****

**ASSIGNMENT 02**

**KANWAL SHEHZADI-------------------------------------------------02-131212-027**

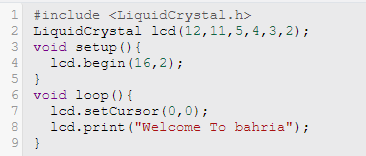
****

**ASSIGNMENT 02**

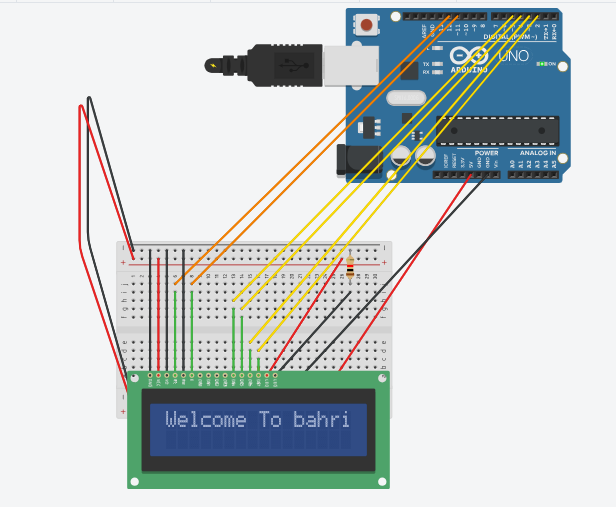
1. **Write a program to display a well come to bahria in LCD display**

**With Arduino:**

**C PROGRAM:**

****

**CIRCUIT:**

****

**With 8051:**

**C CODE:**

#include <htc.h> // Include the header file for your microcontroller (e.g., 8051)

void delay(unsigned int time) {

unsigned int i, j;

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

for(j = 0; j < 1275; j++);

}

void lcd\_command(unsigned char command) {

P20 = 0; // Select command register

P21 = 0; // Set to write mode

P1 = command; // Send the command to data pins

P22 = 1; // Enable the LCD

delay(1);

P22 = 0; // Disable the LCD

}

void lcd\_data(unsigned char dat) {

P20 = 1; // Select data register

P21 = 0; // Set to write mode

P1 = dat; // Send the data to data pins

P22 = 1; // Enable the LCD

delay(1);

P22 = 0; // Disable the LCD

}

void lcd\_init() {

lcd\_command(0x38); // 8-bit, 2-line display, 5x7 font

lcd\_command(0x0E); // Display on, cursor blinking

lcd\_command(0x01); // Clear display

lcd\_command(0x06); // Increment cursor

lcd\_command(0x80); // Set cursor to the first line

}

void main() {

lcd\_init();

lcd\_data('W');

lcd\_data('e');

lcd\_data('l');

lcd\_data('c');

lcd\_data('o');

lcd\_data('m');

lcd\_data('e');

lcd\_data(' ');

lcd\_data('t');

lcd\_data('o');

lcd\_data(' ');

lcd\_data('B');

lcd\_data('a');

lcd\_data('h');

lcd\_data('r');

lcd\_data('i');

lcd\_data('a');

while (1);

}

**ASSEMBLY PROGRAM:**

#include <htc.h> ; Include the header file for your microcontroller (e.g., 8051)

delay\_ms MACRO delay

mov R2, #delay

mov R3, #20

DLY\_LOOP:

DJNZ R3, DLY\_LOOP

DJNZ R2, DLY\_LOOP

ENDM

delay\_us MACRO delay

mov R2, #delay

DLY\_LOOP\_US:

NOP

DJNZ R2, DLY\_LOOP\_US

ENDM

lcd\_command MACRO command

CLR P2.0 ; Select command register

CLR P2.1 ; Set to write mode

MOV A, command

MOV P1, A ; Send the command to data pins

SETB P2.2 ; Enable the LCD

delay\_us 1

CLR P2.2 ; Disable the LCD

ENDM

lcd\_data MACRO data

SETB P2.0 ; Select data register

CLR P2.1 ; Set to write mode

MOV A, data

MOV P1, A ; Send the data to data pins

SETB P2.2 ; Enable the LCD

delay\_us 1

CLR P2.2 ; Disable the LCD

ENDM

lcd\_init MACRO

lcd\_command 0x38 ; 8-bit, 2-line display, 5x7 font

lcd\_command 0x0E ; Display on, cursor blinking

lcd\_command 0x01 ; Clear display

lcd\_command 0x06 ; Increment cursor

lcd\_command 0x80 ; Set cursor to the first line

ENDM

ORG 0x0000

LJMP MAIN

MAIN:

MOV A, #0x20 ; Initialize the LCD data pins

MOV P1, A

ACALL lcd\_init

ACALL lcd\_data, 'W'

ACALL lcd\_data, 'e'

ACALL lcd\_data, 'l'

ACALL lcd\_data, 'c'

ACALL lcd\_data, 'o'

ACALL lcd\_data, 'm'

ACALL lcd\_data, 'e'

ACALL lcd\_data, ' '

ACALL lcd\_data, 't'

ACALL lcd\_data, 'o'

ACALL lcd\_data, ' '

ACALL lcd\_data, 'B'

ACALL lcd\_data, 'a'

ACALL lcd\_data, 'h'

ACALL lcd\_data, 'r'

ACALL lcd\_data, 'i'

ACALL lcd\_data, 'a'

MAIN\_LOOP:

SJMP MAIN\_LOOP

1. **Draw the Block diagram of seven segment display and write its code in C and Assembly**

**C PROGRAM:**

#include<reg51.h>

void msdelay(unsigned int time) // Function for creating delay in milliseconds.

{

unsigned i,j ;

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

for(j=0;j<1275;j++);

}

void main()

{

unsigned char to\_disp[]={0x3F,0x06,0x5B,0x4F,0x66, // Array for hex values (0-F) for

0x6D,0x7D,0x07,0x7F,0x6F} // common anode 7 segment

int k;

while(1)

{

for(k=0;k<10;k++)

{

P2=to\_disp[k];

msdelay(100);

}

}

}

**ASSEMBLY PROGRAM:**

ORG 4000H

DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H, 7FH, 6FH, 0

; Lookup table for digits 0 to 9

ORG 0000H

main: MOV DPTR, #4000H

repeat: CLR A

MOVC A, @A+DPTR ; Copy data from external location to accumulator

MOV P2, A ; Move the pattern of the digit into port P2

ACALL delay ; Call a delay to so that the transition is visible

INC DPTR ; Point to the next pattern

CJNE A, 0, repeat ; Repeat till 0 (Stop bit) is received

SJMP main ; Run this forever till externally stopped

; generate a decent enough delay between transitions

delay:

MOV R0, #08H

LP2: MOV R1, #0FFH

LP1: MOV R2, #0FFH

LP3: DJNZ R2, LP3

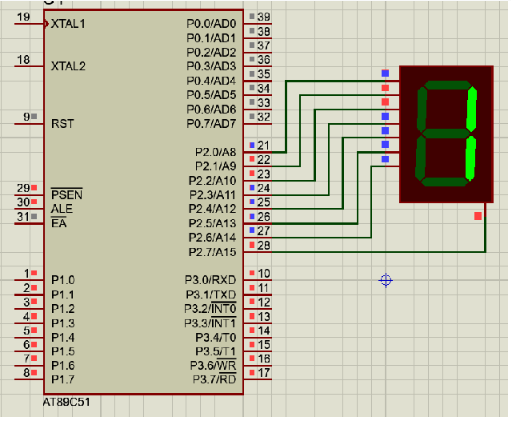
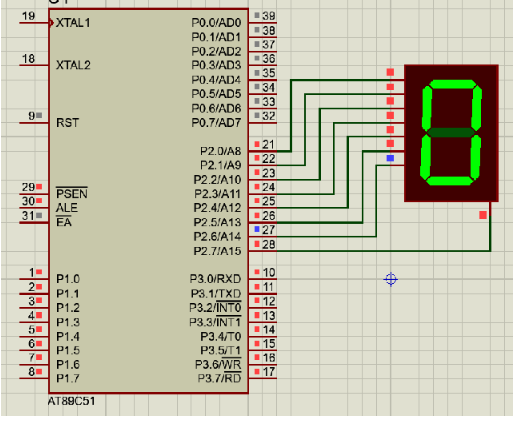
DJNZ R1, LP1

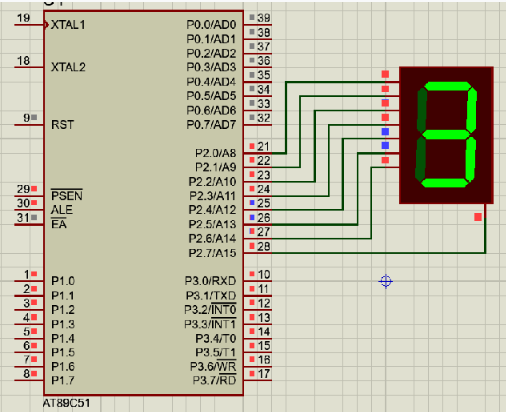
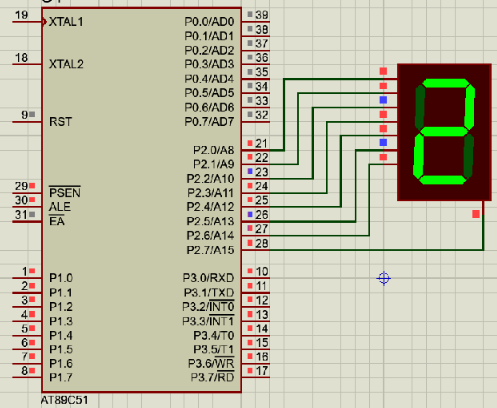
DJNZ R0, LP2

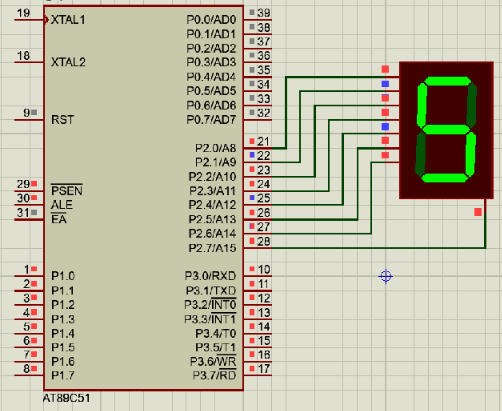
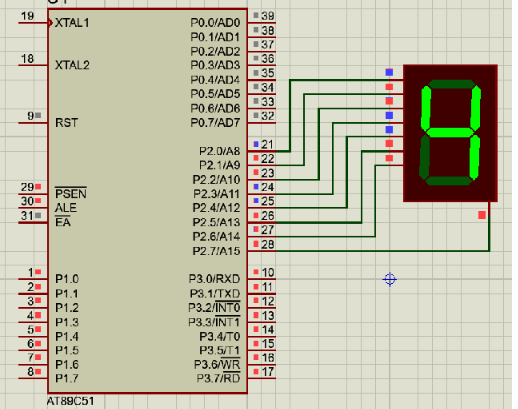
RET

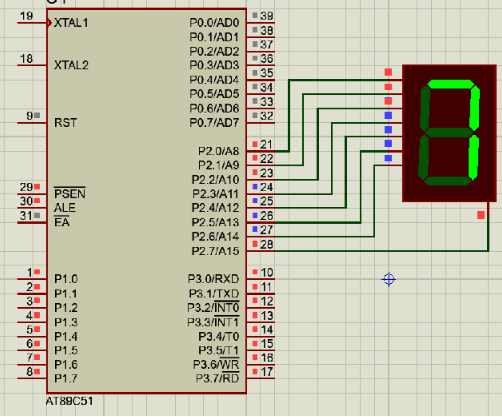
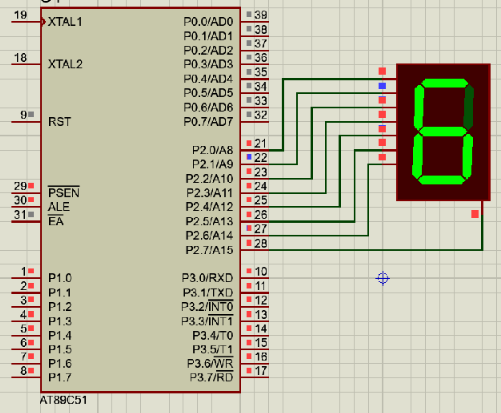
END

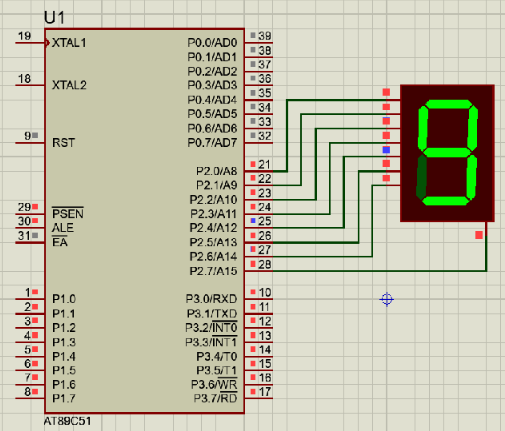
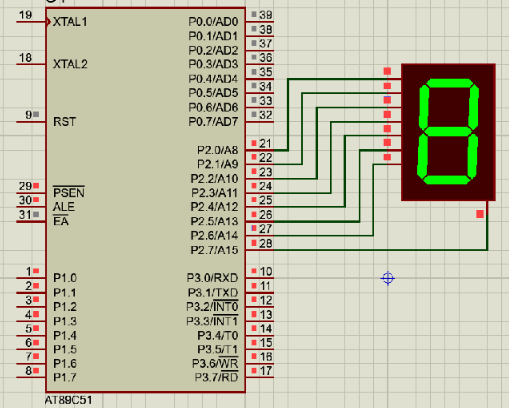
**CIRCUIT:**

****

****

****

****

****

1. **Draw block diagram of two seven segment display**

**Write a program in Assembly and C to display a digit from 0 to 9 in both seven segment display**

**C PROGRAM:**

#include<reg51.h>

void delay(unsigned int t);

void main()

{

unsigned int ch[]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};

unsigned int i,j;

P3=0xc0;

P2=0xc0;

while(1)

{

for(j=0;j<10;j++)

{

for(i=0;i<10;i++)

{

P3=ch[i];

delay(50);

}

if(j!=10)

P2=ch[j+1];

}

if(i==10&&j==10)

P3=0xc0;

P2=0xc0;

}

}

void delay(unsigned int t)

{

unsigned int i,j;

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

for(j=0;j<1275;j++);

}

**ASSEMBLY PROGRAM:**

ORG 0000H

MOV DPTR, #patterns ; Initialize DPTR with the starting address of the patterns

MOV R2, #0Ah ; R2 = 10 (to iterate 10 times)

display\_loop:

MOV A, R2 ; Move the count in R2 to the accumulator

MOVX A, @DPTR ; Move the data from the address pointed by DPTR to the accumulator

MOV P3, A ; Display the value from the array on port P3

ACALL delay ; Call delay function to create a delay

INC DPTR ; Point to the next pattern in the array

DJNZ R2, display\_loop ; Repeat until R2 becomes 0

MOV P3, #0xC0 ; Reset P3 to initial value 0xC0

MOV P2, #0xC0 ; Reset P2 to initial value 0xC0

endless\_loop:

SJMP endless\_loop ; Infinite loop

delay: ; Delay function definition

MOV R7, #5 ; Delay count outer loop

outer\_loop:

MOV R6, #50 ; Delay count inner loop

inner\_loop:

NOP ; No operation, to increase delay

DJNZ R6, inner\_loop

DJNZ R7, outer\_loop

RET

patterns: ; Define patterns array

DB 0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80, 0x90

DB 0 ; Stop bit

END

**CIRCUIT DIAGRAM:**

