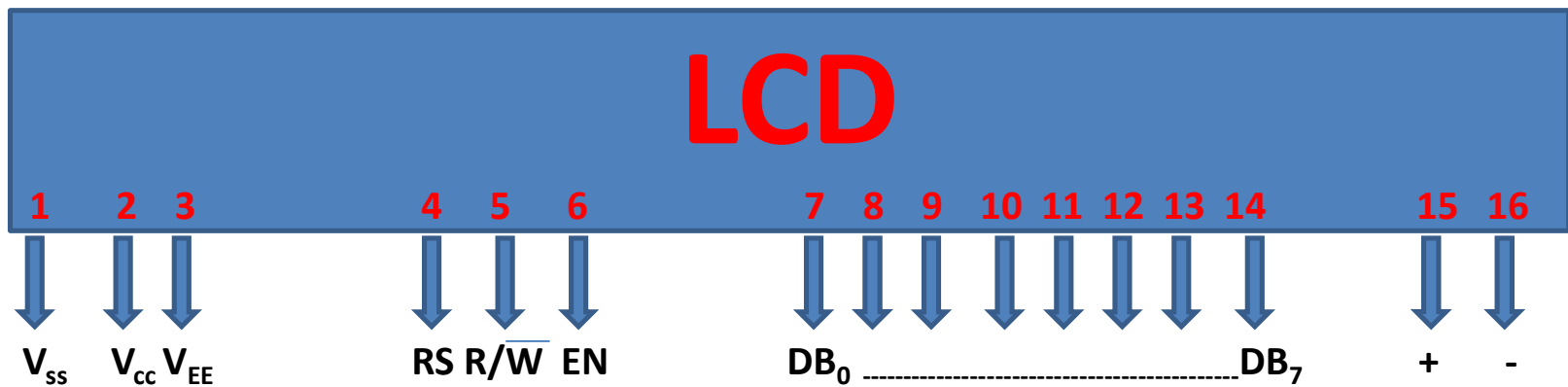


# LCD Interfacing (16X2)



# Pin Descriptions for LCD

Pin	Symbol	I/O	Descriptions
1	VSS	--	Ground
2	VCC	--	+5V power supply
3	VEE	--	Power supply to control contrast
4	RS	I	RS=0 to select command register, RS=1 to select data register
5	R/W	I	R/W=0 for write, R/W=1 for read
6	E	I/O	Enable
7	DB0	I/O	The 8-bit data bus
8	DB1	I/O	The 8-bit data bus
9	DB2	I/O	The 8-bit data bus
10	DB3	I/O	The 8-bit data bus
11	DB4	I/O	The 8-bit data bus
12	DB5	I/O	The 8-bit data bus
13	DB6	I/O	The 8-bit data bus
14	DB7	I/O	The 8-bit data bus

- Send displayed information or instruction command codes to the LCD  
- Read the contents of the LCD's internal registers

used by the LCD to latch information presented to its data bus

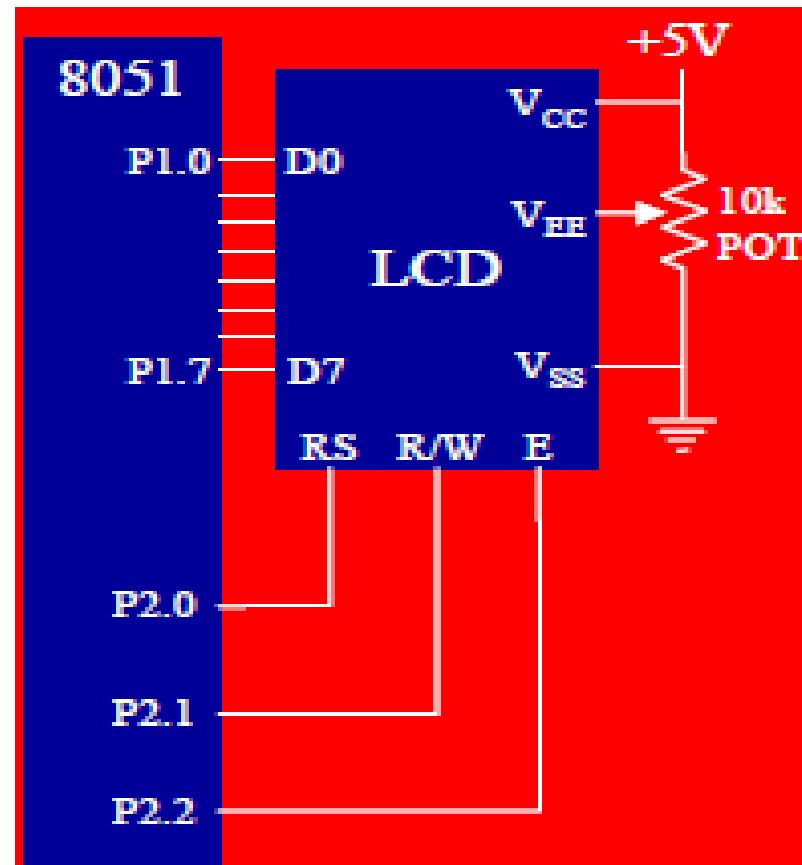
## LCD Command Codes

Code (Hex)	Command to LCD Instruction Register
1	Clear display screen
2	Return home
4	Decrement cursor (shift cursor to left)
6	Increment cursor (shift cursor to right)
5	Shift display right
7	Shift display left
8	Display off, cursor off
A	Display off, cursor on
C	Display on, cursor off
E	Display on, cursor blinking
F	Display on, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to the left
1C	Shift the entire display to the right
80	Force cursor to beginning to 1st line
C0	Force cursor to beginning to 2nd line
38	2 lines and 5x7 matrix

# Addresses of 16x2 LCD

80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

# LCD INTERFACING TO 8051



To send any of the commands to the LCD, make pin RS=0. For data, make RS=1. Then send a high-to-low pulse to the E pin to enable the internal latch of the LCD.

;calls a time delay before sending next data/command  
 ;P1.0-P1.7 are connected to LCD data pins D0-D7  
 ;P2.0 is connected to RS pin of LCD  
 ;P2.1 is connected to R/W pin of LCD  
 ;P2.2 is connected to E pin of LCD

```

ORG 0H
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: ;send command to LCD
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: ;write data to LCD
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
DJNZ R3,HERE2
RET
END

```

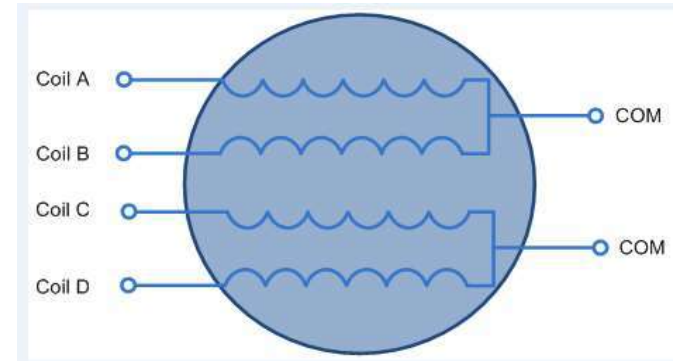
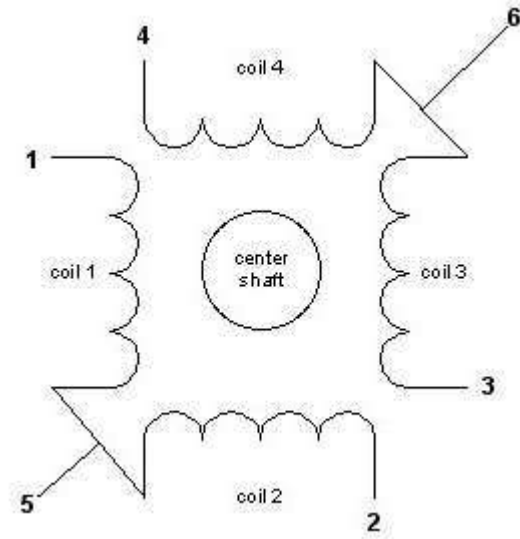
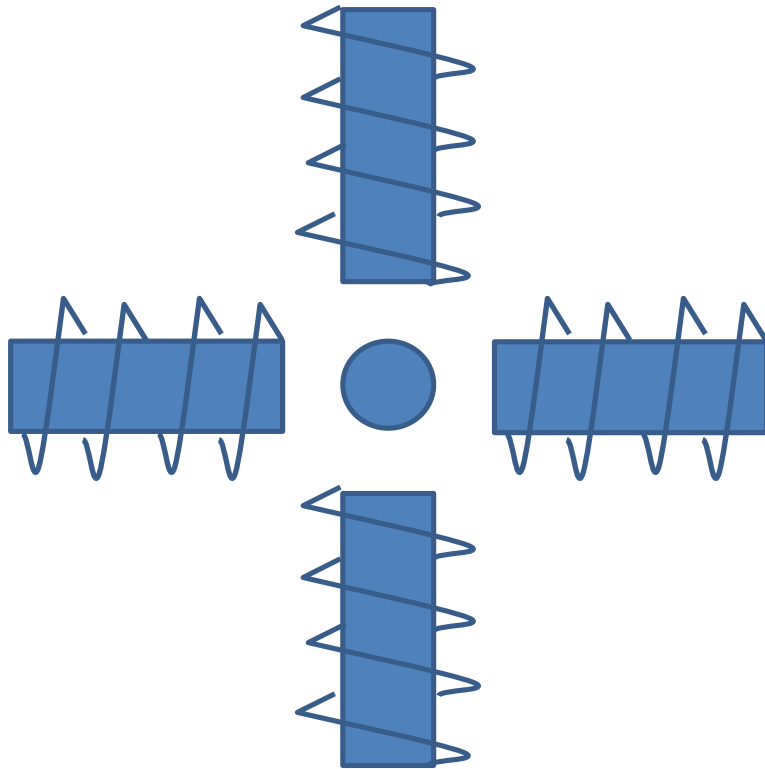
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

# **STEPPER MOTOR INTERFACING**

- A Stepper motor is a widely used device that translates electrical pulses into mechanical movement.
- In applications such as disk drives, dot matrix printers & robotics, the stepper motor is used for position control
- Stepper motors commonly have a permanent magnet 'rotor' (also called shaft) surrounded by a 'stator'.



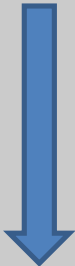

# Stator Windings Configuration







# Driving Stepper motor

## Normal 4 Step Sequence



CLOCKWISE	STEP	WINDING A	WINDING B	WINDING C	WINDING D	COUNTER CLOCKWISE
	1	1	1	0	0	
	2	0	1	1	0	
	3	0	0	1	1	
	4	1	0	0	1	

## Wave Drive 4 step sequence

CLOCKWISE	STEP	WINDING A	WINDING B	WINDING C	WINDING D	COUNTER CLOCKWISE
	1	1	0	0	0	
	2	0	1	0	0	
	3	0	0	1	0	
	4	0	0	0	1	

# Driving Stepper motor

## Half step 8-step sequence Step Sequence

CLOCKWISE	STEP	WINDING A	WINDING B	WINDING C	WINDING D	COUNTER CLOCKWISE
	1	1	0	0	0	
	2	1	1	0	0	
	3	0	1	0	0	
	4	0	1	1	0	
	5	0	0	1	0	
	6	0	0	1	1	
	7	0	0	0	1	
	8	1	0	0	1	

# Stepper Motor Program

Assuming two stepper motors are interfaced to 8051 with ULN2803 drivers.

**Program for normal 4-step sequence ( clock wise )**

ORG 0H

BACK:      MOV A,#0CCH

          MOV P1,A

          ACALL DELAY

          MOV A,#66H

          MOV P1,A

          ACALL DELAY

          MOV A,#33H

          MOV P1,A

          ACALL DELAY

          MOV A,#99H

          MOV P1,A

          ACALL DELAY

          SJMP BACK

END

DELAY:

          MOV R0,#0FFH

LOOP2:    MOV R1,#80H

LOOP1:    DJNZ R1,LOOP1

          DJNZ R0,LOOP2

          RET

# Stepper Motor Program

Assuming two stepper motors are interfaced to 8051 with ULN2803 drivers.

**Program for wave drive 4-step sequence ( clock wise )**

ORG 0H

```
BACK:    MOV A,#88H
          MOV P1,A
          ACALL DELAY
          MOV A,#44H
          MOV P1,A
          ACALL DELAY
          MOV A,#22H
          MOV P1,A
          ACALL DELAY
          MOV A,#11H
          MOV P1,A
          ACALL DELAY
          SJMP BACK
```

END

DELAY:

```
          MOV R0,#0FFH
LOOP2:    MOV R1,#80H
LOOP1:    DJNZ R1,LOOP1
          DJNZ R0,LOOP2
          RET
```

# Stepper Motor Program

Assuming two stepper motors are interfaced to 8051 with ULN2803 drivers.

**Program for half step 8-step sequence ( clock wise )**

ORG 0H

BACK:	MOV A,#88H	MOV A,#22H	
	MOV P1,A	MOV P1,A	
	ACALL DELAY	ACALL DELAY	DELAY:
	MOV A,#0CCH	MOV A,#33H	MOV R0,#0FFH
	MOV P1,A	MOV P1,A	LOOP2: MOV R1,#80H
	ACALL DELAY	ACALL DELAY	LOOP1: DJNZ R1,LOOP1
	MOV A,#44H	MOV A,#11H	DJNZ R0,LOOP2
	MOV P1,A	MOV P1,A	RET
	ACALL DELAY	ACALL DELAY	
	MOV A,#66H	MOV A,#99H	
	MOV P1,A	MOV P1,A	
	ACALL DELAY	ACALL DELAY	
		SJMP BACK	
		END	