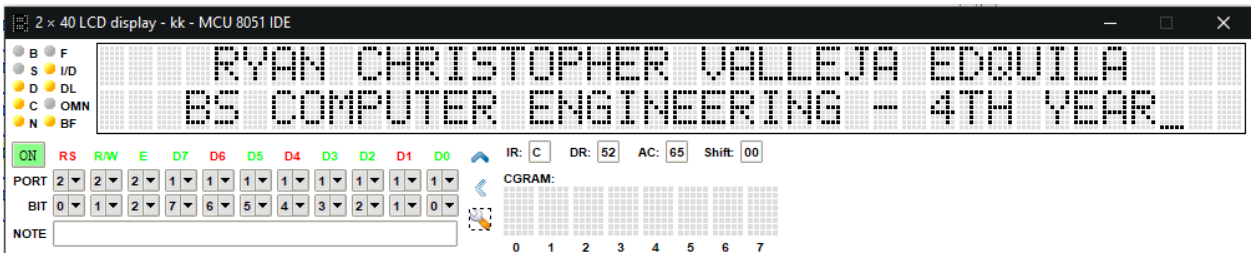
LCD DISPLAY

EEE184.1 M67 - Microprocessor, Microcontroller Systems and Design Laboratory

Ryan Christopher V. Edquila
Department of Electrical Engineering and Technology
Mindanao State University - Iligan Institute of Technology
Iligan City, Philippines
ryanchristopoher.edquila@g.msuiit.edu.ph

This activity displays, sends commands and data to control the seven-segment display, and continuously displays the message "RYAN CHRISTOPHER VALLEJA EDQUILA - 4TH YEAR" on the display. It utilizes subroutines and control signals to ensure proper communication between the microcontroller and the display. The program remains in an infinite loop to continuously display the message.



Code

The provided code is responsible for displaying a specific message on a seven-segment display using a 8051 microcontroller in a simulated environment.

```
ORG 0000H

;
Complete Name. Course/Year

MOV A, #38H
ACALL COMMAND
MOV A, #0EH
ACALL COMMAND
MOV A, #01H                ;CLEAR DISPLAY
ACALL COMMAND
MOV A, #06H
ACALL COMMAND
MOV A, #84H
ACALL COMMAND

MOV A, #'R'
ACALL DATA_DISPLAY
MOV A, #'Y'
ACALL DATA_DISPLAY
MOV A, #'A'
ACALL DATA_DISPLAY
MOV A, #'N'
ACALL DATA_DISPLAY
MOV A, #' '
ACALL DATA_DISPLAY

MOV A, #'C'
ACALL DATA_DISPLAY
MOV A, #'H'
ACALL DATA_DISPLAY
MOV A, #'R'
ACALL DATA_DISPLAY
MOV A, #'I'
ACALL DATA_DISPLAY
MOV A, #'S'
ACALL DATA_DISPLAY
MOV A, #'T'
ACALL DATA_DISPLAY
MOV A, #'O'
ACALL DATA_DISPLAY
```

```
MOV A, #'P'
ACALL DATA_DISPLAY
MOV A, #'H'
ACALL DATA_DISPLAY
MOV A, #'E'
ACALL DATA_DISPLAY
MOV A, #'R'
ACALL DATA_DISPLAY
MOV A, #' '
ACALL DATA_DISPLAY

MOV A, #'V'
ACALL DATA_DISPLAY
MOV A, #'A'
ACALL DATA_DISPLAY
MOV A, #'L'
ACALL DATA_DISPLAY
MOV A, #'L'
ACALL DATA_DISPLAY
MOV A, #'E'
ACALL DATA_DISPLAY
MOV A, #'J'
ACALL DATA_DISPLAY
MOV A, #'A'
ACALL DATA_DISPLAY
MOV A, #' '
ACALL DATA_DISPLAY

MOV A, #'E'
ACALL DATA_DISPLAY
MOV A, #'D'
ACALL DATA_DISPLAY
MOV A, #'Q'
ACALL DATA_DISPLAY
MOV A, #'U'
ACALL DATA_DISPLAY
MOV A, #'I'
ACALL DATA_DISPLAY
MOV A, #'L'
ACALL DATA_DISPLAY
MOV A, #'A'
ACALL DATA_DISPLAY


MOV A, #0C0H
ACALL COMMAND
MOV A, #0C3H
ACALL COMMAND
MOV A, #'B'
ACALL DATA_DISPLAY
MOV A, #'S'
ACALL DATA_DISPLAY
MOV A, #' '
ACALL DATA_DISPLAY

MOV A, #'C'
ACALL DATA_DISPLAY
MOV A, #'O'
ACALL DATA_DISPLAY
MOV A, #'M'
ACALL DATA_DISPLAY
```

```
MOV A, #'P'
ACALL DATA_DISPLAY
MOV A, #'U'
ACALL DATA_DISPLAY
MOV A, #'T'
ACALL DATA_DISPLAY
MOV A, #'E'
ACALL DATA_DISPLAY
MOV A, #'R'
ACALL DATA_DISPLAY
MOV A, #' '
ACALL DATA_DISPLAY

MOV A, #'E'
ACALL DATA_DISPLAY
MOV A, #'N'
ACALL DATA_DISPLAY
MOV A, #'G'
ACALL DATA_DISPLAY
MOV A, #'I'
ACALL DATA_DISPLAY
MOV A, #'N'
ACALL DATA_DISPLAY
MOV A, #'E'
ACALL DATA_DISPLAY
MOV A, #'E'
ACALL DATA_DISPLAY
MOV A, #'R'
ACALL DATA_DISPLAY
MOV A, #'I'
ACALL DATA_DISPLAY
MOV A, #'N'
ACALL DATA_DISPLAY
MOV A, #'G'
ACALL DATA_DISPLAY

MOV A, #' '
ACALL DATA_DISPLAY
MOV A, #'-'
ACALL DATA_DISPLAY
MOV A, #' '
ACALL DATA_DISPLAY

MOV A, #'4'
ACALL DATA_DISPLAY
MOV A, #'T'
ACALL DATA_DISPLAY
MOV A, #'H'
ACALL DATA_DISPLAY
MOV A, #' '

ACALL DATA_DISPLAY
MOV A, #'Y'
ACALL DATA_DISPLAY
MOV A, #'E'
ACALL DATA_DISPLAY
MOV A, #'A'
ACALL DATA_DISPLAY
MOV A, #'R'
ACALL DATA_DISPLAY
```

```
HERE: SJMP HERE

COMMAND:ACALL READY
        MOV P1, A
        CLR P2.0
        CLR P2.1
        SETB P2.2
        CLR P2.2
        RET

DATA_DISPLAY:   ACALL READY
                MOV P1, A
                SETB P2.0
                CLR P2.1
                SETB P2.2
                CLR P2.2
                RET

READY:  SETB P1.7
        CLR P2.0 ; PUT IN DATA REGISTER
        SETB P2.1

BACK:   SETB P2.2
        CLR P2.2
        JB P1.7, BACK
        RET

END
```