# Microprocessor, Microcontroller and Interfacing Techniques Portfolio Assignment II

## Portfolio Assignment 2:

UNIT IV:

**Question 1:**

1. Write an ALP to copy the value 12H into RAM memory location 50H to 5FH using
   a) Direct addressing mode
   b) Register indirect addressing mode without a loop, and
   c) Register indirect addressing mode with a loop

**Solution:**

a) Direct addressing mode

```
MOVA, #12h; load A with value 12h
MOV 50h, A ; copy A to RAM location 50h
MOV 51h, A ; copy A to RAM location 51h
MOV 52h, A ; copy A to RAM location 52h
MOV 53h, A ; copy A to RAM location 53h
MOV 5fh, A ; copy A to RAM location 5fh
```

b) Register indirect addressing mode without a loop, and

```
MOVA, #12h; load A with value 55h
MOVR0, #50h; load the pointer. R0 = 50h
MOV@R0, A; copy A to RAM location R0 points to
INCR0; increment pointer. Now R0 = 51h
MOV@R0, A; copy A to RAM location R0 points to
INCR0; increment pointer. Now R0 = 52h
MOV@R0, A; copy A to RAM location R0 points to
INCR0; increment pointer. Now R0 = 53h
MOV@R0, A; copy A to RAM location R0 points to
INCR0; increment pointer. Now R0 = 54h
MOV@R0, A; copy A to RAM location R0 points to
```

c) Register indirect addressing mode with a loop

```
MOVA, #12h; A = 12h
MOVR0, #50h; load pointer. R0 = 50h, RAM add.
MOVR2, #05; load counter, R2 = 5
AGAIN:
MOV@R0, A; copy 12A to RAM location R0 points to
```

```
INC R0; increment R0 pointer
DJNZR2, AGAIN; loop until counter = zero
```

**Question 2:**

Write an ALP to get the x value from PORT1 and send (x+5)*2 to PORT2, continuously

**Solution:**

```
ORG 0; //ROM locations starts from 0000H
MOV DPTR, #300H; //DPTR is a pointer starting from 300H ( why
choose this adress ? 300h)
MOV A, #0FFH; //Make P1 as an INPUT why choose #0ffh ?
MOV P1, A;
Back: MOV A, P1; //get x from P1
MOVC A, @A+DPTR; //get x2

MOV P2, A
SJMP Back
ORG 300H
Table: DB 0, 1, 4, 9, 16, 25, 36, 49, 64, 81; //where do these
values come from were they chosen arbitrary
END
```

**Question 3:**

UNIT V:

1. Design an 8051 based system to display “SRMIST” in 16x2 LCD display.

**Solution:**

```
MOV A,#38H // Use 2 lines and 5x7 matrix
ACALL CMND
MOV A,#0FH // LCD ON, cursor ON, cursor blinking ON
ACALL CMND
MOV A,#01H //Clear screen
ACALL CMND
MOV A,#06H //Increment cursor
ACALL CMND
MOV A,#82H //Cursor line one , position 2
ACALL CMND
MOV A,#3CH //Activate second line
ACALL CMND
MOV A,#49D
ACALL DISP
MOV A,#54D
ACALL DISP
MOV A,#88D
ACALL DISP
```

```
MOV A,#50D
ACALL DISP
MOV A,#32D
ACALL DISP
MOV A,#76D
ACALL DISP
MOV A,#67D
ACALL DISP
MOV A,#68D
ACALL DISP

MOV A,#0C1H //Jump to second line, position 1
ACALL CMND

MOV A,#67D
ACALL DISP
MOV A,#73D
ACALL DISP
MOV A,#82D
ACALL DISP
MOV A,#67D
ACALL DISP
MOV A,#85D
ACALL DISP
MOV A,#73D
ACALL DISP
MOV A,#84D
ACALL DISP
MOV A,#83D
ACALL DISP
MOV A,#84D
ACALL DISP
MOV A,#79D
ACALL DISP
MOV A,#68D
ACALL DISP
MOV A,#65D
ACALL DISP
MOV A,#89D
ACALL DISP

HERE: SJMP HERE

CMND: MOV P1,A
CLR P3.5
CLR P3.4
```

```
SETB P3.3
CLR P3.3
ACALL DELY
RET

DISP:MOV P1,A
SETB P3.5
CLR P3.4
SETB P3.3
CLR P3.3
ACALL DELY
RET

DELY: CLR P3.3
CLR P3.5
SETB P3.4
MOV P1,#0FFh
SETB P3.3
MOV A,P1
JB ACC.7,DELY

CLR P3.3
CLR P3.4
RET

END
```

*Thank You.*