****

**ASSIGNMENT 01**

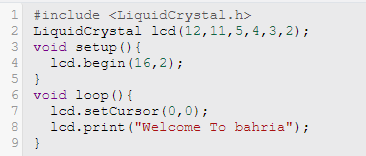
**KANWAL SHEHZADI-------------------------------------------------02-131212-027**

****

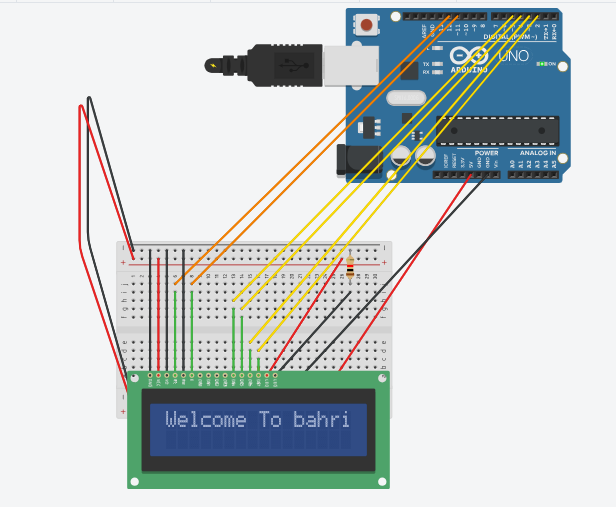
**ASSIGNMENT 01**

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);**

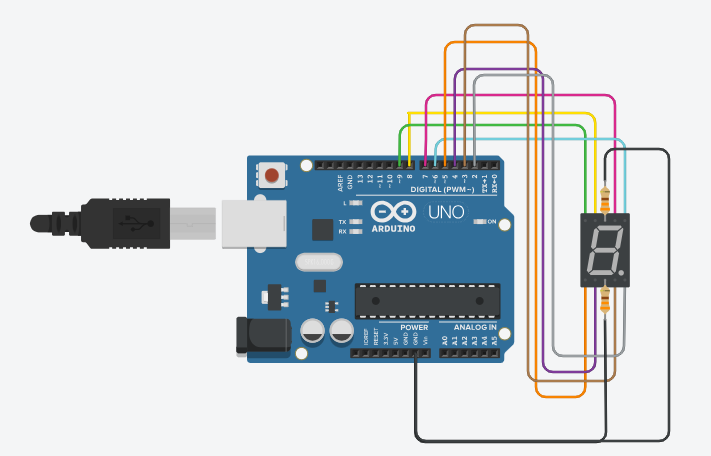
**digitalWrite(8, LOW);**

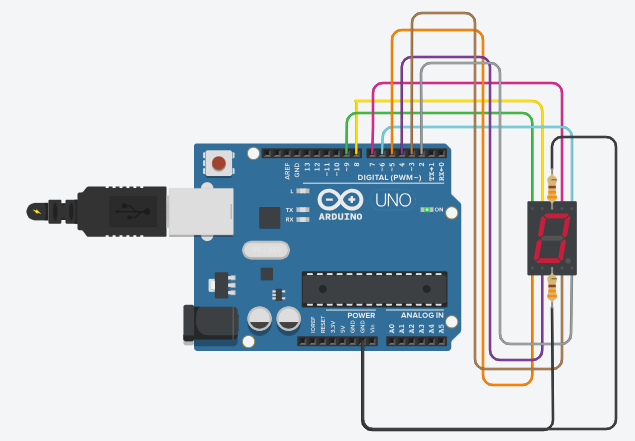
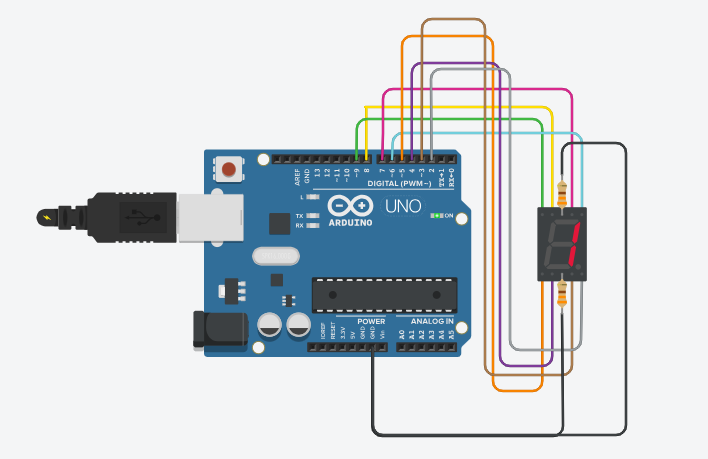
**digitalWrite(9, LOW);**

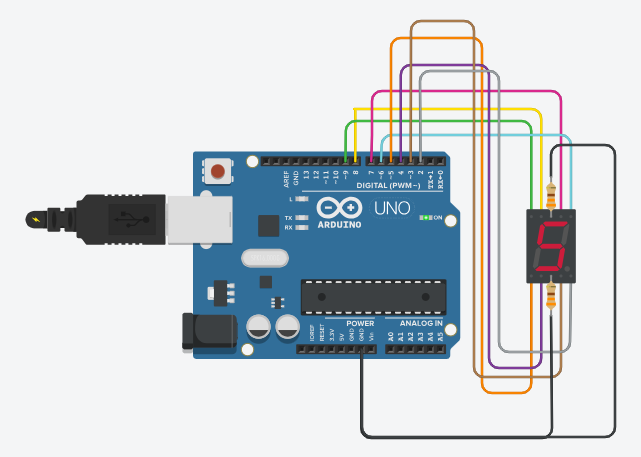
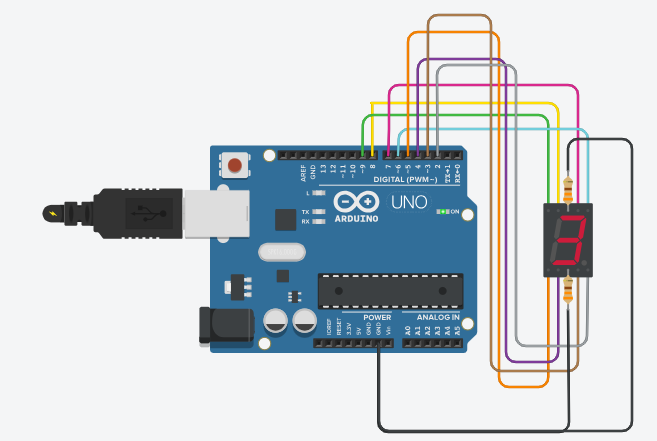
**delay(2000); // Wait for 2000 millisecond(s)**

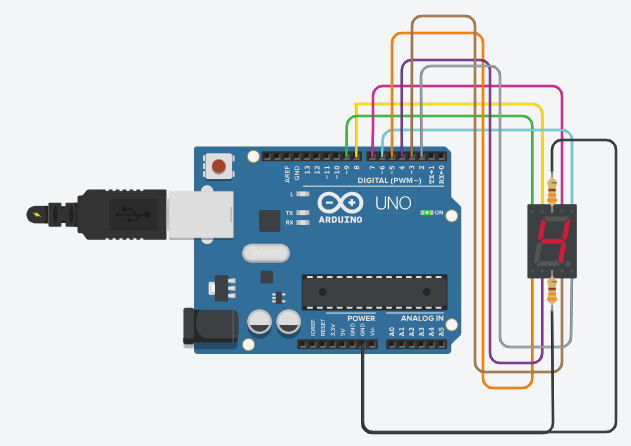
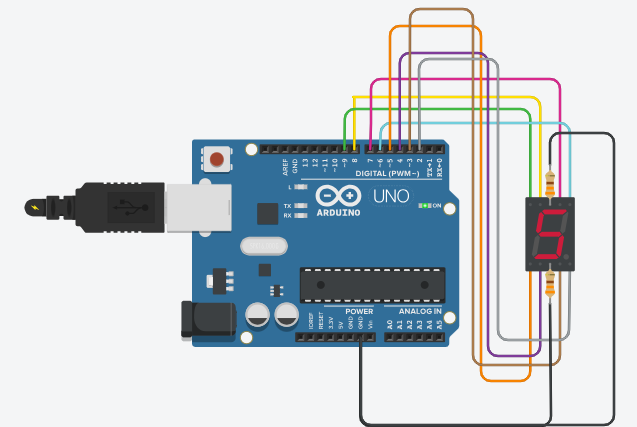
**}**

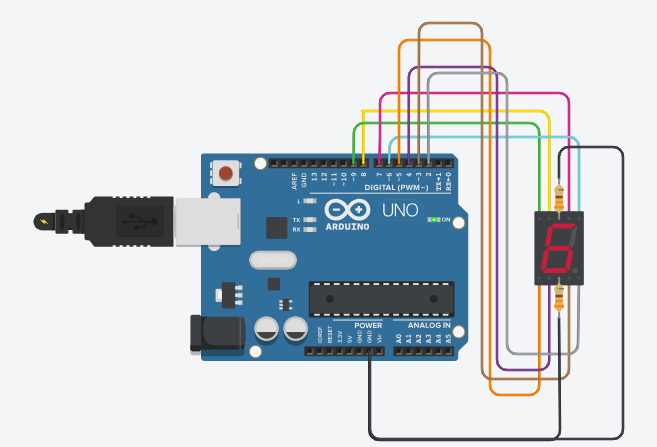
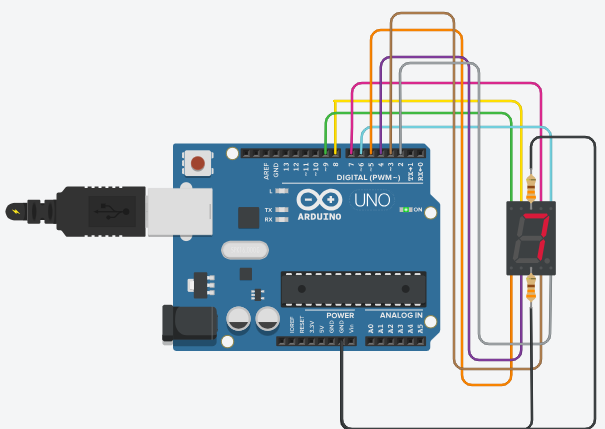
**CIRCUIT:**

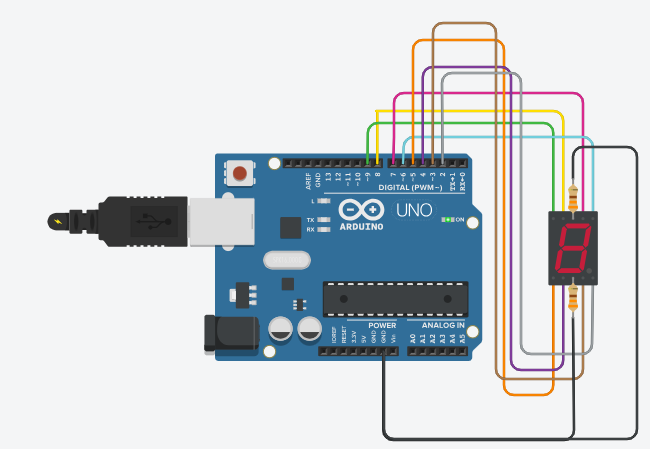
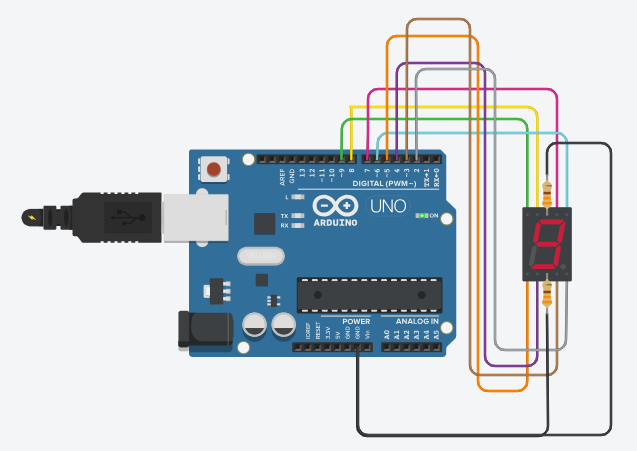
****

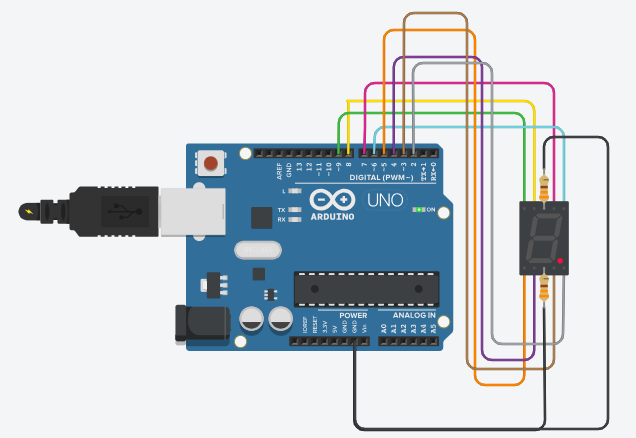
** **

** **

** **

** **

** **

****

**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**

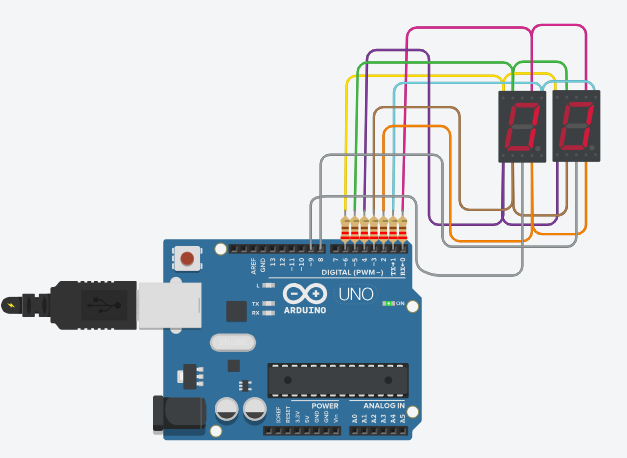
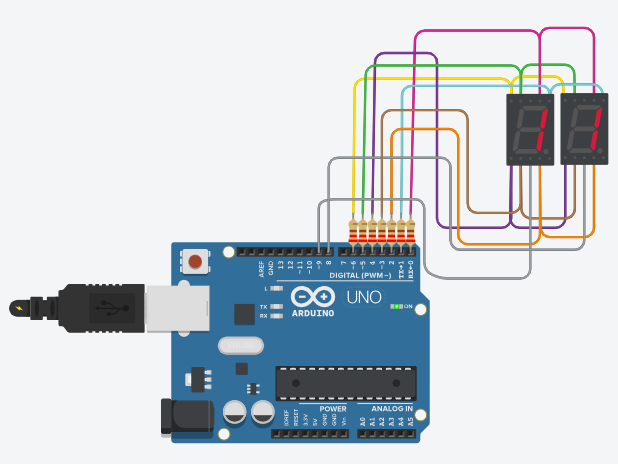
**brne outer\_loop**

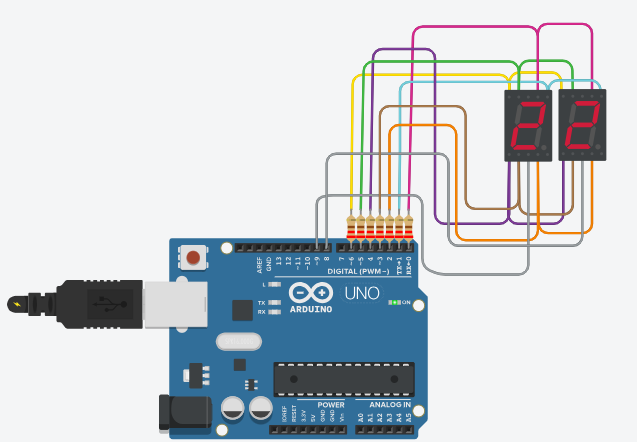
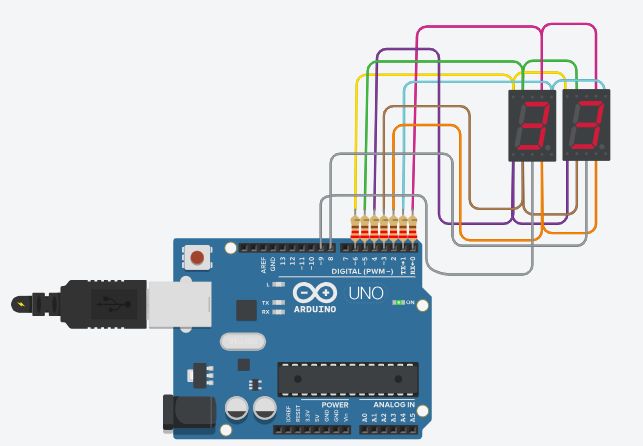
**ret**

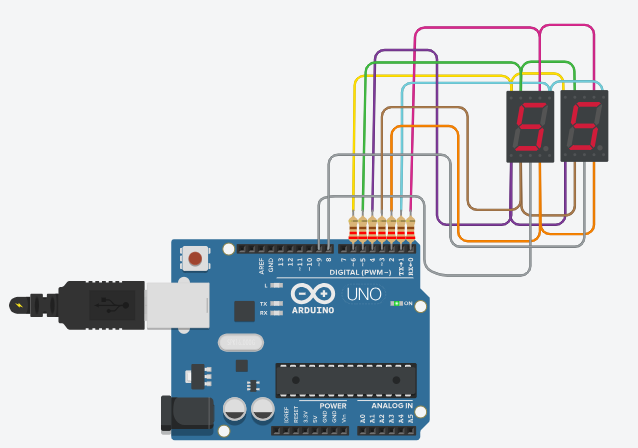
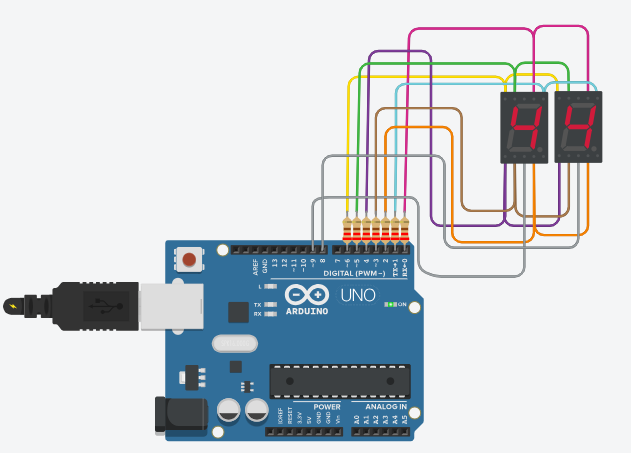
**.end**

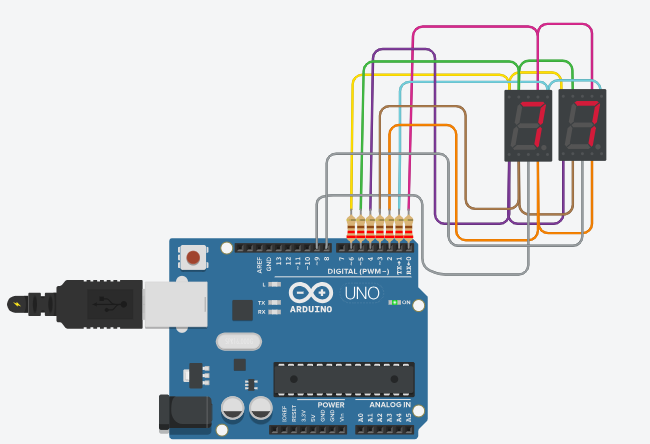
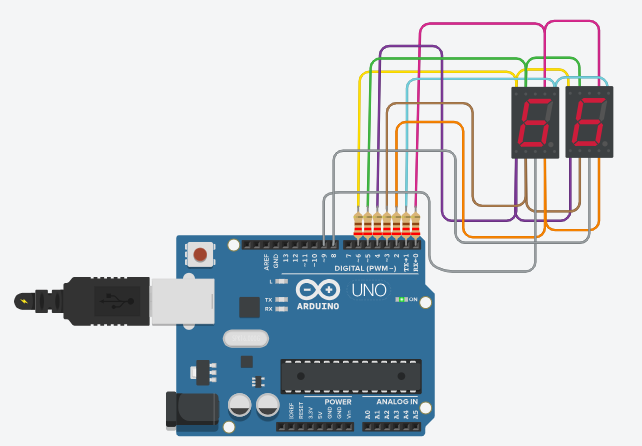
1. **Draw block diagram of two seven segment display**

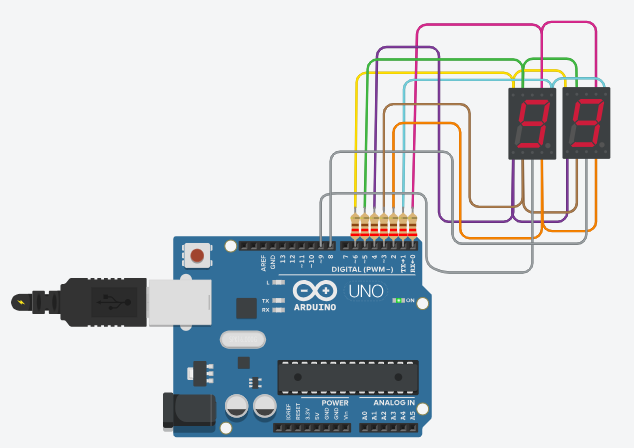
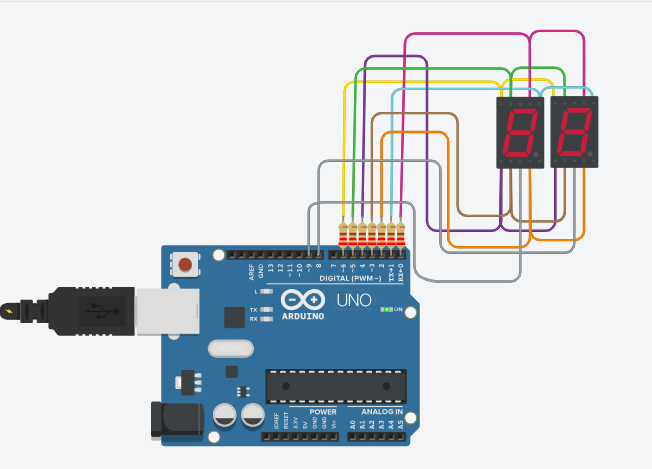
**Write a program in Assembly and C to display a digit from 0 to 9 in both seven segment display**

** **

** **

****

****

****

**C PROGRAM:**

**//Pin for 7 segment**

**char A=0,B=1,C=2,D=3,E=4,F=5,G=6;**

**//Digit**

**char digit1=9;**

**char digit2=8;**

**char myNumber[10] = {**

**0b0111111, //0**

**0b0000110, //1**

**0b1011011, //2**

**0b1001111, //3**

**0b1100110, //4**

**0b1101101, //5**

**0b1111101, //6**

**0b0000111, //7**

**0b1111111, //8**

**0b1101111 //9**

**};**

**void setup(){**

**pinMode(A,OUTPUT);**

**pinMode(B,OUTPUT);**

**pinMode(C,OUTPUT);**

**pinMode(D,OUTPUT);**

**pinMode(E,OUTPUT);**

**pinMode(F,OUTPUT);**

**pinMode(G,OUTPUT);**

**pinMode(digit1,OUTPUT);**

**pinMode(digit2,OUTPUT);**

**}**

**int count=0;**

**void loop() {**

**// Display "0" on the right display (digit2)**

**digitalWrite(digit1, 0);**

**digitalWrite(digit2, 0);**

**number(myNumber[count]);**

**delay(1000);**

**count++;**

**if(count==10){**

**count=0;**

**}**

**}**

**void number(int num){**

**digitalWrite(A,num & 0b0000001);**

**digitalWrite(B,num & 0b0000010);**

**digitalWrite(C,num & 0b0000100);**

**digitalWrite(D,num & 0b0001000);**

**digitalWrite(E,num & 0b0010000);**

**digitalWrite(F,num & 0b0100000);**

**digitalWrite(G,num & 0b1000000);**

**}**

**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**

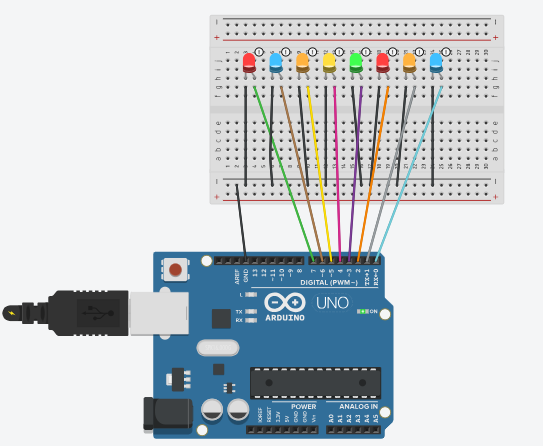
**rjmp loop**

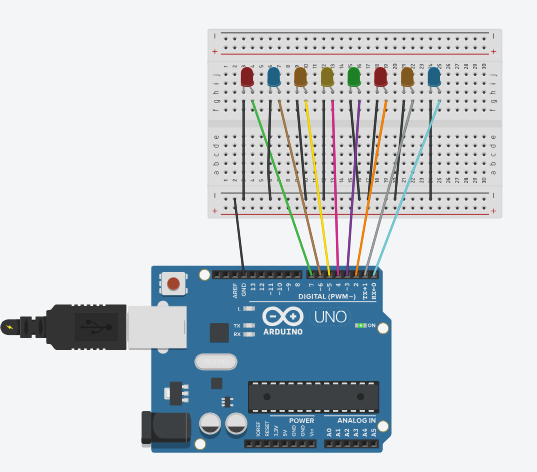
1. **Write a code of LED interface in Assembly and C through different P1, P2, P3**

**Block diagram of 8 LEDs**

1. **Perform Rotation operation**

**CIRCUIT**

****

****

**C CODE**

**void setup()**

**{**

**pinMode(7, OUTPUT);**

**pinMode(6, OUTPUT);**

**pinMode(5, OUTPUT);**

**pinMode(4, OUTPUT);**

**pinMode(3, OUTPUT);**

**pinMode(2, OUTPUT);**

**pinMode(1, OUTPUT);**

**pinMode(0, OUTPUT);**

**}**

**void loop()**

**{**

**digitalWrite(7, HIGH);**

**digitalWrite(6, HIGH);**

**digitalWrite(5, HIGH);**

**digitalWrite(4, HIGH);**

**digitalWrite(3, HIGH);**

**digitalWrite(2, HIGH);**

**digitalWrite(1, HIGH);**

**digitalWrite(0, HIGH);**

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

**digitalWrite(7, LOW);**

**digitalWrite(6, LOW);**

**digitalWrite(5, LOW);**

**digitalWrite(4, LOW);**

**digitalWrite(3, LOW);**

**digitalWrite(2, LOW);**

**digitalWrite(1, LOW);**

**digitalWrite(0, LOW);**

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

**}**

**ASSEMBLY CODE**

**.include "m328pdef.inc" ; Include the ATmega328P register definitions**

**.text**

**.global main**

**main:**

**; Initialize stack pointer**

**ldi r16, high(RAMEND)**

**out SPH, r16**

**ldi r16, low(RAMEND)**

**out SPL, r16**

**; Set up I/O pins as outputs**

**ldi r16, 0xFF ; Initialize r16 with 0xFF (all pins set as outputs)**

**out DDRD, r16 ; Port D (pin 7 to 0) as outputs**

**loop:**

**; Set all pins to HIGH**

**ldi r16, 0xFF ; Initialize r16 with 0xFF (all pins set to HIGH)**

**out PORTD, r16 ; Set all pins on Port D to HIGH**

**call delay\_ms**

**; Set all pins to LOW**

**ldi r16, 0x00 ; Initialize r16 with 0x00 (all pins set to LOW)**

**out PORTD, r16 ; Set all pins on Port D to LOW**

**call delay\_ms**

**rjmp loop ; Repeat the loop**

**delay\_ms:**

**; Delay for approximately 1000 milliseconds**

**ldi r17, 250 ; Load the outer loop counter (about 4ms per iteration)**

**outerLoop:**

**ldi r18, 250 ; Load the inner loop counter**

**innerLoop:**

**dec r18 ; Decrement inner loop counter**

**brne innerLoop ; If not zero, repeat inner loop**

**dec r17 ; Decrement outer loop counter**

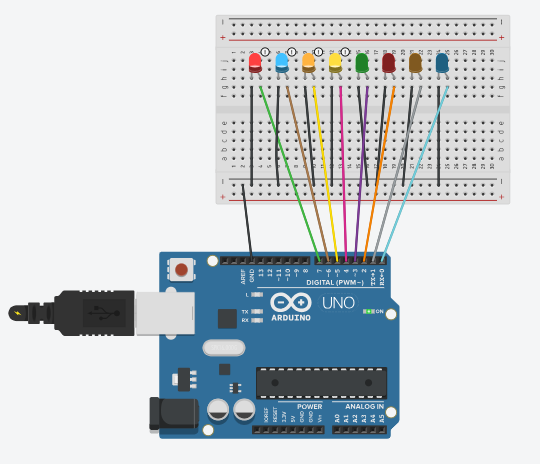
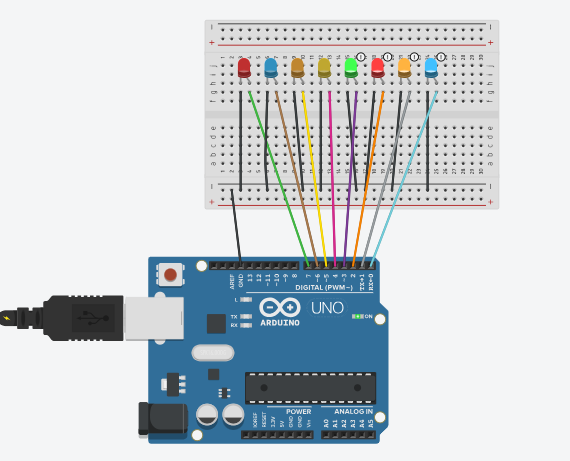
**brne outerLoop ; If not zero, repeat outer loop**

**ret**

**.end**

**2. Perform four on and off**

**CIRCUIT**

****

**C CODE**

**void setup()**

**{**

**pinMode(7, OUTPUT);**

**pinMode(6, OUTPUT);**

**pinMode(5, OUTPUT);**

**pinMode(4, OUTPUT);**

**pinMode(3, OUTPUT);**

**pinMode(2, OUTPUT);**

**pinMode(1, OUTPUT);**

**pinMode(0, OUTPUT);**

**}**

**void loop()**

**{**

**digitalWrite(7, HIGH);**

**digitalWrite(6, HIGH);**

**digitalWrite(5, HIGH);**

**digitalWrite(4, HIGH);**

**digitalWrite(3, LOW);**

**digitalWrite(2, LOW);**

**digitalWrite(1, LOW);**

**digitalWrite(0, LOW);**

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

**digitalWrite(7, LOW);**

**digitalWrite(6, LOW);**

**digitalWrite(5, LOW);**

**digitalWrite(4, LOW);**

**digitalWrite(3, HIGH);**

**digitalWrite(2, HIGH);**

**digitalWrite(1, HIGH);**

**digitalWrite(0, HIGH);**

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

**}**

**ASSEMBLY CODE**

**.include "m328pdef.inc" ; Include the ATmega328P register definitions**

**.def temp = r16 ; Define a temporary register for general use**

**; Initialize the stack pointer and set the reset vector**

**ldi temp, high(RAMEND)**

**out SPH, temp**

**ldi temp, low(RAMEND)**

**out SPL, temp**

**; Set pins 0-7 as outputs in the DDRD register (Data Direction Register D)**

**ldi temp, 0xFF ; 0b11111111, all bits set to 1**

**out DDRD, temp**

**main:**

**; Set pins according to the pattern**

**ldi temp, 0b00111100 ; Initial pattern**

**out PORTD, temp**

**loop:**

**; Delay for 1000 milliseconds**

**ldi temp, 250 ; Delay count for 1 second (adjust based on clock speed)**

**delay\_loop1:**

**rcall delay\_ms**

**dec temp**

**brne delay\_loop1**

**; Invert the pattern**

**ldi temp, 0b11000011 ; Inverted pattern**

**out PORTD, temp**

**; Delay for 1000 milliseconds**

**ldi temp, 250 ; Delay count for 1 second (adjust based on clock speed)**

**delay\_loop2:**

**rcall delay\_ms**

**dec temp**

**brne delay\_loop2**

**rjmp loop ; Repeat the loop**

**; Delay function for approximately 1 ms**

**delay\_ms:**

**ldi temp, 250 ; Delay count for 1 ms (adjust based on clock speed)**

**delay\_us\_loop:**

**dec temp**

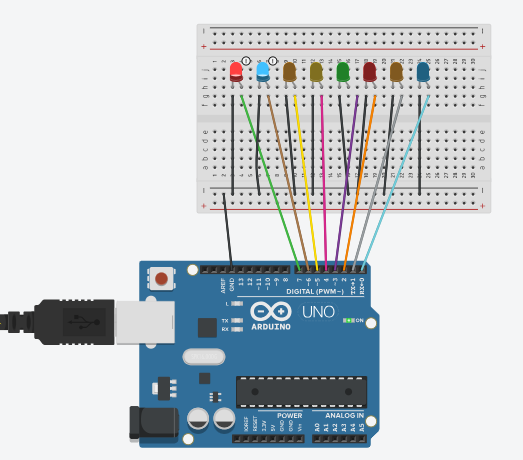
**brne delay\_us\_loop**

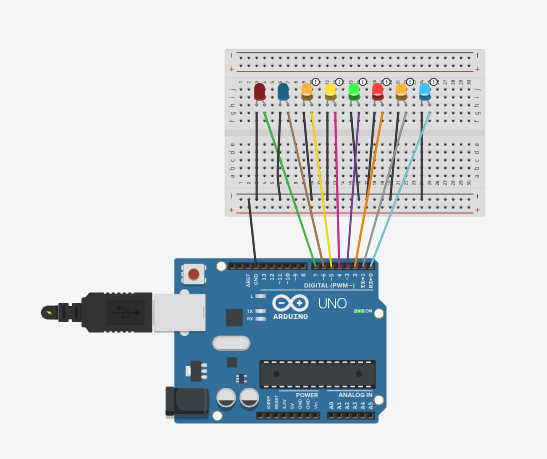
**ret**

**.end**

1. **Two on and six off**

**CIRCUIT**

****

****

**C CODE**

**void setup()**

**{**

**pinMode(7, OUTPUT);**

**pinMode(6, OUTPUT);**

**pinMode(5, OUTPUT);**

**pinMode(4, OUTPUT);**

**pinMode(3, OUTPUT);**

**pinMode(2, OUTPUT);**

**pinMode(1, OUTPUT);**

**pinMode(0, OUTPUT);**

**}**

**void loop()**

**{**

**digitalWrite(7, HIGH);**

**digitalWrite(6, HIGH);**

**digitalWrite(5, LOW);**

**digitalWrite(4, LOW);**

**digitalWrite(3, LOW);**

**digitalWrite(2, LOW);**

**digitalWrite(1, LOW);**

**digitalWrite(0, LOW);**

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

**digitalWrite(7, LOW);**

**digitalWrite(6, LOW);**

**digitalWrite(5, HIGH);**

**digitalWrite(4, HIGH);**

**digitalWrite(3, HIGH);**

**digitalWrite(2, HIGH);**

**digitalWrite(1, HIGH);**

**digitalWrite(0, HIGH);**

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

**}**

**ASSEMBLY CODE**

**.include "m328pdef.inc" ; Include the ATmega328P register definitions**

**.def temp = r16 ; Define a temporary register for general use**

**; Define constants**

**.equ DELAY\_COUNT = 977 ; Delay count for 1000 ms (adjust based on clock speed)**

**; Initialize the stack pointer and set the reset vector**

**ldi temp, high(RAMEND)**

**out SPH, temp**

**ldi temp, low(RAMEND)**

**out SPL, temp**

**main:**

**; Set pins 0-7 as outputs**

**ldi temp, 0xFF ; Set all bits to 1**

**out DDRD, temp**

**loop:**

**; Set pins according to the pattern**

**ldi temp, 0b11000000 ; Initial pattern**

**out PORTD, temp**

**; Delay for 1000 milliseconds**

**ldi temp, DELAY\_COUNT**

**delay\_loop1:**

**rcall delay\_ms**

**dec temp**

**brne delay\_loop1**

**; Invert the pattern**

**ldi temp, 0b00111111 ; Inverted pattern**

**out PORTD, temp**

**; Delay for 1000 milliseconds**

**ldi temp, DELAY\_COUNT**

**delay\_loop2:**

**rcall delay\_ms**

**dec temp**

**brne delay\_loop2**

**rjmp loop ; Repeat the loop**

**; Delay function for approximately 1 ms**

**delay\_ms:**

**ldi temp, 4 ; 4 \* 250 µs = 1 ms**

**delay\_ms\_loop:**

**ldi temp, 250 ; Load inner loop count for 250 µs**

**delay\_us\_loop:**

**dec temp**

**brne delay\_us\_loop**

**dec temp**

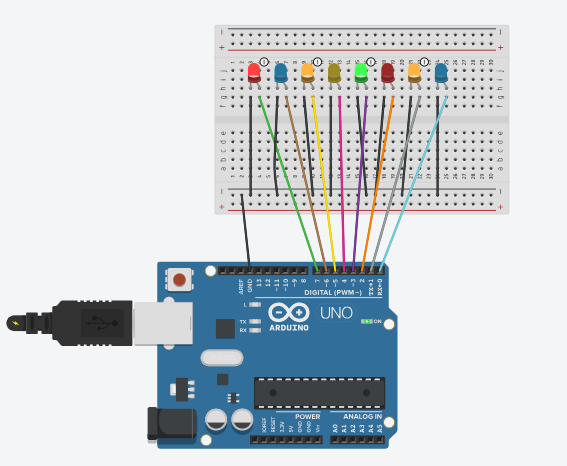
**brne delay\_ms\_loop**

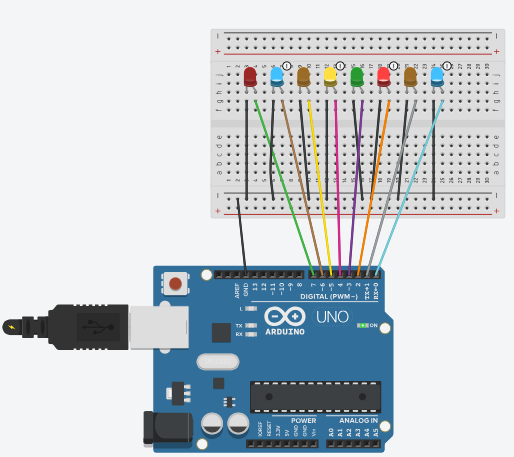
**ret**

**.end**

1. **Blink alternate LEDs at P3 using software delay**

**CIRCUIT**

****

****

**C CODE**

**void setup()**

**{**

**pinMode(7, OUTPUT);**

**pinMode(6, OUTPUT);**

**pinMode(5, OUTPUT);**

**pinMode(4, OUTPUT);**

**pinMode(3, OUTPUT);**

**pinMode(2, OUTPUT);**

**pinMode(1, OUTPUT);**

**pinMode(0, OUTPUT);**

**}**

**void loop()**

**{**

**digitalWrite(7, HIGH);**

**digitalWrite(6, LOW);**

**digitalWrite(5,HIGH);**

**digitalWrite(4, LOW);**

**digitalWrite(3, HIGH);**

**digitalWrite(2, LOW);**

**digitalWrite(1, HIGH);**

**digitalWrite(0, LOW);**

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

**digitalWrite(7, LOW);**

**digitalWrite(6, HIGH);**

**digitalWrite(5, LOW);**

**digitalWrite(4,HIGH);**

**digitalWrite(3, LOW);**

**digitalWrite(2, HIGH);**

**digitalWrite(1, LOW);**

**digitalWrite(0, HIGH);**

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

**}**

**ASSEMBLY CODE**

**; Define register addresses for ATmega328P**

**.equ DDRC, 0x27 ; Data Direction Register for Port C**

**.equ PORTC, 0x28 ; Port C Output Register**

**; Define constants**

**.equ DELAY\_COUNT, 250**

**; Initialize the stack pointer and set the reset vector**

**ldi r16, high(RAMEND)**

**out SPH, r16**

**ldi r16, low(RAMEND)**

**out SPL, r16**

**main:**

**; Set pins 0-7 as outputs**

**ldi r16, 0xFF**

**out DDRC, r16**

**loop:**

**; Set pins according to the pattern**

**ldi r16, 0b10101010 ; Initial pattern**

**out PORTC, r16**

**; Delay for 1000 milliseconds**

**ldi r17, DELAY\_COUNT**

**delay\_loop1:**

**ldi r18, 200**

**delay\_loop2:**

**dec r18**

**brne delay\_loop2**

**dec r17**

**brne delay\_loop1**

**; Invert the pattern**

**ldi r16, 0b01010101 ; Inverted pattern**

**out PORTC, r16**

**; Delay for 1000 milliseconds**

**ldi r17, DELAY\_COUNT**

**delay\_loop3:**

**ldi r18, 200**

**delay\_loop4:**

**dec r18**

**brne delay\_loop4**

**dec r17**

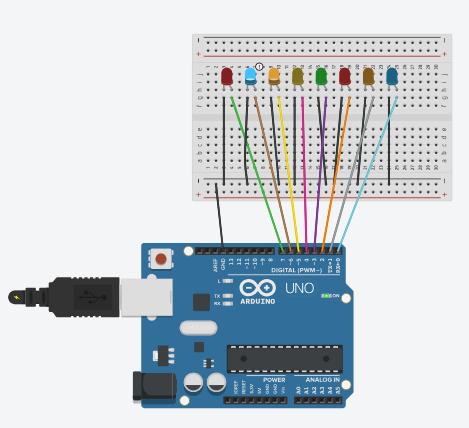
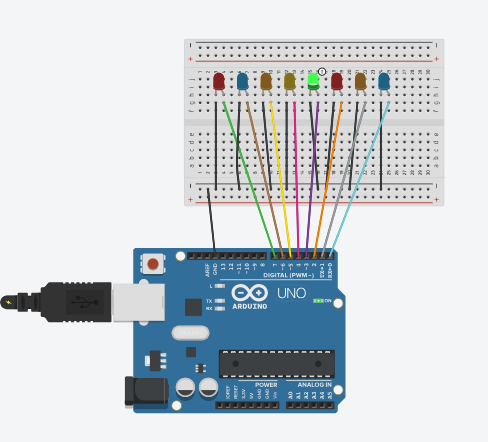
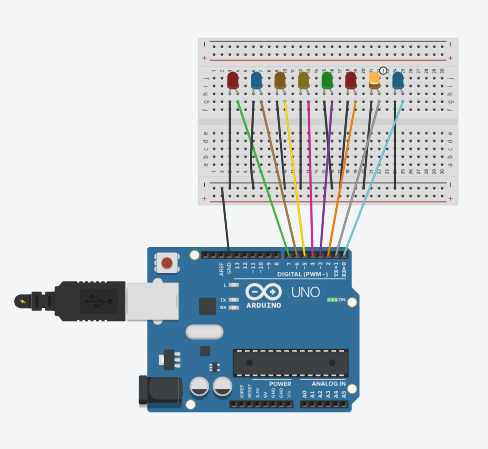
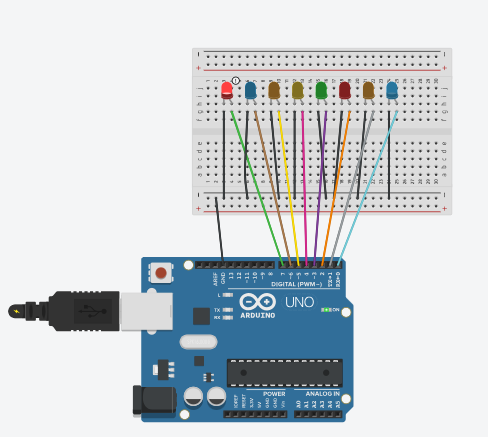
**brne delay\_loop3**

**rjmp loop ; Repeat the loop**

**; End of program**

1. **Blink P0 LEDs in cyclic fashion using software delay**

**CIRCUIT**

****

**C CODE**

**int pins[] = {7, 6, 5, 4, 3, 2, 1, 0};**

**int numPins = sizeof(pins) / sizeof(pins[0]);**

**int currentIndex = 0;**

**void setup() {**

**for (int i = 0; i < numPins; i++) {**

**pinMode(pins[i], OUTPUT);**

**}**

**}**

**void loop() {**

**digitalWrite(pins[currentIndex], HIGH);**

**delay(1000);**

**digitalWrite(pins[currentIndex], LOW);**

**currentIndex = (currentIndex + 1) % numPins;**

**ASSEMBLY CODE**

**; Define constants**

**NUM\_PINS = 8**

**.equ PINS\_PORT, 0x02 ; Replace with the actual port register address**

**.equ PINS\_DDR, 0x03 ; Replace with the actual DDR register address**

**; Initialize the stack pointer and set the reset vector**

**ldi r16, high(RAMEND)**

**out SPH, r16**

**ldi r16, low(RAMEND)**

**out SPL, r16**

**main:**

**; Initialize pin modes (All pins as OUTPUT)**

**ldi r16, (1 << PINS\_DDR)**

**out PINS\_DDR, r16**

**loop:**

**; Set the current pin to HIGH**

**ldi r16, (1 << (currentIndex + PINS\_PORT))**

**out PINS\_PORT, r16**

**; Delay for 1000 milliseconds (implement your delay logic)**

**; This will depend on your specific microcontroller and timer setup**

**; Set the current pin to LOW**

**ldi r16, (1 << PINS\_PORT)**

**out PINS\_PORT, r16**

**; Update the current index**

**adiw r24, 1 ; Increment currentIndex**

**cpi r24, NUM\_PINS ; Compare with NUM\_PINS**

**brne loop ; If not equal, repeat the loop**

**; Implement delay for transitioning to the next pin**

**rjmp loop**

**.equ NUM\_PINS, 8**

**.equ currentIndex, 0**

**.equ delay\_loop\_count, 250 ; Adjust this for your specific delay**

**.equ delay\_ms\_loop, 4**

**delay:**

**ldi r24, delay\_ms\_loop**

**delay\_ms\_loop\_start:**

**ldi r25, delay\_loop\_count**

**delay\_loop\_start:**

**dec r25**

**brne delay\_loop\_start**

**dec r24**

**brne delay\_ms\_loop\_start**

**ret**

**THE END**