SLCD16x2 : 16 Characters 2 lines Serial LCD Module

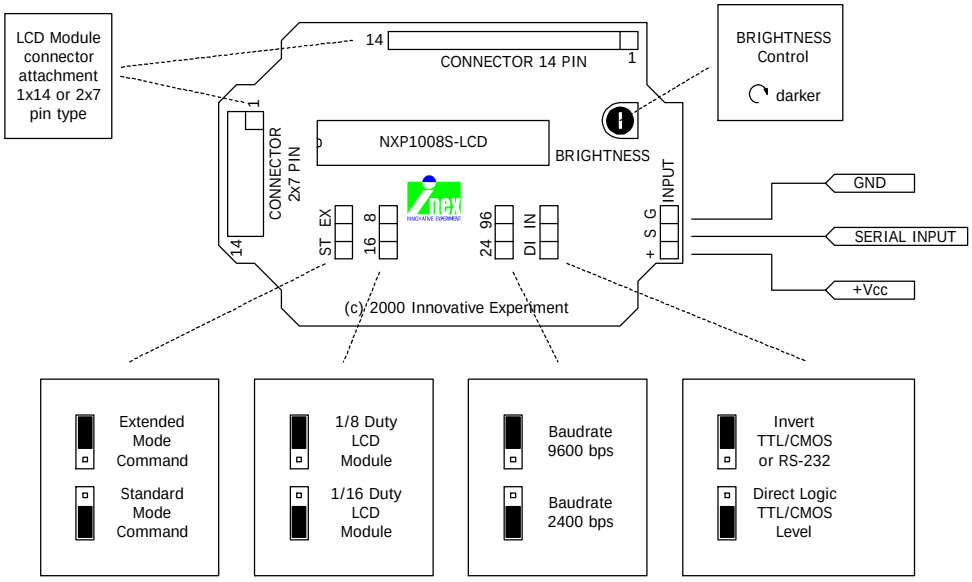
General Description

SLCD16x2 is the 16 characters 2 lines LCD module that communication by serial interface. It received data serially and display on the LCD. Accept serial data at 2400 or 9600 baudrate and accept either TTL or RS-232 level, by 2 jumpers select. Support on standard LCD controller HITACHI HD44780 or SEIKO EPSON SED1278 compatible. Both 1/8 Duty and 1/16 Duty of 1x16 LCD Module can be used by jumper selection too.

Features

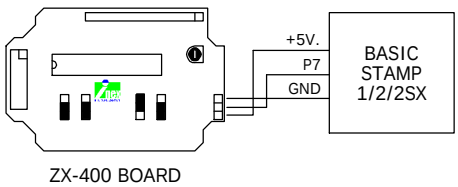
- Serial Input RS-232 or Invert/Non-invert TTL/CMOS logic level.
- 28 byte FIFO buffer, baudrate 2400 or 9600 bps selectable, 8 bit data, no parity, 1 stop bit.
- 1/8 or 1/16 Duty can be selected by jumper.
- Scott Edwards's LCD Serial Backpack ฎ command compatible addition with Extended Command that make LCD control easier.
- Easy to interface with microcontroller such as BASIC STAMP ฎ .
- Operation with 5 to 12 V.DC supply (5.1V. zener build-in)

Connection Diagram



Typical Application

Connect SLCD16x2 with BASIC STAMP 1/2/2E/2SX
Using UART serial 9600bps, Non-invert




```
' Example program for BASIC STAMP 1
        SEROUT 7,T9600,("Hello !")
' Example program for BASIC STAMP 2/2E
        SEROUT 10,84,["Hello !"]
' Example program for BASIC STAMP 2SX
        SEROUT 10,240,["Hello !"]
```

Data and Command sending

Once SLCD16x2 is properly connected and configured. Data and command can be send serially. For data sending, user can send any message such as "Hello" via serial I/O directly, "Hello" message will be shown on your LCD.

For command sending, you can send standard instruction set to LCD (see Figure C) by precede it with the instruction prefix character, ASCII 254 (0FE hex or 11111110 binary). SLCD16x2 treats the byte immediately after prefix as an instruction, then automatically returns to data mode.

An example: To clear screen on LCD, clear instruction is 00000001 binary (or ASCII 1), send [254] and [1] to SLCD16x2 (where parentheses in [ ] symbols mean single bytes set to these values)

COMMAND\DATA BIT	D7	D6	D5	D4	D3	D2	D1	D0
1. Initial LCD	0	0	0	0	0	0	0	0
2. Clear LCD	0	0	0	0	0	0	0	1
3. Returm Home	0	0	0	0	0	0	1	*
4. Entry Mode Setting	0	0	0	0	0	1	I / D	S
5. Display Setting	0	0	0	0	1	D	C	B
6. Shift Display	0	0	0	1	S / C	R / L	*	*
7. Function Setting	0	0	1	*	N	F	*	*
8. Set CGRAM Address	0	1	A5	A4	A3	A2	A1	A0
9. Set DDRAM Address	1	A6	A5	A4	A3	A2	A1	A0

Figure B
Standard instrction command set
(except Initial LCD is addition command.
Initialize make I/D=1, S=0, D=1, C=0, B=0, N=1, F=0, DDRAM Address=00

* Don't care bit

S 0=Automatic cursor shift after byte
1=Cursor not moved
I/D 0=After byte, decrease cursor position
1=After byte, increase cursor position
(when S=1, cursor won't be shifted .)

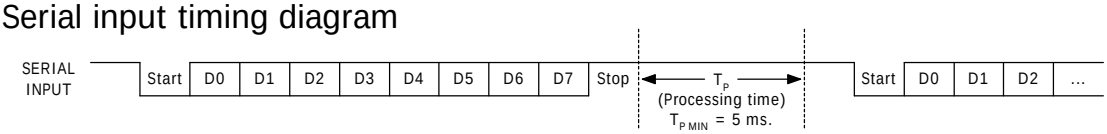
D 0=Display OFF, 1=Display ON
C 0=Cursor OFF, 1=Cursor ON
B 0=Cursor not blink, 1=Cursor blink

S/C 0=Cursor shift, 1=Display Shift
R/L 0=Left shift, 1=Right shift

N 0=1/8 Duty, 1=1/16 Duty
(not recommend to set this bit, use jumper setting instead)
F 0=5x7 dot size, 1=5x10 dot size

A0 to A7 are CGRAM or DDRAM Address

Serial input timing diagram



LCD Characters

Most of the LCD characters (Figure E) cannot be changed because they are store in ROM. However, the first eight symbols, corresponding to ASCII 0 through 7, are store in RAM. By Writing new values to the character-generator RAM (CGRAM), you can alter these characters as you want in 5x8 dots size.

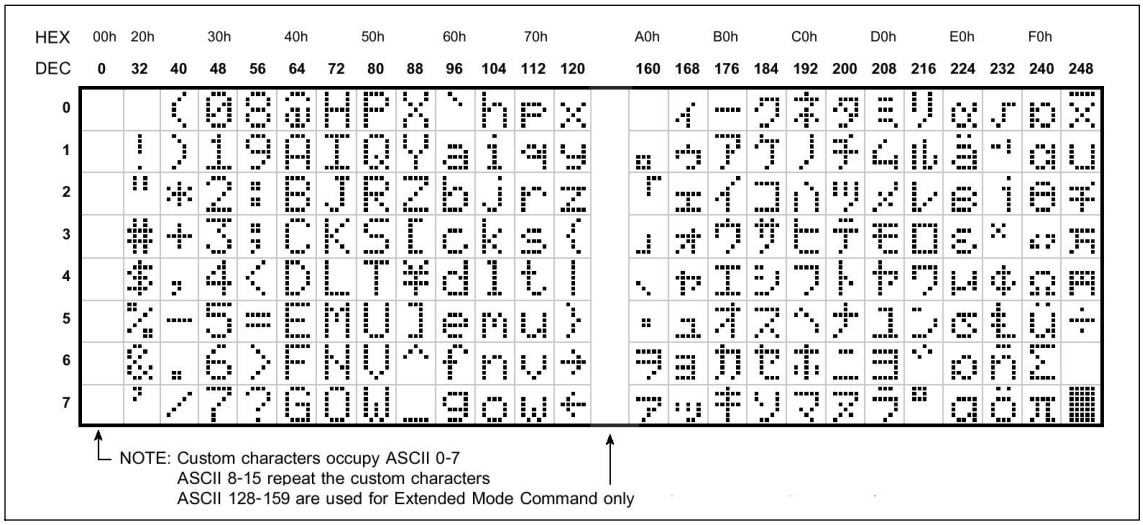


Figure C
LCD character set. (Built-in character on HD44780A or SED1278F0A)

Create your symbols by point to the CGRAM location, then write first line whose bits form the desired pattern, and point to next CGRAM address to write bits later. Repeat this procedure until 8 times (one character), your character is ready to use now.

CGRAM 0 is located on CGRAM Address 00h-07h, CGRAM 1 on 08h-0Fh, CGRAM 2 on 10h-17h, ... until CGRAM 7 on 38h-3Fh. See Figure F for example.

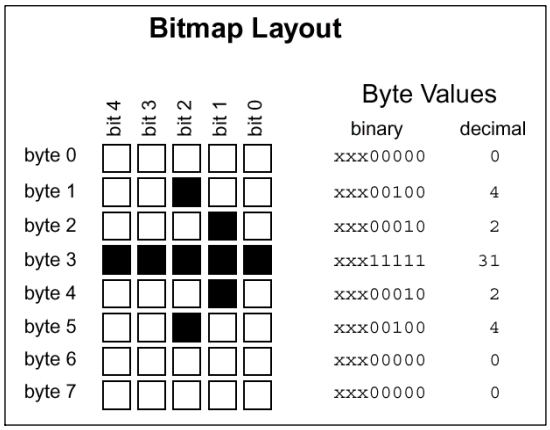


Figure D
Defining custom symbols.
Example: Load arrow symbol on CGRAM 3, a program would send the following bytes to the ZX-400.

[254] , [01011000 b] , [0] ,
[254] , [01011001 b] , [4] ,
[254] , [01011010 b] , [2] ,
[254] , [01011011 b] , [31] ,
[254] , [01011100 b] , [2] ,
[254] , [01011101 b] , [4] ,
[254] , [01011110 b] , [0] ,
[254] , [01011111 b] , [0]

Extended your command with Extended mode.

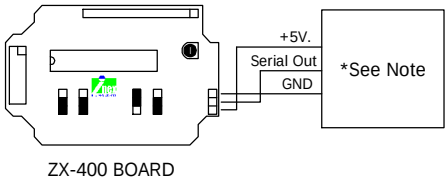
You can control your LCD easier by using Extended Mode Command (to enable this mode, set first left jumper to " EX " position), In this mode,instruction prefix had no needed. Extended command has shown in text below.

ASCII	Instruction / Action	ASCII	Instruction / Action
128	Initial LCD	142	Write CGRAM 0 (*See note)
129	Clear Screen	143	Write CGRAM 1
130	Return Home cursor	144	Write CGRAM 2
131	Cursor not move after byte (S=1)	145	Write CGRAM 3
132	Cursor increase after byte (I/D=1)	146	Write CGRAM 4
133	Cursor decrease after byte (I/D=0)	147	Write CGRAM 5
134	Display ON (D=1)	148	Write CGRAM 6
135	Display OFF (D=0)	149	Write CGRAM 7
136	Display ON with Cursor on	150	Set DDRAM to 00h
137	Display ON with Blink Cursor on	151	Set DDRAM to 10h
138	Shift Cursor to left	152	Set DDRAM to 14h
139	Shift Cursor to right	153	Set DDRAM to 20h
140	Shift Display to left	154	Set DDRAM to 40h
141	Shift Display to right	155	Set DDRAM to 50h
		156	Set DDRAM to 54h

Note. For CGRAM write command (ASCII 142 - 149) , Program would send 8 bytes whose bits form the desired pattern follow the command.
Example: If you want to load arrow symbol as above of this page on CGRAM 3, a program will be modified to use in extended mode as: [145] , [0] , [4] , [2] , [31] , [2] , [4] , [0] , [0].
That's all procedure to do!

Example program

Connect ZX-400 with BASIC STAMP 1/2/2E/2SX
Using UART serial 9600bps, Non-invert



Note: Example connection with microcontrollers

MCS-51
Connect serial input pin to P3.1
BASIC STAMP 1/2/2E/2SX
You can define any pin as serial output.
PIC
You can define any I/O use any pin but depend on programming and internal I/O function register.

Example program

```
' Example Assembly program for MCS-51
' Running at 11.0592MHz, baudrate 9600bps, Standard mode

            ORG   0000H
            AJMP  INITIAL     ; Jump to initial


INITIAL:    MOV   TMOD,#021H  ; Timer 1 Mode 8bit auto reload
            MOV   TH1,#0FDH   ; Set reload value
            MOV   TL1,#0FDH   ;
            MOV   SCON,#040H  ; Set Serial mode 1,
                              ; not receive data
            SETB  TR1         ; Start timer 1

            ACALL DELAY_1s    ; Wait 1 s. for ZX-400 initial

            MOV   R2,#9       ; Set loop 9 times sending
                              ; ( 2 for Command, 7 for Data )
            MOV   DPTR,#TEXT  ; Set pointer to "Hello !"
            CLR   TI          ; Clear TI
SEND:       CLR   A           ; Clear ACC
            MOVC  A,@A+DPTR   ; Get data from pointer
            MOV   SBUF,A      ; Send data to serial
            INC   DPTR        ; Increase pointer to next byte
            JNB   TI,$        ; Wait until sending complete
            CLR   TI          ; Clear TI after sending
            DJNZ  R2,SEND     ; Do loop until finish 9 times
            ; After this, "Hello !" message
            ; was shown on LCD.

            ; Anything you want..

D100ms:     MOV     R7,#100   ; Do 100 times
D100ms_1:   MOV     R6,#0E6H  ; Each loop = 1 ms
D100ms_2:   NOP
            NOP
            DJNZ    R6,DELAY_100ms_2
            DJNZ    R7,DELAY_100ms_1
            RET

DELAY_1s:   MOV     R5,#100   ; Do 100 times
DELAY_1s_1: ACALL   DELAY_10ms
            DJNZ    R5,DELAY_1s_1
            RET

TEXT  DB    254,0,"Hello !"   ; Instruction prefix,Initial
                              ; Send text "Hello !"



' Example PBASIC program for BASIC Stamp I
' Using LCD 16x2 type
' connect I/O pin0 as serial output, invert logic level
' baudrate 2400bps, Standard mode

SYMBOL      I     = 254
SYMBOL      CLR   = 1
SYMBOL      LINE2 = $C0       ' Set address to $40

PAUSE       1000
SEROUT      0,N2400,(I,CLR)         ' Clear screen
SEROUT      0,N2400," Hello Stamp 1! ") ' Show message
SEROUT      0,N2400,(I,LINE2,"Test Line 2 LCD ") ' Test Line 2

END
```

Example program (Continues)

```
' Example PBASIC program for BASIC Stamp 2/2E
' Using LCD 16x2 type
' connect I/O pin8 as serial output, invert logic level
' baudrate 9600bps, Extended mode

CLR         CON 129
LINE2       CON 154           ' Set address to $40
LINE1       CON 150           ' Set address to $00
BLINK       CON 137           ' Cursor on & blink
BAUD9600    CON $4054

PAUSE 1000              ' Delay 1 sec. wait for power-on initial
SEROUT      8,BAUD9600,[CLR]        ' Clear screen
SEROUT      8,BAUD9600,[" Hello Stamp 2! "] ' Show message
SEROUT      8,BAUD9600,[LINE2,"Test Line 2 LCD "] ' Test Line 2
SEROUT      8,BAUD9600,[LINE1,BLINK]

END

' Example PBASIC program for BASIC Stamp 2SX
' Using LCD 16x2 type
' connect I/O pin8 as serial output, invert logic level
' baudrate 9600bps, Extended mode

CLR         CON 129
LINE2       CON 154           ' Set address to $40
LINE1       CON 150           ' Set address to $00
BLINK       CON 137           ' Cursor on & blink
BAUD9600    CON $40F0

PAUSE 1000              ' Delay 1 sec. wait for power-on initial
SEROUT      8,BAUD9600,[CLR]        ' Clear screen
SEROUT      8,BAUD9600,["Hello Stamp 2SX!"] ' Show message
SEROUT      8,BAUD9600,[LINE2,"Test Line 2 LCD "] ' Test Line 2
SEROUT      8,BAUD9600,[LINE1,BLINK]

END

' Example PBASIC program for BASIC Stamp 2SX
' Using LCD 16x1 type
' connect I/O pin8 as serial output, invert logic level
' baudrate 9600bps, Standard mode

I           CON $FE           ' Define prefix instruction $FE  as I
LCD_DATA    VAR BYTE          ' Define Byte variable to keep LCD data.
N           VAR BYTE          ' Define Byte variable

PAUSE 1000              ' Delay 1 sec. wait for power-on initial

FOR N=0 TO 7
  LOOKUP N,[%00010,%00110,%11010,%10010,%10010,%11010,%00110,%00010],LCD_DATA
  ' Data table for generate speaker symbol
  SEROUT 0,16624,[I,%01000000+N,LCD_DATA] ' Send CGRAM0 loop
NEXT

PAUSE 100               ' Delay 100 ms for prevent from data stream overflow

FOR N=0 TO 7
  LOOKUP N,[%00100,%00110,%00101,%00101,%01100,%11100,%01000,%00000],LCD_DATA
  ' Data table for generate note symbol
  SEROUT 0,16624,[I,%01001000+N,LCD_DATA] ' Send CGRAM1 loop
NEXT

PAUSE 100               ' Delay 100 ms for prevent from data stream overflow

MAIN: SEROUT 0,16624,[I,1,"Serial L",I,%11000000,"CD show."]     ' Send message
      PAUSE 2000
      SEROUT 0,16624,[I,1,I,%00001111]   ' Clear screen and set cursor blink
      PAUSE 500
      SEROUT 0,16624,100,[0,1," Speak",I,%11000000,"er CGRAM"]   ' Send message slowly
      PAUSE 4000
      SEROUT 0,16624,[I,1]    ' Clear screen
      PAUSE 1000
      GOTO MAIN
```