Microchip PIC Controlled HD44780 LCD and 3x4 Matrix Keypad

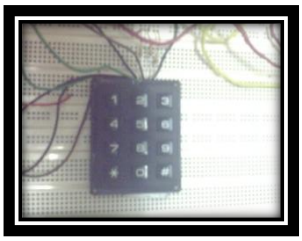
Author: King Seh Horng. This article is published on December 22, 2008.

About the Author

King Seh Horng is an electronic and computer hobbyist from Brunei Darussalam. He is currently pursuing for full time undergraduate studies in Computer Systems Engineering in Curtin University of Technology Sarawak Campus (http://www.curtin.edu.my). Previously, he has accomplished a full-time diploma in Instrumentation and Control Engineering in a national college, namely Maktab Kejuruteraan Jefri Bolkiah (http://www.mkjb.edu.bn). He has successfully completed many projects including web development, microcontroller application design and assembly programming. The author can be contacted through sehhorng@gmail.com or via his website at http://king.iplussoft.com.

Introduction

This article will provide you with the fundamental information on how you integrate the HD44780 controlled LCD display and matrix keypad with Microchip's popular 16F877/A. This project can be further improved for numerous applications such as security door lock system and others requiring a hardware interface to input and verify pass keys.



List of items which may be required for this project:

1. PIC16F877/A main IO board
2. PIC16F877/A IC chip
3. Microchip PIC programmer kit
4. IDCC wires, male and female connectors
5. 2 prototype boards
6. 3x4 matrix keypad (character inputs: 0-9, # & *)
7. 1 to 4 lines HD44780 controlled LCD display with/without backlight
8. Mobile charger 5VDC - 20mA adaptor
9. 3-24VDC mini buzzer
10. Insulated wires and jumper wires
11. Crocodile clips (color coded)
12. 4 1KΩ resistors
13. MPLAB™ software
14. Additional software driver

HD44780 controlled LCD Display

HD44780 is the controller which controls the main settings of the LCD display. It is important to understand the behavior of this controller before getting started with the display.

There will be 14 pins (pin 1 to 14 inclusive) for the LCD display which has no backlight feature. The pin information of this controller is shown in the table below:

No.	Name	Function
1	VSS	Common/Ground
2	VDD	Input voltage
3	VO	Control display contrast (Connected to ground or potentiometer connected to ground)
4	RS	Register Select
5	RW	1=Read, 0=Write
6	E	Enable
7	DB0	Data bit 0
8	DB1	Data bit 1
9	DB2	Data bit 2
10	DB3	Data bit 3
11	DB4	Data bit 4
12	DB5	Data bit 5
13	DB6	Data bit 6
14	DB7	Data bit 7
15	BL1	Backlight (V+)
16	BL2	Backlight (Ground)

RS is set to 0 when issuing a command through the data bits 0-7 to the controller while RS is set to 1 when treating the data bits as 1 byte character code. The following table shows the command that can be sent to the controller to clear screen, change cursor position, create custom CGRAM character & etc.

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	40 to 7F
Set Display Address	1	A	A	A	A	A	A	A	80 to FF

I/D: 1=Increment*, 0=Decrement
S: 1=Display shift on, 0=Display shift off*
D: 1=Display On, 0=Display Off*
U: 1=Cursor underline on, 0=Underline off*
B: 1=Cursor blink on, 0=Cursor blink off*
D/C: 1=Display shift, 0=Cursor move
R/L: 1=Right shift, 0=Left shift
8/4: 1=8 bit interface*, 0=4 bit interface
2/1: 1=2 line mode, 0=1 line mode*
10/7: 1=5x10 dot format, 0=5x7 dot format*
x = Don't care
* = Initialisation settings

Timing is very crucial. It is important to know that you should configure your microcontroller to execute an initialization delay for the LCD display to start up. A delay of not more than 200ms should be sufficient. After the delay, the controller can start receive commands and character bits from the microcontroller.

Microchip PIC Controlled HD44780 LCD and 3x4 Matrix Keypad

Author: King Seh Horng. This article is published on December 22, 2008.

Input Commands

A certain procedure is needed for the HD44780 to register the data bits 0-7. You should already know that setting RS to 0 to send a command while setting RS to 1 to send character code. Enable should be set to 1 during initialization and by default. You should note that enable needs to be triggered for sending each data byte.

Required initialization settings after initialization delay:

RW=Don't care, RS=0, E=1
Data bits (7-0): 00001101
E=0 → Short Delay → E=1 → Short Delay
Purpose: To make display and cursor visible on screen

RW=Don't care, RS=0, E=1
Data bits (7-0): 00000001
E=0 → Short Delay → E=1 → Short Delay
Purpose: Clear screen – To ensure the RAM of the LCD does not use previous display

RW=Don't care, RS=0, E=1
Data bits (7-0): 00111100
E=0 → Short Delay → E=1 → Short Delay
Purpose: To configure the LCD's character type, line number & bit interface

Some examples:

RW=Don't care, RS=0, E=1
Data bits (7-0): 1000000
E=0 → Short Delay → E=1 → Short Delay
Purpose: To move the cursor to the left most position of the first line

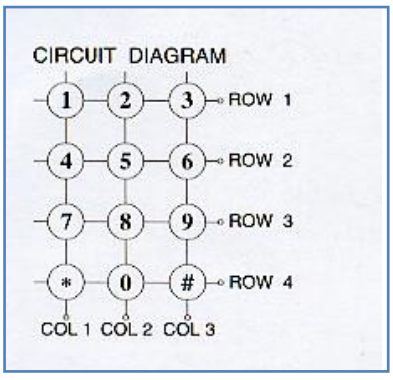
RW=Don't care, RS=0, E=1
Data bits (7-0): 1000001
E=0 → Short Delay → E=1 → Short Delay
Purpose: To move the cursor to the second position from the left of the first line

RW=Don't care, RS=0, E=1
Data bits (7-0): 1100000
E=0 → Short Delay → E=1 → Short Delay
Purpose: To move the cursor to the left most position of the second line

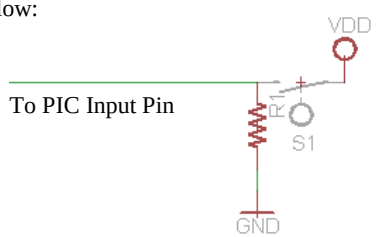
RW=0, RS=1, E=1
Data bits (7-0): 01000001
E=0 → Short Delay → E=1 → Short Delay
Purpose: Write a 'A' character in the cursor position

The following table shows the character code for the data bits 0-7 when RS is set to 1.

3x4 Matrix Keypad



Experimentally, three resistors connected to ground should be linked to the input pins of the microcontroller port. The concept is illustrated below:



Concept: Scanning for keys

Due to the nature of the keypad, it is necessary to save 7 pins of the PIC – 4 output pins and 3 input pins. To minimize the number of pins needed down to 4 pins, you may try finding an external keypad encoder circuit. However, in this experiment, the keypad is connected directly to the PIC microcontroller. COLx will be ON when the respective ROW is ON and a key on that row is pressed. COL0, COL1 and COL2 will remain OFF when no key on the row is pressed.

The following is the pseudo code for scanning only the first two rows, three columns (1,2,3 and 4,5,6):

```
pressed=false; //initial declaration

Function scan(void){
        Set ROW1=1, ROW2=0, ROW3=0, ROW4=0
        Call Short Delay

        If (COL1==1){Key '1'; pressed=1; return;}
        If (COL2==1){Key '2'; pressed=1; return;}
        If (COL3==1){Key '3'; pressed=1; return;}

        Set ROW1=0, ROW2=1, ROW3=0, ROW4=0
        Call Short Delay

        If (COL1==1){Key '4'; pressed=1;return;}
        If (COL2==1){Key '5'; pressed=1;return;}
        If (COL3==1){Key '6'; pressed=1;return;}

.
.
.
}

While(1){

call scan

}
```

A short delay may be required to capture the key pressed correctly. This delay allows the microcontroller to obtain a stable and consistent reading.

You may also need to think of how your microcontroller should handle multiple key inputs. One method is by verifying a push and pull event, in which case only the initial key pressed during that event would be accepted. The pseudo code is as follows:

```
pressed=false; //initial declaration

While(1){

If (COL1==0 && COL2==0 && COL3==0){pressed=false;} //if no key pressed

If(pressed==false){call scan;}

}
```

Related Links

- Official project site: http://king.iplussoft.com
- Assembly source file: http://king.iplussoft.com/asm_pic_lcdkeypad.zip
- Video demonstration: http://www.youtube.com/watch?v=dMFbWLK5kGE
- External JavaScript LCD simulator: http://www.dinceraydin.com/djlcdsim/djlcdsim.html
- Email author: sehhorng@gmail.com

References

ILett, J. (1997, February). *How to use intelligent L.C.Ds.* Retrieved December 21, 2008, from Everyday Practical Electronics: http://www.lcd-linux.sourceforge.net/pdfdocs/lcd1.pdf

Hitachi, L. (1998). *HD44780U (LCD-II).* Retrieved December 20, 2008, from http://web.media.mit.edu/~ayah/documents/hd44780u.pdf