

LCDX Software Guide

Ver 1.0



Table of Contents:

Overview.....	2
LCDX Function Library	2
LCDInitialize	3
DisplayLine.....	3
DisplayAtCell	4
LCDPutCell	4
ClearScreen.....	5
SetContrast.....	5
SetBrightness	6
LCDGetCell.....	6
LCDPutCursorStyle.....	7
LCDMoveCursor	7
SetCustomChar	8
LCDScrollUp.....	10
Beep	10
GetVoltage	11
GetVoltageI.....	11
XPin	12
SetRelay	12
GetRelay	13
SetAllRelays	13
GetAllRelays.....	14
GetKeypad	14
GetKeypadRaw.....	15
Pin Assignments	16
Misc. Specifications.....	16
Warranty Information	17

Overview

LCDX is a BasicX Microcontroller with a built-in 4x20 LCD display. The All-In-One design saves both space and time by allowing you to build powerful display and user input orientated devices with one easy to integrate module. The LCDX features are controlled using the “LCDX Function Library” or by user modified/supplied code.

LCDX Function Library

The LCDX Function Library is a Basic file containing Subroutines and Functions for controlling the various LCDX features. Including this file into your BasicX project makes the LCDX features appear as standard built-in BasicX language commands (example: Beep, PutRelay, LCDScrollUp). The LCDX function library is supplied on CD with the LCDX development kit, it can also be downloaded from the LCDX page at BasicX.com

LCDInitialize

Syntax

Call LCDInitialize

Arguments

None

Description

Initializes the 4x20 LCD display and loads default setting for contrast, backlight, etc. All relay outputs are set to 0, backlight level is set to 254 but not turned on and contrast is set to 0 (max contrast). LCDInitialize Must be called once at the beginning of all LCDX programs.

Example

Call LCDInitialize

DisplayLine

Syntax

Call DisplayLine(Text , Row)

Arguments

Item	Type	Direction	Description
Text	String	Input	Text string to display
Row	Byte	Input	Row to display text on

Description

Writes a string starting at a specified row. Range of Row is 1 to 4.

Example

Call DisplayLine("Hello" , 1) ' Print "Hello" on line 1

DisplayAtCell

Syntax

Call DisplayAtCell(Text, Row, Column)

Arguments

Item	Type	Direction	Description
Text	String	Input	Text string to display
Row	Byte	Input	Row to display text on
Column	Byte	Input	Column to start text at

Description

Writes a string starting at a specified row and column. Useful for updating a data field without needing to rewrite the whole line or screen. Range of Row is 1 to 4 and Column is 1 to 20.

Example

Call DisplayAtCell("Hello" , 1, 5) ‘ Print "Hello" on line 1 starting at cell 5

LCDPutCell

Syntax

Call LCDPutCell (Data Byte, Row, Column)

Arguments

Item	Type	Direction	Description
Data Byte	Byte	Input	Character to display
Row	Byte	Input	Row to display text on
Column	Byte	Input	Column to start text at

Description

Writes 1 character to a specified cell on the LCD display. Useful for displaying custom graphic characters. Data is from 0-255, the range of Row is 1 to 4 and Column is 1 to 20.

Example

Const Data = 200

Call LCDPutCell(Data , 1, 5) ‘ Print the character representation of 200 on line 1 starting at cell 5

ClearScreen

Syntax

Call ClearScreen

Arguments

None

Description

Clears the LCD display

Example

Call ClearScreen ' Clear the screen

SetContrast

Syntax

Call SetContrast (Value)

Arguments

Item	Type	Direction	Description
Value	Byte	Input	New contrast value

Description

Changes the display contrast of the LCD characters. Range is from 0 (Max) to 255 (Min).

Example

Const Data = 0

Call SetContrast(Data) ' Set LCD contrast to maximum

SetBrightness

Syntax

Call SetBrightness (Value)

Arguments

Item	Type	Direction	Description
Value	Byte	Input	New brightness value

Description

Changes the displays backlight brightness level. Range is from 0 (off) to 255 (Max).

Example

Const Data = 255

Call SetBrightness (Data Byte) ' Set LCD backlight brightness level to max

LCDGetCell

Syntax

F = LCDGetCell(Row , Column)

Arguments

Item	Type	Direction	Description
Row	Byte	Input	Row of Cell
Column	Byte	Input	Column Of Cell
F	Byte	Output	Cell Byte

Description

Returns the byte value of a character at a specified cell.

Example

Dim Data As Byte

Data = LCDGetCell (1, 1) ' The value of character at Row 1 Column 1 is returned in Data

LCDPutCursorStyle

Syntax

Call LCDPutCursorStyle(CursorStyle)

Arguments

Item	Type	Direction	Description
CursorStyle	Byte Constant	Input	Changes LCD cursor style

Description

Changes the LCD cursor style to one of three built-in types. HiddenCursor, UnderscoreCursor, and BlinkingBlockCursor. These names are constants that reside in the LCDX library so there is no need to create your own

Example

Call LCDPutCursorStyle(HiddenCursor) ' Set LCD cursor to hidden

LCDMoveCursor

Syntax

Call LCDMoveCursor (Row, Column)

Arguments

Item	Type	Direction	Description
Row	Byte	Input	Row
Column	Byte	Input	Column

Description

Moves the cursor to the specified position

Example

Call LCDMoveCursor (2, 1) ' Move cursor to Row 2 Column 1

SetCustomChar

Syntax

Call SetCustomChar(Value, BitMap())

Arguments

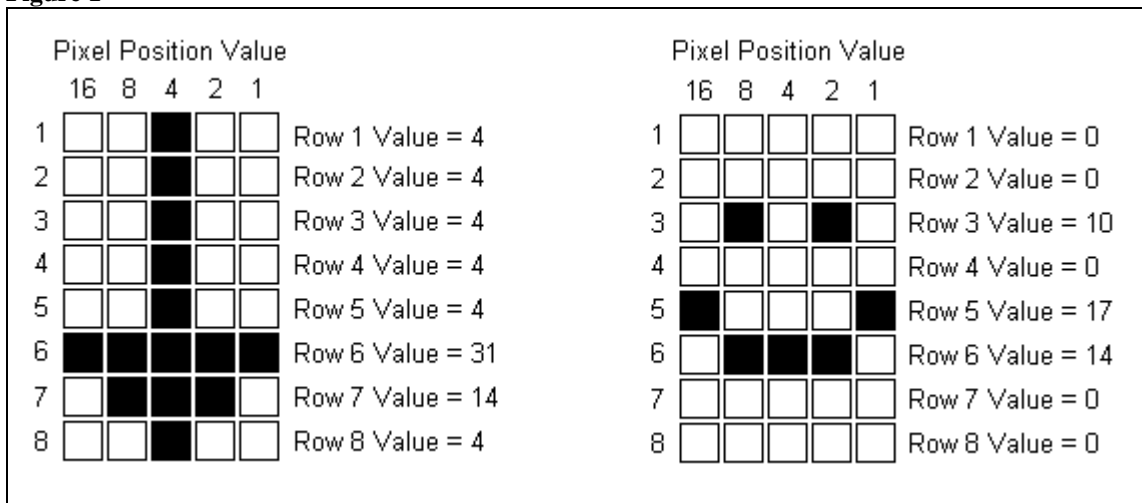
Item	Type	Direction	Description
Value	Byte	Input	Number for new character
Bitmap	Byte array	Input	Data array for new character

Description

The LCDX contains 8 user definable characters. Each of these six characters (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. **Figure 1** shows how each of those eight bytes correspond to a character.

The allowable character numbers are in range 0 to 7. Each character is eight pixels high and five pixels wide. The bitmap is an eight-byte array. Element one of the array corresponds to the top row of pixels, element two is the next row down, and so forth. Pixel values are taken from the lower five bits of each byte. The upper three bits are ignored.

Figure 1



Example

```
Dim Happy(1 to 8) As Byte
Dim D_Arrow(1 to 8) As Byte
```

```
Sub Main()
```

```
' Make a happy face
```

```
Happy(1) = 0
Happy(2) = 0
Happy(3) = 10
Happy(4) = 0
Happy(5) = 17
Happy(6) = 14
Happy(7) = 0
Happy(8) = 0
```

```
' Make a down arrow
```

```
D_Arrow(1) = 4
D_Arrow(2) = 4
D_Arrow(3) = 4
D_Arrow(4) = 4
D_Arrow(5) = 4
D_Arrow(6) = 21
D_Arrow(7) = 14
D_Arrow(8) = 4
```

```
' Get the LCD ready
```

```
Call LCDInitialize
```

```
' Set Backlight to max
```

```
Call SetBrightness(255)
```

```
' Set Contrast to 200
```

```
Call SetContrast(220)
```

```
' Store the new characters in the LCD
```

```
Call SetCustomChar(0, Happy)
```

```
Call SetCustomChar(7, D_Arrow)
```

```
' Use DisplayLine command to print "Down Arrow" on line 1
```

```
Call DisplayLine("Down Arrow ", 1)
```

```
' Use DisplayAtCell to add the down arrow we made to the text
```

```
Call DisplayAtCell(Chr(7), 1, 11)
```

```
' Use DisplayLine to print "Happy Face" and the "Happy" character on line 3
```

```
Call DisplayLine("Happy Face " & Chr(0), 3)
```

```
End Sub
```

LCDSrollUp

Syntax

Call LCDScrollUp

Arguments

None

Description

This causes the current text on the LCD screen to scroll up

Example

Call LCDScrollUp ' Scroll up the current LCD display

Beep

Syntax

Call Beep(Frequency , Duration)

Arguments

Item	Type	Direction	Description
Frequency	Long	Input	Desired Frequency
Duration	Integer	Input	Duration of sound

Description

Plays a sound through the LCDX speaker at a user determined frequency and duration

Example

Call Beep(10000 , 1000) ' Make a sound

GetVoltage

Syntax

Value = GetVoltage(ADC_Channel)

Arguments

Item	Type	Direction	Description
Value	Single	Output	Returned voltage
ADC_Channel	Byte	Input	ADC Channel number 1-8

Description

Reads the ADC value present on ADC inputs 1-8. Returned voltage is non-dimensional, in range 0.0 to 1.0. Channel is range 1 to 8.

Example

```
Dim Value As Single
Const ADC_Channel As Byte = 1
Value = GetVoltage(ADC_Channel)
```

GetVoltageI

Syntax

Value = GetVoltageI(ADC_Channel)

Arguments

Item	Type	Direction	Description
Value	Single	Output	Returned voltage
ADC_Channel	Byte	Input	ADC Channel number 1-8

Description

Reads the ADC value present on ADC inputs 1-8. Returned voltage is an Integer ranging from 0 to 1023. Channel is range 1 to 8.

Example

```
Dim Value As Integer
Const ADC_Channel As Byte = 1
Value = GetVoltageI(ADC_Channel)
```

XPin

Syntax

Pin = XPin(LCDXPin)

Arguments

Item	Type	Direction	Description
LCDXPin	Byte	Output	The Silkscreen pin # printed on LCDX board
Pin	Byte	Input	The actual pin number used by the BasicX

Description

Xpin converts an LCDX silkscreen pin number to the proper pin number used by BasicX. Range is 1-10. 1-8 are the ADC channels , 9&10 are free IO pins marked EE on the back of the LCDX.

Example

Call PutPin(Xpin(8), 1) ' Set ADC channel 8 to logic high
Foo = GetPin(Xpin(1)) ' Read the State of ADC channel 8

SetRelay

Syntax

Call SetRelay(Channel, State)

Arguments

Item	Type	Direction	Description
Channel	Byte	Output	Relay driver output channel
State	Byte	Output	State to set the relay driver

Description

Controls the output state of the on-board relay driver chip. Channel number range is 1-8. State is 0 or 1.

Example

Call SetRelay(1, 1) ' Activate the load connected on relay driver channel 1

GetRelay

Syntax

State = GetRelay (Channel)

Arguments

Item	Type	Direction	Description
Channel	Byte	Output	Relay driver output channel
State	Byte	Input	Current output state of channel

Description

Reads the output state of the on-board relay driver chip. Channel number range is 1-8. Returned state is logic 0 or 1.

Example

Call SetRelay(1, 1) ‘ Activate the load connected on relay driver channel 1

SetAllRelays

Syntax

Call SetAllRelays (State)

Arguments

Item	Type	Direction	Description
State	Byte	Output	The state to set all relay drivers

Description

Simultaneously sets the output state of all on-board relay driver outputs. State is 0-255

Example

Call SetAllRelays (255) ‘ Turn on all relay driver outputs

GetAllRelays

Syntax

State = GetAllRelays

Arguments

Item	Type	Direction	Description
State	Byte	Output	The current state of all relay drivers

Description

Simultaneously reads the output state of all on-board relay driver outputs. Returned state is 0-255

Example

State = GetAllRelays ‘ Read all relay driver outputs

GetKeypad

Syntax

Key = GetKeypad

Arguments

Item	Type	Direction	Description
Key	Byte	Input	Keypad data

Description

Reads the keypad provided with the LCDX development kit and returns a byte value equivalent of the marked Keypad layout. Keypad keys 0-9 return 0-9, Up arrow returns 10, down arrow returns 11, 2nd key returns 12, clear returns 13, Help 14 and Enter 15. No key press or read error returns 255. See **Figure 2** for connections.

Example

Key = GetKeypad ‘ Read the keypad

GetKeypadRaw

Syntax

Key = GetKeypadRaw

Arguments

Item	Type	Direction	Description
Key	Byte	Input	The returned keypad data

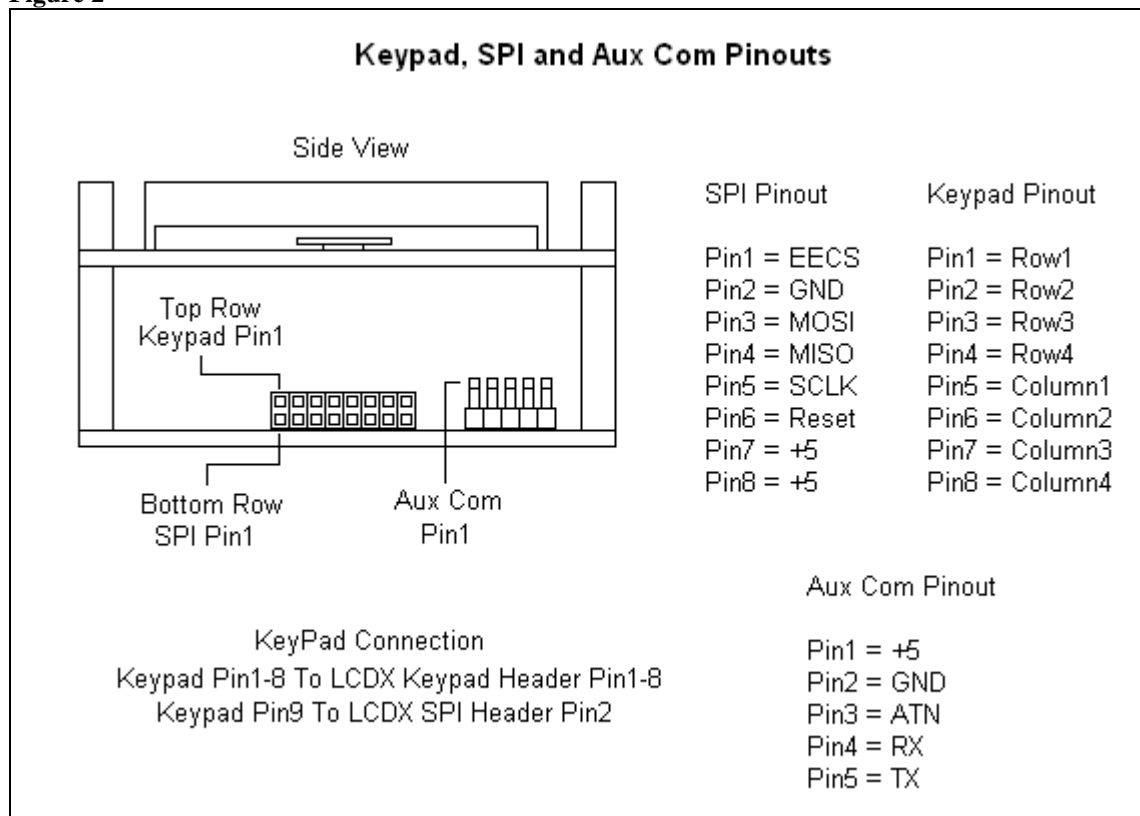
Description

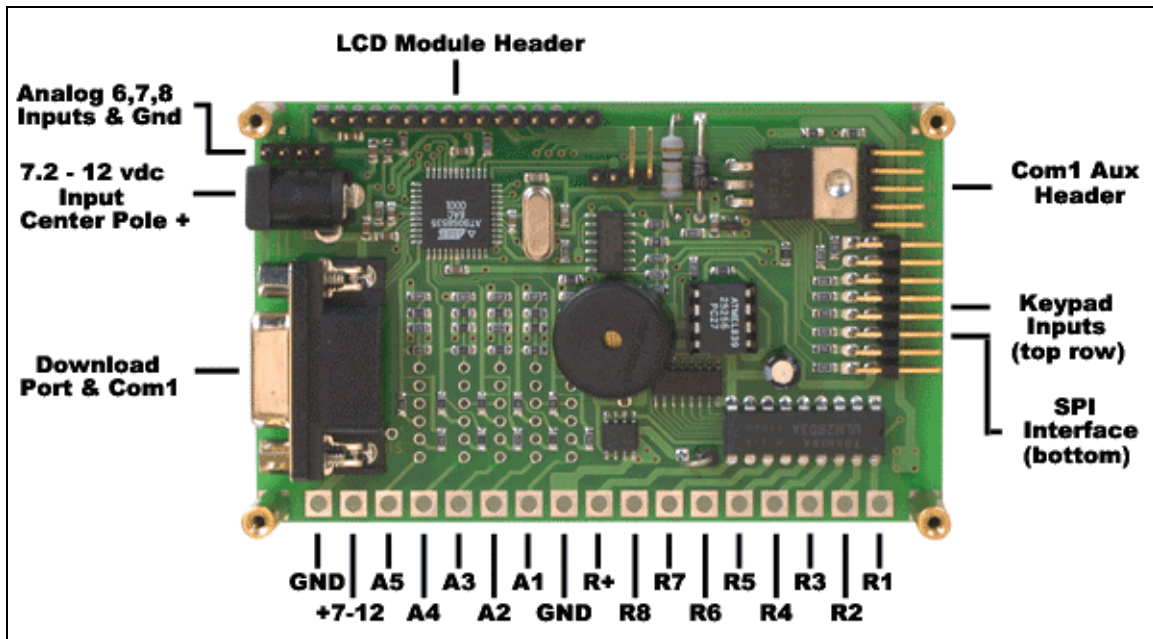
Reads a 4 x 4 matrix keypad connected to the keypad input pins and return its Raw data. Returned keypad data is in byte format. 255 = No key pressed. See **Figure 2** for connections.

Example

Key = GetKeypadRaw ' Read the keypad

Figure 2





Pin Assignments

Software to Hardware Pin Map			
Device	Function	Software Pin Name	Comments
LCD Data Line	LCD Data/Command Input	40	
LCD Read Line	LCD Read/Write Input	42	
LCD Enable	LCD Enable Input	43	
LCD Bus	LCD Data Buss	19-22	Shared With KeyPad Input
Relay Output	Relay Output Driver Chip	15	Uses SPI Buss 15 is Chip Enable
ADC Inputs 1-8	Analog Inputs	30-37	Reversed order. ie ADC1=Pin37
EE1	Aux I/O pin	16	Also Int1
EE2	Aux I/O pin	17	Also Int0
Keypad	KeyPad Interface	23-26	LCD Bus used as other 4 inputs

Misc. Specifications

Power Requirements	7.2 to 12V dc @ 34mA (200mA max backlight on)
Serial I/O	RS232 or Inverted TTL (1200 to 115,200 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

Warranty Information

NetMedia warrants this product against defects in materials and manufacturing defects for a period of 90 days from date of original purchase. 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. To return a product for warranty consideration, send the product with a copy of the original invoice, your contact information including phone number and a brief description of the problem to:

NetMedia, Inc. Attn: Warranty Repair
10940 N. Stallard Place
Tucson, AZ 85737.

The warranted item or its replacement will be returned via standard shipping. For expedited or foreign shipping please include your credit card number with your return shipping instructions.