Driving LCD from MMBasic

lcd.pdf

HD4478

This module is part of the original MMBasic library. It is reproduced here with kind permission of Hugh Buckle and Geoff Graham. Be aware it may reference functionality which has changed or is deprecated in the latest versions of MMBasic.

Demonstrates how to talk to a “standard” LCD display based on the Hitachi HD44780 controller and compatibles. Later versions of MMBasic have inbuilt commands which are simple and faster. If you have anything that is non-standard (8 bit bus, shared bus etc.) then these pieces of code will give you a good insight into how to drive the LCD discreetly. The attached file shows the assumed connectivity.

LCD.BAS

```
' demonstration program
' send 2 lines to the LCD
InitLCD
PrintLCD 1, "  Hello  World"
PrintLCD 2, "  Maximite LCD"



''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' LCD driver for MMBasic 3.1 or later (uses defined subroutines)
' Geoff Graham, Jan 2012
'
' Will drive a standard 16 x 2 LCDs
' For example:  futurlec.com       LCD16X2
'               altronics.com.au   Z7001
'               jaycar.com.au      QP5512
'
' To use:
'  - Setup the LCD with the command:    InitLCD
'  - Display a line using the command:  PrintLCD LineNbr, Text$
'
' See documentation (LCD.PDF) for the schematic
' Maximite pin 11 is RS, pin 12 is EN
' pins 13 to 16 are D4 to D7
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Initialise the LCD
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Sub InitLCD
  For i = 11 To 16 : SetPin i, 9 : Next i  ' all open collector
  LCD_Nibble &B0011, 0, 5                  ' reset
  LCD_Nibble &B0011, 0, 5                  ' reset
  LCD_Nibble &B0011, 0, 5                  ' reset
```

```
  LCD_Nibble &B0010, 0, 2                  ' 4 bit mode
  LCD_Nibble &B0010 : LCD_Nibble &B1100    ' 4 bits, 2 lines
  LCD_Nibble &B0000 : LCD_Nibble &B1100    ' display on, no cursor
  LCD_Nibble &B0000 : LCD_Nibble &B0110    ' increment on write
  LCD_Nibble &B0000 : LCD_Nibble &B0001    ' clear the display
  Pause 2
End Sub


''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Display a line on the LCD
' argument #1 is the line to be used (1 or 2)
' argument #2 is the line to display (can be any length)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Sub PrintLCD ( LineNumber, Line$ )
  Local i, c

  ' first send the cursor position (in two nibbles)
  LCD_Nibble &B1000 + (LineNumber - 1) * 4 : LCD_Nibble 0

  ' then send the text character by character (two nibbles per character)
  For i = 1 To 16
    c = Asc(Mid$(Line$ + Space$(16), i, 1))
    LCD_Nibble Int(c/16), 1 : LCD_Nibble c, 1
  Next i
End Sub


''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Send the lower 4 bits (called a nibble) to the LCD
' argument #1 is the nibble to send
' argument #2 is true if data, false if command (default is command)
' argument #3 is delay after the data has been sent (default is zero)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Sub LCD_Nibble ( Data, Flag, Wait_mSec )
  Pin(11) = Flag
  Pin(13) = Data And &B00000001
  Pin(14) = Data And &B00000010
  Pin(15) = Data And &B00000100
  Pin(16) = Data And &B00001000
  Pin(12) = 1 : Pin(12) = 0
  Pause Wait_mSec
End Sub
```

LCD1_1.BAS

```
' demonstration program
' send lines to the LCD
```

```
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' LCD driver for MMBasic 3.1 or later (uses defined subroutines)
' Geoff Graham, Jan 2012.
' Modified by James Deakins to work with 20 x 4 LCDs, Feb 2012.
' Should also work with 8 x 1 and other configs, but not tested.
'
' Will drive standard 16 x 2 and 20 x 4 LCDs
' For example:  futurlec.com       LCD16X2
'               altronics.com.au   Z7001
'               jaycar.com.au      QP5512
'
' To use:
'  - Define LCD size with the command:  InitLCDVariables columns, lines
'  - Setup the LCD with the command:    InitLCD
'  - Display a line using the command:  PrintLCD LineNbr, Text$
'
' File LCD_LIB.BAS will provide an extended range of functions.
'
' See documentation (LCD.PDF) for the schematic
' Maximite pin 11 is RS, pin 12 is EN
' pins 13 to 16 are D4 to D7
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

DIM C_LineStart(4)  ' an array to hold the start addresses for each line
C_Lines = 0         ' variable to hold the number of lines on the LCD
C_LineLength = 0    ' variable to hold the line length of the LCD
'
' **** Change the parameters to indicate whether you have a 16 x 2
' **** or a 20 x 4 LCD screen.

InitLCDVariables(20, 4) ' <<<<<<<<<<-------

InitLCD

' display simple lines, based on how many lines the LCD screen has

PrintLCD 1, "  Hello  World"
IF C_Lines > 1 THEN PrintLCD 2, "  Maximite LCD"
IF C_Lines > 2 THEN PrintLCD 3, "with a 20 x 4 LCD,"
IF C_Lines > 3 THEN PrintLCD 4, "using Subroutines."
ENDIF

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Initialise the variables for this application
' (C_Prefix indicates constants).
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
SUB InitLCDVariables(LCD_LineLength, LCD_Lines)

    C_LineLength = LCD_LineLength
    C_Lines = LCD_Lines
```

```
    C_LineStart(1) = 0        ' Start address for line 1
    C_LineStart(2) = 64       ' Start address for line 2
    C_LineStart(3) = 20       ' Start address for line 3
    C_LineStart(4) = 84       ' Start address for line 4

END SUB


'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Initialise the LCD hardware
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
SUB InitLCD
    FOR i = 11 TO 16 : SETPIN i, 9 : NEXT i     ' all open collector
    LCD_Nibble &B0011, 0, 5                     ' reset
    LCD_Nibble &B0011, 0, 5                     ' reset
    LCD_Nibble &B0011, 0, 5                     ' reset
    LCD_Nibble &B0010, 0, 2                     ' 4 bit mode
    LCD_Nibble &B0010 : LCD_Nibble &B1100       ' 4 bits, 2 lines
    LCD_Nibble &B0000 : LCD_Nibble &B1100       ' display on, no cursor
    LCD_Nibble &B0000 : LCD_Nibble &B0110       ' increment on write
    LCD_Nibble &B0000 : LCD_Nibble &B0001, 0, 2 ' clear the display, pause 2
END SUB

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Move cursor to a location on the LCD screen.
' argument #1 is the line to be used (1 to 4)
' argument #2 is the column to be used (1 to 20)
' This Subroutine isn't used by the other code, but
' is included because it's simple and useful.
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
SUB LCDLocate(Line, Column)
    LOCAL curPosn

    curPosn = 128 + C_LineStart(Line) + Column - 1
    LCD_Nibble INT(curPosn/16) : LCD_Nibble curPosn

END SUB


'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Display a line on the LCD
' argument #1 is the line to be used (1 to 4)
' argument #2 is the line to display (can be any length)
' Note: The text has to be padded out with spaces, otherwise
' 'odd' characters are displayed on the LCD.
' The code does the padding for you.
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
SUB PrintLCD ( LineNumber, Line$ )
    LOCAL i, c, fillSpaces, fullLine$, curPosn
```

```
    curPosn = 128 + C_LineStart(LineNumber)

    ' first, send the cursor position (in two nibbles)
    LCD_Nibble INT(curPosn/16) : LCD_Nibble curPosn

    fillSpaces = C_LineLength - LEN(Line$) ' is the line full?
    IF fillSpaces > 0 THEN
        fullLine$ = Line$ + SPACE$(fillSpaces)
    ELSE
        fullLine$ = Line$
    ENDIF

    ' then send the text character by character (two nibbles per character)
    FOR i = 1 TO C_LineLength
        c = ASC(MID$(fullLine$, i, 1))
        LCD_Nibble INT(c/16), 1 : LCD_Nibble c, 1
    NEXT i
END SUB


''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Send the lower 4 bits (called a nibble) to the LCD
' argument #1 is the nibble to send
' argument #2 is true if data, false if command (default is command)
' argument #3 is delay after the data has been sent (default is zero)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
SUB LCD_Nibble ( Data, Flag, Wait_mSec )
  PIN(11) = Flag
  PIN(13) = Data AND &B00000001
  PIN(14) = Data AND &B00000010
  PIN(15) = Data AND &B00000100
  PIN(16) = Data AND &B00001000
  PIN(12) = 1 : PIN(12) = 0
  IF Wait_mSec > 0 THEN PAUSE Wait_mSec       ' modified 30/1/2012. Restore
to Wait_mSec when Pause 0 bug corrected.
END SUB
```