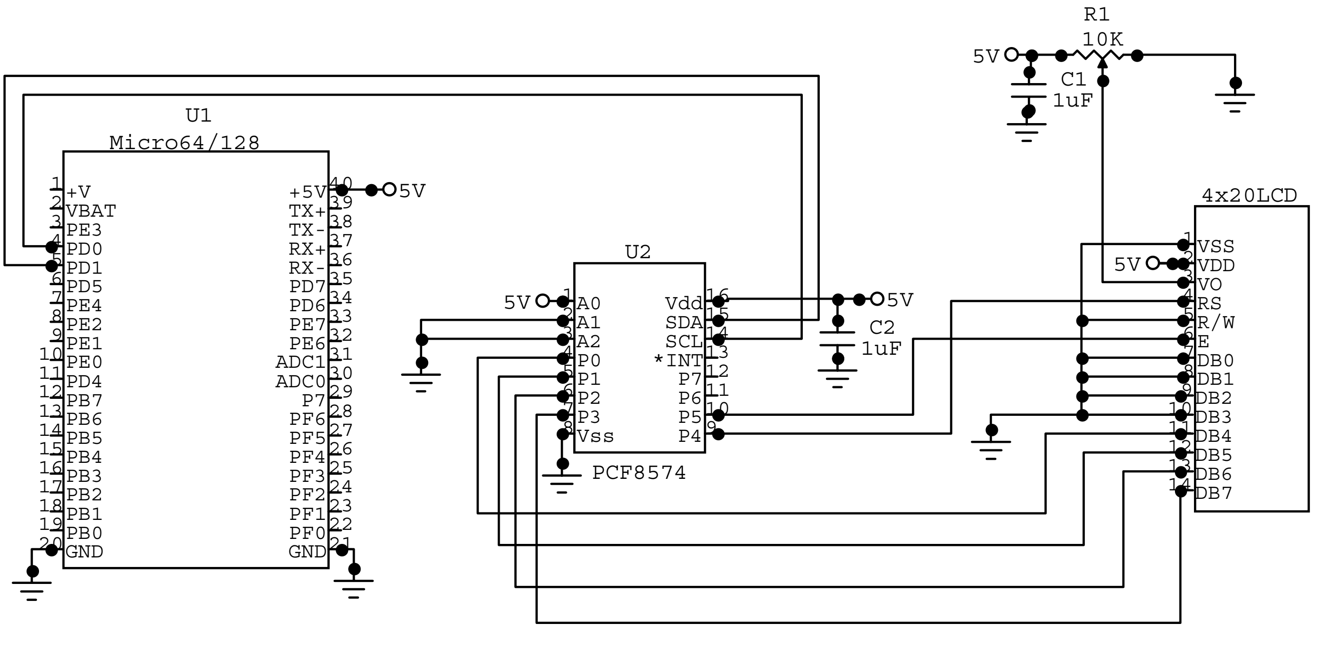
Micromint, Inc	AN755
	Micro64/128
I^2C LCD	12/10/04

Introduction: Micromint's Domino modules have built-in firmware to control an LCD through the I^2C bus. Since the Micro64/128 is a drop in replacement for the Domino 2 this application note demonstrates how to connect and write to an LCD over the I^2C bus.

Background: In order to connect an LCD to the I^2C bus a Philip's PCF8574 I^2C I/O expander set to the proper slave address needs to be connected between the LCD and the Micro64/128. The following schematic demonstrates how to connect the I/O expander.



How it works: The BASCOM-AVR program demonstrates how to write and access some of the Hitachi PN LM044L 4X20 LCD features using the I^2C Byte transfer function in Micro64/128's Utilities. Using a Philips/Signetics PCF8574 I^2C I/O expander with the slave address 01000010 we can communicate to a parallel LCD screen serially by sending it nibbles of information. The I^2C I/O expander only has enough I/O to control the LCD screen in nibble mode. This program allows you to enter a string and display it on the LCD, turn the display and the cursor on and off, make the cursor blink, and clear the entire screen.

Program Listing:

```
'***************************************************************************
'Project : I2C 4x20 LCD for the Micro64
```

```
'Company : Micromint, Inc.
'
'**********************************************************************

$regfile = "m64def.dat"
$baud1 = 9600


'Configure the serial port.
Config Com2 = Dummy , Synchrone = 0 , Parity = None , Stopbits = 1 , Databits = 8 , Clockpol = 0
'Configure PORTD.6 as an ouput and the rest of the port as inputs.
Ddrd = 64
Portd.6 = 1                   'Enable the transmitter
'Open the serial port
Open "com2:" For Random As #1
'Declare Variables & Constants
Const Lcdaddr = &H42
Dim B As Byte
Dim S As String * 80
Dim X As Byte
Dim Strlen As Byte
Dim Char As Byte
Dim Upnib As Byte
Dim Lownib As Byte
'The data sent over the I2C bus must be loacted at $HFFD in order for the
'utilities to work.
Dim I2cdatatx As Byte At &HFFD
'The Slave Address must be loacted at $HFFB in order for the utilities to work.
Dim Sladdr As Byte At &HFFB
'The Utilities puts the results from a read over the I2C bus at address &HFFE
Dim I2cdatarx As Word At &HFFE

'A Call to &HFC28 enables the I2C bus to 100kHz.
$asm
  !Call &HFC28;
$end Asm
Waitms 500                    'wait for 500 mS for LCD to come up
Gosub Init_lcd                'Initialize the LCD
Do
   Print #1 , "LCD Menu"
   Print #1 , "1 - Print to the LCD Screen"
   Print #1 , "2 - Clear the Screen"
   Print #1 , "3 - LCD On"
   Print #1 , "4 - LCD Off "
   Print #1 , "5 - Turn the cursor off"
   Print #1 , "6 - Turn the cursor on"
   Print #1 , "7 - Make the cursor not blink"
   Print #1 , "8 - Make the cursor blink"
   Input #1 , B
   Select Case B
    Case 1 : Gosub Displaystring
    Case 2 : Gosub Clear_lcd
    Case 3 : Gosub Lcdon
    Case 4 : Gosub Lcdoff
    Case 5 : Gosub Curseoff
    Case 6 : Gosub Curseon
    Case 7 : Gosub Curseblinkoff
    Case 8 : Gosub Curseblinkon
  End Select

Loop



'****************************************************************************
Init_lcd:
   Sladdr = Lcdaddr

   ' E   RS  DB7  DB6  DB5  DB4
   ' P5  P4   P3   P2   P1   P0
   Gosub Set4bitmode

   I2cdatatx = &B00100010      '4 bit mode upper nibble
   $asm
     !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
  I2cdatatx = &B00101100     '4 bit, 2line, 4X7 Matrix , lower nibble
  $asm
    !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin

  I2cdatatx = &B00100000     'Auto Increment, Shift Right upper nibble
  $asm
    !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
  I2cdatatx = &B00100110     'Auto Increment, Shift Right lower nibble
  $asm
    !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin

  Gosub Lcdon
  Gosub Clear_lcd

Return
'****************************************************************************
Clear_lcd:
  Sladdr = Lcdaddr
  I2cdatatx = &B00100000     'Clear the Display upper nibble
  $asm
    !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
  I2cdatatx = &B00100001     'Clear the Display lower nibble
  $asm
    !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
Return
'****************************************************************************
 Lcdon:
  Sladdr = Lcdaddr
  I2cdatatx = &B00100000     'LCD on upper nibble
  $asm
    !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
  I2cdatatx = &B00101100     'LCD on lower nibble
  $asm
    !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
Return
'****************************************************************************
Lcdoff:
  Sladdr = Lcdaddr
  I2cdatatx = &B00100000     'LCD off upper nibble
  $asm
    !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
  I2cdatatx = &B00101010     'LCD off lower nibble
  $asm
    !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
Return
'****************************************************************************
Curseoff:
  Sladdr = Lcdaddr
  I2cdatatx = &B00100000     'Cursor Off upper nibble
  $asm
    !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
  I2cdatatx = &B00101100     'Cursor Off lower nibble
  $asm
```

```
   !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
Return
'****************************************************************************
Curseon:
  Sladdr = Lcdaddr
  I2cdatatx = &B00100000     'Cursor On upper nibble
  $asm
   !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
  I2cdatatx = &B00101110     'Cursor On lower nibble
  $asm
   !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
Return
'****************************************************************************
Curseblinkoff:
  Sladdr = Lcdaddr
  I2cdatatx = &B00100000     'Turn off the blinking upper nibble
  $asm
   !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
  I2cdatatx = &B00101100     'Turn off the blinking lower nibble
  $asm
   !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
Return
'****************************************************************************
Curseblinkon:
  Sladdr = Lcdaddr
  I2cdatatx = &B00100000     'Turn on the blinking upper nibble
  $asm
   !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
  I2cdatatx = &B00101101     'Turn on the blinking lower nibble
  $asm
   !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
Return
'****************************************************************************
Lowlcdepin:
  Sladdr = Lcdaddr
  I2cdatatx = &B00000000     'Low the E pin to move in the nibble
  $asm
   !Call &HFCBD
  $end Asm
  Waitms 10
Return
'****************************************************************************
Set4bitmode:

  Sladdr = Lcdaddr
  I2cdatatx = &B00100010     '4 bit mode only
  $asm
   !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
Return
'****************************************************************************
Cursorline1:
  Sladdr = Lcdaddr
  I2cdatatx = &B00101000     'Move Cursor to beginning of line 1 upper nibble
  $asm
   !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
  I2cdatatx = &B00100000     'Move Cursor to beginning of line 1 lower nibble
  $asm
```

```
    !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
Return
'****************************************************************************
Cursorline2:
  Sladdr = Lcdaddr
  I2cdatatx = &B00101100     'Move Cursor to beginning of line 2 upper nibble
  $asm
    !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
  I2cdatatx = &B00100000     'Move Cursor to beginning of line 2 lower nibble
  $asm
    !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
Return
'****************************************************************************
Cursorline3:
  Sladdr = Lcdaddr
  I2cdatatx = &B00101001     'Move Cursor to beginning of line 3 upper nibble
  $asm
    !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
  I2cdatatx = &B00100100     'Move Cursor to beginning of line 3 lower nibble
  $asm
    !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
Return
'****************************************************************************
Cursorline4:
  Sladdr = Lcdaddr
  I2cdatatx = &B00101101     'Move Cursor to beginning of line 4 upper nibble
  $asm
    !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
  I2cdatatx = &B00100100     'Move Cursor to beginning of line 4 lower nibble
  $asm
    !Call &HFCBD
  $end Asm
  Gosub Lowlcdepin
Return
'****************************************************************************
Displaystring:
   Gosub Cursorline1
   Sladdr = Lcdaddr
   X = 0
   Input #1 , "Please enter a string" , S
   Strlen = Len(s)          'get the lenght of the string to determine how many times to loop
   Do                       'begin a the loop to send out the charecters to the display.
     Char = Right(s , Strlen)
     'Prepare the upper nibble
     I2cdatatx = 0
     I2cdatatx.0 = Char.4
     I2cdatatx.1 = Char.5
     I2cdatatx.2 = Char.6
     I2cdatatx.3 = Char.7
     I2cdatatx.4 = 1
     I2cdatatx.5 = 1
     $asm
       !Call &HFCBD
     $end Asm
     Gosub Lowlcdepin
     'Prepare the lower nibble
     I2cdatatx = 0
     I2cdatatx.0 = Char.0
     I2cdatatx.1 = Char.1
     I2cdatatx.2 = Char.2
     I2cdatatx.3 = Char.3
     I2cdatatx.4 = 1
     I2cdatatx.5 = 1
```

```
      $asm
        !Call &HFCBD
      $end Asm
      Gosub Lowlcdepin
      If X = 19 Then Gosub Cursorline2
      If X = 39 Then Gosub Cursorline3
      If X = 59 Then Gosub Cursorline4
      Decr Strlen             'decrease by one
      X = X + 1
    Loop Until Strlen = 0


Return

'****************************************************************************

Close #1
```