Interfacing 16x2 LCD with 8051 microcontroller

Name: Nirvan Tamhane

Aim: To interface and program the 8051-microcontroller for displaying letters, symbols, and numbers on a 16x2 LCD using Keil µvision 5 software.

Theory:

Display units are the most important output devices in embedded projects and electronics products. 16x2 LCD is one of the most used display unit. 16x2 LCD means that there are two rows in which 16 characters can be displayed per line, and each character takes 5X7 matrix space on LCD. In this tutorial we are going to connect 16X2 LCD module to the 8051 microcontroller (AT89S52). Interfacing LCD with 8051 microcontroller might look quite complex to newbies, but after understanding the concept it would look very simple and easy. Although it may be time taking because you need to understand and connect 16 pins of LCD to the microcontroller.

Category	Pin NO.	Pin Name	Function
Power Pins	1	VSS	Ground Pin, connected to Ground
	2	VDD or VCC	Voltage Pin +5V
Contrast Pin	3	V0 or VEE	Contrast Setting, connected to Vcc thorough a variable resistor.
Control Pins	4	RS	Register Select Pin, RS=0 Command mode, RS=1 Data mode
	5	RW	Read/ Write pin, RW=0 Write mode, RW=1 Read mode
	6	E	Enable, a high to low pulse need to enable the LCD

Data Pins	7-14	D0-D7	Data Pins, Stores the Data to be displayed on LCD or the command instructions
Backlight Pins	15	LED+ or A	To power the Backlight +5V
	16	LED- or K	Backlight Ground

RS: RS is the register select pin. We need to set it to 1, if we are sending some data to be displayed on LCD. And we will set it to 0 if we are sending some command instruction like clear the screen (hex code 01).

RW: This is Read/write pin, we will set it to 0, if we are going to write some data on LCD. And set it to 1, if we are reading from LCD module. Generally, this is set to 0, because we do not have need to read data from LCD. Only one instruction "Get LCD status", need to be read sometimes.

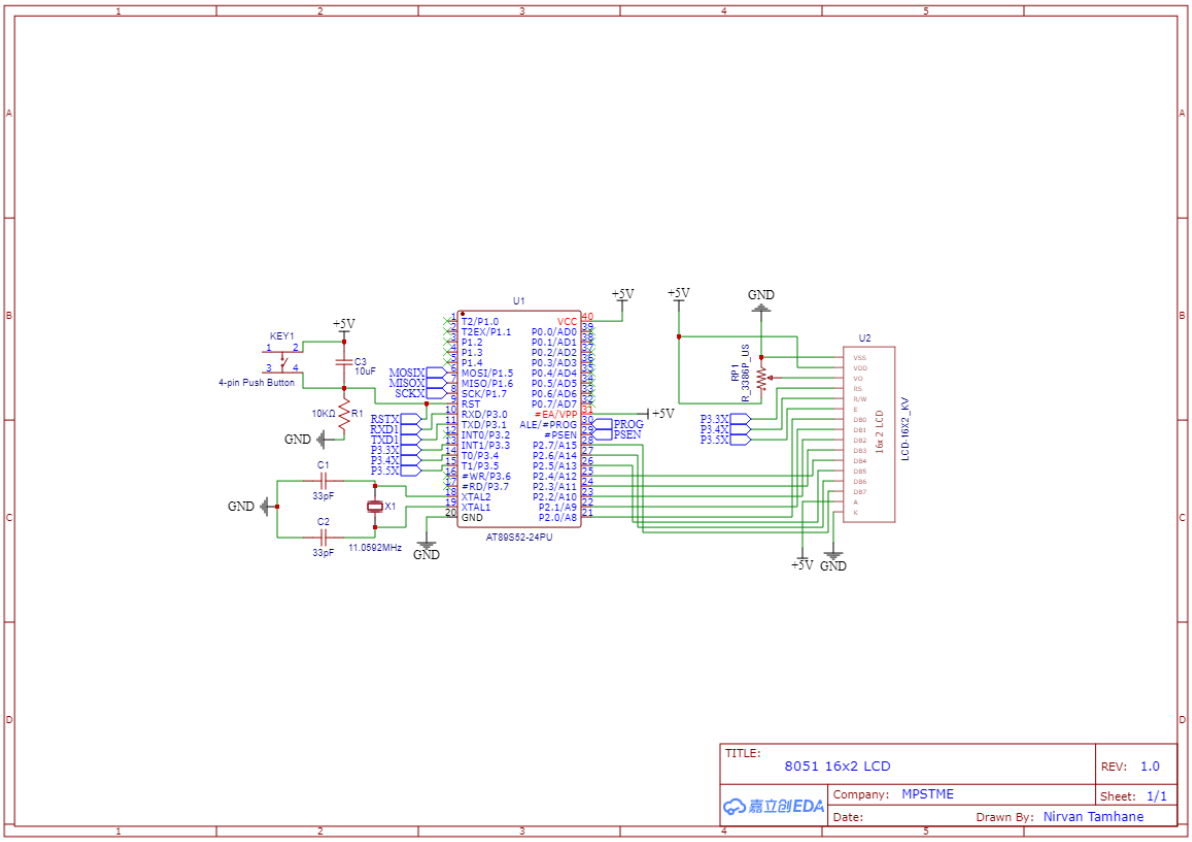
E: This pin is used to enable the module when a high to low pulse is given to it. A pulse of 450 ns should be given. That transition from HIGH to LOW makes the module ENABLE.

There are some pre-set command instructions in LCD, we have used them in our program below to prepare the LCD (in lcd_init() function). Some important command instructions are given below:

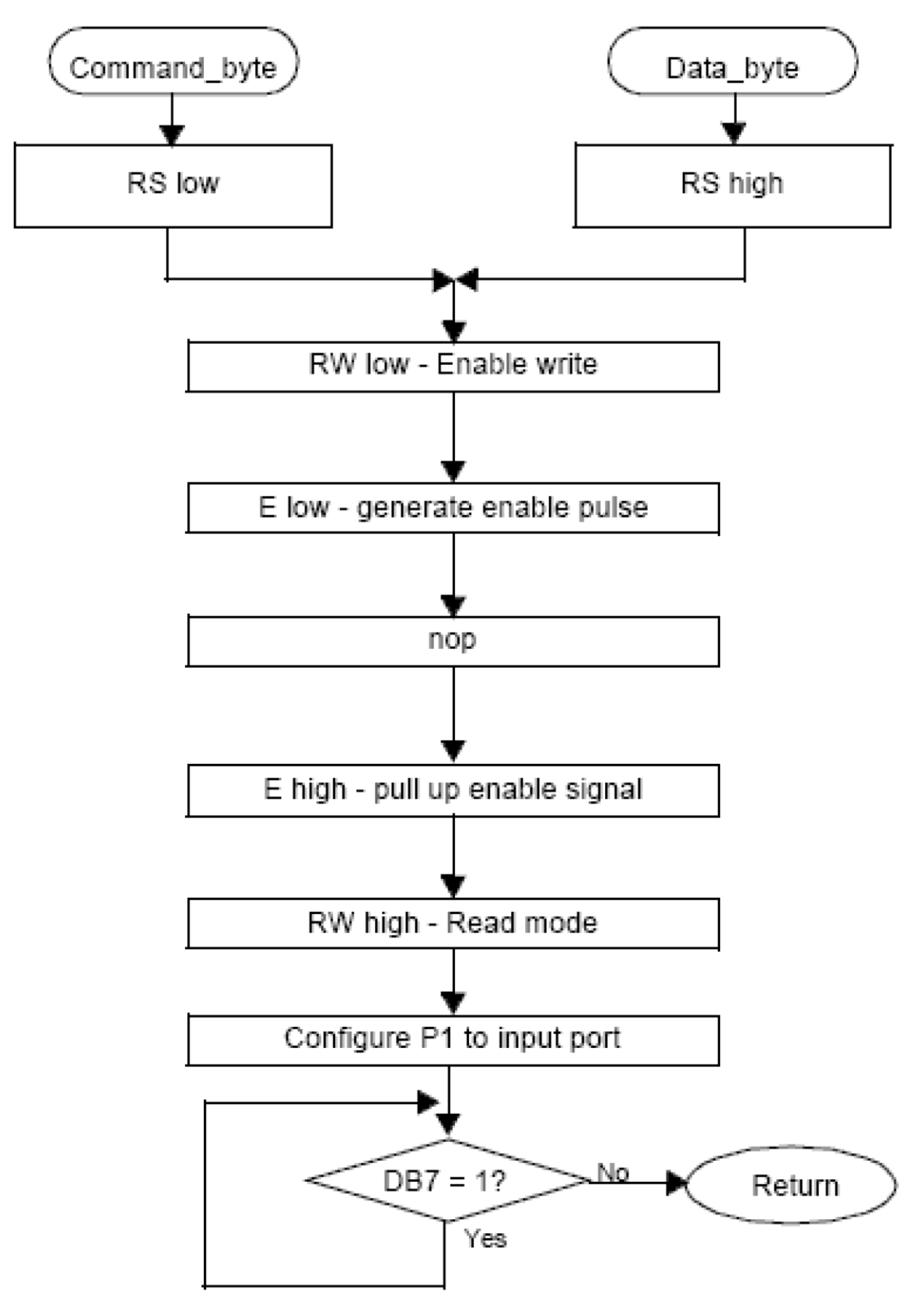
Hex Code	**Command to LCD Instruction Register**
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home
04	Decrement cursor (shift cursor to left)
06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left

0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

Interfacing/Schematic:



Program flowchart:



Embedded C Program:

```
ORG   100H
MOV A,  #38H                ;INIT. LCD 2 LINES, 5X7 MATRIX
ACALL  COMNWRT              ;call command subroutine
ACALL  DELAY                ;give LCD some time
MOV A, #0EH                 ;display on, cursor on
ACALL  COMNWRT              ;call command subroutine
ACALL  DELAY                ;give LCD some time
MOV A, #01                  ;clear LCD
ACALL  COMNWRT              ;call command subroutine
ACALL  DELAY                ;give LCD some time
MOV A, #06H                 ;shift cursor right
ACALL  COMNWRT              ;call command subroutine
ACALL  DELAY                ;give LCD some time
MOV A, #84H                 ;cursor at line 1, pos. 4
ACALL  COMNWRT              ;call command subroutine
ACALL  DELAY                ;give LCD some time
MOV A, #'N'                 ;display letter N
ACALL  DATAWRT              ;call display subroutine
ACALL  DELAY                ;give LCD some time
MOV A, #'O'                 ;display letter O
ACALL  DATAWRT              ;call display subroutine
AGAIN:  SJMP AGAIN          ;stay here

COMNWRT:  MOV P1, A                    ;copy reg A to port 1
CLR  P2.0                   ;RS=0 for command
CLR  P2.1                   ;R/W=0 for write
SETB  P2.2                  ;E=1 for high pulse
ACALL  DELAY                ;give LCD some time
```

```
CLR  P2.2                    ;E=0 for H-to-L pulse
RET

DATAWRT:   MOV P1, A                    ;copy reg A to port 1
SETB  P2.0                   ;RS=1 for data
CLR  P2.1                    ;R/W=0 for write
SETB  P2.2                   ;E=1 for high pulse
ACALL  DELAY                 ;give LCD some time
CLR  P2.2                    ;E=0 for H-to-L pulse
RET

DELAY:    MOV R3, #50                   ;50 or higher for fast CPUs
HERE2: MOV R4, #255          ;R4 = 255
HERE: DJNZ  R4, HERE         ;stay until R4 becomes 0
```

Conclusion – I was able to print the letters, symbols, and numbers on the LCD and learned the interfacing of the 16x2 LCD with 8051-microcontroller using Keil µvision 5 software.