

## Corrected codes of Proteus 3 Practicals

### LCD Code :

```
#include <reg51.h> // Include register definition file for AT89C51
#define LCD P0 // Define LCD data port
sbit RS = P1^0; // Register Select pin
sbit RW = P1^1; // Read/Write pin
sbit E = P1^2; // Enable pin
void Delay(unsigned int ms);
void LCD_Command(unsigned char cmd);
void LCD_Char(unsigned char Data);
void LCD_Init(void);
void LCD_String(char *str);
void main() {
    LCD_Init(); // Initialize LCD
    LCD_String("ISQUAREIT"); // Display message
    while (1); // Loop forever
}

void LCD_Init(void) {
    Delay(20); // Wait for more than 15 ms after power on
    LCD_Command(0x38); // 2 Lines, 5x7 Matrix
    Delay(5); // Wait for a bit after this command
    LCD_Command(0x0C); // Display ON, Cursor OFF
    LCD_Command(0x06); // Increment cursor
    LCD_Command(0x01); // Clear display
    Delay(2); // Wait for clear command to complete
}

void LCD_Command(unsigned char cmd) {
    RS = 0; // Command mode
    RW = 0; // Write mode
    LCD = cmd; // Send command
    E = 1; // Enable high
    Delay(1); // Delay for processing
    E = 0; // Enable low
    Delay(5); // Wait for command to be executed
```

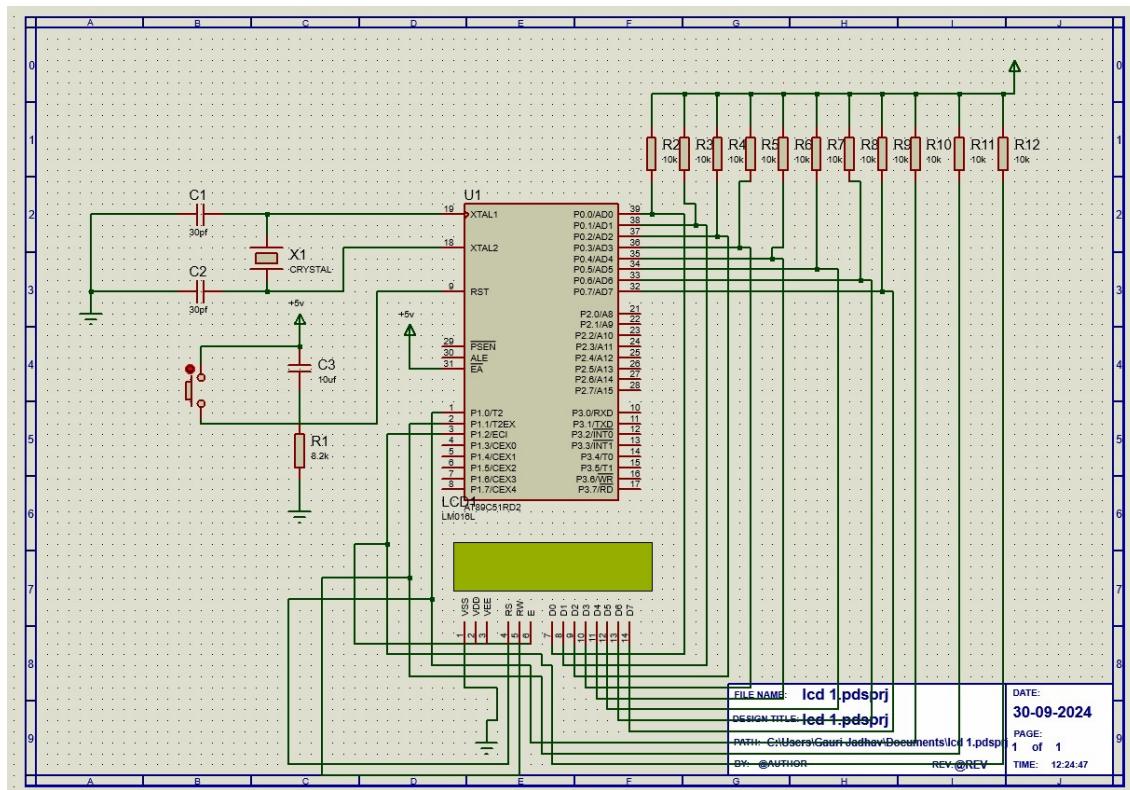
```
}
```

```
void LCD_Char(unsigned char Data) {
    RS = 1; // Data mode
    RW = 0; // Write mode
    LCD = Data; // Send data
    E = 1; // Enable high
    Delay(1); // Delay for processing
    E = 0; // Enable low
    Delay(5); // Wait for data to be written
}
```

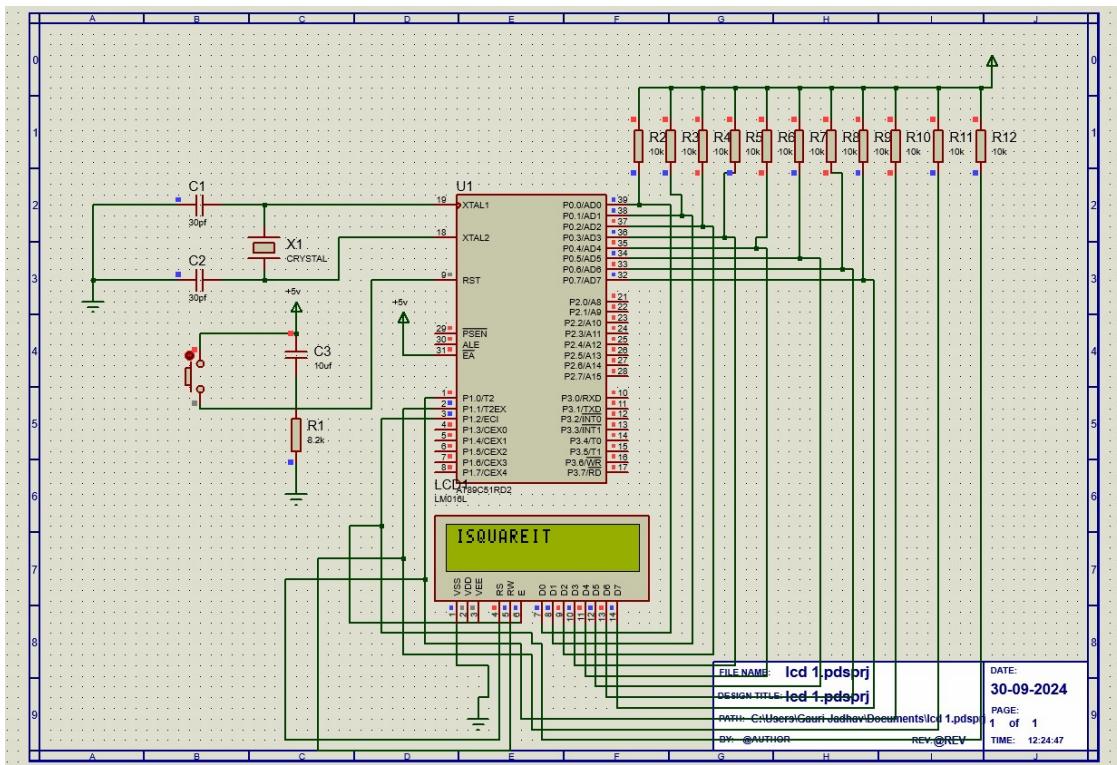
```
void LCD_String(char *str) {
    while (*str) {
        LCD_Char(*str++);
    }
}
```

```
void Delay(unsigned int ms) {
    unsigned int i, j;
    for (i = 0; i < ms; i++) {
        for (j = 0; j < 1275; j++); // Approx delay
    }
}
```

## Interfacing Diagram:



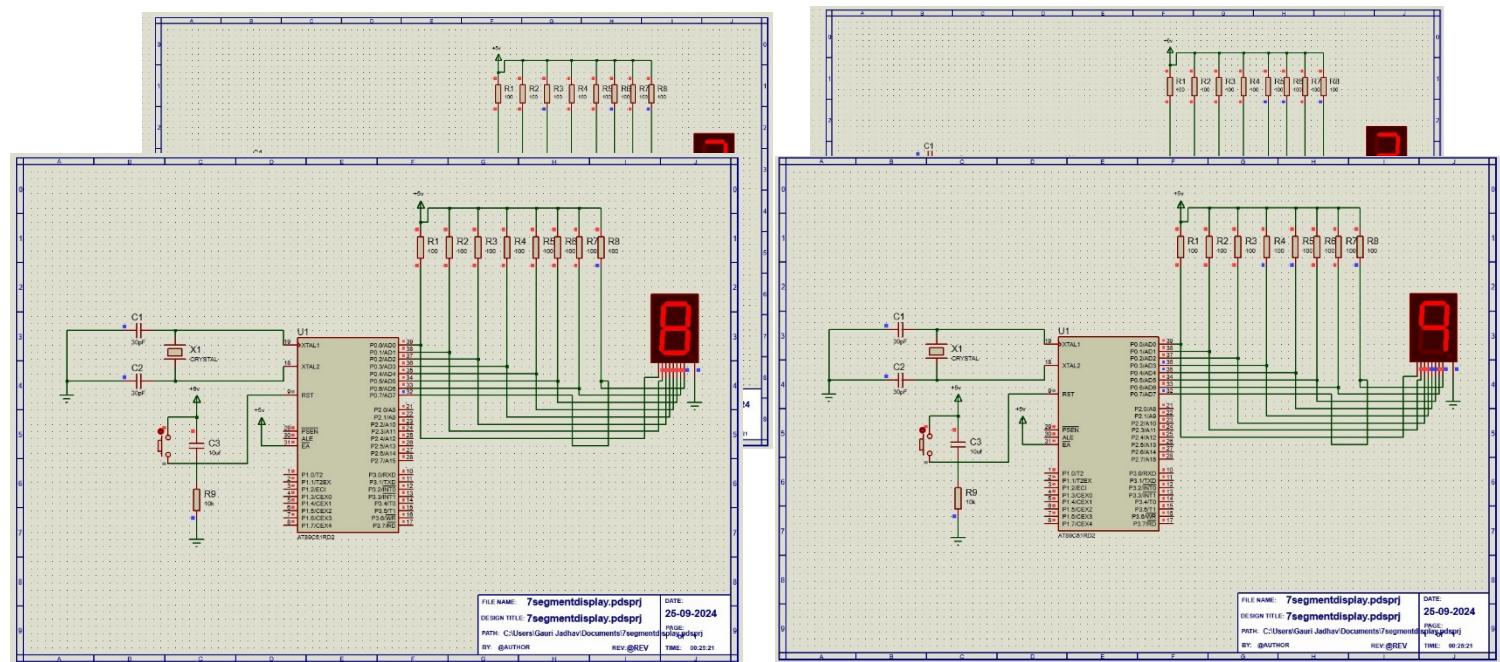
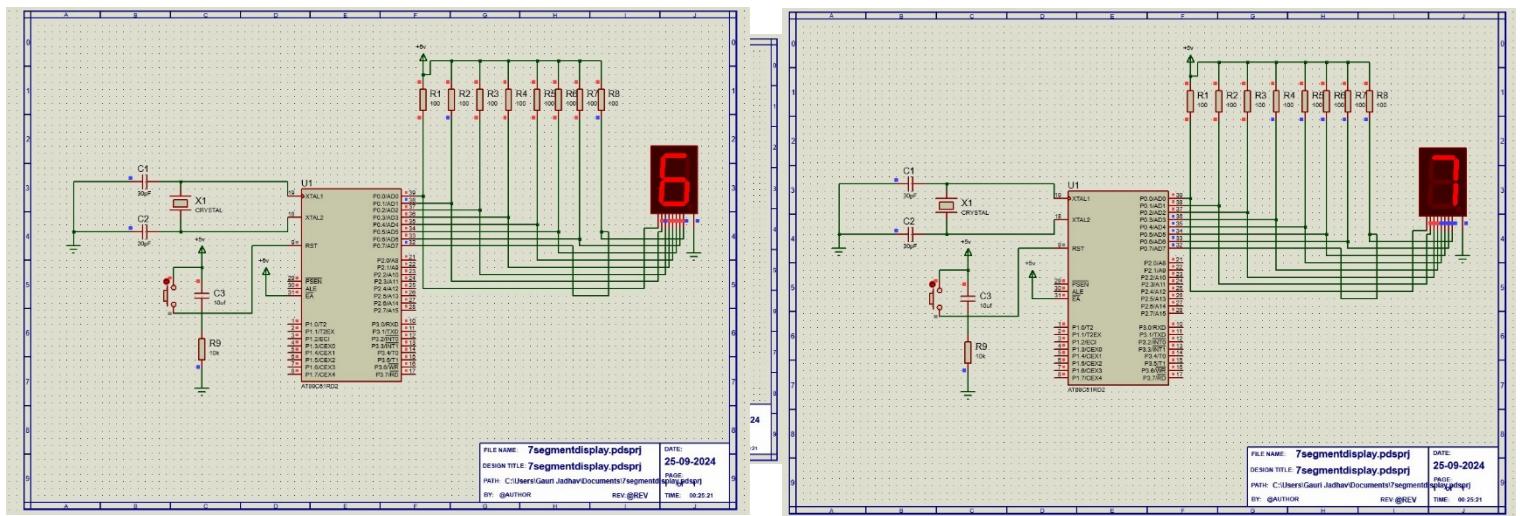
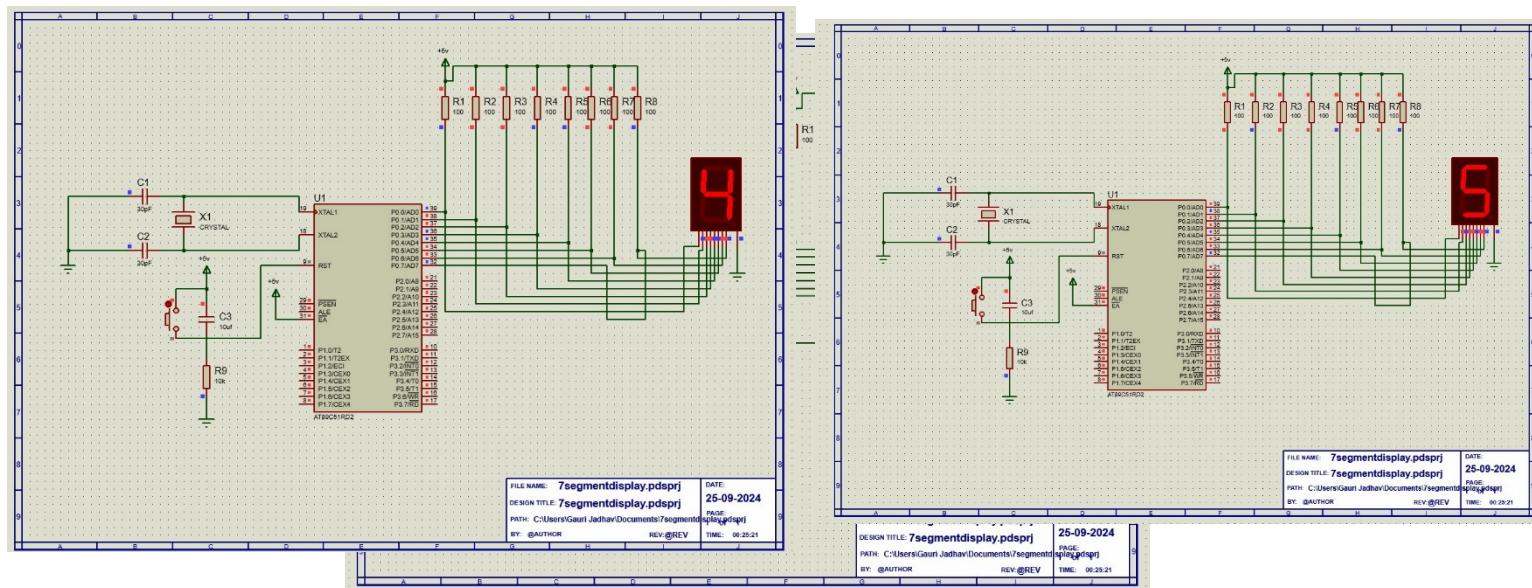
Ouput:



## 7 Segment Code :

```
#include <reg51.h>
void delay (int k);
void main()
{
unsigned char i;
unsigned char arr[10] = {0x3f, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F,
0x67};
P0 = 0x00;
while (1)
{
for (i=0;i<10;i++)
{
P0 = arr[i];
delay(100);
}
}
}
void delay (int k)
{
int i, j;
```

for (i=0; i<k; i++)

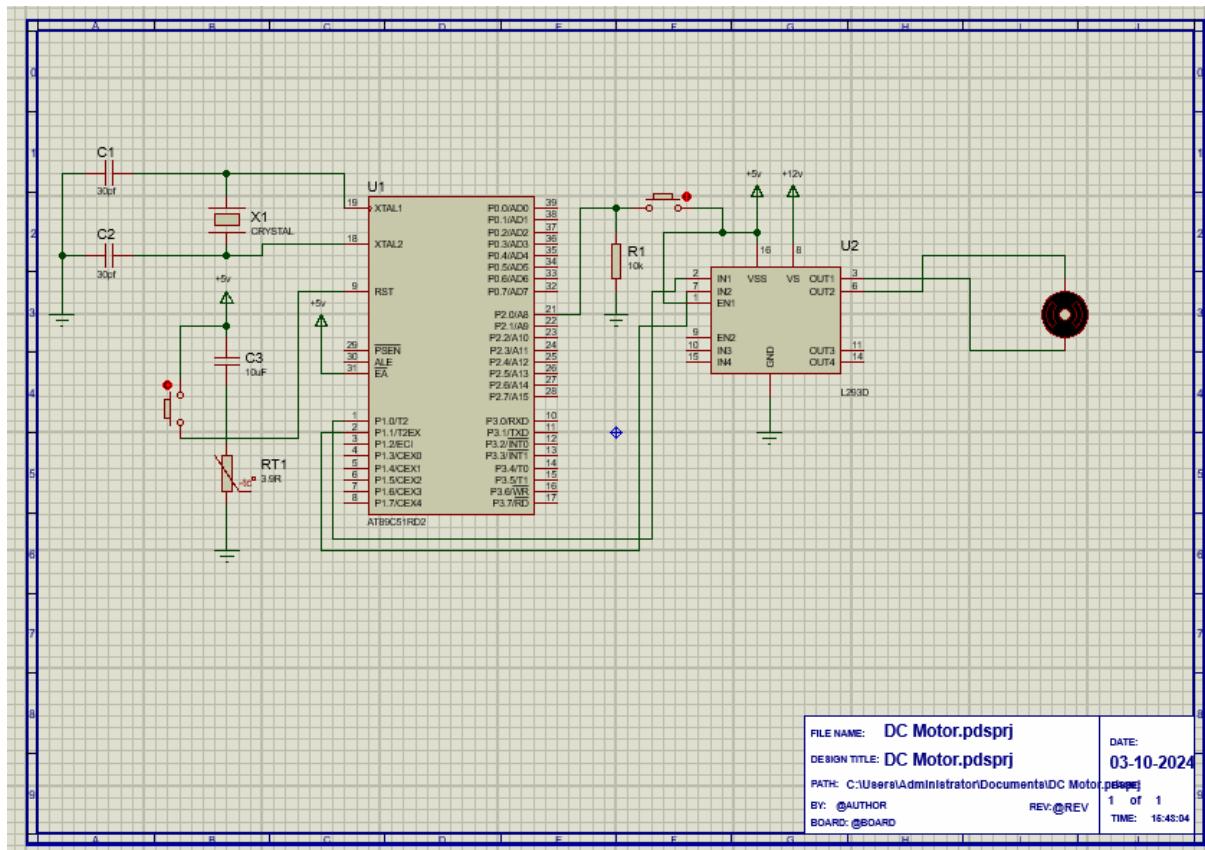


## **DC Motor :**

```
#include <reg51.h>
sbit motor_pin1 = P1^0;
sbit motor_pin2 = P1^1;
sbitbutton_reverse = P2^0;
bit motor_direction = 0;
void delay(unsigned int time) {
    unsigned int i, j;
    for(i = 0; i < time; i++) {
        for(j = 0; j < 1275; j++);
    }
}
void low_speed_delay(unsigned int time) {
    unsigned int i, j;
    for(i = 0; i < time; i++) {
        for(j = 0; j < 100000; j++); // Increase delay significantly to slow
down motor
    }
}
void main() {
    motor_pin1 = 1;
    motor_pin2 = 0;
    while(1) {
        if(button_reverse == 0) {
            delay(50);
            while(button_reverse == 0);
        }
    }
}
```

```
motor_direction= !motor_direction;
if(motor_direction == 1) {
    motor_pin1 = 0;
    motor_pin2 = 1;
} else {
    motor_pin1 = 1;
    motor_pin2 = 0;
}
delay(50);
}
motor_pin1 = 1;
motor_pin2 = 0;
low_speed_delay(10); // Motor ON for a long time to achieve low
speed
motor_pin1 = 0;
motor_pin2 = 0;    // Motor OFF to simulate slow rotation
low_speed_delay(10); // OFF delay to reduce effective RPM
}}
```

Interfacing Diagram :



**Output:**

