 69                                     'Stays in loop until temp = 69
         READ temp,char                                         'reads data from internal EEprom
         SELECT  temp                                   'watches the temp variable
         CASE 19                                        'If temp = 19 then do commands below
         Inst = 128+64                                  'sets Inst variable to 192 which is line 2 of LCD
       LCDCMD 0,inst                                    'SENDS Out the command to set the text to line 2
         CASE 36                                        'If temp = 19 then do commands below
       Inst = 128+20                                    'sets Inst variable to 148 which is line 3 of LCD
       LCDCMD 0,inst                                    'SENDS Out the command to set the text to line 3
         CASE 54                                        'If Temp = 54 then do commands below
       Inst = 128+84                                    'sets Inst variable to 212 which is line 4 of LCD
       LCDCMD 0,inst                                    'SENDS Out the command to set the text to line 4
         ENDSELECT                                      'ends case select
' -----[ Subroutines ]----------------------------------------------------
out:                                                    'Label
                  LCDOUT 0,non_op,[char]                        'Sends charter to LCD
   PAUSE 100                                            ' This number adjust the rate of displaying the
text
 NEXT                                                   'continues to next line when temp = 69
 LCDCMD Lcd_pin,blinking_cur_on                         'turns on blinking curser
         STOP                                                   'stops stamp
```