Course Title: Peripherals, Interfacing and Embedded Systems Lab (CSE-4640)

Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT), Gazipur

Lab # 2

Interfacing with LEDs Lights through 8255A in MDA 8086 Kit.

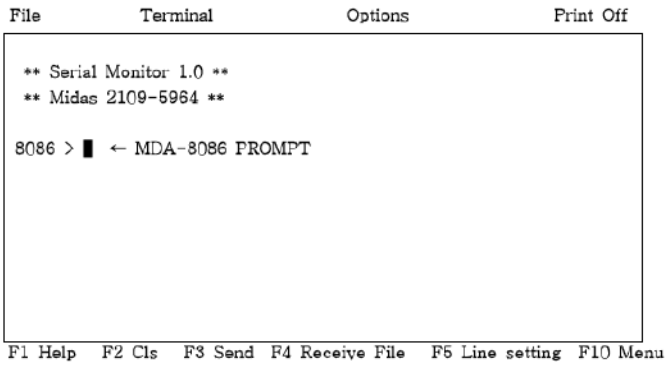
Objective:

To understand the basic interfacing concept through implementing a model for with LEDs through 8255A interfacing of MDA 8086 trainer Kit.

Theory:

- **Serial Monitor Mode**
  Serial monitor is the basic monitor program mode to have data communication between MDA-8086 and Computer (i.e., PC). To use serial monitor mode, move jumper P1 which located on the PCB like the following figure.

To connect the serial interface of the MDA-8086 with the Computer (PC) interface and run the WinComm program in PC to have the following screen shot after pressing RES key.



- **Operation Serial Monitor Command**
  User can only use command which stored at serial monitor using WinComm. Serial monitor can execute to command when user type the command and then press ENTER key. If there is no any command at serial monitor, error message will be displayed with bell sound and serial monitor prompt will be displayed again.
  8086 >?

  HELP Command

  E segment : offset.....: Enter Data To Memory

  D segment : offset length.....: Dump Memory Contents

  R [register name].....: Register Display & Change

  M address1, length, address2.....: Move Memory From 1 to 2

  F address, length, data.....: Fill Memory With Any Data

  L Return key.....: Program Down Load

  G segment : offset.....: Execute Program

  T.....: Program 1 step execute

- **Basic Command Syntax:**

1. Memory Modify Command

Syntax: E segment: offset
Purpose: This command is used to enter data to memory.
Example:

```
8086 > E 0000:1000
        0000:1000 FF ? 11
        0000:1001 FF ? 22
        0000:1002 FF ? 33
        0000:1003 FF ? 44
        0000:1004 FF ? 55
        0000:1005 FF ? / (Offset decrement)
        0000:1004 55 ?
```

2. Memory Display Command

Syntax: D segment: offset
Purpose: This command is used to display the data stored in memory.
Example:

```
8086> D 0000:1000
0000:1000  11  22 33 44  55 FF FF FF - FF FF FF FF FF FF FF FF
0000:1020  FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF
```

3. Display Register Command

Syntax: R

Purpose: The R command is used to display the 8086 processor registers

Example:

8086 > R

```
AX=0000  BX=0000  CX=0000  DX=0000
SP=0540   BP=0000   SI=0000   DI=0000
DS=0000   ES=0000  SS=0000   CS=0000
IP=1000    FL=0000 = . . . . . . . .
```

To change individual register:-

8086 > R AX

```
AX = 0000  1234
```

8086 > R BX

```
BX  = 0000  4567
```

8086 > R CX

```
CX = 0000  7788
```

8086 > R DX

```
DX =0000  1111
```

4. Program Download and Execute Command

```
 The L command moves object data in hexa format from an external devices to
memory.

8086 >L↵
   Down load start !!    ← ( Note : See section 5. Serial monitor experiment)

:14100000B83412BB7856B90010BA00208BF08BFBBD0030BC08
:0910140000408EDA8ED18EC0CCB2
:00
   OK Completed !!

☞  Set IP
8086 >R IP↵
IP=1000↵
8086 >T↵
        AX=1234  BX=4567  CX=7788  DX=1111
        SP=0540  BP=0000  SI=0000  DI=0000
        DS=0000  ES=0000  SS=0000  CS=0000
        IP=1003   FL=0100  = . . . t . . . . .
         ↓
      Next address

☞  Execute program command

       Segment  Offset
          ↓      ↓
8086 >G 0000:1000↵
Execute Address = 0000:1000
```

- **LEDs in MDA 8086 Kit**

There are 4 LEDs namely RED (L1), GREEN (L2), YELLOW (L3) and RED (L4) inside the MDA–8086 trainer kit and can be modeled to design a simpler application. This requires 8255 PPI ports which are already connected to the 4 LEDs internally. Through a code we can access these ports and provide binary or hex values to switch on the required LED (on or off). In order to turn a particular LED ON, a logical '1' should be provided to a particular port. Note that only Port B of 8255A PPI is used in the following example code to control the LEDs.

RED **L1**	GREEN **L2**
YELLOW **L3**	RED **L4**

Different ports of Programmable Peripheral Interface (PPI) 8255A is used for switching on LEDs, the address of the ports are:

Port A: 19h	Port C: 1Dh
Port B: 1Bh	Control Register: 1FH

To control the LEDs, Port B will be used for the value to select the LEDs and Port A and C will be set with constant values. In-case of Port B, pass a value of '11110001' to select the L1 LED and pass a value of '11110010' to select the L2 LED and so on.

Setup jumper cap of P2 as following:

- **Example:** Example code to illuminate LEDs with a sequence of L1, L2, L3 and L4.

```
CODE    SEGMENT
        ASSUME      CS:CODE, DS:CODE, ES:CODE, SS:CODE
PPIC_C  EQU     1FH             ; Control register  address
PPIC    EQU     1DH             ; Port C address
PPIB    EQU     1BH             ; Port B address
PPIA    EQU     19H             ; Port A address

        ORG     1000H
        MOV     AL, 10000000B   ; Control register initialization with a 8255 Mode set
        OUT     PPIC_C, AL
        MOV     AL, 11111111B
        OUT     PPIA, AL
        MOV     AL, 00000000B
        OUT     PPIC, AL
L1:     MOV     AL, 11110001B
L2:     OUT     PPIB, AL        ; Select a L1 LED with AL value and make it ON
        CALL    TIMER
        SHL     AL, 1
        TEST    AL, 00010000B   ; Perform Logical AND and set Zero Flag (ZF), SF, PF
        JNZ     L1
        OR      AL, 11110000B   ; Perform Logical OR and store the result in AL
        JMP     L2
        INT     3               ; Single-step interrupt
TIMER:  MOV     CX, 1
TIMER2:         PUSH    CX
                MOV     CX, 0
TIMER1:         NOP
                NOP
                NOP
                NOP
                LOOP    TIMER1
                POP     CX
                LOOP    TIMER2
                RET

CODE    ENDS
        END
```

Tasks to do:

1. Write a program to illuminate LEDs for the implementation of Simple Traffic Control Light Signaling model. Vehicles will be in STOP (with red color), WAIT (with yellow color) & GO (i.e., with green color) states for a moderately long time and vice-versa: