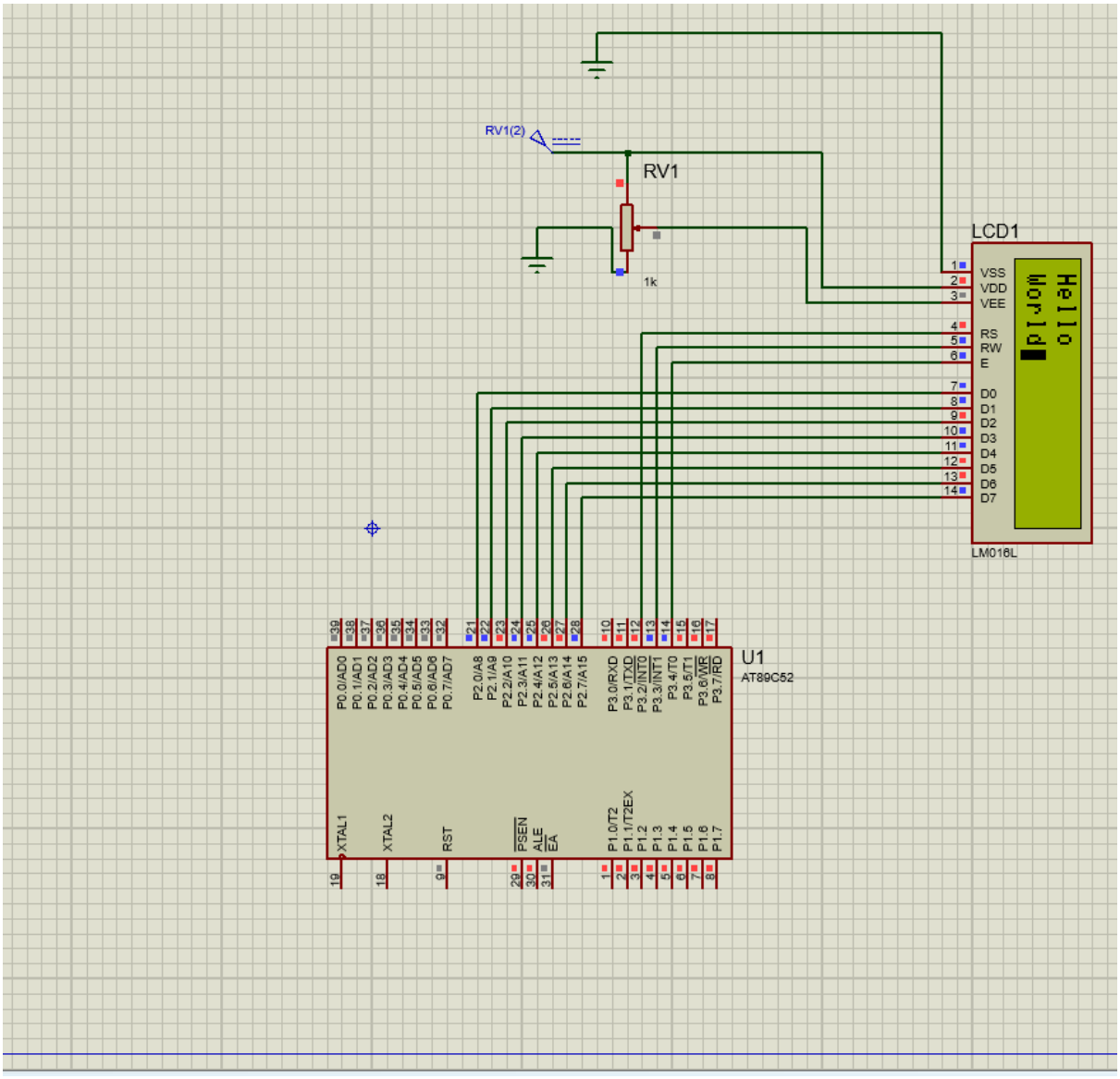
OUTPUT:



Code:

```
#include <reg51.h> // Header file for 89C52
// Lm016L
//Control Signals for LCD
sbit RS = P3^2;
sbit RW = P3^3;
sbit EN = P3^4;

#define LCD_PORT P2


// Delay function in milliseconds
```

```
void delay(unsigned int time)
{
  unsigned int i, j;
  for(i=0; i<time; i++)
    for(j=0; j<1275; j++);
}

// Send Command to Lcd

void sendCommand(unsigned char command){
    LCD_PORT = command;
    RW = 0;
    RS = 0;
    EN = 1;
        delay(1);
    EN = 0;

}

// send data to lcd to display
void sendData(unsigned char dat){

    LCD_PORT = dat;
    RW = 0;
    RS = 1;
    EN = 1;
        delay(1);
    EN = 0;

}

// initialize lcd
void init(){
    sendCommand(0x38); //2 lines ,5x7
    delay(12);
    sendCommand(0x0F); //display on,cursor on
    delay(12);
}

void run(char* message1,char* message2){
    unsigned short int i;
    while(1){
        sendCommand(0x01); //clear screen
            delay(10);
        sendCommand(0x80); //1st line
            delay(10);
        i=0;
```

```
        while(message1[i] != '\0'){
            sendData(message1[i]); //send Data
            i++;
            delay(30);
        };

        i=0;

        sendCommand(0xC0); //2nd line
        delay(10);

        while(message2[i] != '\0'){
            sendData(message2[i]); //send Data
            i++;
            delay(30);
        };

    }

}


void main() {
    init();
    run("Hello","World");
}
```

CODE Timer

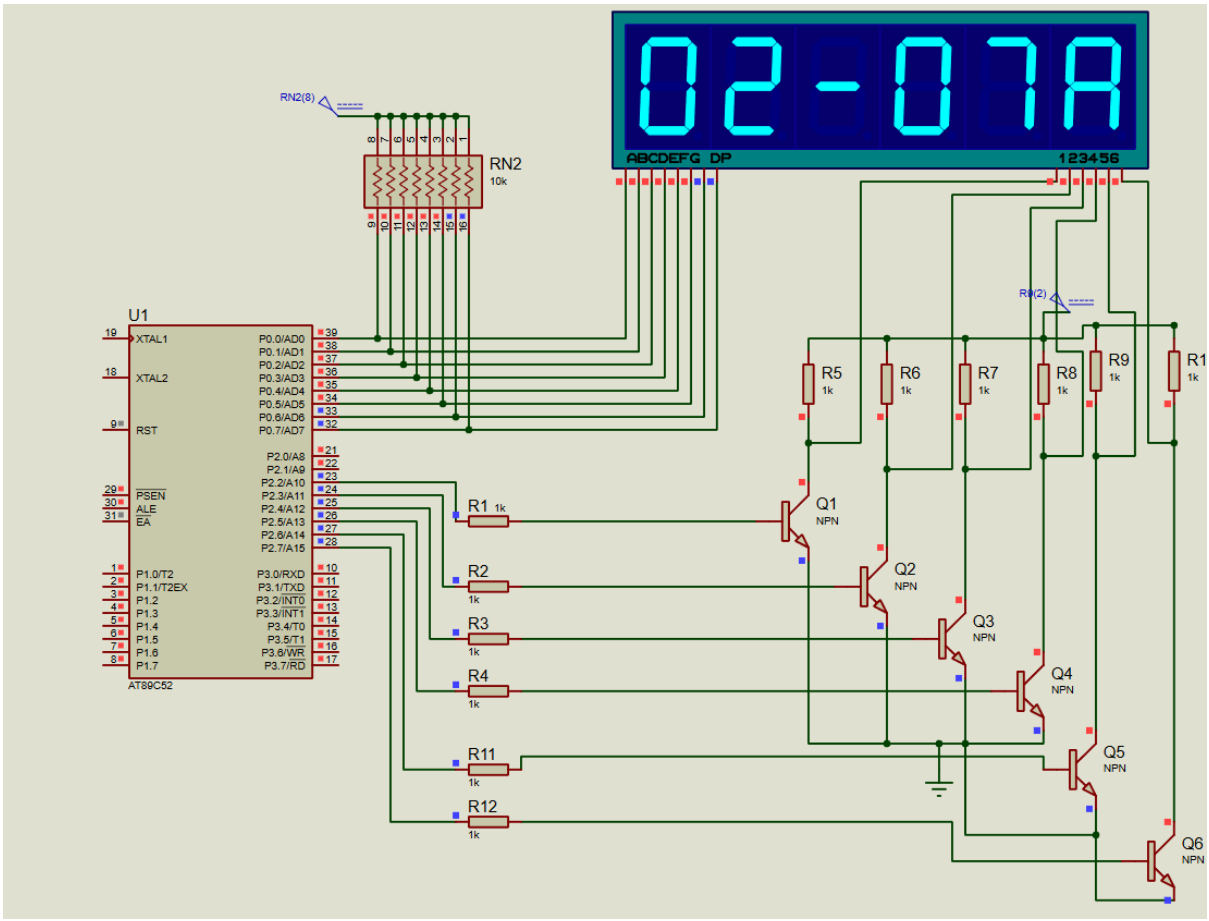
#include <reg51.h> // Header file for 89C52 Timer

```
// Define segment LEDs
sbit LED0 = P0^0;
sbit LED1 = P0^1;
sbit LED2 = P0^2;
sbit LED3 = P0^3;
sbit LED4 = P0^4;
sbit LED5 = P0^5;
sbit LED6 = P0^6;
sbit LED7 = P0^7;

// Define control pins for the 7-segment displays
sbit CON0 = P2^2;
sbit CON1 = P2^3;
sbit CON2 = P2^4;
sbit CON3 = P2^5;
sbit CON4 = P2^6;
sbit CON5 = P2^7;

    unsigned int noOfTimes = 20;
    unsigned char partOfDay = 11;
    unsigned char time[] = {2, 7,0};

// Digit patterns for a common cathode 7-segment display (0-9)
unsigned char digits[] = {
    0x3F, // 0
    0x06, // 1
    0x5B, // 2
    0x4F, // 3
    0x66, // 4
    0x6D, // 5
    0x7D, // 6
    0x07, // 7
    0x7F, // 8
    0x6F,// 9
    0x40, //:,index 10
    0x77, //index 11 ,A
    0x73 //index 12,P
};

void delay_three_ms(){
  TH1=0xF4;
  TL1=0x48;
  TR1=1;
  while (TF1 != 0);
```

```
  TR1=0;
  TF1=0;
}

// Delay function
void delay_ms() {//default 50ms
  TL0 =0x00;
  TH0 =0x4c;
  TR0 =1;
}

void isr_timer() interrupt 1 {
    delay_ms();
    if(noOfTimes == 0) {
        noOfTimes = 20;
        time[2]++;
        if (time[2] < 60) return;
        time[1]++;
        time[2] = 0;
        if (time[1] < 60) return;
        time[1] = 0;
        time[0]++;
        if(time[0]<12) return;
        time[0]=0;
        partOfDay = partOfDay==11 ? 12 :11;
    }else{
        noOfTimes--;
        return;
    }
}


void init(){
  IE = 0x82;
  TMOD =0x01;
  delay_ms();
}

// Function to display a digit on a specific 7-segment display
void display_digit(unsigned char digit, unsigned char position) {
    // Clear all control pins
    CON0 = 0;
    CON1 = 0;
    CON2 = 0;
    CON3 = 0;
    CON4 = 0;
    CON5 = 0;

    // Output the segment data
```

```
    P0 = digits[digit];

    // Activate the selected display
    if (position == 0) CON0 = 1;
    if (position == 1) CON1 = 1;
    if (position == 2) CON2 = 1;
    if (position == 3) CON3 = 1;
    if (position == 4) CON4 = 1;
    if (position == 5) CON5 = 1;

    delay_three_ms();
}

void main() {
  int i = 80;
  init();
    while (1) {
        display_digit((time[0]/10), 0);
        display_digit((time[0] % 10), 1);
        display_digit(10, 2);
        display_digit((time[1]/10), 3);
        display_digit((time[1]%10), 4);
        display_digit(partOfDay,5);
    }
}
```

OUTPUT

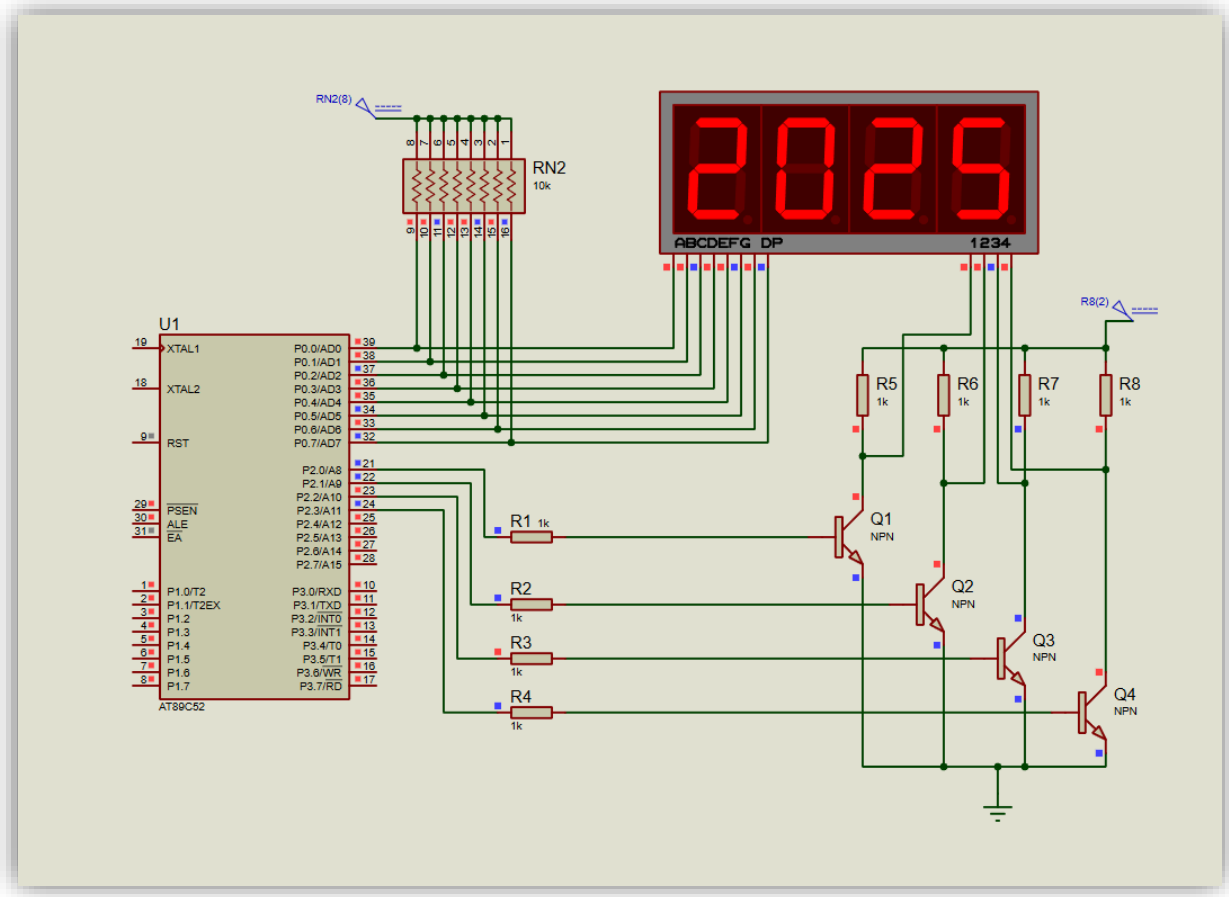

02-07A
ABCDEFG DP
123456
RN2(8)
RN2
10k
U1
AT89C52
XTAL1
XTAL2
RST
PSEN
ALE
EA
P1.0/T2
P1.1/T2EX
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
R1 1k
R2 1k
R3 1k
R4 1k
R11 1k
R12 1k
R5 1k
R6 1k
R7 1k
R8 1k
R9 1k
Q1 NPN
Q2 NPN
Q3 NPN
Q4 NPN
Q5 NPN
Q6 NPN

Code:

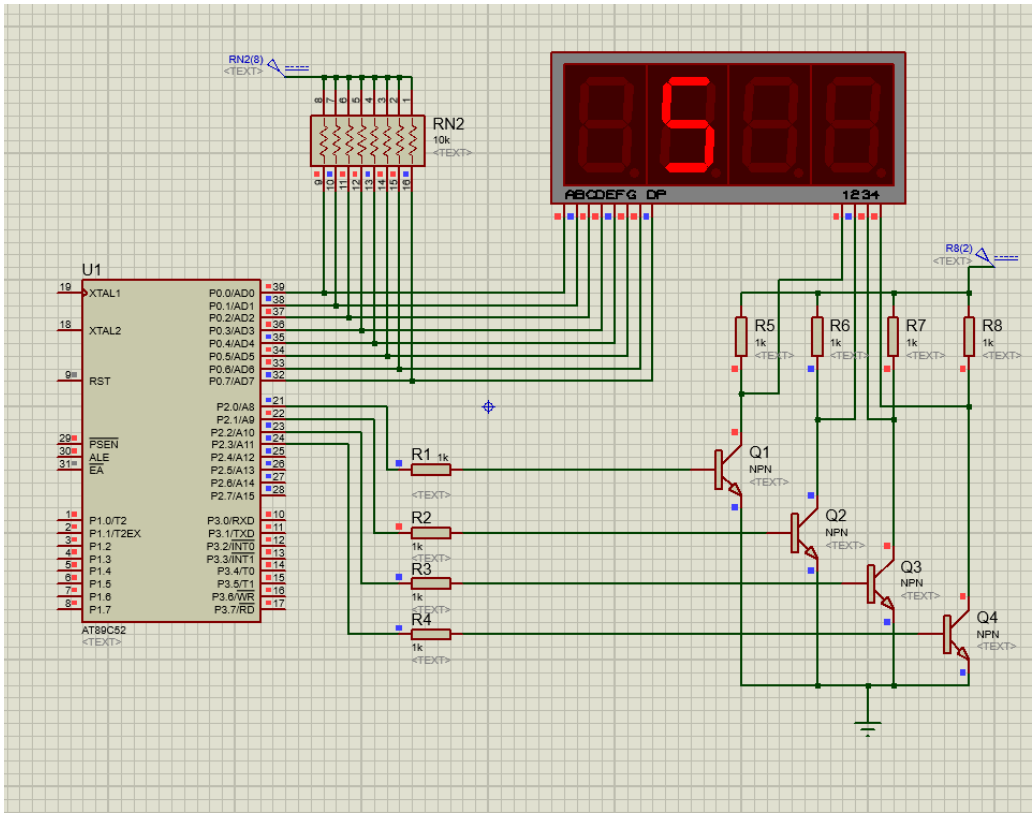
```
#include <reg51.h> // Header file for 89C52

// Define segment LEDs
sbit LED0 = P0^0;
sbit LED1 = P0^1;
sbit LED2 = P0^2;
sbit LED3 = P0^3;
sbit LED4 = P0^4;
sbit LED5 = P0^5;
sbit LED6 = P0^6;
sbit LED7 = P0^7;

// Define control pins for the 7-segment displays
sbit CON0 = P2^0;
sbit CON1 = P2^1;
sbit CON2 = P2^2;
sbit CON3 = P2^3;

// Digit patterns for a common cathode 7-segment display (0-9)
unsigned char digits[] = {
    0x3F, // 0
    0x06, // 1
    0x5B, // 2
    0x4F, // 3
    0x66, // 4
    0x6D, // 5
    0x7D, // 6
    0x07, // 7
    0x7F, // 8
    0x6F// 9
    };

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

// Function to display a digit on a specific 7-segment display
void display_digit(unsigned char digit, unsigned char position) {
    // Clear all control pins
    CON0 = 0;
```

```
    CON1 = 0;
    CON2 = 0;
    CON3 = 0;

    // Output the segment data
    P0 = digits[digit];

    // Activate the selected display
switch (position) {
        case 0: CON0 = 1; break;
        case 1: CON1 = 1; break;
        case 2: CON2 = 1; break;
        case 3: CON3 = 1; break;
        default: break;
    }


    // Small delay for persistence
    delay_ms(1);
}

void main() {
    unsigned char year[] = {2, 0, 2, 5}; // Digits of the year 2025

    while (1) {
        // Display each digit on the corresponding 7-segment display
        display_digit(year[0], 0); // Display '2' on the first display
        display_digit(year[1], 1); // Display '0' on the second display
        display_digit(year[2], 2); // Display '2' on the third display
        display_digit(year[3], 3); // Display '5' on the fourth display
    }
}
```

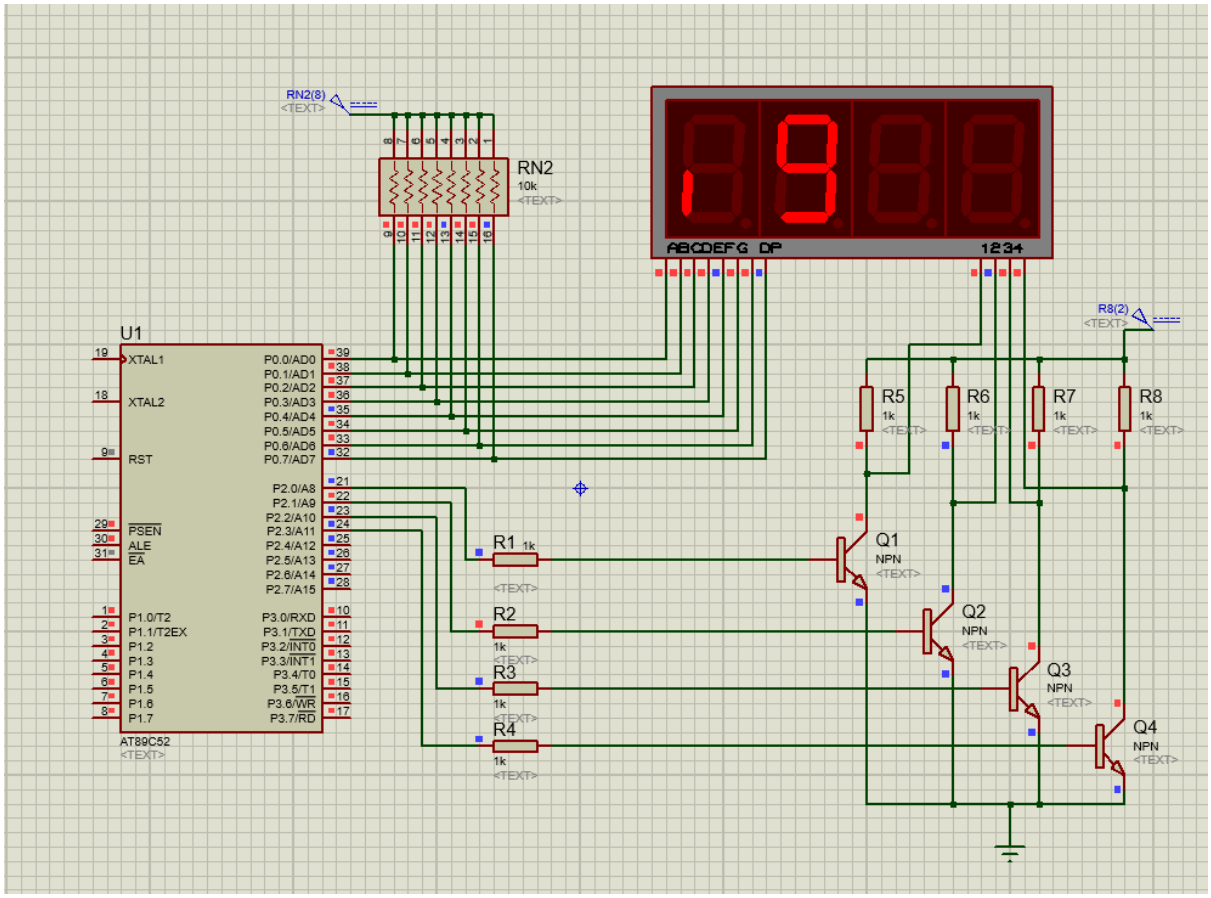
OUTPUT:


RN2(8)
RN2
10k
2025
ABCDEFG DP
1234
U1
XTAL1
XTAL2
RST
PSEN
ALE
EA
P1.0/T2
P1.1/T2EX
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
AT89C52
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
R1 1k
R2 1k
R3 1k
R4 1k
R5 1k
R6 1k
R7 1k
R8 1k
R8(2)
Q1 NPN
Q2 NPN
Q3 NPN
Q4 NPN

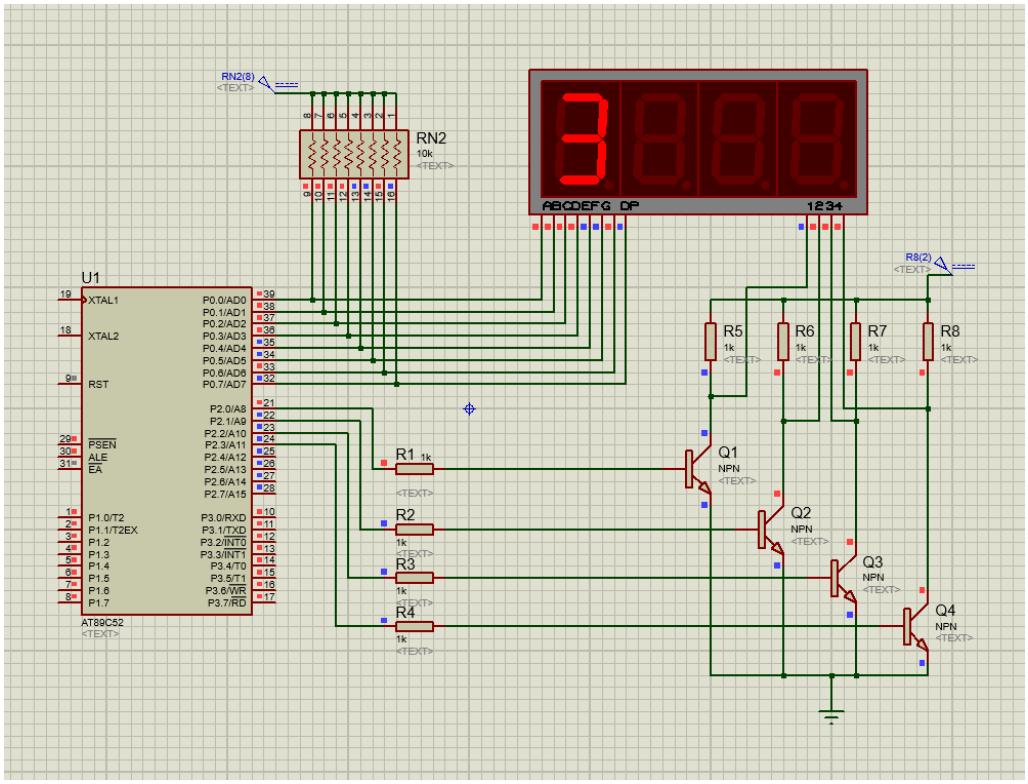
Output – 2.1:


RN2(8)
RN2
10k
ABCDEFG DP
1234
U1
XTAL1
XTAL2
RST
PSEN
ALE
EA
P1.0/T2
P1.1/T2EX
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
AT89C52
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
R1
R2
R3
R4
1k
R5
R6
R7
R8
R8(2)
Q1
Q2
Q3
Q4
NPN

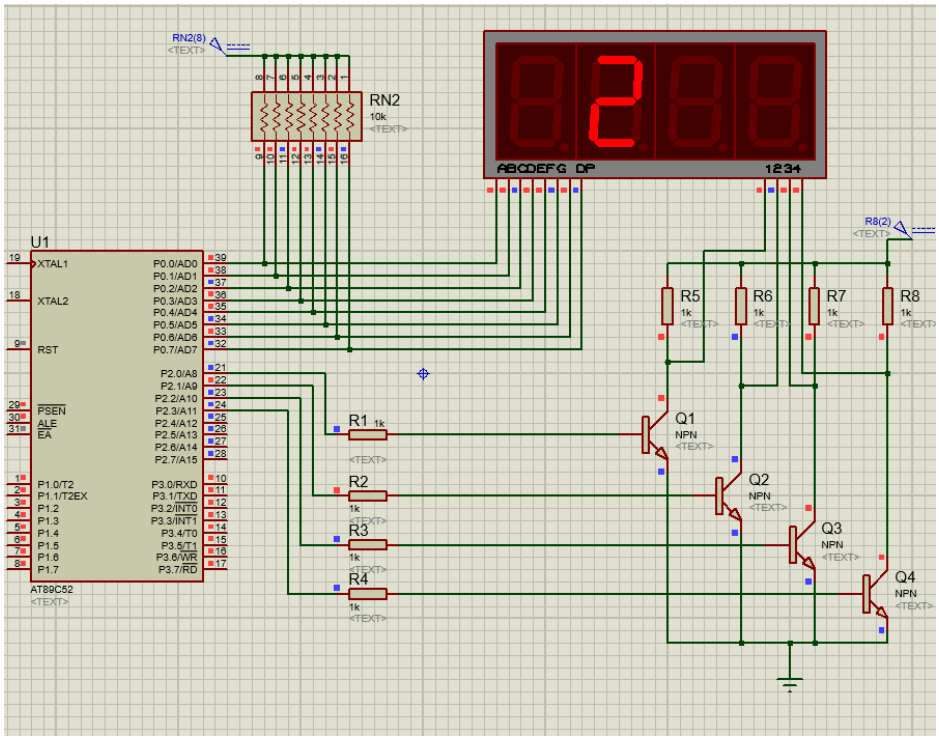
Output – 2.2:



OUTPUT – 2.3:



Output 2.4



Output 2.5: