Lab 4

LCD Interfacing with AVR Microcontroller

Objectives

- Overview of controller to controller communication.
- To understand LCD interfacing with AVR microcontroller

Software

- Microchip Studio
- AVR Studio
- Proteus ISIS
- AVRDUDE

Hardware

- Microcontroller: ATMega328p × 1
- Breadboard: Standard × 1
- 16 × 2 LCD: 1602A × 1
- Variable Resistor: Up to 5kΩ × 1

Theory

Introduction To LCD

LCD provides an output interface to AVR microcontroller. It comes in various configurations such as 20 x 2, 16 x 1, 40 x 1, 16 x 2 etc. with 5 x 8 or 5 x 11 dots resolution. Out of these, 16 x 2 is widely used.

LCD also has a microcontroller in it. This microcontroller is already coded to display characters on the screen. Therefor to use an LCD a controller to controller interface is required.

LCD control unit (Controller)

The LCD controller is responsible for all the operations with the LCD. These operations may include (but are not limited to):

- movement of curser,
- clearing display,
- printing characters,

- special character generation etc.

HD44780U and S6A0069 are common examples of LCD control unit. An LCD controller's internal memory comprises of Character Generator ROM (CGROM), Character Generator RAM (CGRAM) and Display Data RAM (DDRAM).

- Character Generator ROM (CGROM) contain predefined character fonts table.
- Character Generator RAM (CGRAM) stores customized character table
- Display Data RAM (DDRAM) stores display data represented in 8-bit character codes.

LCD Pin Configuration

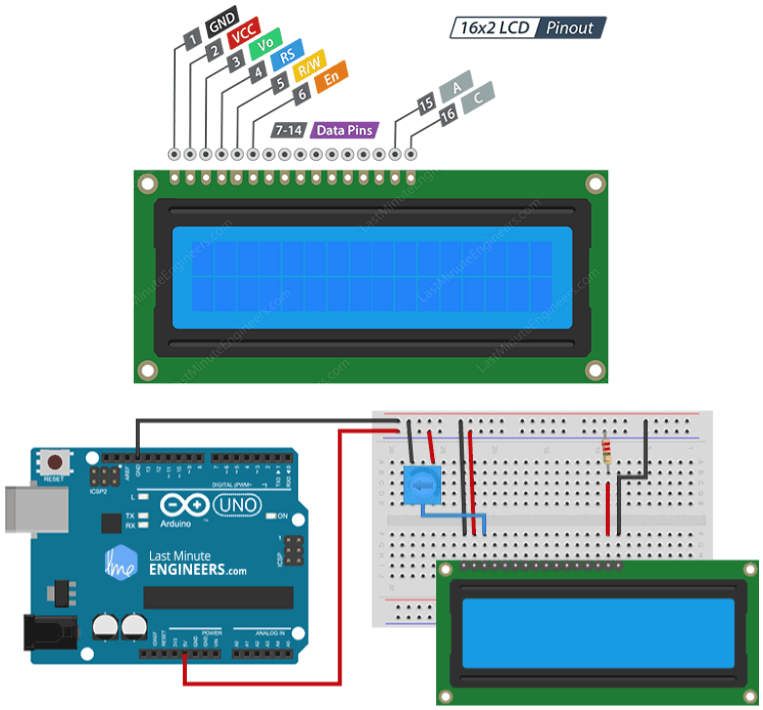

Figure 5.1 – Basic Pin-out of 1630A LCD

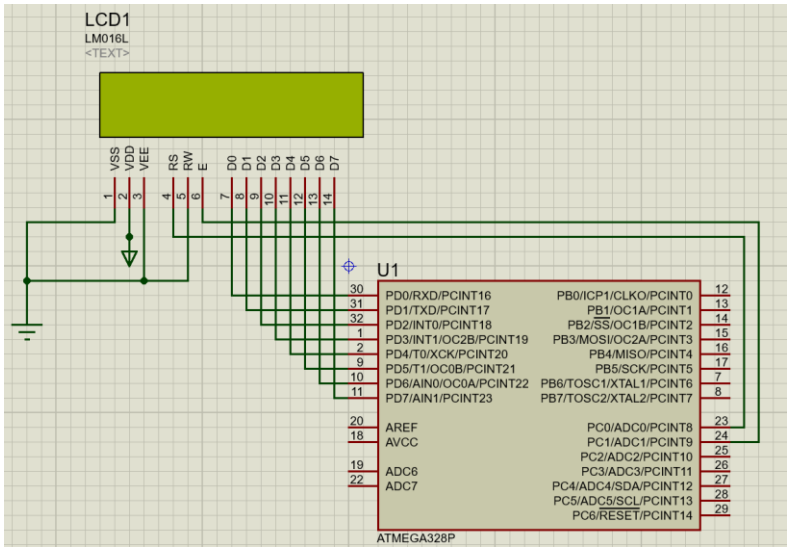

Figure 5.2 – Basic Wiring of LCD with ATmega328P chip

The table below provides the description of the pins:

Pin	Symbol	Function	Description
1	V_{SS}	Ground	Keep it at 0 V
2	V_{CC}	Main power supply	Keep it at +5 V
3	V_{EE}	Power supply for contrast control	Contrast can be adjusted by providing a variable resistor through V_{CC}
4	RS	Register select	RS = 0 means that D0-D7 will be interpreted as Command RS = 1 means that D0-D7 will be interpreted as Data
5	R/W	Read/write	R/W = 0 to write to the LCD register R/W = 1 to read from the LCD register Keep it grounded for normal display operation
6	EN	Enable	This should be toggled to High then Low for any data/command to be passed to the LCD This pin should remain high for a minimum of 450ns
7	D0	Data OR Command pins	These pins can be used to send both commands and data depending on the value of the RS pin. Data can be sent both in 8-bit mode, utilizing all 8 pins, or in the 4-bit mode, utilizing only pins D0-D3, in case there is a constraint with regards to available number of pins on the producer
8	D1		
9	D2		
10	D3		
11	D4		
12	D5		
13	D6		
14	D7		
15	LED+	Backlight power supply	+5 V
16	LED-	Backlight ground	0 V

Table 5.1 – LCD Pin-out Details

LCD Communication Modes

LCD has 2 communication modes. The LCD can be interfaced using 3 control pins (EN, R/W, RS) with either 8 pins or 4 pins for commands or data.

1. **8-bit communication**

   Data and commands can be sent by using 8 data pins simultaneously (D0-D7). Each instruction executes in 1 cycle for 1-byte instruction.

2. **4-bit communication**

   Data and commands can be sent by using only 4 data pins (D0-D3). Each instruction normally executes in 2 cycles for 1-byte instruction i.e. 1 nibble per cycle

LCD Interfacing

R/W pin is used select between the read or write operation. You can either send commands to configure the LCD or send data to be displayed on the LCD. Since same pins (D0 to D7) are used for both command and data, the RS pin is used to choose between them.

Sending Commands to LCD:

1. Clear the RS pin (logic zero)
2. Pass the appropriate Command to the data pins (via connected buffer/port)

3. The command should be sent as per command syntax specified in the Datasheet (see Table 5.2)
4. Set the EN pin (logic one) and hold for at least 450ns
5. Clear the EN pin (logic zero)

Sending Data to the LCD

1. Set the RS pin (logic one)
2. Pass the appropriate Data to the data pins (via connected buffer/port)
3. The data to be displayed should be sent in the form of standard 8-bit ASCII code
4. Set the EN pin (logic one) and hold for at least 450ns
5. Clear the EN pin (logic zero)

LCD commands	Binary Code	Code (Hex)
Clear display screen	0000 0001	1
Return home	0000 0010	2
Decrement cursor (shift cursor to left)	0000 0100	4
Increment cursor (shift cursor to right)	0000 0110	6
Shift display right	0000 0101	5
Shift display left	0000 0111	7
Display off, cursor off	0000 1000	8
Display off, cursor on	0000 1010	A
Display on, cursor off	0000 1100	C
Display on (cursor not blinking)	0000 1110	E
Display on (cursor blinking)	0000 1111	F
Shift cursor position to left	0001 0000	10
Shift cursor position to right	0001 0100	14
Shift the entire display to the left	0001 1000	18
Shift the entire display to the right	0001 1100	1C
Force cursor to beginning to 1st line	1000 0000	80
Force cursor to beginning to 2nd line	1100 0000	C0
Force cursor to beginning to 3rd line	1001 0000	90
Force cursor to beginning to 4th line	1101 0000	D0
2 lines and 5×7 matrix (8-bit mode)	0011 1000	38
2 lines and 5×7 matrix (4-bit mode)	0010 1000	28

Table 5.2 – 1602A LCD Command List

Pre-Lab Tasks

Task-1

Read the theory section of this lab thoroughly and implement the circuit show in figure 5.2 on Proteus.

Task-2

Write an assembly language program that is able to display the numbers '0' to '9' on an LCD as connected in figure 5.2. *[hint: look up ASCII table]*

Task-3

Consider the basic wiring shown between an ATmega328P chip and an LCD in figure 5.2. Write and execute a C-program on Proteus that is able print your name on the first row of the LCD and your roll-number on the second row of the LCD

In-Lab Tasks

Task-1

Wire your Arduino Uno / Nano / ATmega328P to an LCD on your breadboard and execute the program performed in Task-2 from 'Pre-Lab' Tasks

Task-2

Consider your controller connected to an 8×dipswitch, a push-button and an LCD. Program your controller in a way that whenever the push-button is pressed, the input from the dipswitch should be pushed on to the LCD as an ASCII character. Execute your code on Proteus.

Task-3

Implement task 2 on your available hardware

Post Lab

Task-1

Implement the in-lab task for the 4-pin communication mode on Proteus. Additionally, add a 5×dip-switch array at an input. Use the dip switch to pass 8-bit ASCII code, one nibble at a time, to be displayed on the LCD. Toggle the 5th switch to indicate that one nibble is ready to load.

Task-2

Implement the previous task on available hardware

Critical Analysis / Conclusion (To be filled in by the student)

Lab Assessment (To be filled by the lab-instructor)

Pre-Lab	/5	/25
In-Lab	/5	
Results	/5	
Viva	/5	
Critical Analysis	/5	

Comments:

Instructor Name

Instructor Signature