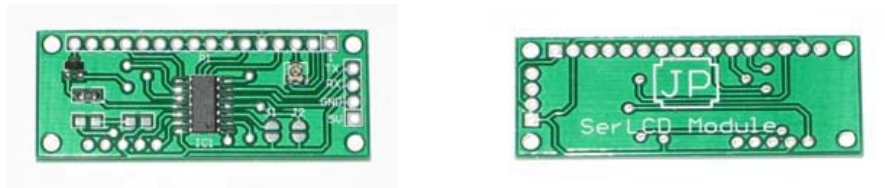


# JP Serial LCD Module



JP serial LCD module is a simple and cost effective interface controller module. It supports 16x2, 16x4, 20x2 and 20x4 LCD display base on HD44780 controller. Also it will work with any microcontroller capable of accepting asynchronous serial data.

## SPECS

Power: 5V VDC <10mA (not include LCD display Backlight)

Size: 50mm x 20mm (2.0" x 0.75")

Speed: 4800 or 19200 Baud.

## PIN FUNCTIONS

**P1 Connector:** P1 connector will be connected to a 16x2, 16x4, 20x2 or 20x4 LDC Display.

Pin1:	GND
Pin2:	Vcc
Pin3:	Vee
Pin4:	RS
Pin5:	R/W
Pin6:	En
Pin7 – Pin14:	D0 – D7
Pin15	LED+
Pin16	LED-

**P2 Connector:** P2 connector will be connected to power and a MCU

**RX:** Serial input connection to JP Module. The module allows 4800 or 19200 baud based on J1 settings.

**TX:** Serial output connection to JP Module. The module allows 4800 or 19200 baud based on J1 settings.

J1 Settings:

J1	J2	Baud
x	n/a	4800
-	n/a	19200

- = connected

x = disconnected

**GND:** Power supply and serial ground. This MUST also be connected to ground on the device to allow the serial data to be sent to the module.

**5V:** Supply voltage to module. Vin may be 5.0V ( $\pm 0.5V$ ), with 10 milliamps of current.

## JP Serial LCD Module Commands:

Those commands are ASCII value between 1 and 30. It needs a pause to be pressed.

Commands	Dec	Hex
CGRam 1 Address	1	01

CGRam 2 Address	2	02
CGRam 3 Address	3	03
CGRam 4 Address	4	04
CGRam 5 Address	5	05
CGRam 6 Address	6	06
CGRam 7 Address	7	07
CGRam 8 Address	8	08
CGRam 1 setting	9	09
CGRam 2 setting	10	0A
CGRam 3 setting	11	0B
CGRam 4 setting	12	0C
CGRam 5 setting	13	0D
CGRam 6 setting	14	0E
CGRam 7 setting	15	0F
CGRam 8 setting	16	10
Clear Display	17	11
Return cursor to home	18	12
Turn cursor off	19	13
Turn underline cursor on	20	14
Turn blink cursor on	21	15
Move cursor left	22	16
Move cursor right	23	17
Turn LCD Display on	24	18
Turn LCD Display off	25	19
Shift Display left	26	1A
Shift Display right	27	1B
Turn backlight on	28	1C
Turn backlight off	29	1D

### **JP Serial LCD Module Characters:**

JP Serial LCD Module supports all of ASCII value between 32 – 127.

### **JP Serial LCD Module out put row and column:**

JP Serial LCD Module output row and column control number are between 128 and 254.

HD44780 the addresses of the first position of Lines 1 & 2 are 0 and 64, respectively. The first position of Line 3 and Line 4 are extension of the addresses of Line 1 and Line 2.

16x4 LCD: Line 1 = 0, Line 2 = 64, Line 3 = 16, Line 4 = 80

20x4 LCD: Line 1 = 0, Line 2 = 64, Line 3 = 20, Line 4 = 84

Those address must be used the LCD Command \$80, or 128.

16x4 LCD: Line 1 = 128 + 0, Line 2 = 128 + 64, Line 3 = 128 + 16, Line 4 = 128 + 80

20x4 LCD: Line 1 = 128 + 0, Line 2 = 128 + 64, Line 3 = 128 + 20, Line 4 = 128 + 84

**16x2 LCD Display Character Address Code: (128 + Position number)**

Display Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DD RAM Address	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79

**20x2 LCD Display Character Address Code: (128 + Position number)**

Display Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
DD RAM Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
DD RAM Address	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83

**16x4 LCD Display Character Address Code: (128 + Position number)**

Display Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DD RAM Address	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
DD RAM Address	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
DD RAM Address	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95

**20x2 LCD Display Character Address Code: (128 + Position number)**

Display Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
DD RAM Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
DD RAM Address	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83
DD RAM Address	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
DD RAM Address	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103

**Custom user Characters:**

First of all, an 8 bytes array must be created. (CGRam[8])

□ □ □ □ □	CGRam[0]=%00000000	□ ■ ■ ■ □	CGRam[0] = %00001110
□ □ □ □ □	CGRam[1]=%00000000	■ □ □ □ ■	CGRam[1] = %00010001
□ □ □ □ □	CGRam[2]=%00000000	□ □ □ □ ■	CGRam[2] = %00000001
□ □ □ □ □	CGRam[3]=%00000000	□ ■ □ □ ■	CGRam[3] = %00001101
□ □ □ □ □	CGRam[4]=%00000000	■ □ ■ □ ■	CGRam[4] = %00010101
□ □ □ □ □	CGRam[5]=%00000000	■ □ □ □ ■	CGRam[5] = %00010001
□ □ □ □ □	CGRam[6]=%00000000	□ ■ ■ ■ □	CGRam[6] = %00001110
□ □ □ □ □	CGRam[7]=%00000000	□ □ □ □ □	CGRam[7] = %00000000

Usually, Bottom row is empty for underline.  
Then CGRam command and this array are sent to JP Serial LCD Module. You will see a Character that you made on the LCD screen. (See sample code for detail)

**JP Serial LCD Module Default setting (internal):**

- A. Cursor is turned off.
- B. Backlight is turned on.

**SERIAL DATA FORMAT**

The serial data format is eight data bits, no parity and 1 stop bit (8N1). Characters are sent using standard ASCII values. Baud rate may be 4800 and 19200 depending on the settings of J1.

## LIABILITY WARNING

This device should be used only for experimental purposes. It has **NOT** gone through extensive testing and it could erase or corrupt some or all data on media cards that are inside the device. You assume to take your own risk when you purchase this device, and release the responsibility and liability from the manufacturer with no harm.

## REGULATORY WARNING

This device is intended solely for experimental purpose, it is not in finished product form and is NOT FCC approved. If you wish to install these modules into non-experimental final finished products, you will be responsible to have the modules approved by the FCC at your own cost.

# Basic Stamp ® Example Programs

```
'=====
' File.....JP Serial LCD Module Test.BSP
' Purpose.....This test code for JP SerLCD Module
' Auther.....Jianping Sun
' Website.....www.JianpingUSA.com
' Email.....JP@JianpingUSA.com
' Started.....Aug 08, 2009
' Updated.....
'=====
'{$STAMP BS2sx}
'{$PBASIC 2.5}
```

```
SOUT  PIN    0                ' serial output for JP SerLCD Module
TXT   VAR    Byte (13)
i     VAR    Byte
j     VAR    Byte
CGRam VAR    Byte (8)
```

```
#SELECT $STAMP
#CASE BS2, BS2E, BS2PE
  T1200  CON  813
  T2400  CON  396
  T4800  CON  188
  T9600  CON   84
  T19K2  CON   32
  T38K4  CON    6
#CASE BS2SX, BS2P
  T1200  CON  2063
  T2400  CON  1021
  T4800  CON   500
  T9600  CON   240
  T19K2  CON   110
  T38K4  CON    45
#CASE BS2PX
  T1200  CON  3313
  T2400  CON  1646
  T480   CON   813
  T9600  CON   396
```

```
T19K2    CON    188
T38K4    CON    84
#ENDSELECT
```

```
Inverted    CON    $4000
Baud        CON    T4800 '+ Inverted
```

```
TXT(0) = "H"
TXT(1) = "E"
TXT(2) = "L"
TXT(3) = "L"
TXT(4) = "O"
TXT(5) = ","
TXT(6) = " "
TXT(7) = "W"
TXT(8) = "O"
TXT(9) = "R"
TXT(10) = "L"
TXT(11) = "D"
TXT(12) = "!"
```

```
'=====
'                                     Main program
'=====
```

```
Main:
  PAUSE 3000
  GOSUB ScreenTest
  PAUSE 2000
  GOSUB LcdDisplay_on_off
  PAUSE 1000
  GOSUB BacklightTest
  PAUSE 100
  GOSUB CousorTest
  PAUSE 500
  GOSUB CGRamTest1
  PAUSE 2000
  GOSUB CGRamTest2
  PAUSE 1000
  GOSUB ShiftTest
END
```

```
'=====
'                                     Full Screen Test
'=====
```

```
ScreenTest:
  SEROUT SOUT, Baud, [18]           ' Clear screen
  PAUSE 5
  SEROUT SOUT, Baud, [" JianpingUSA "]
  SEROUT SOUT, Baud, [128 + 64]     ' Move sursor to 2nd row
  PAUSE 5
  SEROUT SOUT, Baud, [" JP Serial LCD "]
  SEROUT SOUT, Baud, [128 + 16]    ' Move sursor to 3rd row
  PAUSE 5
  SEROUT SOUT, Baud, [" Module testing "]
  SEROUT SOUT, Baud, [128 + 80]    ' Move sursor to 4th row
  PAUSE 5
  SEROUT SOUT, Baud, [" (16x4 or 20x4) "]
```

```
    PAUSE 500
RETURN
```

```
=====
'          LCD Display turn on/off
=====
```

```
LCDDisplay_on_off:
    SEROUT SOUT, Baud, [17]          ' Clear screen
    PAUSE 5
    SEROUT SOUT, Baud, [128 + 64]    ' Move sursor to 2nd row
    PAUSE 5
    SEROUT SOUT, Baud, [" LCD Display "]
    SEROUT SOUT, Baud, [128 + 16]    ' Move sursor to 3rd row
    PAUSE 5
    SEROUT SOUT, Baud, ["Turn on/off test"]
    PAUSE 500
    FOR i = 0 TO 5
        SEROUT SOUT, Baud, [25]      ' Turn LCD Display off
        PAUSE 500
        SEROUT SOUT, Baud, [24]      ' Turn LCD Display on
        PAUSE 500
    NEXT
RETURN
```

```
=====
'          Create Custemer CGRam
=====
```

```
CGRamTest2:

    SEROUT SOUT, Baud, [17]          ' Clear screen
    PAUSE 5
    SEROUT SOUT, Baud, [19]          ' Turn cousor off
    PAUSE 5                          ' First char
    CGRam(0) = %00000000
    CGRam(1) = %00000000
    CGRam(2) = %00000000
    CGRam(3) = %00010000            ' X
    CGRam(4) = %00000000
    CGRam(5) = %00000000
    CGRam(6) = %00000000
    CGRam(7) = %00000000            'This row empty FOR underline
    SEROUT SOUT, Baud, [9]          ' 1st character set number
    PAUSE 5
    FOR i = 0 TO 7
        SEROUT SOUT, Baud, [CGRam(i)]
    NEXT
    PAUSE 5

=====
'          Second char

    CGRam(0) = %00000000
    CGRam(1) = %00000000
    CGRam(2) = %00010000            'X
    CGRam(3) = %00011000            'XX
    CGRam(4) = %00010000            'X
    CGRam(5) = %00000000
    CGRam(6) = %00000000
    CGRam(7) = %00000000            'this row empty FOR underline
    SEROUT SOUT, Baud, [10]        ' 2nd character set number
```

```

PAUSE 5
FOR i = 0 TO 7
  SEROUT SOUT, Baud, [CGRam(i)]
NEXT
PAUSE 5

```

---

```

CGRam(0) = %00000000          ' 3rd char
CGRam(1) = %00010000          'X
CGRam(2) = %00011000          'XX
CGRam(3) = %00011100          'XXX
CGRam(4) = %00011000          'XX
CGRam(5) = %00010000          'X
CGRam(6) = %00000000
CGRam(7) = %00000000          ' This row empty FOR underline
SEROUT SOUT, Baud, [11]       ' 3rd character set number
PAUSE 5
FOR i = 0 TO 7
  SEROUT SOUT, Baud, [CGRam(i)]
NEXT
PAUSE 5

```

---

```

CGRam(0) = %00010000          ' 4th char
CGRam(1) = %00011000          'X
CGRam(2) = %00011100          'XX
CGRam(3) = %00011110          'XXX
CGRam(4) = %00011110          'XXXX
CGRam(5) = %00011000          'XXX
CGRam(6) = %00010000          'XX
CGRam(7) = %00000000          ' This row empty FOR underline
SEROUT SOUT, Baud, [12]       ' 4th character set number
PAUSE 5
FOR i = 0 TO 7
  SEROUT SOUT, Baud, [CGRam(i)]
NEXT
PAUSE 5

```

---

```

CGRam(0) = %00011000          ' 5th char
CGRam(1) = %00011100          'XX
CGRam(2) = %00011110          'XXX
CGRam(3) = %00011111          'XXXX
CGRam(4) = %00011110          'XXXXX
CGRam(5) = %00011100          'XXX
CGRam(6) = %00011000          'XX
CGRam(7) = %00000000          ' this row empty FOR underline
SEROUT SOUT, Baud, [13]       ' 5th character set number
PAUSE 5
FOR i = 0 TO 7
  SEROUT SOUT, Baud, [CGRam(i)]
NEXT
PAUSE 5

```

---

```

CGRam(0) = %00011100          ' 6th char
                                'XXX

```

```

CGRam(1) = %00011110      'XXXX
CGRam(2) = %00011111      'XXXXX
CGRam(3) = %00011111      'XXXXX
CGRam(4) = %00011111      'XXXXX
CGRam(5) = %00011110      'XXXX
CGRam(6) = %00011100      'XXX
CGRam(7) = %00000000      'this row empty FOR underline
SEROUT SOUT, Baud, [14]   ' 6th character set number
PAUSE 5
FOR i = 0 TO 7
  SEROUT SOUT, Baud, [CGRam(i)]
NEXT
PAUSE 5

```

---

```

CGRam(0) = %00011110      ' 7th char
CGRam(1) = %00011111      ' XXXX
CGRam(2) = %00011111      ' XXXXX
CGRam(3) = %00011111      ' XXXXX
CGRam(4) = %00011111      ' XXXXX
CGRam(5) = %00011111      ' XXXXX
CGRam(6) = %00011110      ' XXXX
CGRam(7) = %00000000      ' this row empty FOR underline
SEROUT SOUT, Baud, [15]   ' 7th character set number
PAUSE 5
FOR i = 0 TO 7
  SEROUT SOUT, Baud, [CGRam(i)]
NEXT
PAUSE 5

```

---

```

CGRam(0) = %00011111      ' 8th char
CGRam(1) = %00011111      ' XXXXX
CGRam(2) = %00011111      ' XXXXX
CGRam(3) = %00011111      ' XXXXX
CGRam(4) = %00011111      ' XXXXX
CGRam(5) = %00011111      ' XXXXX
CGRam(6) = %00011111      ' XXXXX
CGRam(7) = %00000000      ' this row empty FOR underline
SEROUT SOUT, Baud, [16]   ' 8th character set number
PAUSE 5
FOR i = 0 TO 7
  SEROUT SOUT, Baud, [CGRam(i)]
  PAUSE 5
NEXT
PAUSE 5

```

```

SEROUT SOUT, Baud, [" Custom Chars "]
PAUSE 500
SEROUT SOUT, Baud, [128 + 68]
PAUSE 5

```

---

```

FOR i = 1 TO 8
  SEROUT SOUT, Baud, [i]
NEXT
PAUSE 5

```

```

'-----
FOR i = 1 TO 3
  SEROUT SOUT, Baud, [128 + 16]
  PAUSE 100
  SEROUT SOUT, Baud, [i]
NEXT

FOR j = 0 TO 15
  FOR i = 4 TO 5
    SEROUT SOUT, Baud, [128 + 16 + j]
    PAUSE 50
    SEROUT SOUT, Baud, [i]
  NEXT
NEXT

FOR i = 6 TO 8
  SEROUT SOUT, Baud, [128 + 16 + j]
  PAUSE 5
  SEROUT SOUT, Baud, [i]
  SEROUT SOUT, Baud, [i - 5]
  PAUSE 50
NEXT
NEXT
'-----

FOR i = 1 TO 3
  SEROUT SOUT, Baud, [128 + 80]
  PAUSE 100
  SEROUT SOUT, Baud, [i]
NEXT

FOR j = 0 TO 15
  FOR i = 4 TO 5
    SEROUT SOUT, Baud, [128 + 80 + j]
    PAUSE 50
    SEROUT SOUT, Baud, [i]
  NEXT
NEXT

FOR i = 6 TO 8
  SEROUT SOUT, Baud, [128 + 80 + j]
  PAUSE 5
  SEROUT SOUT, Baud, [i]
  SEROUT SOUT, Baud, [i - 5]
  PAUSE 50
NEXT
NEXT

RETURN
'=====
'
'                               Cursor test
'=====
'

CousorTest:

SEROUT SOUT, Baud, [17]           ' Clear screen
PAUSE 5
SEROUT SOUT, Baud, [20]         ' Turn underline cursor on
PAUSE 5

```

```

SEROUT SOUT, Baud, ["Move cursor from"]
SEROUT SOUT, Baud, [128 + 64]
PAUSE 5
SEROUT SOUT, Baud, [" left to right "]

SEROUT SOUT, Baud, [128 + 16]
PAUSE 5
SEROUT SOUT, Baud, ["Move cursor from"]
SEROUT SOUT, Baud, [128 + 80]
PAUSE 5
SEROUT SOUT, Baud, [" right to left "]
SEROUT SOUT, Baud, [128 ]
PAUSE 1000
FOR i = 1 TO 15
    SEROUT SOUT, Baud, [23]
    PAUSE 500
NEXT
SEROUT SOUT, Baud, [21]                ' Turn Blink cursor ON
PAUSE 5
SEROUT SOUT, Baud, [128 + 32 ]
PAUSE 1000
FOR i = 0 TO 15
    SEROUT SOUT, Baud, [22]
    PAUSE 500
NEXT
SEROUT SOUT, Baud, [19]                ' Turn off cursor
RETURN
=====
'                                     Backlight test
=====
BacklightTest:
SEROUT SOUT, Baud, [17]                ' Clear screen
PAUSE 5
SEROUT SOUT, Baud, [128 + 64]
PAUSE 5
SEROUT SOUT, Baud, [" Backlight test "]
FOR i = 0 TO 5
    SEROUT SOUT, Baud, [29]            'Turn backlight off
    PAUSE 500
    SEROUT SOUT, Baud, [28]            'Turn backlight on
    PAUSE 500
NEXT
RETURN
=====
'                                     Display default CDRam Characters
=====
CGRamTest1:
SEROUT SOUT, Baud, [17]                ' Clear screen
PAUSE 5
SEROUT SOUT, Baud, [18]                ' Return cursor to home position, return a shifted
                                        ' diaplay to original position. Dispaly data RAM is
                                        ' unaffected

PAUSE 5
SEROUT SOUT, Baud, ["JP SerLCD Module"]
SEROUT SOUT, Baud, [128 + 64]

```

```
PAUSE 5
SEROUT SOUT, Baud, [" Default Chars "]
SEROUT SOUT, Baud, [128 + 20]
PAUSE 5
FOR i = 1 TO 8
  SEROUT SOUT, Baud, [i]
  PAUSE 10
NEXT
RETURN
```

```
'=====
'                               Shift test
'=====
```

```
ShiftTest:
SEROUT SOUT, Baud, [17]           ' Clear screen
PAUSE 5
SEROUT SOUT, Baud, [128 + 64 + 14] ' Move sursor to 2nd row, 12 col
PAUSE 5
FOR i = 0 TO 12
  SEROUT SOUT, Baud, [26]         ' Shift dispaly left
  PAUSE 5
  SEROUT SOUT, Baud, [128 + 64 + 14 + i] ' Move sursor to 2nd row, 12 col
  PAUSE 5
  SEROUT SOUT, Baud, [txt(i)]
  PAUSE 200
NEXT
RETURN
```

```
'=====
```

**Jianping Electronics**  
[jp@jianpingusa.com](mailto:jp@jianpingusa.com)  
[www.jianpingusa.com](http://www.jianpingusa.com)