**DIGITAL CLOCK ON LCD**

**AIM:**

To write an assembly language program to display digital clock on LCD with using Proteus.

**SOFTWARES REQUIRED:**

* Proteus software

**PROGRAM:**

ORG 0000H ; Start address of the program

MOV R7, #00H ; Initialize hours (HH)

MOV R6, #00H ; Initialize minutes (MM)

MOV R5, #00H ; Initialize seconds (SS)

ACALL INIT\_LCD ; Initialize the LCD

MAIN\_LOOP:

ACALL UPDATE\_LCD ; Update the time on the LCD

ACALL DELAY\_1\_SEC ; Wait for 1 second

ACALL INCREMENT\_TIME ; Increment time (HH:MM:SS)

SJMP MAIN\_LOOP ; Repeat the process

; Subroutine to initialize the LCD

INIT\_LCD:

MOV A, #38H

ACALL CMD\_WRITE ; 8-bit mode, 2 lines, 5x7 matrix

ACALL DELAY\_SHORT

MOV A, #0CH

ACALL CMD\_WRITE ; Display ON, Cursor OFF

ACALL DELAY\_SHORT

MOV A, #06H

ACALL CMD\_WRITE ; Auto-increment cursor

ACALL DELAY\_SHORT

MOV A, #01H

ACALL CMD\_WRITE ; Clear display

ACALL DELAY\_SHORT

RET

; Subroutine to increment time

INCREMENT\_TIME:

INC R5 ; Increment seconds (SS)

CJNE R5, #60, DONE\_SEC ; If seconds < 60, continue

MOV R5, #00H ; Reset seconds to 00

INC R6 ; Increment minutes (MM)

CJNE R6, #60, DONE\_SEC ; If minutes < 60, continue

MOV R6, #00H ; Reset minutes to 00

INC R7 ; Increment hours (HH)

CJNE R7, #24, DONE\_SEC ; If hours < 24, continue

MOV R7, #00H ; Reset hours to 00

DONE\_SEC:

RET

; Subroutine to update the LCD with the current time

UPDATE\_LCD:

MOV A, #80H

ACALL CMD\_WRITE ; Move cursor to the first line of the LCD

MOV A, R7 ; Load hours (HH) into accumulator

ACALL DISPLAY\_TWO\_DIGIT ; Display hours (HH)

ACALL DISPLAY\_COLON ; Display ':'

MOV A, R6 ; Load minutes (MM) into accumulator

ACALL DISPLAY\_TWO\_DIGIT ; Display minutes (MM)

ACALL DISPLAY\_COLON ; Display ':'

MOV A, R5 ; Load seconds (SS) into accumulator

ACALL DISPLAY\_TWO\_DIGIT ; Display seconds (SS)

RET

; Subroutine to display two-digit numbers on the LCD

DISPLAY\_TWO\_DIGIT:

MOV B, #10 ; Divide the value in A by 10

DIV AB ; Quotient in A (tens), remainder in B (ones)

ADD A, #30H ; Convert tens digit to ASCII

ACALL DISPLAY\_CHAR ; Display the tens digit

MOV A, B ; Move the remainder (ones digit) to A

ADD A, #30H ; Convert ones digit to ASCII

ACALL DISPLAY\_CHAR ; Display the ones digit

RET

; Subroutine to display colon ':' on the LCD

DISPLAY\_COLON:

MOV A, #3AH ; ASCII value of ':'

ACALL DISPLAY\_CHAR ; Display ':'

RET

; Subroutine to display a character on the LCD

DISPLAY\_CHAR:

MOV P2, A ; Send ASCII character to data pins (P2 connected to D0-D7 of LCD)

SETB P3.2 ; Set RS to 1 (data register)

CLR P3.3 ; Set RW to 0 (write mode)

SETB P3.4 ; Set E to 1 (Enable high)

NOP ; Small delay

CLR P3.4 ; Set E to 0 (Enable low)

ACALL DELAY\_SHORT ; Short delay after sending character

RET

; Subroutine to write command to the LCD

CMD\_WRITE:

MOV P2, A ; Send command to data pins (P2 connected to D0-D7 of LCD)

CLR P3.2 ; Set RS to 0 (command register)

CLR P3.3 ; Set RW to 0 (write mode)

SETB P3.4 ; Set E to 1 (Enable high)

NOP ; Small delay

CLR P3.4 ; Set E to 0 (Enable low)

ACALL DELAY\_SHORT ; Short delay after sending command

RET

; Short delay for LCD commands and data

DELAY\_SHORT:

MOV R0, #250 ; Adjust this value for a short delay

DELAY\_SHORT\_LOOP:

DJNZ R0, DELAY\_SHORT\_LOOP

RET

; Subroutine for 1-second delay

DELAY\_1\_SEC:

MOV R3, #50 ; Outer loop for delay

DELAY\_LOOP:

MOV R4, #255 ; Inner loop for delay

DELAY\_LOOP\_INNER:

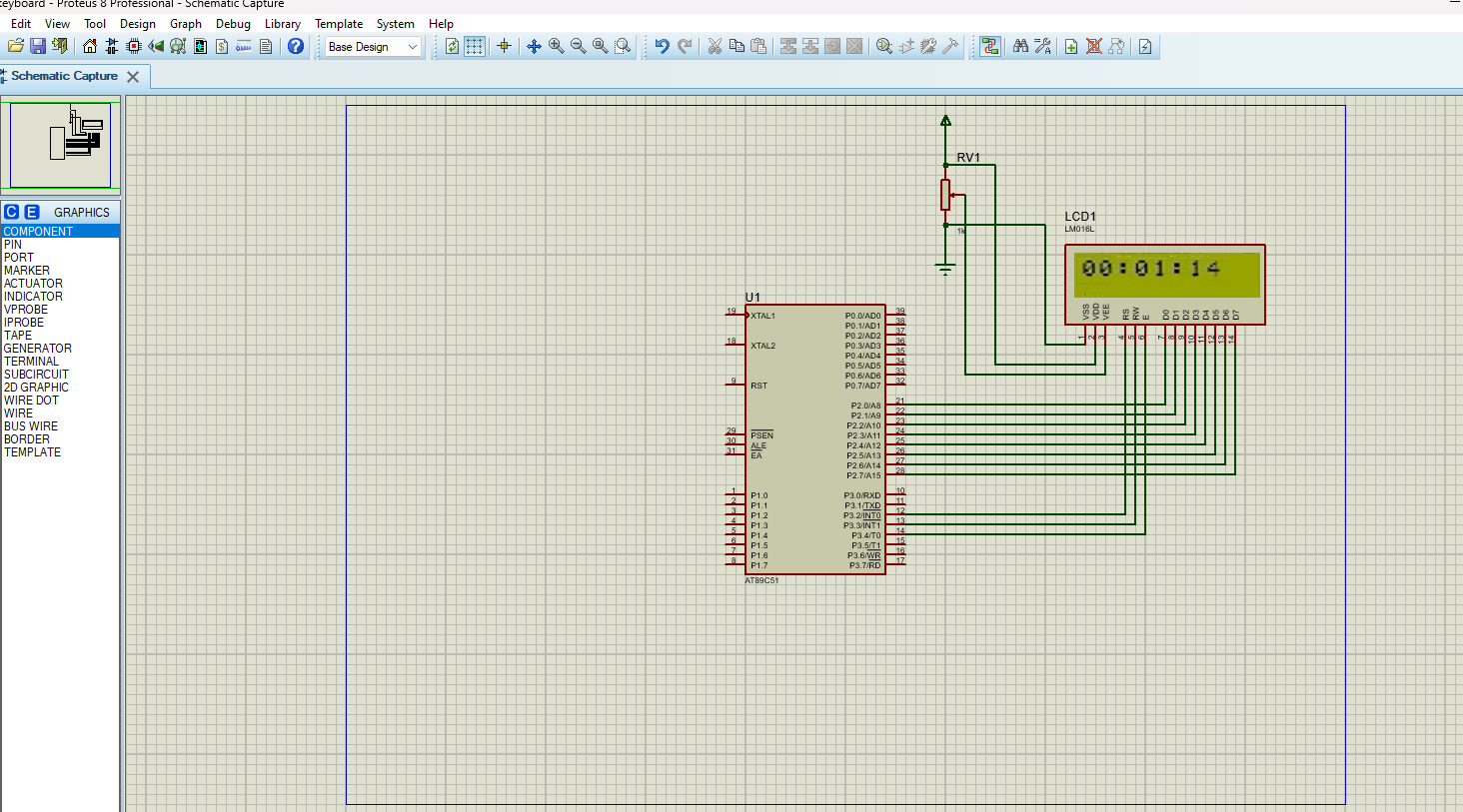
DJNZ R4, DELAY\_LOOP\_INNER

DJNZ R3, DELAY\_LOOP

RET

END

**CIRCUIT DIAGRAM:**



**OUTPUT:**

* When this program is run, the LCD will display the current time in the format HH:MM.
* Every second, the display will update to increment the seconds value.
* After reaching 59 seconds, the seconds will reset to 00, and the minutes will increment.
* Similarly, when the minutes reach 59 and increment again, they will reset to 00, and the hours will increment.
* The hours will increment from 00 to 23 in a 24-hour format. When the hours reach 23 and the next second occurs, the hours, minutes, and seconds will all reset to 00:00:00.

**RESULT:**

Thus, the assembly language program to display digital clock on LCD with using Proteus was executed.