

Programming PIC Microcontrollers in PicBasic Pro – LCD Lesson 2

Cornerstone Electronics Technology and Robotics II

- **Administration:**
 - Prayer
- **PicBasic Pro Programs Used in This Lesson:**
 - General PicBasic Pro Program Listing:
<http://www.cornerstonerobotics.org/picbasic.php>
 - Lab 1 lcd1 as .pdf file:
<http://www.cornerstonerobotics.org/code/LCD1.pdf>
 - Lab 1 lcd2 as .pdf file:
<http://www.cornerstonerobotics.org/code/LCD2.pdf>
- **LCD Basics Continued:**
 - 4-bit Data Transfer:
 - In the last lesson, we focused on the 8-bit data transfer through the use of all eight data lines to the LCD.
 - To conserve pins in the PIC chip, we will transfer the data using only four data lines (D4 – D7).
 - Instead of sending one byte (8-bits), we will send two nibbles (4-bits).
 - Note: Underlined commands refer to the LCD Command Control Codes table in the previous lesson.
 - Set RS to LOW
 - Set Display On/Off & Cursor commands as 00001111.
 - Set the Function Set command to 00101000.
 - 8/4 = 0, 4-bit Interface
 - 2/1 = 1, 2 Line Mode
 - 10/7 = 0, 5x7 Dot Matrix
 - This sets the Function Set commands to 4-bit data, two lines, and 5x7 dot matrix format.
 - The 8-bit ASCII standard for the letter A is 01000001. To send A in 4-bit data format, type in 0100, then Enable, then type in 0001, then Enable.
- **LCD PicBasic Programming:**
 - New PICBasic Command:
 - **LCDOUT:**
 - Format:

LCDOUT *Item*{,*Item*...}

Display *Items* on an intelligent Liquid Crystal Display. If a pound sign (#) precedes an *Item*, the ASCII representation for each digit is sent to the LCD.

 - Other:
 - A program should wait for at least half a second before sending the first command to an LCD. It can take quite a while for an LCD to start up.

- Commands are sent to the LCD by sending a \$FE followed by the command. Some useful commands are listed in the following table:

PicBasic Pro LCD Command Table	
Command	Operation
LCDOUT \$FE, 1	Clear display
LCDOUT \$FE, 2	Return home
LCDOUT \$FE, \$0C	Cursor off
LCDOUT \$FE, \$0E	Underline cursor on
LCDOUT \$FE, \$0F	Blinking cursor on
LCDOUT \$FE, \$10	Move cursor left one position
LCDOUT \$FE, \$14	Move cursor right one position
LCDOUT \$FE, \$18	Display shift left
LCDOUT \$FE, \$1C	Display shift right
LCDOUT \$FE, \$80	Move cursor to beginning of first line
LCDOUT \$FE, \$C0	Move cursor to beginning of second line
LCDOUT \$FE, \$94	Move cursor to beginning of third line
LCDOUT \$FE, \$D4	Move cursor to beginning of fourth line

- Note that there are commands to move the cursor to the beginning of the different lines of a multi-line display. For most LCDs, the displayed characters and lines are not consecutive in display memory - there can be a break in between locations. For most 16x2 displays, the first line starts at \$80 and the second line starts at \$C0.
- The command:

LCDOUT \$FE, \$80 + 4

sets the display to start writing characters at the forth position of the first line.

- Examples:

LCDOUT \$FE,1,"Hello" ' Clear display and show "Hello"

LCDOUT \$FE,\$C0,"World" ' Jump to second line and show "World"

LCDOUT B0,#B1 ' Display B0 and decimal ASCII value of B1

- **Compare HEX and Binary Command Codes to Control an LCD:**
 - Review the table below and note how the binary commands used in the Lesson LCD1 (see the LCD Command Control Codes Table) match the HEX command codes used to control the LCD.

LCD Command Control Codes

Command	Hex	Binary
Clear Display	1	%00000001
Return Home	2	%00000010
Cursor Off	\$0C	%00001100
Underline Cursor On	\$0E	%00001110
Blinking Cursor On	\$0F	%00001111
Move Cursor Beginning First Line	\$80	%10000000
Move Cursor Beginning Second Line	\$C0	%11000000

Table Comparing HEX and Binary Command Codes

- **LCD Command Control Codes from Lesson LCD1:**

Command	Binary								Hex
	D7	D6	D5	D4	D3	D2	D1	D0	
Clear Display	0	0	0	0	0	0	0	1	01
Display & Cursor Home	0	0	0	0	0	0	1	X	02 or 03
Character Entry Mode	0	0	0	0	0	1	I/D	S	04 to 07
Display On/Off & Cursor	0	0	0	0	1	D	U	B	08 to 0F
Display/Cursor Shift	0	0	0	1	D/C	R/L	x	x	10 to 1F
Function Set	0	0	1	8/4	2/1	10/7	x	x	20 to 3F
Set CGRAM Address	0	1	A	A	A	A	A	A	10 to 7F
Set Display Address	1	A	A	A	A	A	A	A	80 to FF

I/D: 1=Increment*, 0=Decrement	R/L: 1=Right Shift, 0=Left Shift
S: 1=Display Shift On, 0=Display Shift Off	8/4: 1=8-bit Interface*, 0=4-bit Interface
D: 1=Display On, 0=Display Off*	2/1: 1=2 Line Mode, 0=1 Line Mode*
U: 1=Cursor Underline On, 0=Underline Off*	10/7: 1=5x10 Dot Matrix, 0=5x7 Dot Matrix
B: 1=Cursor Blink On, 0=Cursor Blink Off*	
D/C: 1=Display Shift, 0=Cursor Move	X=Don't Care *= Initialization Settings

- **LCD Display Addresses:**

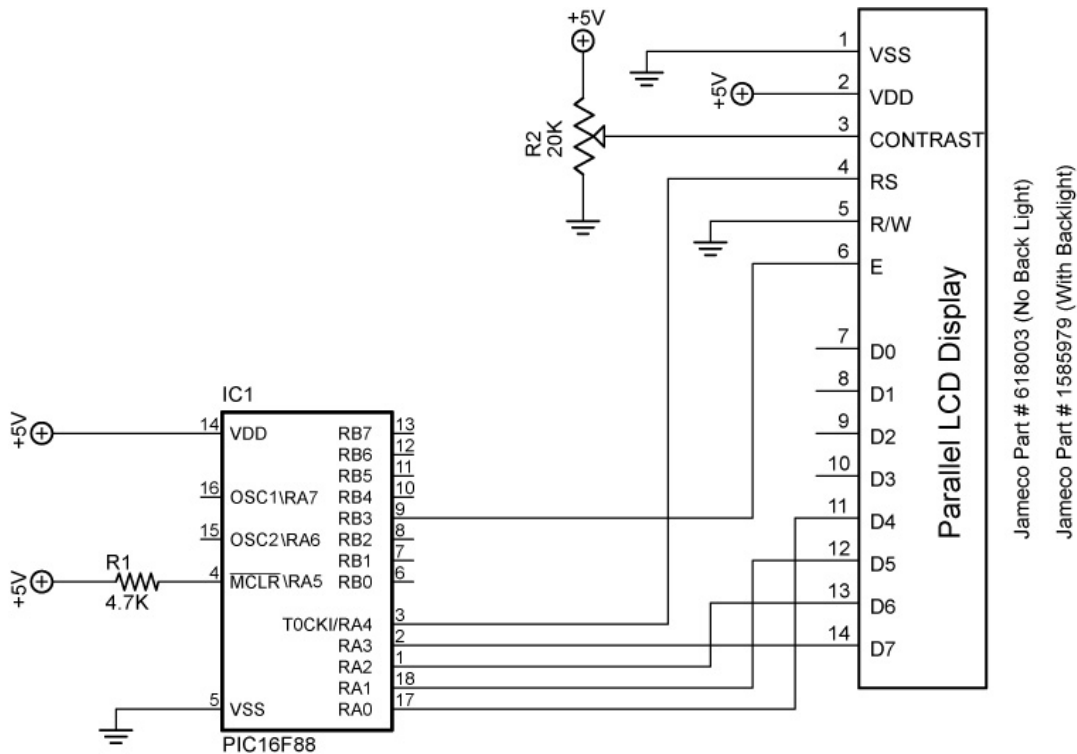
0000000	0000001	0000010	0000011	0000100	0000101	0000110	0000111	0001000	0001001	0001010	0001011	0001100	0001101	0001110	0001111
1000000	1000001	1000010	1000011	1000100	1000101	1000110	1000111	1001000	1001001	1001010	1001011	1001100	1001101	1001110	1001111

7- Bit Display Addresses for 16 Character 2 Line Display in Binary

- Perform LCD2 LAB 1 – lcd1.pbp and lcd2.pbp, LCD Commands

Cornerstone Electronics Technology and Robotics II LCD Lesson 2 LAB 1 – lcd1.pbp and lcd2.pbp, LCD Commands

- **Purpose:** The purpose of this lab is to acquaint the student with the PicBasic Pro command **LCDOUT** .
- **Apparatus and Materials:**
 - 1 – Analog/Digital Trainer or Breadboard
 - PIC16F88 Microcontroller
 - Hantronix HDM16216H-5-300S 16x2 LCD, Jameco #618003
 - 20 K Potentiometer
 - 4.7 K Resistor
- **Procedure:**
 - Wire the following circuit:



lcd1 and lcd2

- Notice that only D4 – D7 data inputs are used; we will use 4-bit data transfer to save pins.
- Open lcd1.pbp from your folder and program your chip. Add your own text. Save your program as **lcd10.pbp**.
- Open lcd2.pbp from your folder and program your chip. Run the program on the breadboard.

Command	Operation	Activity	Completed
\$FE, 1	Clear display	1	
\$FE, 2	Return home	2	
\$FE, \$0C	Cursor off	2	
\$FE, \$0E	Underline cursor on	2	
\$FE, \$0F	Blinking cursor on	2	
\$FE, \$10	Move cursor left one position	5	
\$FE, \$14	Move cursor right one position	5	
\$FE, \$18	Display shift left	4	
\$FE, \$1C	Display shift right	4	
\$FE, \$80	Move cursor to beginning of first line	3	
\$FE, \$C0	Move cursor to beginning of second line	3	
\$FE, \$94	Move cursor to beginning of third line	-	n/a
\$FE, \$D4	Move cursor to beginning of fourth line	-	n/a

- LCD Command Table Experiments: Use the table above to check off each command as it is completed.
 - Activity 1:
 - Open **lcd1.pbp**; save it as **lcd11.pbp**.
 - Revise the program so that “Hello World” remains on the LCD screen for 2 seconds, then make the LCD blank using the \$FE,1 command.
 - Activity 2:
 - Start with **lcd11.pbp** and save the new program as **lcd12.pbp**
 - Revise **lcd12.pbp** such that:
 - “Hello World” remains on the LCD for 2 seconds
 - Clear Display
 - Return the cursor home and blinking for 2 more seconds. Each LCD command may be put on a separate line.
 - Now underline the cursor for 2 seconds
 - Clear display
 - Activity 3:
 - Start again with **lcd11.pbp** and save the new program as **lcd13.pbp**
 - Remember:

LCDOUT \$FE, \$80 + 4

sets the display to start writing characters at the forth position of the first line.

Cornerstone Electronics Technology and Robotics II
LCD Lesson 2 LAB 1 – lcd1.pbp and lcd2.pbp, LCD Commands Continued

- Activity 3 Continued:
 - Revise **lcd13.pbp** such that:
 - “Hello” begins on the first line, 6 positions in from the right.
 - Pause 1 second
 - “World” begins on the second line, 6 positions in from the right
 - Pause 1 second
 - Clear display for 1 second
 - Repeat the whole sequence 3 times
- Activity 4:
 - Start with **lcd1.pbp** and save the new program as **lcd14.pbp**
 - Revise **lcd14.pbp** such that:
 - “Hello World” shifts to the right one position at a time for 4 positions
 - “Hello World” shifts to the left one position at a time for 4 positions
 - Repeat both shifts 3 times
- Activity 5:
 - Start with **lcd1.pbp** and save the new program as **lcd15.pbp**
 - Delete the main code in **lcd15.pbp**
 - Revise **lcd15.pbp** such that:
 - Display a variable name “Resistor1 = ”
 - Move over to the right one blank space and display the value of the variable r. At this point, you will have to input a value for r, such as, r = 330. To display the value of r in the **LCDOUT** command, place a # sign immediately before r.

LCDOUT \$FE,1,“ Resistor1 = ”, #r

- **Challenge:** Display your first name on the first line and your second name on the second line. The first letters of your first and last names are to appear in the far right positions on the LCD and remain for 2 seconds. Then your first and last names are to scroll across the screen right to left.