****

**ASSIGNMENT 01**

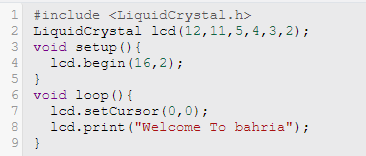
**KANWAL SHEHZADI-------------------------------------------------02-131212-027**

****

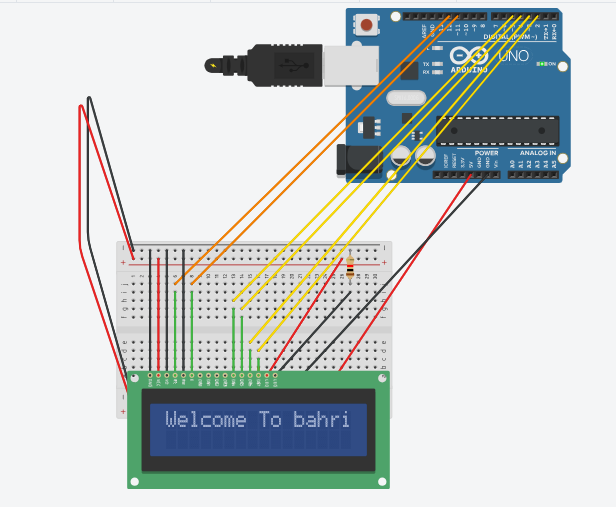
**ASSIGNMENT 01**

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

**PROGRAM:**

****

**CIRCUIT:**

****

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

**C PROGRAM:**

void setup()

{

pinMode(2, OUTPUT);

pinMode(3, OUTPUT);

pinMode(4, OUTPUT);

pinMode(5, OUTPUT);

pinMode(6, OUTPUT);

pinMode(7, OUTPUT);

pinMode(8, OUTPUT);

pinMode(9, OUTPUT);

pinMode(0, OUTPUT);

}

void loop()

{

digitalWrite(2, LOW);

digitalWrite(3, HIGH);

digitalWrite(4, HIGH);

digitalWrite(5, HIGH);

digitalWrite(6, HIGH);

digitalWrite(7, HIGH);

digitalWrite(8, HIGH);

digitalWrite(9, LOW);

delay(2000); // Wait for 2000 millisecond(s)

digitalWrite(2, LOW);

digitalWrite(3, HIGH);

digitalWrite(4, LOW);

digitalWrite(5, LOW);

digitalWrite(6, HIGH);

digitalWrite(7, LOW);

digitalWrite(8, LOW);

digitalWrite(9, LOW);

delay(2000); // Wait for 2000 millisecond(s)

digitalWrite(2, LOW);

digitalWrite(3, LOW);

digitalWrite(4, HIGH);

digitalWrite(5, HIGH);

digitalWrite(6, HIGH);

digitalWrite(7, HIGH);

digitalWrite(8, LOW);

digitalWrite(9, HIGH);

delay(2000); // Wait for 2000 millisecond(s)

digitalWrite(2, LOW);

digitalWrite(3, HIGH);

digitalWrite(4, HIGH);

digitalWrite(5, LOW);

digitalWrite(6, HIGH);

digitalWrite(7, HIGH);

digitalWrite(8, LOW);

digitalWrite(9, HIGH);

delay(2000); // Wait for 2000 millisecond(s)

digitalWrite(2, LOW);

digitalWrite(3, HIGH);

digitalWrite(4, LOW);

digitalWrite(5, LOW);

digitalWrite(6, HIGH);

digitalWrite(7, LOW);

digitalWrite(8, HIGH);

digitalWrite(9, HIGH);

delay(2000); // Wait for 2000 millisecond(s)

digitalWrite(2, LOW);

digitalWrite(3, HIGH);

digitalWrite(4, HIGH);

digitalWrite(5, LOW);

digitalWrite(6, LOW);

digitalWrite(7, HIGH);

digitalWrite(8, HIGH);

digitalWrite(9, HIGH);

delay(2000); // Wait for 2000 millisecond(s)

digitalWrite(2, LOW);

digitalWrite(3, HIGH);

digitalWrite(4, HIGH);

digitalWrite(5, HIGH);

digitalWrite(6, LOW);

digitalWrite(7, HIGH);

digitalWrite(8, HIGH);

digitalWrite(0, HIGH);

delay(1000); // Wait for 1000 millisecond(s)

digitalWrite(2, LOW);

digitalWrite(3, HIGH);

digitalWrite(4, LOW);

digitalWrite(5, LOW);

digitalWrite(6, HIGH);

digitalWrite(7, HIGH);

digitalWrite(8, LOW);

digitalWrite(9, LOW);

delay(2000); // Wait for 2000 millisecond(s)

digitalWrite(2, LOW);

digitalWrite(3, HIGH);

digitalWrite(4, HIGH);

digitalWrite(5, HIGH);

digitalWrite(6, HIGH);

digitalWrite(7, HIGH);

digitalWrite(8, HIGH);

digitalWrite(9, HIGH);

delay(2000); // Wait for 2000 millisecond(s)

digitalWrite(2, LOW);

digitalWrite(3, HIGH);

digitalWrite(4, HIGH);

digitalWrite(5, LOW);

digitalWrite(6, HIGH);

digitalWrite(7, HIGH);

digitalWrite(8, HIGH);

digitalWrite(9, HIGH);

delay(2000); // Wait for 2000 millisecond(s)

digitalWrite(2, HIGH);

digitalWrite(3, LOW);

digitalWrite(4, LOW);

digitalWrite(5, LOW);

digitalWrite(6, LOW);

digitalWrite(7, LOW);

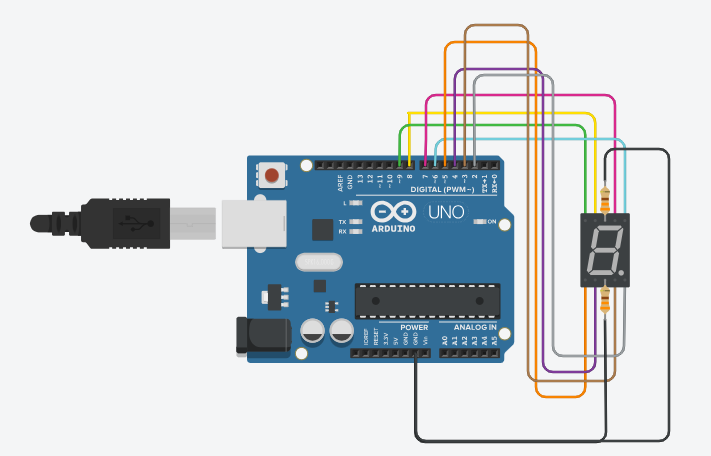
digitalWrite(8, LOW);

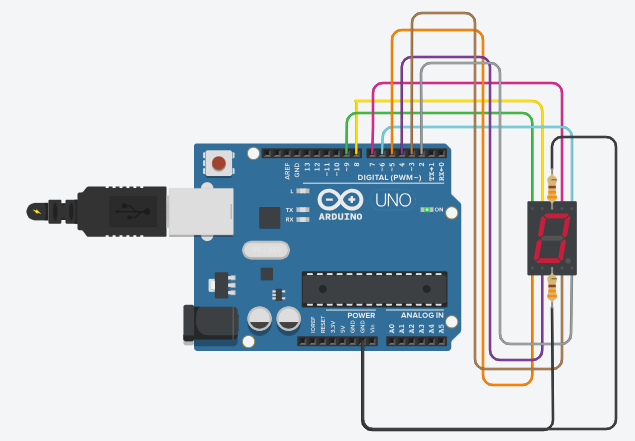
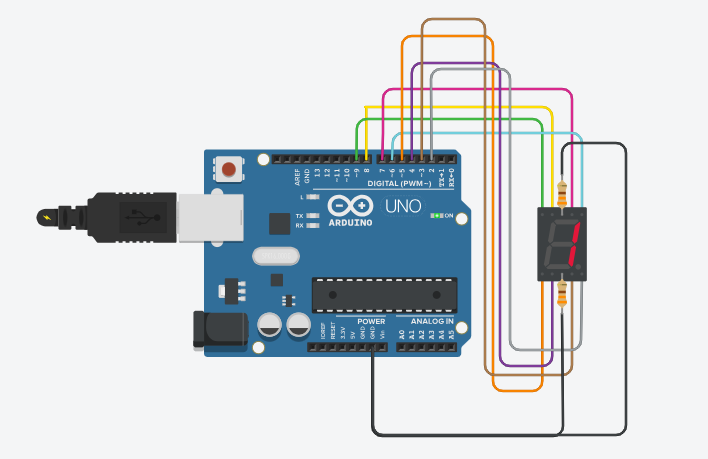
digitalWrite(9, LOW);

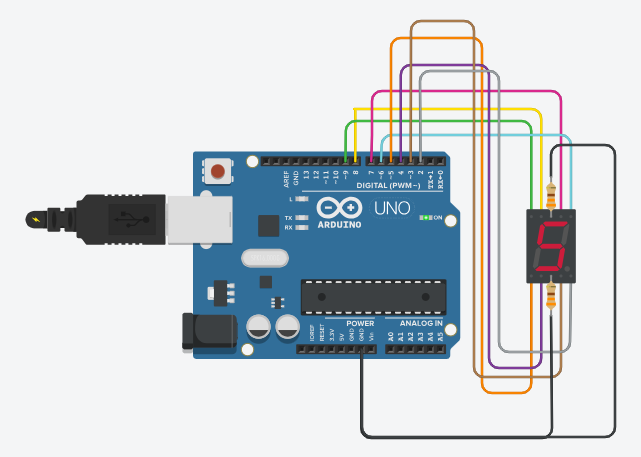
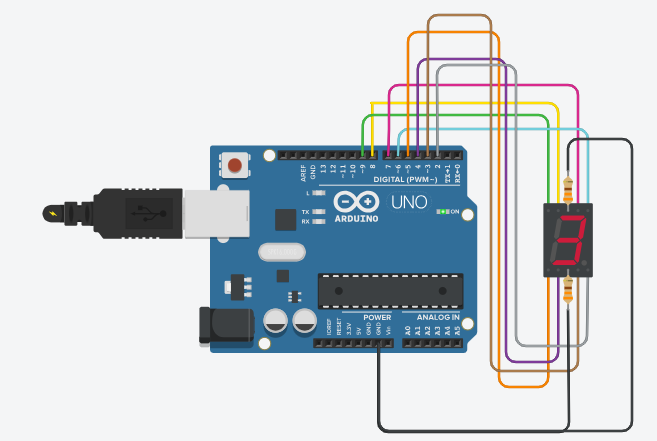
delay(2000); // Wait for 2000 millisecond(s)

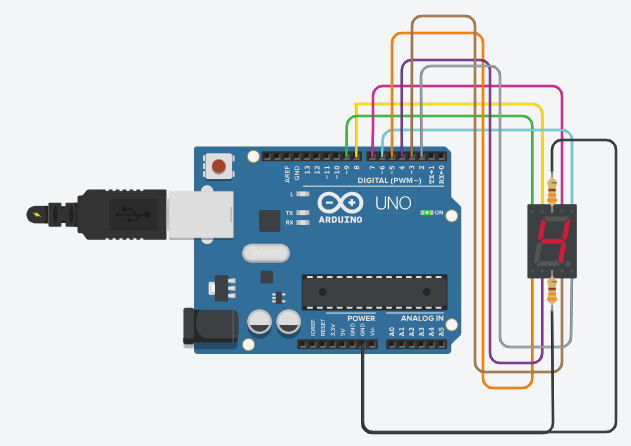
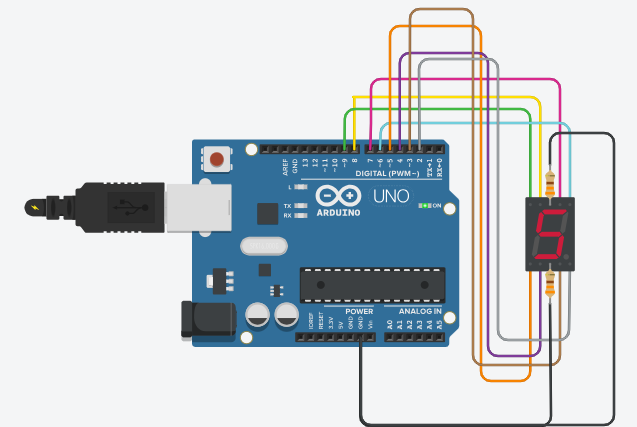
}

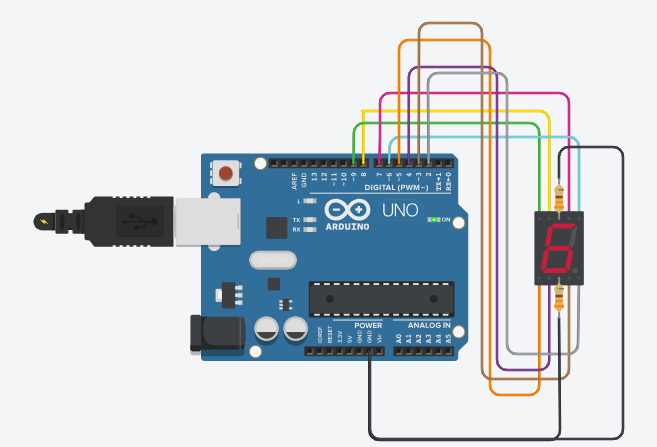
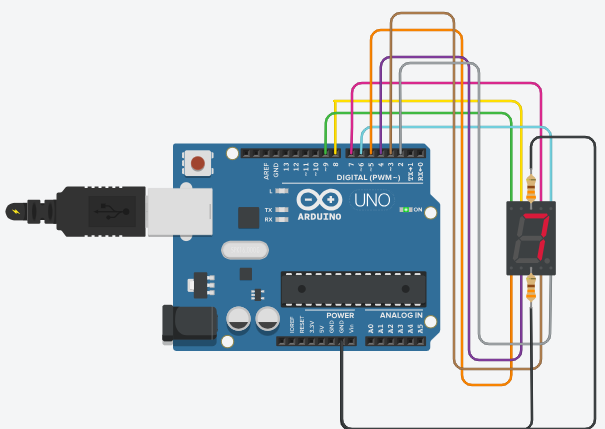
**CIRCUIT:**

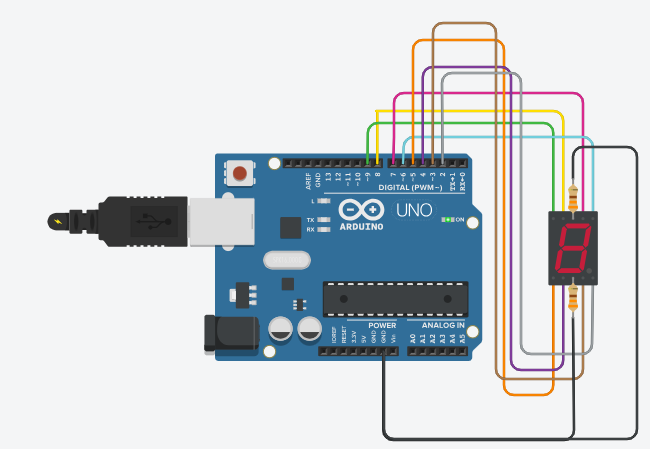
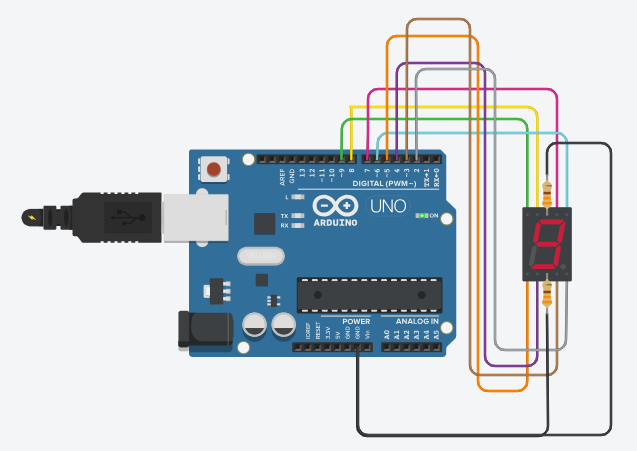
****

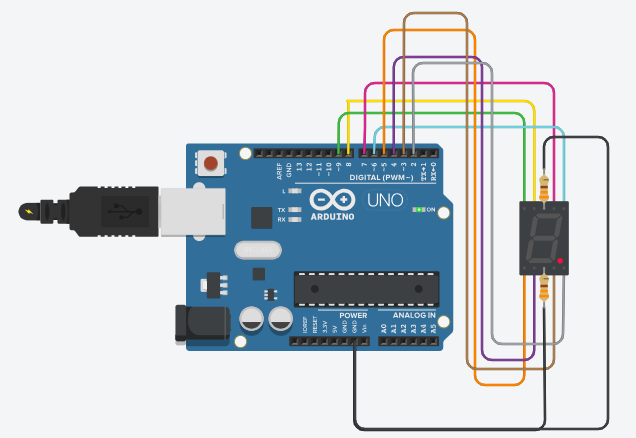
** **

** **

** **

** **

** **

****

**ASSEMBLY PROGRAM:**

**.**include "m328pdef.inc"

// Define constants for port addresses

.equ SEG\_PORT = PORTD

.equ DELAY = 2000 // Delay in milliseconds

.global main

main:

; Initialize port D for output

ldi r16, 0xFF

out SEG\_PORT, r16

loop:

; Display '1'

ldi r16, 0b11111100

out SEG\_PORT, r16

call delay\_ms

out SEG\_PORT, r16

call delay\_ms

; Display '2'

ldi r16, 0b01100000

out SEG\_PORT, r16

call delay\_ms

out SEG\_PORT, r16

call delay\_ms

; Display '3'

ldi r16, 0b11011010

out SEG\_PORT, r16

call delay\_ms

out SEG\_PORT, r16

call delay\_ms

; Add code for displaying other digits

rjmp loop

delay\_ms:

ldi r18, 250 ; Initialize outer loop counter

outer\_loop:

ldi r17, 184 ; Initialize inner loop counter

inner\_loop:

dec r17

brne inner\_loop

dec r18

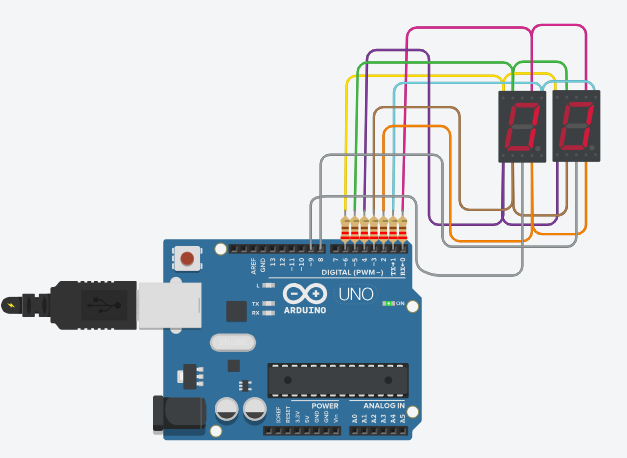
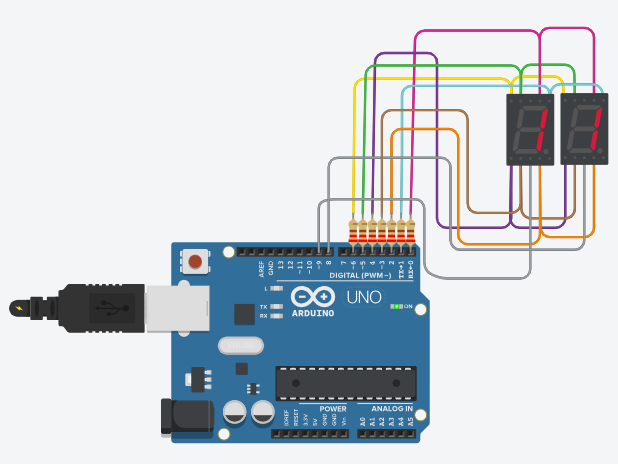
brne outer\_loop

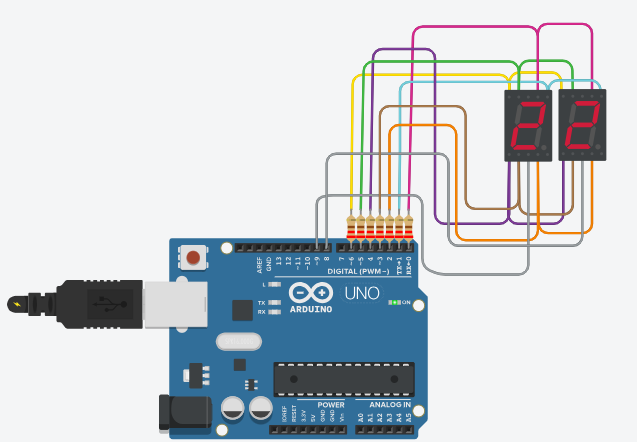
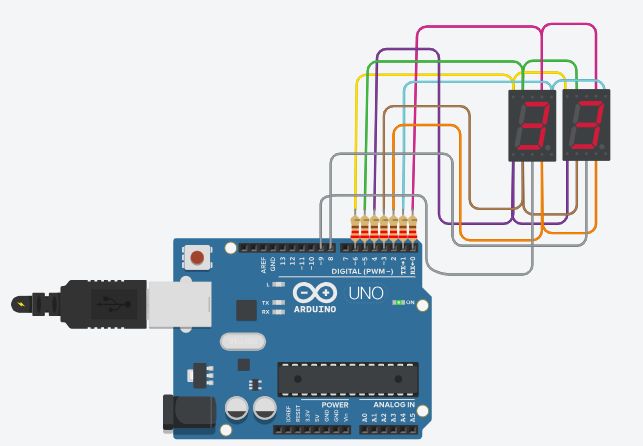
ret

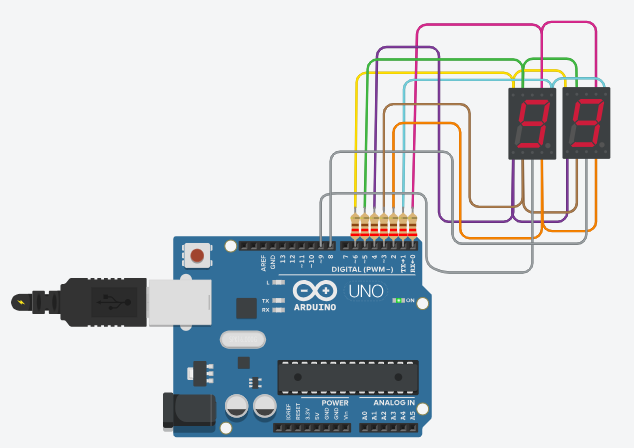
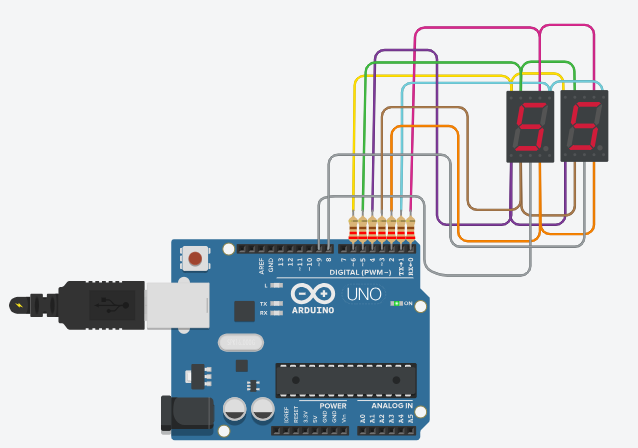
.end

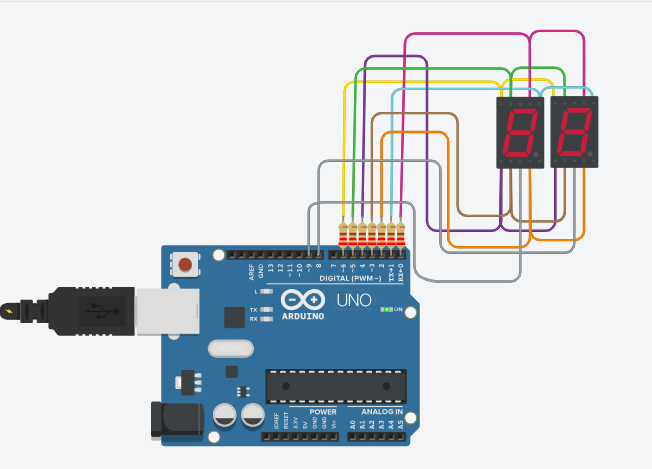
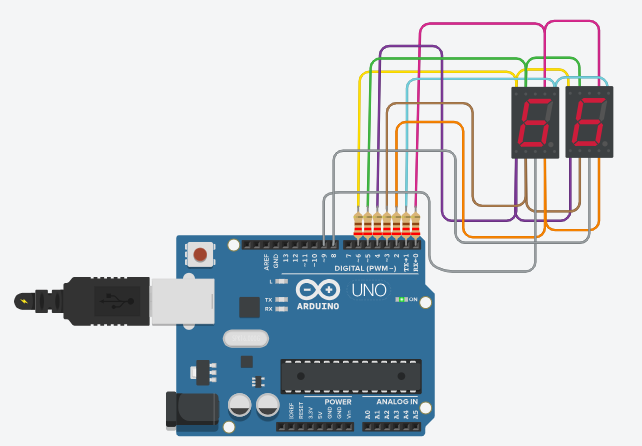
1. **Draw block diagram of two seven segment display**

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

** **

** **

****

****

**C PROGRAM:** //a,b,c,d,e,f,g

byte segValue[10][7] = {

{0,0,0,0,0,0,1}, //0

{1,0,0,1,1,1,1}, //1

{0,0,1,0,0,1,0}, //2

{0,0,0,0,1,1,0}, //3

{1,0,0,1,1,0,0}, //4

{0,1,0,0,1,0,0}, //5

{0,1,0,0,0,0,0}, //6

{0,0,0,1,1,1,1}, //7

{0,0,0,0,0,0,0}, //8

{0,0,0,0,1,0,0} //9

};

byte segPin[8]={2,3,4,5,6,7,8,9}; //{a,b,c,d,e,f,g,dp}

byte digitPin[2] = {A0,A1}; //segment

void setup() {

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

pinMode(segPin[i], OUTPUT);

}

pinMode(digitPin[0], OUTPUT);

pinMode(digitPin[1], OUTPUT);

digitalWrite(digitPin[0], LOW);

digitalWrite(digitPin[1], LOW);

}

void loop()

{

for (int i = 0; i<100; i++)

{

display\_N(i);

delay(1);

}

}

void display\_N(int num)

{

// Convierte unidades y decenas

int und = num % 10;

int dec = (num % 100) / 10;

// Refresco dinamico

for(int i=0; i<100; i++)

{

segOutput(1, und, 1);

segOutput(0, dec, 1);

delay(2);

}

Serial.print(dec);

Serial.println(und);

}

//LED

void segClear(){

for(int i=0;i<8;i++){

digitalWrite(segPin[i], HIGH);

}

}

//LED

void segOutput(int d, int Number, int dp){

segClear();

digitalWrite(digitPin[d], HIGH);

for(int i=0;i<7;i++){

digitalWrite(segPin[i], segValue[Number][i]);

}

digitalWrite(segPin[7], dp);

delayMicroseconds(1000);

digitalWrite(digitPin[d], LOW);

}

**ASSEMBLY PROGRAM:**

#define A 0

#define B 1

#define C 2

#define D 3

#define E 4

#define F 5

#define G 6

#define DIGIT1 9

#define DIGIT2 8

.data

myNumber: .byte 0b0111111, 0b0000110, 0b1011011, 0b1001111, 0b1100110, 0b1101101, 0b1111101, 0b0000111, 0b1111111, 0b1101111

.text

.global main

main:

ldi count, 0 ; Initialize count to 0

loop:

; Display "0" on the right display (digit2)

ldi r16, LOW(DIGIT1)

out PORTB, r16 ; Set digit1 pin low

ldi r16, 0x00 ; Clear display

out PORTA, r16 ; Set all segment pins low

ldi r16, myNumber

add r16, count

ld r17, X

out PORTA, r17 ; Display the current number

ldi r16, 0x01 ; Set digit2 pin high

out PORTB, r16

; Delay

ldi r16, 250

ldi r17, 5

delay\_loop:

dec r17

brne delay\_loop

dec r16

brne delay\_loop

; Increment count

inc count

cpi count, 10

breq reset\_count

; Repeat the loop

rjmp loop

reset\_count:

ldi count, 0

; Repeat the loop

rjmp loop

1. **Write a code of LED interface in Assembly and C through different P1, P2, P3**

**Block diagram of 8 LEDs**

1. **Perform Rotation operation**

**C CODE**

#include <htc.h>

void delay(int ms) {

int i, j;

for (i = 0; i < ms; i++) {

for (j = 0; j < 100; j++);

} }

void main() {

P21=0x00;

P20=0x00;

P30=0x00;

P10=0x00;

P11=0x00;

P31=0x00;

P22=0x00;

P12=0x00;

while (1) {

P21 = 1

delay(1)

P21 = 0

P20 = 1

delay(1)

P20 = 0

P30 = 1

delay(1)

P30 = 0

P10 = 1

delay(1)

P10 = 0

P11 = 1

delay(1)

P11 = 0

P31 = 1

delay(1)

P31 = 0

P22 = 1

delay(1)

P22 = 0

P12 = 1

delay(1)

P12 = 0 }

}

**ASSEMBLY CODE**

#include <htc.h>

delay:

movlw .100

delay\_loop:

decfsz W, F

goto delay\_loop

return

main:

clrf PORTA

clrf PORTB

clrf PORTC

main\_loop:

bsf PORTB, 1

call delay

bcf PORTB, 1

bsf PORTB, 0

call delay

bcf PORTB, 0

bsf PORTC, 0

call delay

bcf PORTC, 0

bsf PORTA, 0

call delay

bcf PORTA, 0

bsf PORTA, 1

call delay

bcf PORTA, 1

bsf PORTC, 1

call delay

bcf PORTC, 1

bsf PORTB, 2

call delay

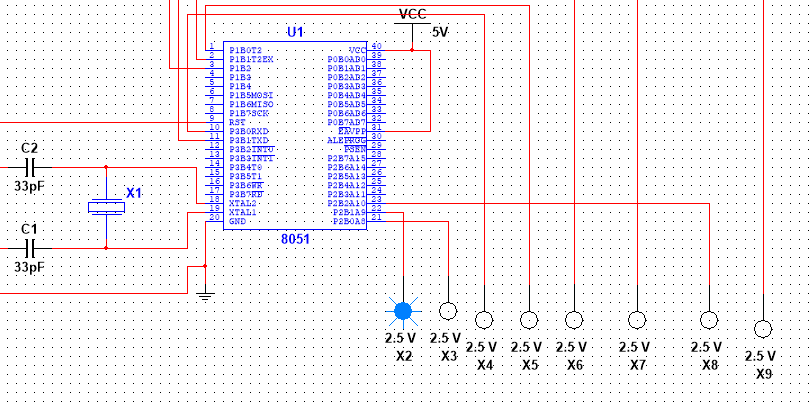
bcf PORTB, 2

bsf PORTA, 2

call delay

bcf PORTA, 2

goto main\_loop

****

1. **Two on and six off**

**C Code:**

#include <htc.h>

void delay(int ms) {

int i, j;

for (i = 0; i < ms; i++) {

for (j = 0; j < 100; j++);

}

}

void main() {

P21=0x00;

P20=0x00;

P30=0x00;

P10=0x00;

P11=0x00;

P31=0x00;

P22=0x00;

P12=0x00;

while (1) {

P21 = 1

P20 = 1

delay(1)

P21 = 0

P20 = 0

P30 = 1

P10 = 1

P11 = 1

P31 = 1

P22 = 1

P12 = 1

delay(1)

P30 = 0

P10 = 0

P11 = 0

P31 = 0

P22 = 0

P12 = 0

}

}

**ASSEMBLY CODE:**

; Define constants and variables

LED1 equ P21

LED2 equ P20

LED3 equ P30

LED4 equ P10

LED5 equ P11

LED6 equ P31

LED7 equ P22

LED8 equ P12

; Delay function

delay:

movlw .100

delay\_loop:

decfsz W, F

goto delay\_loop

return

; Main program

main:

clrf LED1

clrf LED2

clrf LED3

clrf LED4

clrf LED5

clrf LED6

clrf LED7

clrf LED8

main\_loop:

bsf LED1, 0

bsf LED2, 0

call delay

bcf LED1, 0

bcf LED2, 0

bsf LED3, 0

bsf LED4, 0

bsf LED5, 0

bsf LED6, 0

bsf LED7, 0

bsf LED8, 0

call delay

bcf LED3, 0

bcf LED4, 0

bcf LED5, 0

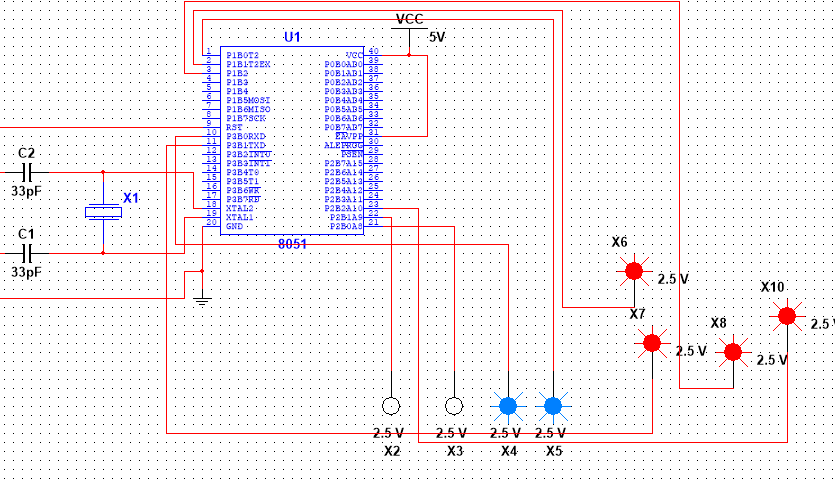
bcf LED6, 0

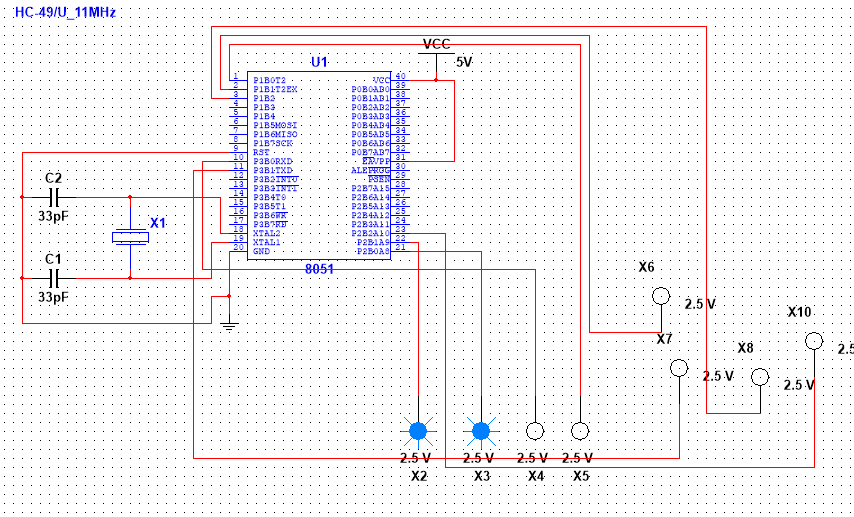
bcf LED7, 0

bcf LED8, 0

goto main\_loop ; Repeat main\_loop

**OUTPUT:**

****

****

**(3)Perform four on and off.**

**C code:**

#include <htc.h>

void delay(int ms) {

int i, j;

for (i = 0; i < ms; i++) {

for (j = 0; j < 100; j++);

}

}

void main() {

P21=0x00

P20=0x00

P30=0x00

P10=0x00

P11=0x00

P31=0x00

P22=0x00

P12=0x00

while (1) {

P21 = 1

P20 = 1

P30 = 1

P10 = 1

delay(1)

P21 = 0

P20 = 0

P30 = 0

P10 = 0

P11 = 1

P12 = 1

P31 = 1

P22 = 1

delay(1);

P11 = 0

P31 = 0

P22 = 0

P12 = 0

}

}

**ASSEMBLY CODE:**

; Define constants and variables

LED1 equ P21 ; Define the symbolic names for LEDs

LED2 equ P20

LED3 equ P30

LED4 equ P10

LED5 equ P11

LED6 equ P31

LED7 equ P22

LED8 equ P12

; Delay function

delay:

movlw .100 ; Load W with delay value

delay\_loop:

decfsz W, F ; Decrement W and skip if zero

goto delay\_loop

return

; Main program

main:

clrf LED1

clrf LED2

clrf LED3

clrf LED4

clrf LED5

clrf LED6

clrf LED7

clrf LED8

main\_loop:

bsf LED1, 0

bsf LED2, 0

bsf LED3, 0

bsf LED4, 0

call delay

bcf LED1, 0

bcf LED2, 0

bcf LED3, 0

bcf LED4, 0

bsf LED5, 0

bsf LED8, 0

bsf LED6, 0

bsf LED7, 0

call delay

bcf LED5, 0

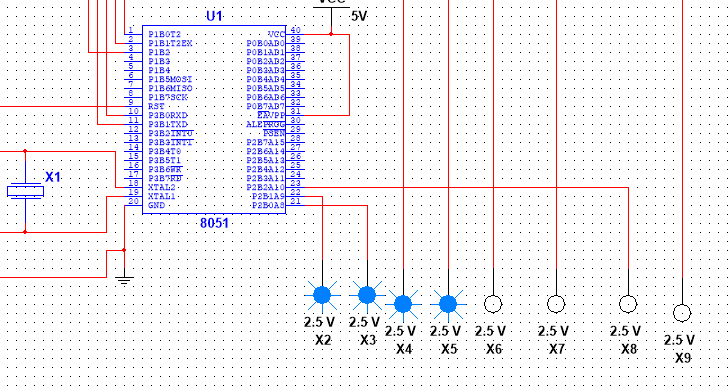
bcf LED6, 0

bcf LED7, 0

bcf LED8, 0

goto main\_loop

**Output:**

****