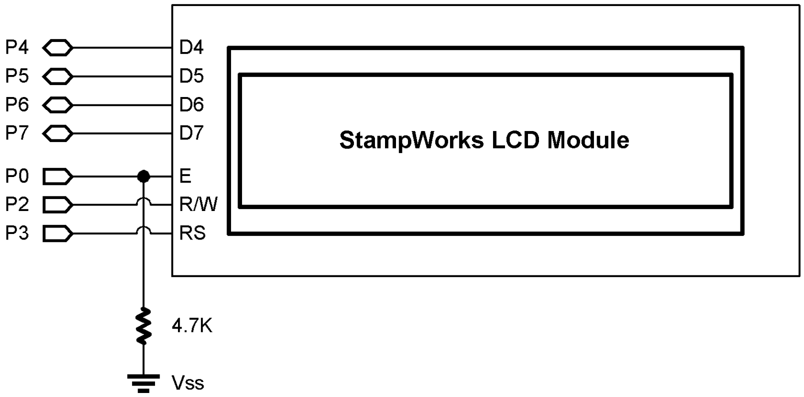
Experiment #13: Reading the LCD RAM

This program demonstrates the use of the LCD's CGRAM space as external memory.

New PBASIC elements/commands to know:

- `InA, InB, InC, InD`

Building The Circuit



```
' ================================================================================
'
'   File...... Ex13 - LCD Read.BS2
'   Purpose... Read data from LCD
'   Author.... Parallax
'   E-mail.... stamptech@parallaxinc.com
'   Started...
'   Updated... 01 MAY 2002
'
'   {$STAMP BS2}
'
' ================================================================================
' -----------------------------------------------------------------------------
' Program Description
' -----------------------------------------------------------------------------

' This program demonstrates how to read data from the LCD's display or CGRAM
' areas.
'
' The connections for this program conform to the BS2p LCDIN and LCDOUT
' commands.  Use this program for the BS2, BS2e or BS2sx.  There is a separate
' program for the BS2p.


' -----------------------------------------------------------------------------
' I/O Definitions
' -----------------------------------------------------------------------------

E               CON     0                       ' LCD Enable pin  (1 = enabled)
RW              CON     2                       ' LCD Read/Write pin (1 = write)
RS              CON     3                       ' Register Select (1 = char)
LCDdirs         VAR     DirB
LCDbusOut       VAR     OutB                    ' 4-bit LCD data bus
LCDbusIn        VAR     InB


' -----------------------------------------------------------------------------
' Constants
' -----------------------------------------------------------------------------

ClrLCD          CON     $01                     ' clear the LCD
CrsrHm          CON     $02                     ' move cursor to home position
CrsrLf          CON     $10                     ' move cursor left
CrsrRt          CON     $14                     ' move cursor right
DispLf          CON     $18                     ' shift displayed chars left
DispRt          CON     $1C                     ' shift displayed chars right
DDRam           CON     $80                     ' Display Data RAM control
CGRam           CON     $40                     ' Custom character RAM


' -----------------------------------------------------------------------------
' Variables
' -----------------------------------------------------------------------------

char            VAR     Byte                    ' character sent to LCD
index           VAR     Byte                    ' loop counter
rVar            VAR     Word                    ' for random number
addr            VAR     Byte                    ' address to write/read
tOut            VAR     Byte                    ' test value to write to LCD
```

```
tIn             VAR     Byte                    ' test value to read from LCD
temp            VAR     Word                    ' temp value for numeric display
width           VAR     Nib                     ' width of number to display


' ------------------------------------------------------------------------------
' Initialization
' ------------------------------------------------------------------------------

Initialize:
  DirL = %11111101                              ' setup pins for LCD

LCD_Init:
  PAUSE 500                                     ' let the LCD settle
  LCDbusOut = %0011                             ' 8-bit mode
  PULSOUT E, 1
  PAUSE 5
  PULSOUT E, 1
  PULSOUT E, 1
  LCDbusOut = %0010                             ' 4-bit mode
  PULSOUT E, 1
  char = %00001100                              ' disp on, crsr off, blink off
  GOSUB LCD_Command
  char = %00000110                              ' inc crsr, no disp shift
  GOSUB LCD_Command


' ------------------------------------------------------------------------------
' Program Code
' ------------------------------------------------------------------------------

Main:
  char = ClrLCD                                 ' clear the LCD
  GOSUB LCD_Command

  FOR index = 0 TO 14                           ' create display
    LOOKUP index, ["ADDR=??  ???/???"], char
    GOSUB LCD_Write
  NEXT

Loop:
  RANDOM rVar                                   ' generate random number
  addr = rVar.LowByte & $3F                     ' create address (0 to 63)
  tOut = rVar.HighByte                          ' create test value (0 to 255)

  char = CGRam + addr                           ' set CGRAM pointer
  GOSUB LCD_Command
  char = tOut
  GOSUB LCD_Write                               ' move the value to CGRAM
  PAUSE 100                                     ' wait a bit, then go get it

  char = CGRam + addr                           ' set CGRAM pointer
  GOSUB LCD_Command
  GOSUB LCD_Read                                ' read value from LCD
  tIn = char

  ' display results

  char = DDRam + 5                              ' show address at position 5
  GOSUB LCD_Command
  temp = addr
  width = 2
  GOSUB Put_Val

  char = DDRam + 9                              ' show output at position 8
  GOSUB LCD_Command
  temp = tOut
  width = 3
  GOSUB Put_Val

  char = DDRam + 13                             ' show input at position 12
  GOSUB LCD_Command
  temp = tIn
  width = 3
  GOSUB Put_Val
  PAUSE 1000
  GOTO Loop                                     ' do it again

  END


' ------------------------------------------------------------------------------
' Subroutines
' ------------------------------------------------------------------------------

Put_Val:
  FOR index = (width - 1) TO 0                  ' display digits left to right
    char = (temp DIG index) + 48                ' convert digit to ASCII
    GOSUB LCD_Write                             ' put digit in display
  NEXT
  RETURN
```

```
LCD_Command:
  LOW RS                                        ' enter command mode

LCD_Write:
  LCDbusOut = char.HighNib                      ' output high nibble
  PULSOUT E, 1                                  ' strobe the Enable line
  LCDbusOut = char.LowNib                       ' output low nibble
  PULSOUT E, 1
  HIGH RS                                       ' return to character mode
  RETURN


LCD_Read:
  HIGH RS                                       ' data command
  HIGH RW                                       ' read
  LCDdirs = %0000                               ' make data lines inputs
  HIGH E
  char.HighNib = LCDbusIn                       ' get high nibble
  LOW E
  HIGH E
  char.LowNib = LCDbusIn                        ' get low nibble
  LOW E
  LCDdirs = %1111                               ' return data lines to outputs
  LOW RW
  RETURN
```

Behind The Scenes

This program demonstrates the versatility of the BASIC Stamp's I/O lines and their ability to be reconfigured mid-program. Writing to the LCD was covered in the last two experiments. To read data back, the BASIC Stamp's I/O lines must be reconfigured as inputs. This is no problem for the BASIC Stamp. Aside from the I/O reconfiguration, reading from the LCD requires an additional control line: RW. In most programs this line can be tied low to allow writing to the LCD. For reading from the LCD the RW line is made high.

The program generates an address and data using the **RANDOM** function. The address is kept in the range of 0 to 63 by masking out the highest bits of the **LowByte** returned by the **RANDOM** function. The **HighByte** is used as the data to be written to and read back from the LCD.

The data is stored in the LCD's CGRAM area. This means -- in this program -- that the CGRAM memory cannot be used for custom characters. In programs that require less than eight custom characters the remaining bytes of CGRAM can be used as off-board memory.

Reading data from the LCD is identical to writing: the address is set and the data is retrieved. For this to take place, the LCD data lines must be reconfigured as inputs. Blipping the E (enable) line makes the data (one nibble at a time) available for the BASIC Stamp. Once again, **HighNib** and **LowNib** are used, this time to build a single byte from the two nibbles returned during the read operation.

When the retrieved data is ready, the address, output data and input data are written to the LCD for examination. As short subroutine, **Put_Val**, handles writing numerical values to the LCD. To use this routine, move the cursor to the desired location, put the value to be displayed in **temp**, the number of characters to display in **width**, then call **Put_Val**. The subroutine uses the **DIG** operator to extract a digit from **temp** and adds 48 to convert it to ASCII so that it can be displayed on the LCD.