DIGITAL CLOCK ON LCD

NAME: S.G.DEVSACHIN
REG NO: 192111088
COURSE CODE: ECA1487, Embedded Systems for IOT Applications

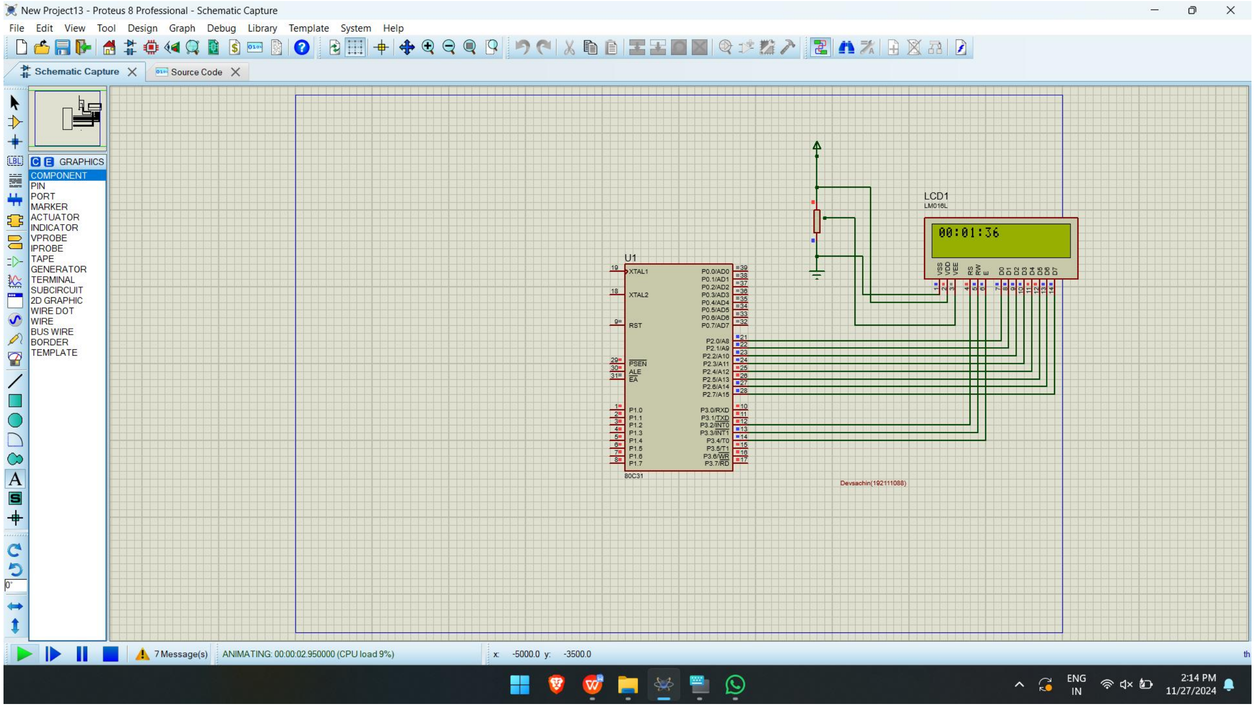
PROGRAM:

```
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
```

OUTPUT:



New Project13 - Proteus 8 Professional - Schematic Capture

00:01:36

7 Message(s)

ANIMATING: 00:00:02.950000 (CPU load 9%)

x: -5000.0 y: -3500.0