# MICROCONTROLLER

UNIT-V
Lecture-4

# 8031 System With 8255

- In 8031-based system
- external program ROM is an absolute must
- the use of 8255 is most welcome
- this is due to the fact that 8031 to external program ROM, we lose the two ports P0 and P2, leaving only P1
- Therefore, connecting an 8255 is the best way to gain some extra ports.
- Shown in Figure 15-8

# Contd.

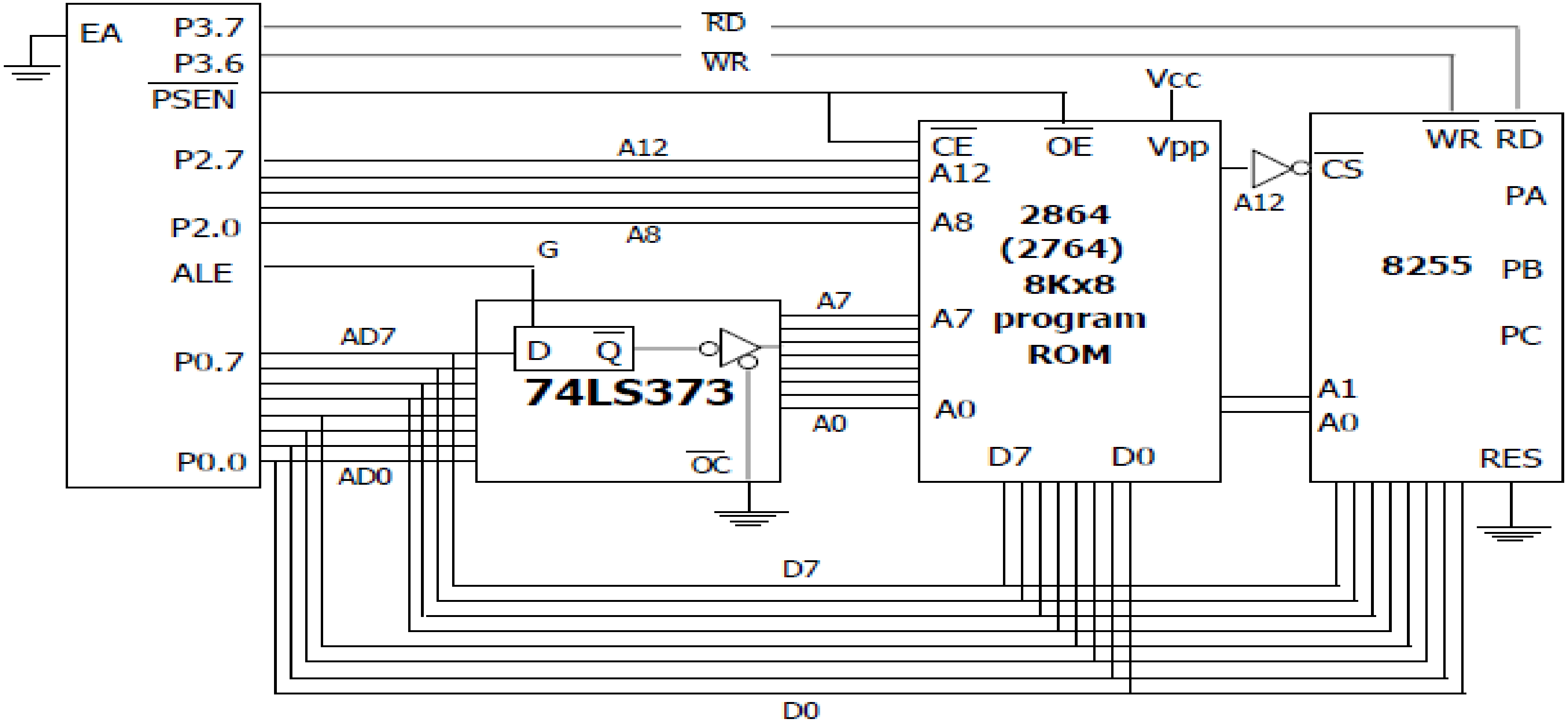
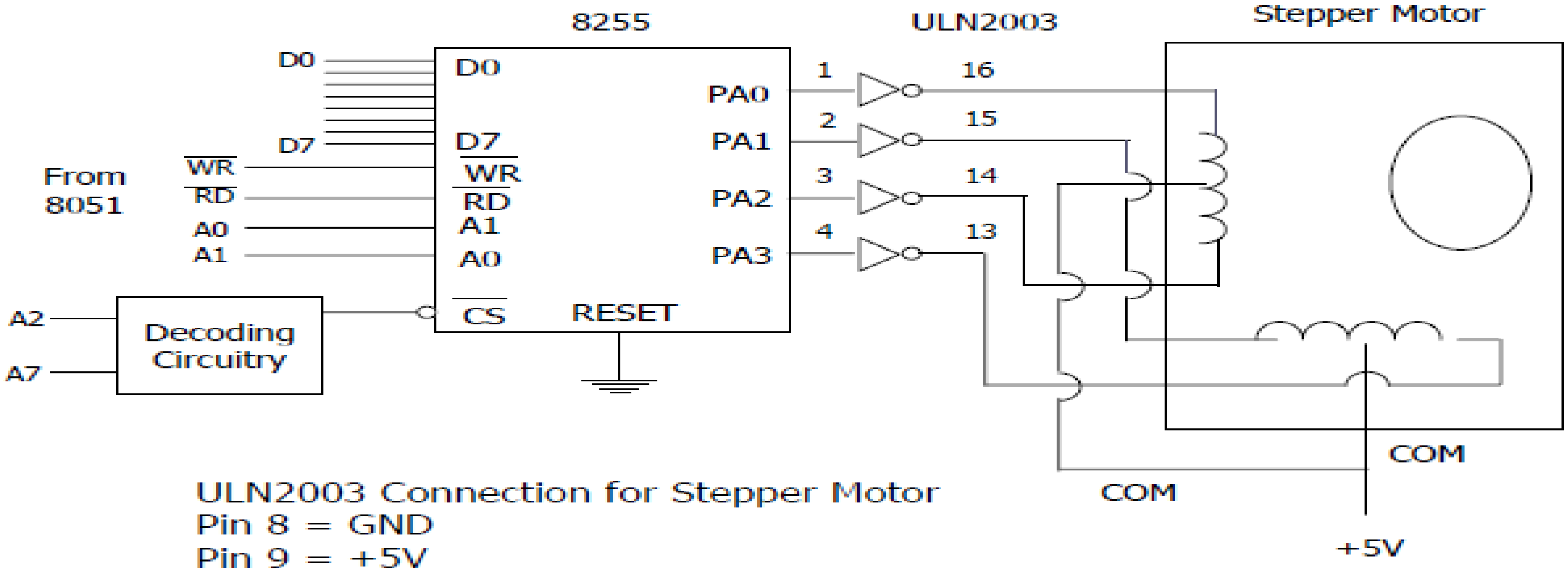


Figure 15-8. 8031 Connection to External Program ROM and the 8255

# Stepper Motor Connection To The 8255

- Here show stepper motor connection to the 8255 and programming in Fig 15-9



ULN2003 Connection for Stepper Motor
Pin 8 = GND
Pin 9 = +5V

Use a separate power supply for the motor

**Figure 15-9. 8255 Connection to Stepper Motor**

# Contd.

```
     MOV   A,#80H     ;control word for PA=out
     MOV   R1,#CRPORT ;control reg port
    address
     MOVX  @R1,A      ;configure PA=out
     MOV   R1,#APORT  ;load PA address
     MOV   A,#66H ;A=66H,stepper motor
    sequence
AGAIN MOVX  @R1,A    ;issue motor sequence to
  PA
     RR    A     ;rotate sequence for
  clockwise
     ACALL DELAY        ;wait
     SJMP AGAIN
```

# LCD Connection To The 8255

- Program 15-1: Shows how to issue commands and data to an LCD. See Figure 15-10
- must put a long delay before issue any information to the LCD
- Program 15-2: A repeat of Program 15-1 with the checking of the busy flag
- Notice that no DELAY is used in the main program

# Contd.

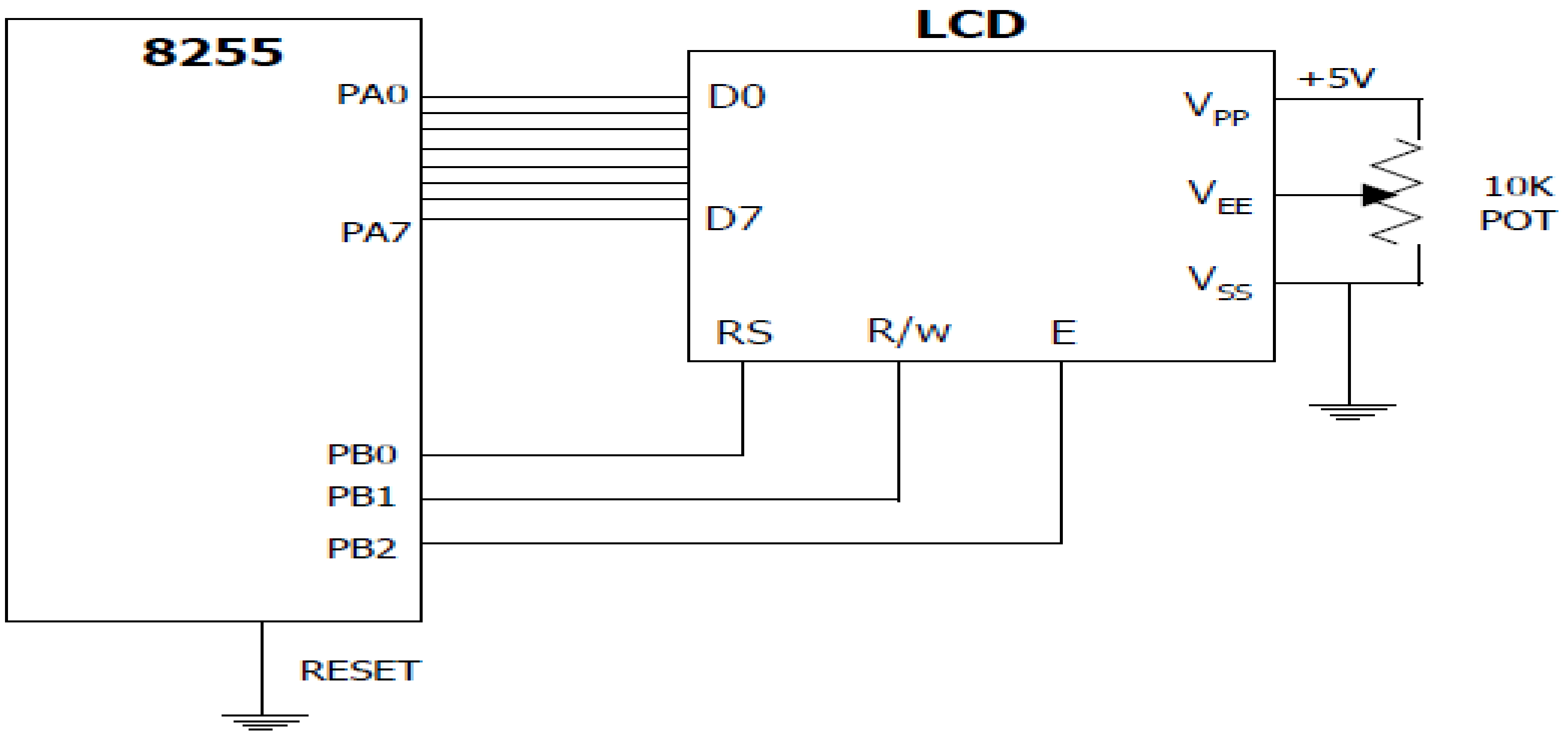
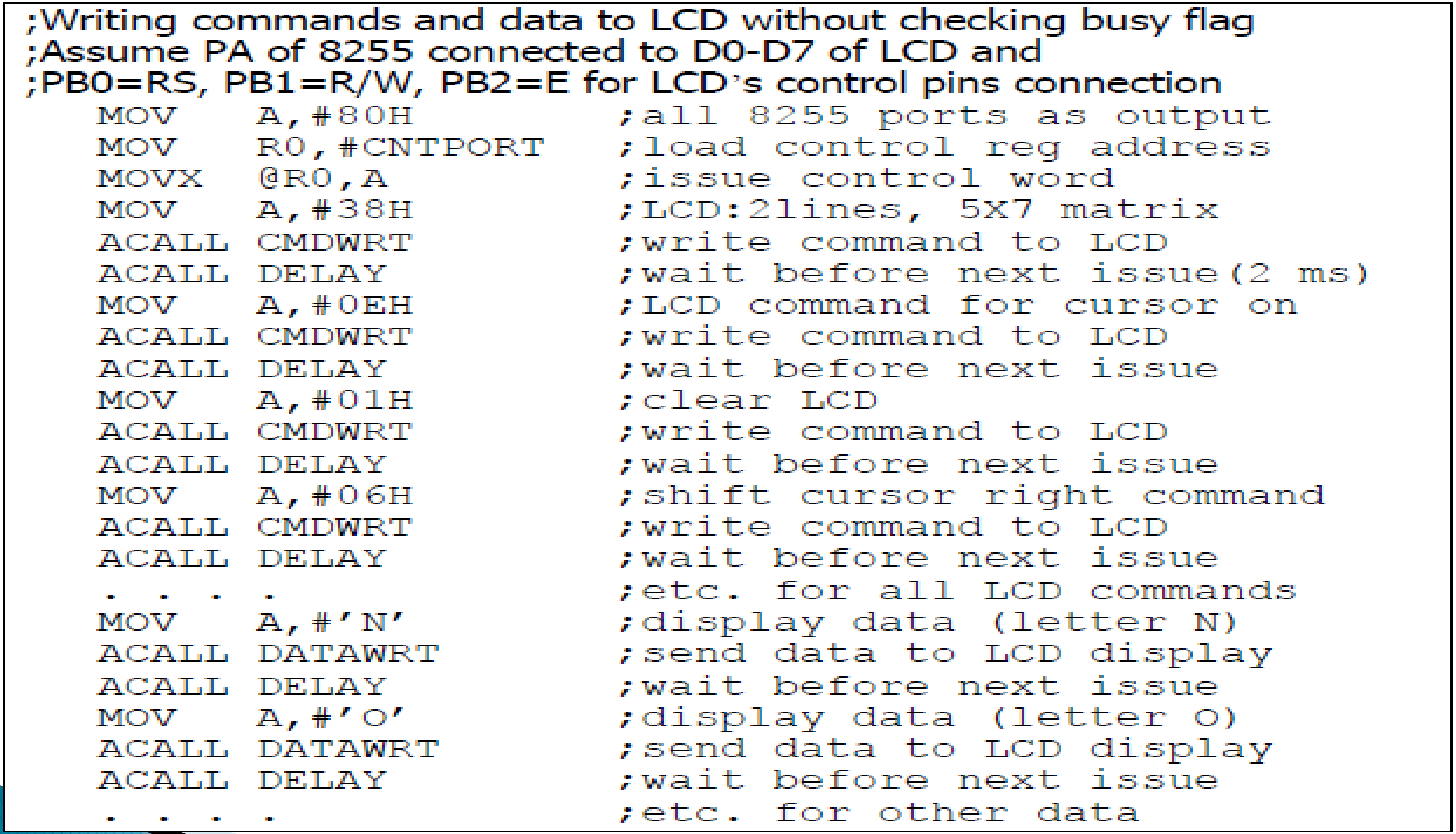


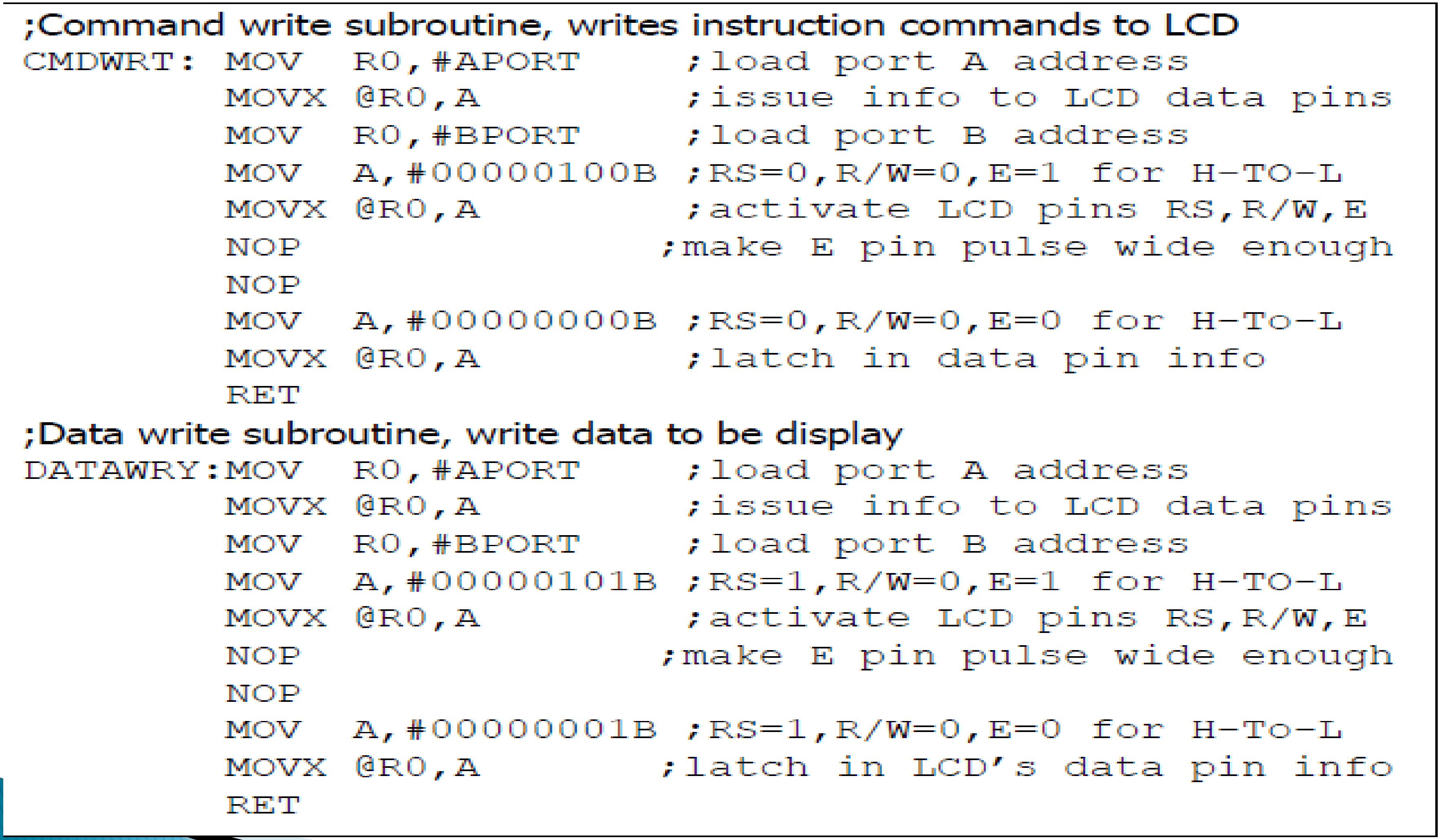
Figure 15-10. LCD Connection

# Contd.



```
;Writing commands and data to LCD without checking busy flag
;Assume PA of 8255 connected to D0-D7 of LCD and
;PB0=RS, PB1=R/W, PB2=E for LCD's control pins connection
   MOV    A,#80H          ;all 8255 ports as output
   MOV    R0,#CNTPORT     ;load control reg address
   MOVX   @R0,A           ;issue control word
   MOV    A,#38H          ;LCD:2lines, 5X7 matrix
   ACALL  CMDWRT          ;write command to LCD
   ACALL  DELAY           ;wait before next issue(2 ms)
   MOV    A,#0EH          ;LCD command for cursor on
   ACALL  CMDWRT          ;write command to LCD
   ACALL  DELAY           ;wait before next issue
   MOV    A,#01H          ;clear LCD
   ACALL  CMDWRT          ;write command to LCD
   ACALL  DELAY           ;wait before next issue
   MOV    A,#06H          ;shift cursor right command
   ACALL  CMDWRT          ;write command to LCD
   ACALL  DELAY           ;wait before next issue
   . . . .                ;etc. for all LCD commands
   MOV    A,#'N'          ;display data (letter N)
   ACALL  DATAWRT         ;send data to LCD display
   ACALL  DELAY           ;wait before next issue
   MOV    A,#'O'          ;display data (letter O)
   ACALL  DATAWRT         ;send data to LCD display
   ACALL  DELAY           ;wait before next issue
   . . . .                ;etc. for other data
```

# Contd.



```
;Command write subroutine, writes instruction commands to LCD
CMDWRT: MOV   R0,#APORT     ;load port A address
        MOVX  @R0,A         ;issue info to LCD data pins
        MOV   R0,#BPORT     ;load port B address
        MOV   A,#00000100B  ;RS=0,R/W=0,E=1 for H-TO-L
        MOVX  @R0,A         ;activate LCD pins RS,R/W,E
        NOP                ;make E pin pulse wide enough
        NOP
        MOV   A,#00000000B  ;RS=0,R/W=0,E=0 for H-To-L
        MOVX  @R0,A         ;latch in data pin info
        RET
;Data write subroutine, write data to be display
DATAWRY:MOV   R0,#APORT     ;load port A address
        MOVX  @R0,A         ;issue info to LCD data pins
        MOV   R0,#BPORT     ;load port B address
        MOV   A,#00000101B  ;RS=1,R/W=0,E=1 for H-TO-L
        MOVX  @R0,A         ;activate LCD pins RS,R/W,E
        NOP                ;make E pin pulse wide enough
        NOP
        MOV   A,#00000001B  ;RS=1,R/W=0,E=0 for H-To-L
        MOVX  @R0,A        ;latch in LCD's data pin info
        RET
```