Serial LCD+2 Users Guide Version 1.1

Note: LCD+2 replaces the older LCD+.

The Serial LCD+2 is a 4x20 LCD display with a built-in bi-directional serial interface. The Serial LCD+2 is controlled using standard RS-232 serial signals from a host computer or microcontroller supporting 2400, 4800, 9600, 19,200, 38,400 and 57,600 Kbps serial data rates.

Table of Contents

Table of Contents 1
Important Warranty Information 1
LCD+2 Improvements 2
LCD+2 Pin Definitions 2
  Serial I/O and +5 Power tap 3
  Power Input 3
  COM Jumper 3
  ADC Inputs 3
  Relay Driver Outputs 3
  Matrix KeyPad Input 3
Initial power-up test 4
    Initial power-up test continued 4
Keypad Interface 4
ADC Interface 5
Defining custom characters 5
Relay Interface 6
LCD+2 Demo Software 6
LCD+ Control Codes 7
Retrieving the LCD+ EEPROM settings 7
Specifications 8
Example Code 8
BX-24 Example 8
Required Connections: 8
Bezel Option 10
Dimensions 11

Important Warranty Information

NetMedia, Inc. warrants this product against defects in materials and manufacturing workmanship for a period of 90 days. Warranted returns will be repaired or replaced at the option of NetMedia, Inc. Products showing signs of alterations or mechanical damage are not eligible for warranty replacement. For warranty returns please contact your distributor or NetMedia directly. Please have a copy of your invoice handy.

NetMedia, Inc. Attn: Warranty Repair
10940 N. Stallard Place
Tucson, AZ 85737.
(520) 544-4567

The warranted item or its replacement will be returned via standard shipping. For expedited or foreign shipping extra shipping charges will apply.

LCD+2 Improvements

LCD+2 is a direct replacement for the popular LCD+ serial display. Customer feedback played a key role in the design and layout of the new LCD+2. To maintain backwards compatibility for our OEMs the footprint and dimensions of the LCD+2 are identical to that of the older LCD+.

Hardware Added:

1. On-Board DB-9 serial connector.
2. Barrel connector for 9-12vdc input.

Hardware Changes:

1. Jumper selectable input range was removed from ADC inputs 6-8.
2. ADC inputs 6-8 have been moved to the header beside the power jack.
3. ADC inputs, Relay Outputs and power connection holes have been aligned for use with Euro connectors.
4. +5 Power input was removed from the solder header row.

Software Changes:

1. Backlight, Contrast and Beeper frequency settings are no longer stored in EEPROM. At power up the default settings are; Backlight = 64, Contrast = 128 and Bell/Buzzer Frequency = 124. Changes to these settings will be lost at power down.

LCD+2 Pin Definitions

Figure 1 shows a top down view the LCD display module has been removed to make the connections more visible.

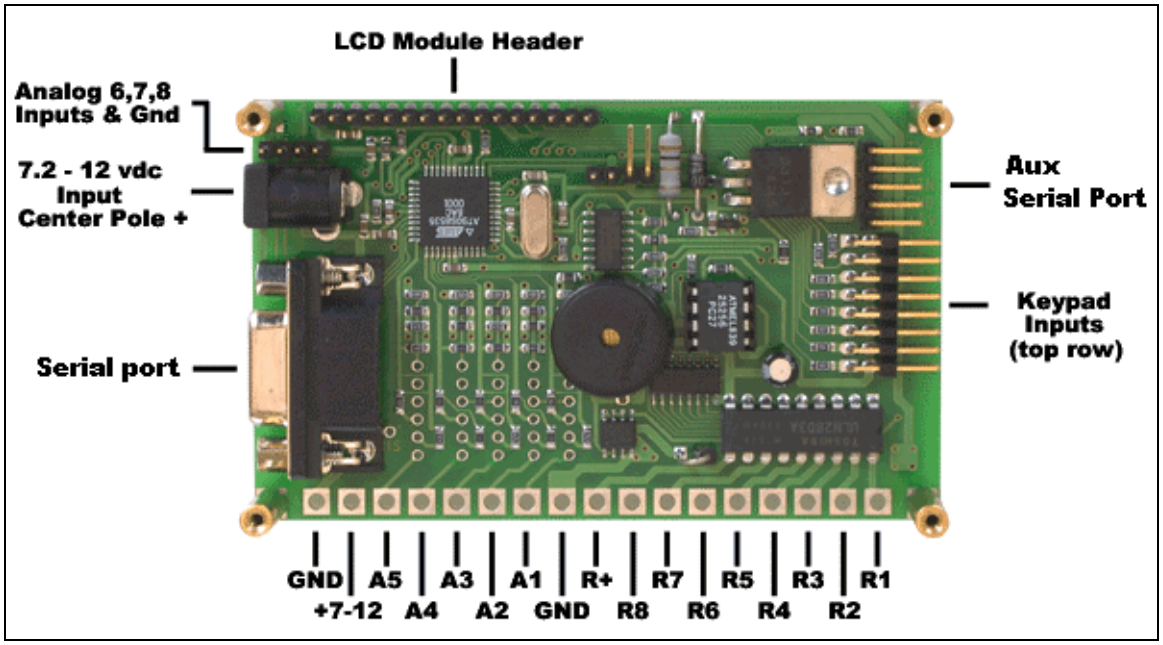


Figure 1

Serial I/O and +5 Power tap

Serial is connected to the LCD+2 via the DB-9 serial connector. An auxiliary serial connection is also provided. The DB-9 and Aux Serial inputs contain the same signals and are tied together on the PCB board. The Aux serial I/O header is made up of 5 connections. The first connection marked **TX** is the LCD+2 serial output. **TX** connects to the host computers/microcontrollers serial input. The Next connection marked **RX** is the LCD+2 serial Input. **RX** connects to the host computers/microcontrollers serial output. The middle connection marked **ATN** is not currently implemented and should be left unconnected. The connection marked **GND** is the serial I/O ground connection and connects to the host computers/microcontrollers serial ground connection. The connection marked **+5v** is a courtesy +5volt 20mA voltage source provided by the LCD+2 voltage regulator. Note all four of the LCD+2 **+5v** courtesy connections are tied to the same voltage regulator, the combined load for all of these connections must not exceed 20mA.

Power Input

The LCD+2 is powered using one of the three provided power connections. The barrel connection expects 5.5v – 12vdc with the center of the plug being positive (Center Pole Positive). The same 5.5v – 12vdc can be used to power the LCD on pads V+ and GND. The LCD+2 can also be powered using a regulated +5vdc supply on the pins marked +5 and GND. The two **+5.5 to +12V** inputs are tied to the LCD+2 modules onboard regulator. The pin marked **+5v** ties to the +5v buss and is used to bypass the LCD+2 onboard regulator when a regulated 5volt source is supplied.

*Note: To insure proper backlight function when running the LCD+2 from a regulated 5volt source, you must connect both the regulator input (**+5.5 to +12V**) and the 5volt input (pine marked **+5**) connections together.*

Important Note: Reversing the positive and negative power connections will cause immediate damage to the LCD+ and void your warranty!

COM Jumper

The **COM** jumper connection sets the LCD+ to its inverted serial receive mode. If a jumper is installed across COM the LCD+2 will read logically inverted serial inputs. COM factory default setting = no jumper installed.

ADC Inputs

The eight ADC inputs are labeled **A1 – A5 & A6 - A8**. By default all ADC input are set to read voltage in the 0 to 5volt range. *Note exceeding this range will damage the ADC inputs.*

Above each of the ADC inputs **A1 – A5** there are provisions for input voltage range jumpers, the use of these optional jumpers will be explained later in this text. ADC inputs **A6 - A8** are 0-5v only.

Relay Driver Outputs

There are ten Relay Driver connections (Labeled on the underside of the board as R1-R8, RV and GND). The connections labeled **R1 – R8** are the relay driver chip outputs, the **RV** connection provides access to the ULN2803A relay driver chip's internal back EMF protection diodes. **GND** is the common ground.

Matrix KeyPad Input

The KeyPad input connections (Visible in figure 1) are the upper most eight of the 8x2 header connection. The lower most eight of the 8x2 header connection (Not visible in figure 1) are used by the factory for programming and should be left unconnected.

Initial power-up test
Before connecting the LCD+2 to your PC or microcontroller you should do a power-on check, this insures your power connections are correct and prevents you from damaging your PC should there be a problem. On initial power up the LCD+2 should display "9600 BAUD". If after connecting power to the LCD+2 you do not see the power-up message check your power connections, your power supply may not be providing enough output current.

Initial power-up test continued

Once you have seen the power-up message you are ready to connect the LCD+2 to your computer. With your computers power off, connect the LCD+2 to your computer's serial port using a DB-9 strait through serial cable. Now you can turn on your computer and run a serial terminal program such as the LCD+2 demo program or HyperTerminal.

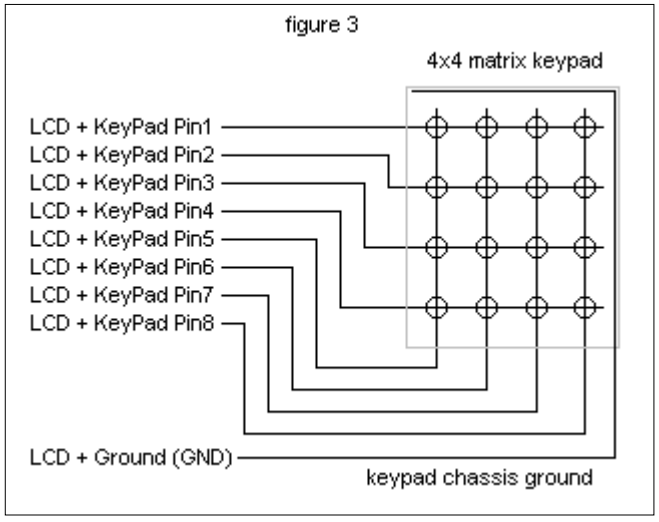
If everything is connected and configured properly, all keystrokes sent through the program's text window will appear on the LCD+2 display.

Keypad Interface

The keypad interface supports matrix keypads up to 4x4 in size (16 keys). Instead of predefining the keypad keys serial data format as 0 though 15, each key of the keypad's keys is serially represented by a user definable byte value. This user definable value or "Tag" is stored within the EEPROM as a 0 - 15byte array. Each byte of the array corresponds to a key on the keypad (i.e. key 0 corresponds to byte0 of the array). Whenever a key is pressed, the stored byte representation for that key number is serially sent. *Note: The keypad is not a polled feature. Key press events are immediately sent out the serial TX pin.*

Figure 3 shows how to interface a common 4x4 matrix keypad. Pin1 of the matrix keypad header is located on the top row closest to the Aux serial header. When using a smaller keypad size the connection/connections for the unused row and or column is simply left disconnected. For example if the keypad in **figure 3** had four rows but only three columns the connection to Pin8 (from the now missing column) would be left disconnected.

figure 3



Keypad Options

The keypad feature is supported by five user definable options or "modes". Setting these modes is done by sending a CTRL-X followed by your command byte containing the desired "modes". Only the lower 5 bits of the mode command byte are used. As shown is Table 1, placing a 1 in any of these lower 5 bits turns its corresponding option on, and a 0 turns it off

Table 1

Mode Byte	Name	Description
B0	1 = "Key Beeps"	Beep buzzer during each key press
B1	1 = "Key Press format"	Send 1 byte for key down and 1 for key up
B2	1 = "LCD Echo"	Echo key press data ASCII representation to LCD display
B3	1 = "Mask Key Presses"	Display all key presses as asterisks on LCD
B4	1 = "Auto Backlight"	Turns on backlight with any key press and off 4 seconds after last key press

Changing the Keypad's key "Tags"

The LCD+2 keypad Tags may be serially changed by sending a CTRL-T (20 Decimal) followed by 16 bytes containing the new keypad values.

Using the Keypad interface for input

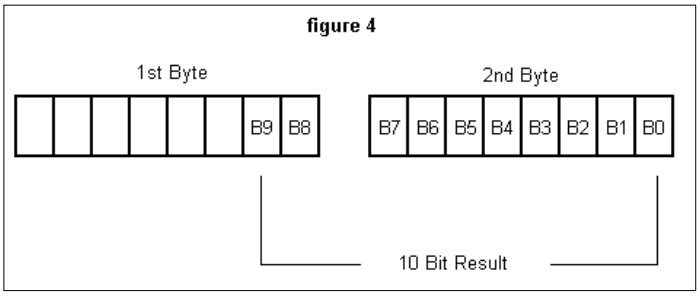
If a CTRL-Y is sent to the LCD+2, the keypad input pins are read as a port. The logical state of the eight keypad input pins are then returned in byte format.

ADC Interface

The LCD+2 is equipped with eight 10Bit Analog to digital converters (ADCs). Although they are labeled 1 – 8 on the LCD+2 board, the ADCs are serially requested as 0 –7. An ADC reading request is initiated by sending the ADC function byte (Ctrl-V or 22 Decimal) immediately followed by a byte containing the requested ADC channel number (0-7 Decimal).

After taking the ADC measurement, the LCD+2 sends the 10bit result out the TX pin in two bytes (seen in figure 4). The first byte contains the extra two bits of the 10Bit result in the bit 0 and 1 position, the second byte contains the bottom eight bits in proper order. After reception, the host computer or microcontroller can shift these two bytes into a 10bit result.

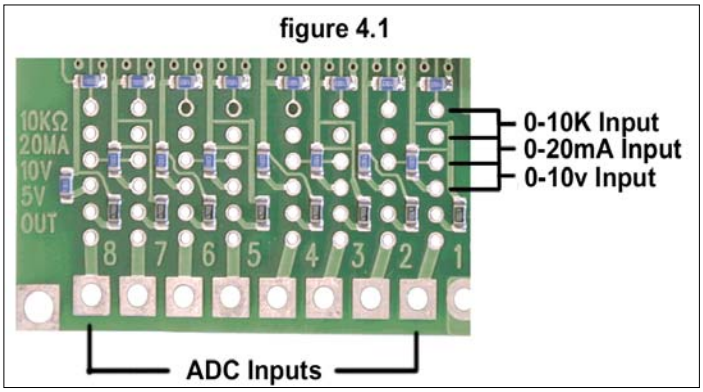
figure 4



ADC channels 1-5 each have four jumper selectable input ranges. By default (with no Jumper installed) this range is 0-5v. The additional ranges shown in Figure 4.1 are set using jumper wire or installing a header-pin strip and using conventional 100mil jumpers. Only the top four holes are used, the bottom most two should be left unconnected.

Note: Never install more than one jumper per row.

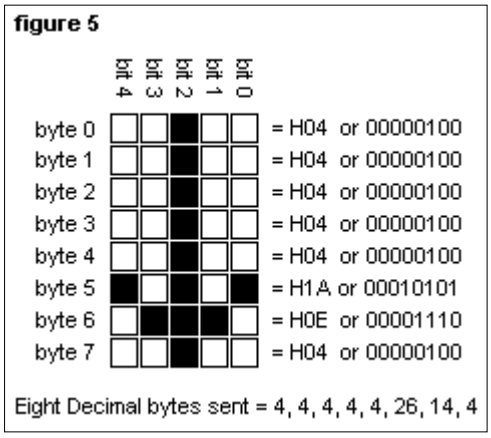
figure 4.1



Defining custom characters

The standard LCD+2 character also contains eight user definable characters. Each of these eight characters (CTRL 0 – 7) is 5 pixels wide and 8 pixels tall and can be used for making things like arrows, custom symbols or any other non-standard character. Writing a new custom character to EEPROM is accomplished by sending the custom character write command byte (CTRL-S or Decimal 19), the characters number (0-7 Decimal) followed by the eight bytes that makeup the new character. **Figure 5** shows how each of those eight bytes correspond to a character.

figure 5

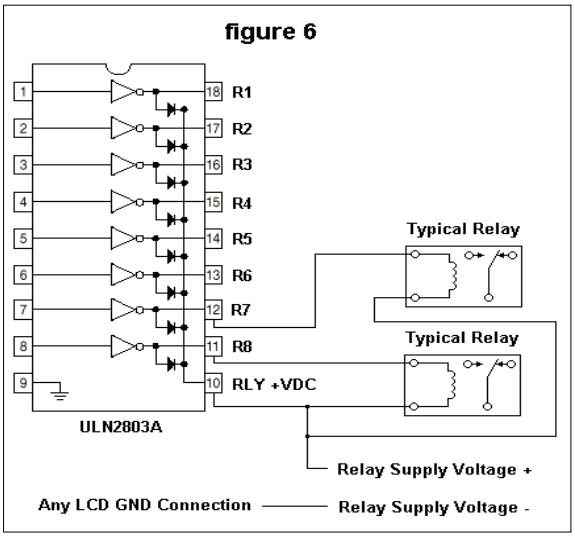


Relay Interface

The ULN2803A relay driver chip used in the LCD+2 consists of eight open collector Darlington transistors (NPN Type).

Figure 6 shows the proper method of connecting relays to the LCD+2.

A set relay command is initiated by sending the relay function byte (CTRL-R or 18 Decimal) followed by a relay set byte. Each relay's state is represented by its own Bit in the relay set byte (R1 = B0 etc). For example, if you wanted to turn both relay 1 and 8 on and 2-7 off, the relay set byte would be 10000001 or 129 Decimal.



LCD+2 Demo Software

The LCD+2 demo program is provided to help first time users test and become acquainted with the LCD+2's many features. This demo program was designed to run on computers running Microsoft Windows versions 95 and up. The demo is not compatible with Asian Windows versions.

Once you have installed the demo program you will need to set your computers comport before use. On the demo programs first screen, click the button labeled "Select/Configure Comport". In the new box marked "Comport properties", select the comport number that corresponds to the one your LCD+2 is or will be connected to. Then make sure 9600 baud is selected (all LCD+2 modules are shipped in 9600 baud mode). Before clicking the OK button, note the setting of the "Local Echo" option box, this option should to be set to on.

With your LCD+2 module connected and powered up, click on the button marked open port. If properly attached/configured, the demo program will download the LCD+2 module's current EEPROM settings and display them under both the "KeyPad" and "LCD settings" tabs.

If you are unsure of what a particular control item does, hold your mouse cursor over it and a "tool tip" should pop up and explain it.

LCD+ Control Codes

Control Code	Function	Total bytes needed + Command +Data	Return Data
Ctrl-A	Cursor Home	1Byte(01Hex or 1Dec.)	None
Ctrl-B	Set/Adjust Backlight Brightness	2Bytes(02Hex or 2Dec.) + 0-255	None
Ctrl-C	Set/Adjust Contrast	2Bytes(03Hex or 3Dec.) + 0-255	None
Ctrl-D	Hide Cursor	1Byte(04Hex or 4Dec.)	None
Ctrl-E	Underline Cursor	1Byte(05Hex or 5Dec.)	None
Ctrl-F	Block Cursor	1Byte(06Hex or 6Dec.)	None
Ctrl-G	Sound Bell/Buzzer	1Byte(07Hex or 7Dec.)	None
Ctrl-H	Backspace	1Byte(08Hex or 8Dec.)	None
Ctrl-I	Horizontal tab	1Byte(09Hex or 9Dec.)	None
Ctrl-J	Line feed	1Byte(0Ahex or 10Dec.)	None
Ctrl-K	Reverse line feed	1Byte(0Bhex or 11Dec.)	None
Ctrl-L	Form feed/Clear screen	1Byte(0Chex or 12Dec.)	None
Ctrl-M	Carriage return	1Byte(0Dhex or 13Dec.)	None
Ctrl-N	Backlight on	1Byte(0Ehex or 14Dec.)	None
Ctrl-O	Backlight off	1Byte(0Fhex or 15Dec.)	None
Ctrl-P	Set cursor position	2Bytes(10Hex or 16Dec.) + 1-80	None
Ctrl-Q	Clear column	1Byte(11Hex or 17Dec.)	None
Ctrl-R	Set Relays	2Bytes(12Hex or 18Dec.) + 0-255	None
Ctrl-S	Define custom character	3Bytes(13Hex or 19Dec.) + 0-7 + 1Bytes	None
Ctrl-T	Download Keypad Tags	17Bytes(14Hex or 20Dec.) + 16 New keys	None
Ctrl-U	Set Baud rate	2Bytes(15Hex or 21Dec.) + Dec191=2400, Dec95=4800, Dec47 = 9600, Dec31=14,400, Dec23=19,200, Dec15= 28,800, Dec11= 38,400, Dec7=57,600	None
Ctrl-V	Read ADC inputs	2Bytes(16Hex or 22Dec.) + 0-7	2 Bytes
Ctrl-W	Change Bell/Buzzer Frequency	2Bytes(17Hex or 23Dec.) + 0-255	None
Ctrl-X	Set Keypad modes	2Bytes(18Hex or 24Dec.) + 0-255	None
Ctrl-Y	Read Keypad input as port	1Byte(19Hex or 25ec.)	1 Byte
Ctrl-Z	Get LCD+ EEPROM settings	1Byte(1Ahex or 26ec.)	20 Bytes
None	Display custom character 0	1Byte(80Hex or 128Dec.)	None
None	Display custom character 1	1Byte(81Hex or 129Dec.)	None
None	Display custom character 2	1Byte(82Hex or 130Dec.)	None
None	Display custom character 3	1Byte(83Hex or 131Dec.)	None
None	Display custom character 4	1Byte(84Hex or 132Dec.)	None
None	Display custom character 5	1Byte(85Hex or 133Dec.)	None
None	Display custom character 6	1Byte(86Hex or 134Dec.)	None
None	Display custom character 7	1Byte(87Hex or 135Dec.)	None

Retrieving the LCD+ EEPROM settings

The LCD+2 settings and configuration data is stored within an on-board EEPROM. Access to this data is provided through the CTRL-Z command . Whenever a CTRL-Z is sent, the LCD+ responds by sending a 20byte packet of data containing the current EEPROM settings. The table below shows the order and type of data returned.

Byte order	Data
1	Bell/Buzzer Current Frequency value (0-255)
2	Current LCD Backlight value (0-255)
3	Current LCD contrast value (0-255)
4	Current Keypad “mode” settings (000XXXXX) as Byte
5-20	Current Keypad “Tags” (0-255) x 16

Specifications

Power Requirements	5.5 to 12V dc @ 9mA (200mA max backlight on)
Serial I/O	RS232 or Inverted TTL (2400 to 57,600 8,N,1)
ADC input Leakage	50nA Typical
Relay Driver Outputs	500mA max per output
Keypad Input	4x4 Matrix type
LCD type	4 x 20 Supertwist
Temperature limits	0° to 50° C Operating and -10° to 60° storage

Example Code

BX-24 Example

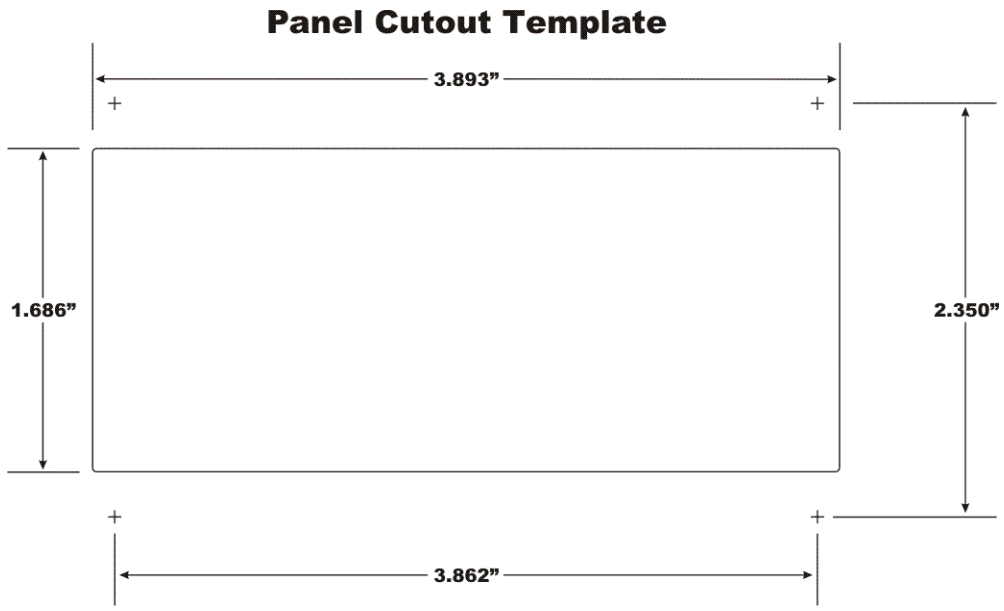
Required Connections:

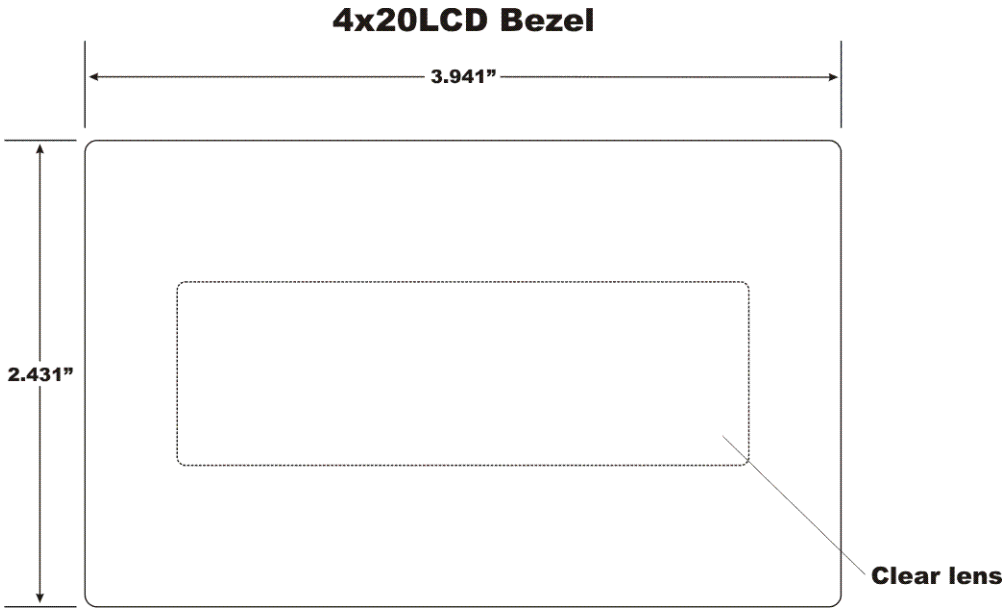
LCD+2 TX to BX-24 Pin12
LCD+2 RX to BX-24 Pin11
LCD+2 GND to BX-24 Pin23
LCD+2 5.5-15volt input connected to appropriate power source.

```
Dim Com3_Out(1 To 40) As Byte 'Com3 Out Buffer
Dim Com3_In(1 To 30) As Byte  'Com3 In Buffer
Dim I As Byte
Const Clear_Screen As Byte = 12
Const Postion_Cursor As Byte = 16
Const Brightness As Byte = 2
Const Sound_Bell As Byte = 7
Const Bell_Freq As Byte = 23

'--------------------------------------------------------------------------------

Sub Main()

     'Open Com3 serial port queues
     Call OpenQueue(Com3_Out, 40)
     Call OpenQueue(Com3_In, 30)

     'Define pins for Com3 (RX = pin12, TX= pin11)
     Call DefineCom3(12, 11, bx1000_1000)
     'Open Com3 at 9600,N,8,1
     Call OpenCom(3,9600, Com3_In, Com3_Out)
     'Wait for a bit
     Call Delay(0.5)

     'Clear LCDs screen
     Call PutQueuestr(Com3_Out,Chr(Clear_Screen))

     'Put cursor at position 27 and print "READY?"
     Call PutQueuestr(Com3_Out,Chr(Postion_Cursor) & Chr(27))
     Call PutQueueStr(Com3_Out, "READY?")
     Call Sleep(1.0)
```

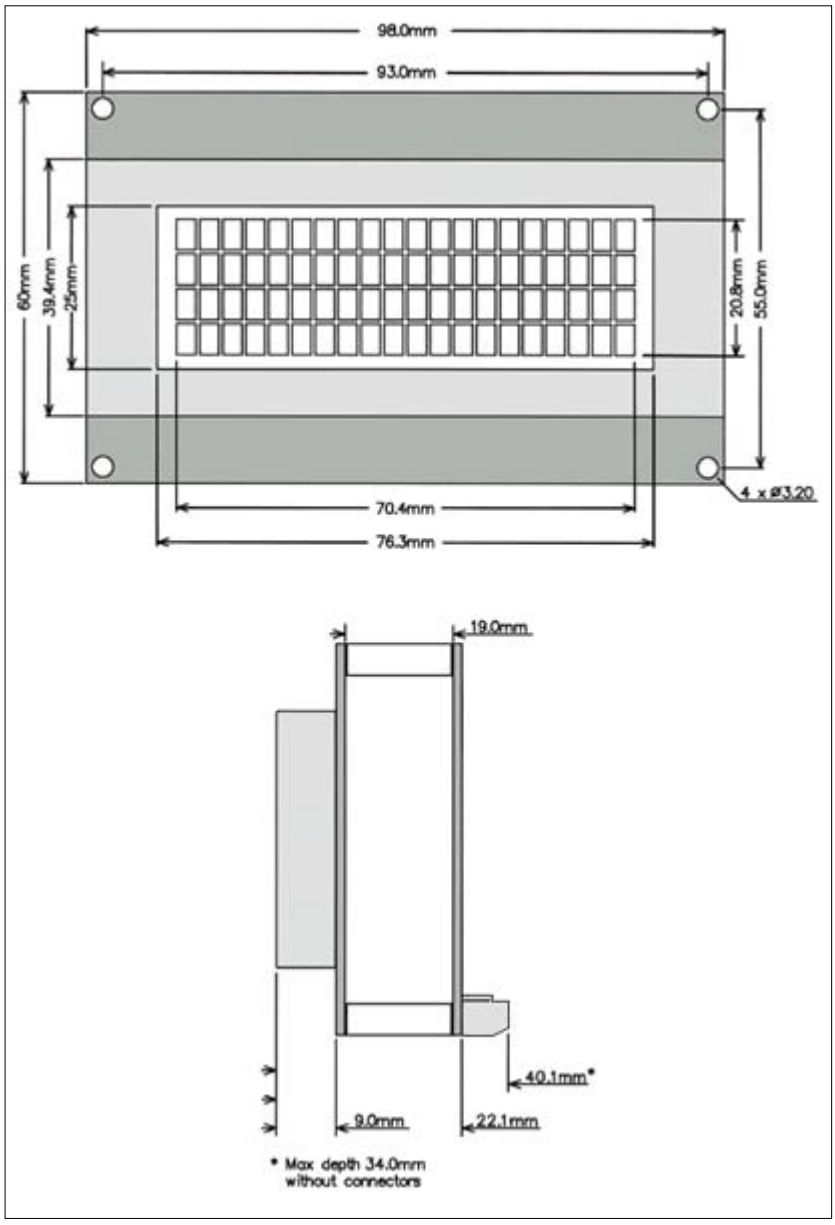
```
    'Clear LCD screen
    Call PutQueuestr(Com3_Out,Chr(Clear_Screen))

    Call PutQueuestr(Com3_Out,Chr(Postion_Cursor) & Chr(28))
    Call PutQueueStr(Com3_Out, "AIM")
    Call Sleep(1.0)

    'Clear LCD screen
    Call PutQueuestr(Com3_Out,Chr(Clear_Screen))

    Call PutQueuestr(Com3_Out,Chr(Postion_Cursor) & Chr(28))
    Call PutQueueStr(Com3_Out, "FIRE!")
    Call Sleep(0.5)

    'Increment the bell frequency from 1 to 230 and sound bell on each freq.
    For I = 1 to 230
        Call PutQueuestr(Com3_Out,Chr(Bell_Freq) & Chr(I))
        Call Sleep(15)
        Call PutQueuestr(Com3_Out,Chr(Sound_Bell))
    Next

        'Clear LCD screen
        Call PutQueuestr(Com3_Out,Chr(Clear_Screen))

        Call PutQueuestr(Com3_Out,Chr(Postion_Cursor) & Chr(27))
        Call PutQueueStr(Com3_Out, "BOOM!!!")

    'Increment the Backlight level from 1 to 220
    For I = 1 to 220
        Call PutQueuestr(Com3_Out,Chr(Brightness) & Chr(I))
        Call Sleep(5)
    Next

        Call Sleep(1.0)

        Call PutQueuestr(Com3_Out,Chr(Postion_Cursor) & Chr(25))
        Call PutQueueStr(Com3_Out, "You sunk my")
        Call PutQueuestr(Com3_Out,Chr(Postion_Cursor) & Chr(65))
        Call PutQueueStr(Com3_Out, "BATTLESHIP!")

        Call Sleep(3.0)


        Call PutQueuestr(Com3_Out,Chr(Postion_Cursor) & Chr(4))
        Call PutQueueStr(Com3_Out, "The New LCD+")
        Call PutQueuestr(Com3_Out,Chr(Postion_Cursor) & Chr(23))
        Call PutQueueStr(Com3_Out, "CAN DO IT ALL!")
        Call PutQueuestr(Com3_Out,Chr(Postion_Cursor) & Chr(40))
        Call PutQueueStr(Com3_Out, "ADCs, Relays, Keypad")
        Call PutQueuestr(Com3_Out,Chr(Postion_Cursor) & Chr(61))
        Call PutQueueStr(Com3_Out, "Sound & I/O Input")

        Do

        Loop

End Sub
```

Bezel Option





Dimensions


98.0mm
93.0mm
60mm
39.4mm
25mm
20.8mm
55.0mm
4 x Ø3.20
70.4mm
76.3mm
19.0mm
40.1mm*
9.0mm
22.1mm
* Max depth 34.0mm
without connectors