

LIQUID CRYSTAL DISPLAY (LCD)

- LCDs provide a better user interface as they can display ASCII messages
- Pin Description for LCD
 - LCD modules usually have 14 pins.

Pin No.	Name	Function	Description
1	V_{SS}	Power	GND
2	V_{DD}	Power	+ 5 V
3	V_{EE}	Contrast adjust	0 – 5 V
4	RS	Register select	Signal to select data or command register of the LCD. RS = 0 select command register (for write); Busy flag, address counter (for read) RS = 1 select data register (for write)
5	R/W	Read / Write	Signal to read/write data from/to LCD. RW = 0 Write to LCD RW=1 read from LCD
6	E	Enable (Strobe)	High to low pulse is applied to this pin to enable LCD to accept (latch) data/command present on its data lines D0-D7
7-14	D0-D7	Data lines D0 (Pin-7-LSB), D7 (Pin-14-MSB)	Bidirectional data lines used to send data/command to LCD; or read LCD internal registers, D7 is also used as a busy flag, D4 -D7 are used in the 4 bit operation

Using Simulator

- In the lab exercise, we used the LCD in 4-bit Mode.
- To use the simulator for the reset of labs, the LCD must be in 8-bit Mode.
- To change mode, RS and E should be remapped to other port pins, using the DI button at the top left of the peripheral panel.

The screenshot displays the EdSim5101 simulator interface. The top window shows assembly code for a clock application. The middle window displays the register file and memory. The bottom window shows the peripheral panel with various hardware components.

Assembly Code:

```
ORG 00H
MOV R0, #20
MOV R1, #0 ;Set time val
MOV R2, #0 ;minutes
MOV R3, #0 ;hours
ACALL SETDIS ;initialise the
MOV TMOD, #0x01
REPEAT: MOV TH0, #0x3C
MOV TL0, #0xB0
SETB TR0
WAIT: JNB TF0, WAIT
CLR TR0
CLR TF0
DJNZ R0, REPEAT
MOV TH0, #0x3C
MOV TL0, #0xB0
SETB TR0
MOV R0, #19
CPL P2.3 ;output every se
ACALL INCT ;Increment ti
ACALL DIST ;Display time
```

Register File:

R/O	W/O	TH0	TL0	R7	B
0x00	0x00	0x00	0x00	0x00	0x00
R6				0x00	ACC
R5				0x00	PSW
R4				0x00	IP
R3				0x00	IE
R2				0x00	PCON
R1				0x07	DPH
R0				0x0A	DPL
					SP

Memory:

addr	0x00	0x00	value
0	0A	07	00
1	00	00	00
2	00	00	00
3	00	00	00
4	00	00	00
5	00	00	00
6	00	00	00
7	00	00	00
8	00	00	00
9	00	00	00
A	00	00	00
B	00	00	00
C	00	00	00
D	00	00	00
E	00	00	00
F	00	00	00

Peripheral Panel:

- DI button (highlighted with a red box and arrow)
- LD button
- AND Gate Disabled
- Key Bounce Disabled
- Standard
- 8-bit UART @ 4800 Baud
- Rx Reset
- Tx Send
- 0.0 V input
- MAX
- MIN
- Motor Enabled
- ADC
- 0.0 V output
- Scope
- DAC
- BF
- AC
- 0x00
- IR
- 0x00
- DR
- 0x00

Using Simulator cont.

- Change the port P1.2 LCD E to P2.2 and the port P1.3 LCD RS to P2.0.

EdSim51DI - Dynamic Interface

Restart required for settings to take effect.

P 0 . 0 Keypad Row 0	P 1 . 3 DAC DB3	P 2 . 2 ADC DB2
P 0 . 1 Keypad Row 1	P 1 . 3 LCD DB3	P 2 . 3 SW 3
P 0 . 2 Keypad Row 2	P 1 . 3 LCD R	P 2 . 3 ADC DB3
P 0 . 3 Keypad Row 3	P 1 . 4 LED 4	P 2 . 4 SW 4
P 0 . 4 Keypad Column 0	P 1 . 4 Seg. e	P 2 . 4 ADC DB4
P 0 . 5 Keypad Column 1	P 1 . 4 DAC DB4	P 2 . 5 SW 5
P 0 . 6 Keypad Column 2	P 1 . 4 LCD DB4	P 2 . 5 ADC DB5
P 0 . 7 Display-select Decoder C5	P 1 . 5 LED 5	P 2 . 6 SW 6
P 0 . 7 DAC WR	P 1 . 5 Seg. f	P 2 . 6 ADC DB6
P 1 . 0 LED 0	P 1 . 5 DAC DB5	P 2 . 7 SW 7
P 1 . 0 Seg. a	P 1 . 5 LCD DB5	P 2 . 7 ADC DB7
P 1 . 0 DAC DB0	P 1 . 6 LED 6	P 3 . 0 Motor Control Bit 0
P 1 . 0 LCD DB0	P 1 . 6 Seg. g	P 3 . 0 Ext. UART Tx
P 1 . 1 LED 1	P 1 . 6 DAC DB6	P 3 . 1 Motor Control Bit 1
P 1 . 1 Seg. b	P 1 . 6 LCD DB6	P 3 . 1 Ext. UART Rx
P 1 . 1 DAC DB1	P 1 . 7 LED 7	P 3 . 2 ADC INTR
P 1 . 1 LCD DB1	P 1 . 7 Seg. dp	P 3 . 3 AND Gate Output
P 1 . 2 LED 2	P 1 . 7 DAC DB7	P 3 . 3 Display-select Input 0
P 1 . 2 Seg. c	P 1 . 7 LCD DB7	P 3 . 4 Display-select Input 1
P 1 . 2 DAC DB2	P 2 . 0 SW 0	P 3 . 5 Motor Sensor
P 1 . 2 LCD DB2	P 2 . 0 ADC DB0	P 3 . 6 ADC WR
P 1 . 2 LCD E	P 2 . 1 SW 1	P 3 . 7 ADC RD
P 1 . 3 LED 3	P 2 . 1 ADC DB1	P 3 . 7 Comparator Output
P 1 . 3 Seg. d	P 2 . 2 SW 2	

Change it to P 2 . 2

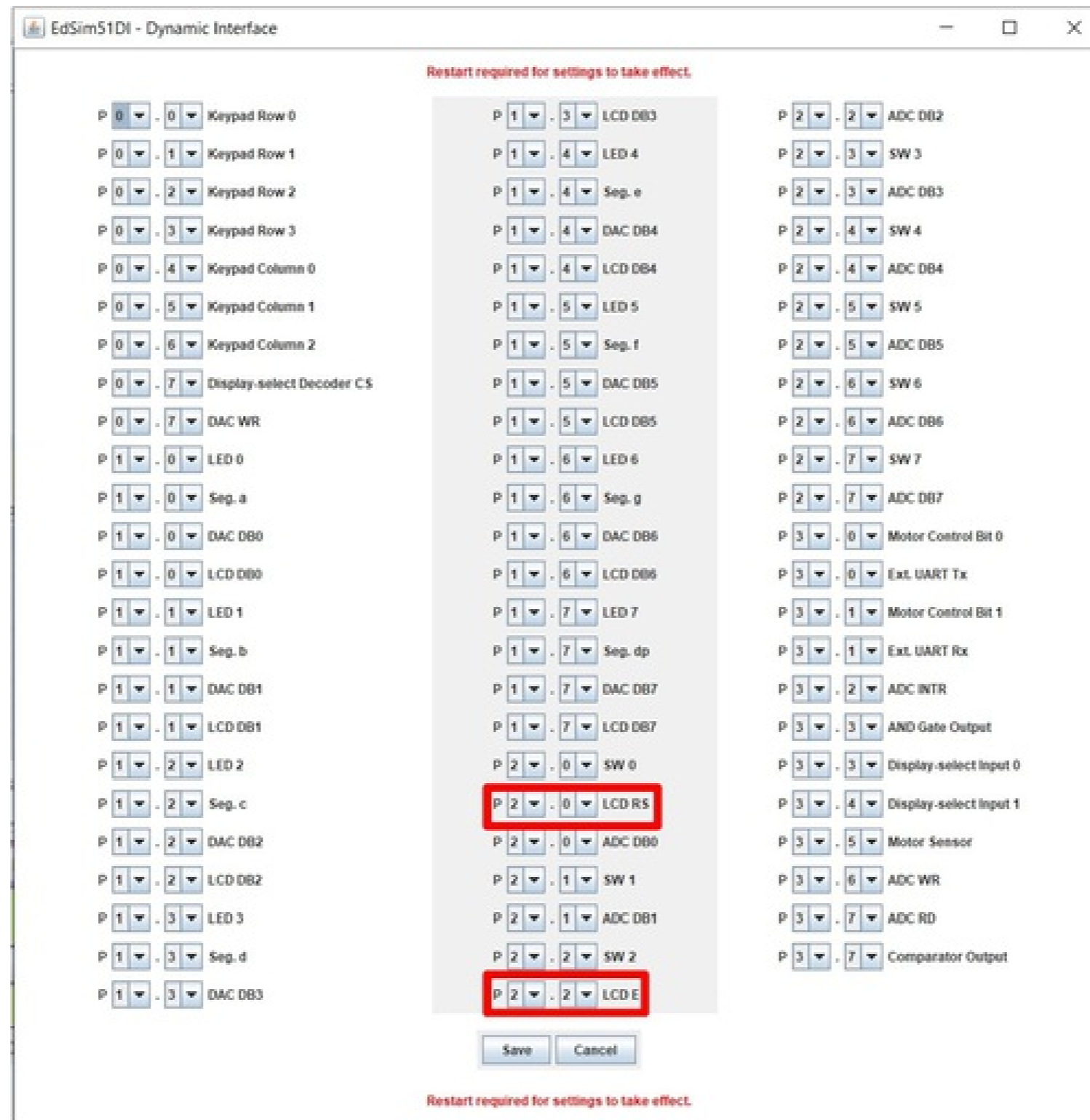
Change it to P 2 . 0

Save Cancel

Restart required for settings to take effect.

Using Simulator cont.

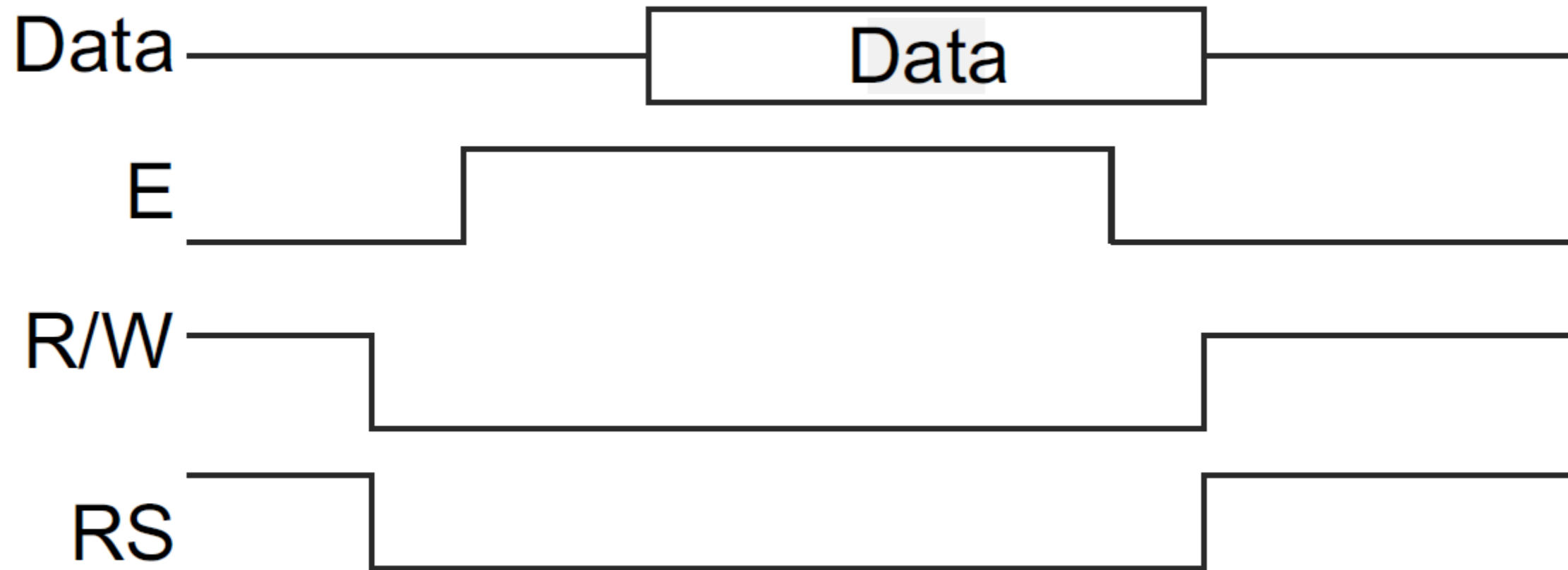
- Save the new settings and restart the program to see the changes.



LCD commands

Command	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	Command Code (Hex)	Execution Time (Max.) $f_{cp} = 250 \text{ kHz}$	
Clear display	0	0	0	0	0	0	0	0	0	1	01	1.64 ms	
Cursor home	0	0	0	0	0	0	0	0	1	x	02	1.64 ms	
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	04–Shift cursor left 06–Shift cursor right 05–Shift display right 07–Shift display left	40 μ s	
Display on/off control	0	0	0	0	0	0	1	D	U	B	08–Display off, Cursor off 0A–Display off, Cursor on 0C–Display on, Cursor off 0E–Display on, Cursor blink off 0F–Display on, Cursor blink	40 μ s	
Cursor/Display Shift	0	0	0	0	0	1	D/C	R/L	x	x	10–Shift cursor left 14–Shift cursor right 18–Shift display left 1C–Shift display right	40 μ s	
Function set	0	0	0	0	1	DL	N	F	x	x	28–2line,5X7matrix,4 line 38–2line,5X7matrix,8 line	40 μ s	
Set CGRAM address	0	0	0	1	CGRAM address							40 μ s	
Set DDRAM address	0	0	1	DDRAM address							80– Set cursor at beginning of line 1	40 μ s	
Read “BUSY” flag (BF)	0	1	BF	DDRAM address								40 μ s	
Write to CGRAM or DDRAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0		40 μ s	
Read from CGRAM or DDRAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0		40 μ s	
I/D 1 = Increment (by 1)			0 = Decrement (by 1)					R/L 1 = Shift right			0 = Shift left		
S 1 = Display shift on			0 = Display shift off					DL 1 = 8-bit interface			0 = 4-bit interface		
D 1 = Display on			0 = Display off					N1 = Display in two lines			0 = Display in one line		
U 1 = Cursor on			0 = Cursor off					F 1 = Character format 5 \times 10 dots			0 = 5 \times 7 dots		
B 1 = Cursor blink on			0 = Cursor blink off					D/C 1 = Display shift			0 = Cursor shift		

LCD timing



Display message 'HI'

- Initialize the LCD by sending a set of commands to it
 1. Configure data bus as 4-bit or 8-bit mode
 2. Select character font, (i.e.) dots/character
 3. Configure display and cursor type, i.e. display ON or not, cursor blinking or not
 4. Configure display and cursor movement, i.e. left shift or right shift or shift off
 5. Configure display position
 6. Clear display
- (The commands are sent by making $RS = 0$ and $R/W = 0$, and high to low pulse on E pin, a delay should be provided between two commands to ensure that the LCD has executed the previous command.)
- Send ASCII values of the characters to be displayed one character at a time with a delay between them.
- (Data characters are sent to the LCD by making $RS = 1$ and $R/W=0$, and high to low pulse on E pin, with a delay between two consecutive data characters).

Example

- Write an assembly-language program to generate a square wave on P1.1 with an ON time (high time) of 1 ms. Assume the crystal frequency is 11.0592 MHz.
- Hint: Count to be loaded in to TH and TL = FC66H.

```
MOV TMOD, #10H           ; configure Timer 1 in Mode 1
REPEAT:
MOV TL1, #66H            ; load count in TH1-TL1
MOV TH1, #0FCH           ;
SETB P1.1                ; set P1.1 high to get ON part of
                          ; square wave
SETB TR1                 ; start Timer 1
HERE: JNB TF1, HERE      ; wait until timer overflows
CLR P1.1                  ; clear P1.1 to get OFF part of square wave
CLR TR1                   ; stop timer
CLR TF1                   ; clear overflow flag
SJMP REPEAT              ; reload timer registers and repeat
```