Interfacing Seven Segment Display with MSP430 MCU

Spring 2025

Submitted by: **Mohsin Sajjad**

Registration No: **22pwsce2149**

Class Section: **A**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Mohsin Sajjad

Student Signature: ______________

Submitted to:

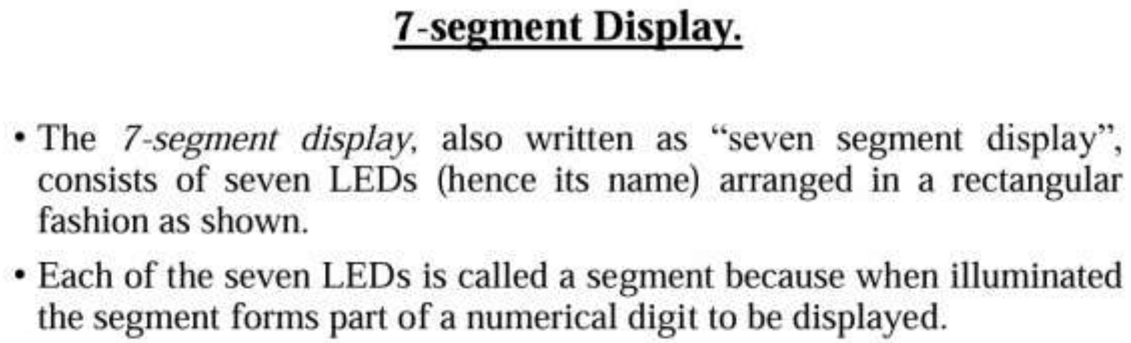
Engr. Faheem Jan

Month Day, Year (02 03, 2025)

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

LAB No 4



7-segment Display.

- The *7-segment display*, also written as “seven segment display”, consists of seven LEDs (hence its name) arranged in a rectangular fashion as shown.
- Each of the seven LEDs is called a segment because when illuminated the segment forms part of a numerical digit to be displayed.

7-segment Display

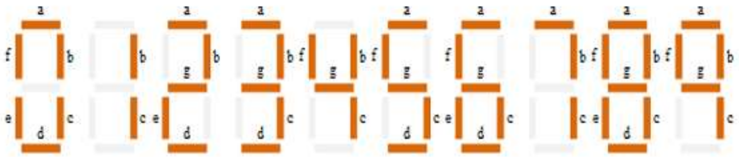
Common Cathode:

In a common cathode (CC) 7-segment display, all the cathodes of the LEDs are connected to ground (0V). To turn on a segment, you apply a HIGH (1) signal to the corresponding anode.

Common Anode:

In a **common anode (CA) 7-segment display**, all anodes are connected to **Vcc (power supply)**. To turn on a segment, you apply a **LOW (0)** signal to the corresponding cathode.

Digital Segments for all Numbers

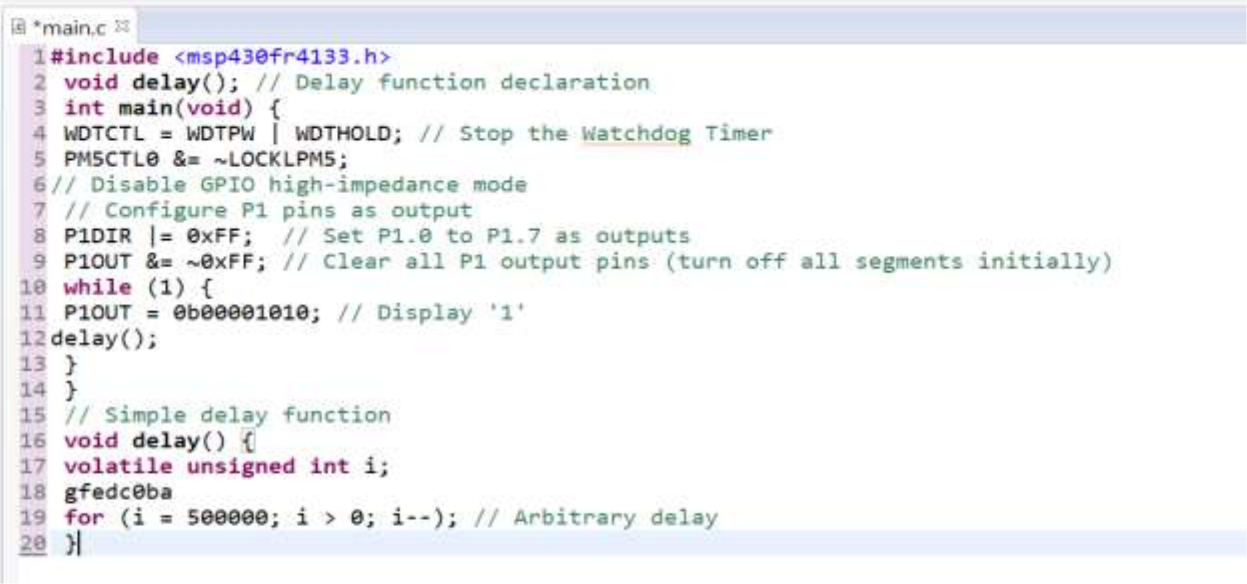


TASKS:

TASK1:

Run all the program given in the lecture.

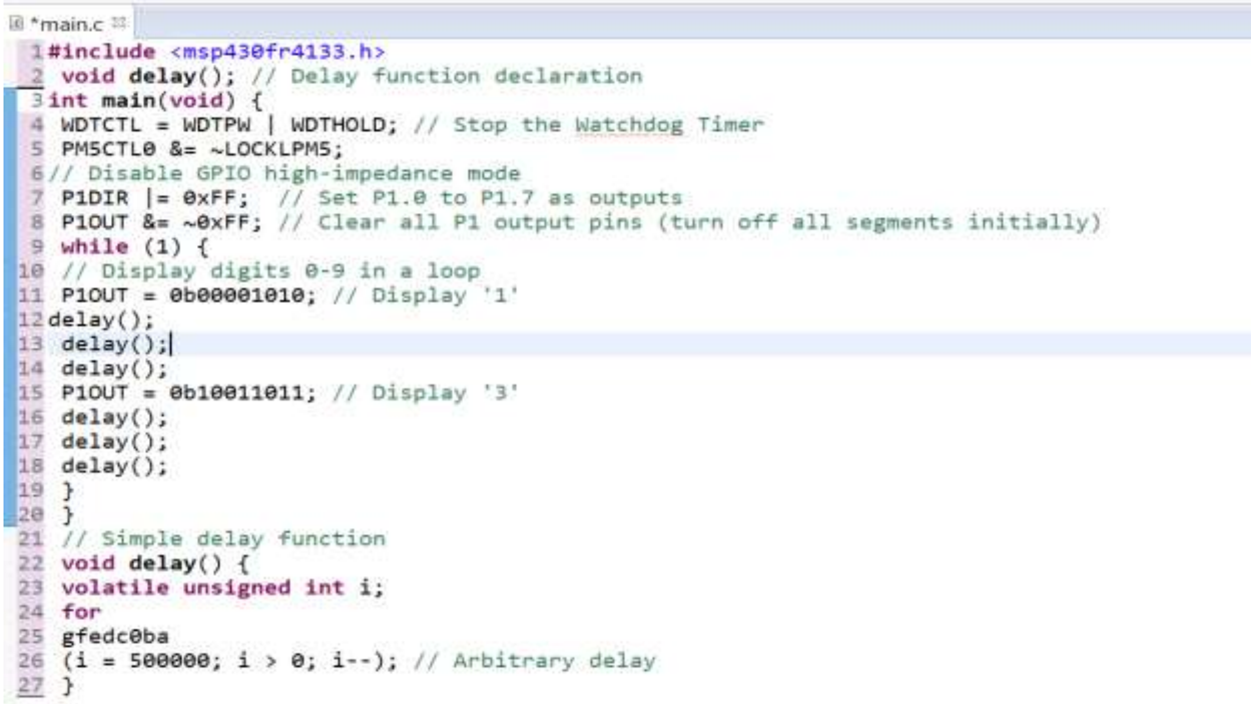
Display 1:



```c
#include <msp430fr4133.h>
 void delay(); // Delay function declaration
 int main(void) {
 WDTCTL = WDTPW | WDTHOLD; // Stop the Watchdog Timer
 PM5CTL0 &= ~LOCKLPM5;
// Disable GPIO high-impedance mode
 // Configure P1 pins as output
 P1DIR |= 0xFF;  // Set P1.0 to P1.7 as outputs
 P1OUT &= ~0xFF; // Clear all P1 output pins (turn off all segments initially)
 while (1) {
 P1OUT = 0b00001010; // Display '1'
delay();
 }
 }
 // Simple delay function
 void delay() {
 volatile unsigned int i;
 gfedc0ba
 for (i = 500000; i > 0; i--); // Arbitrary delay
 }
```

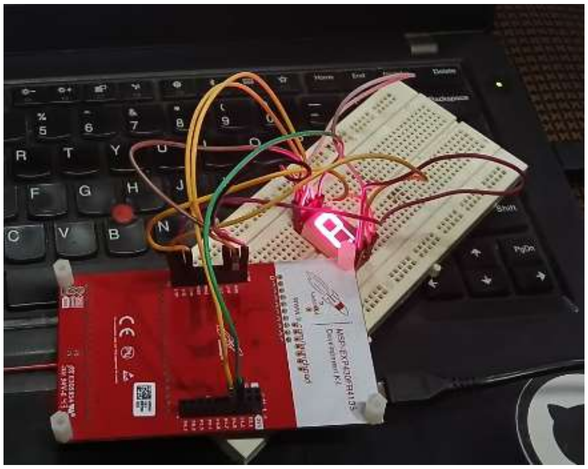
Output:

Display 1 and 3:

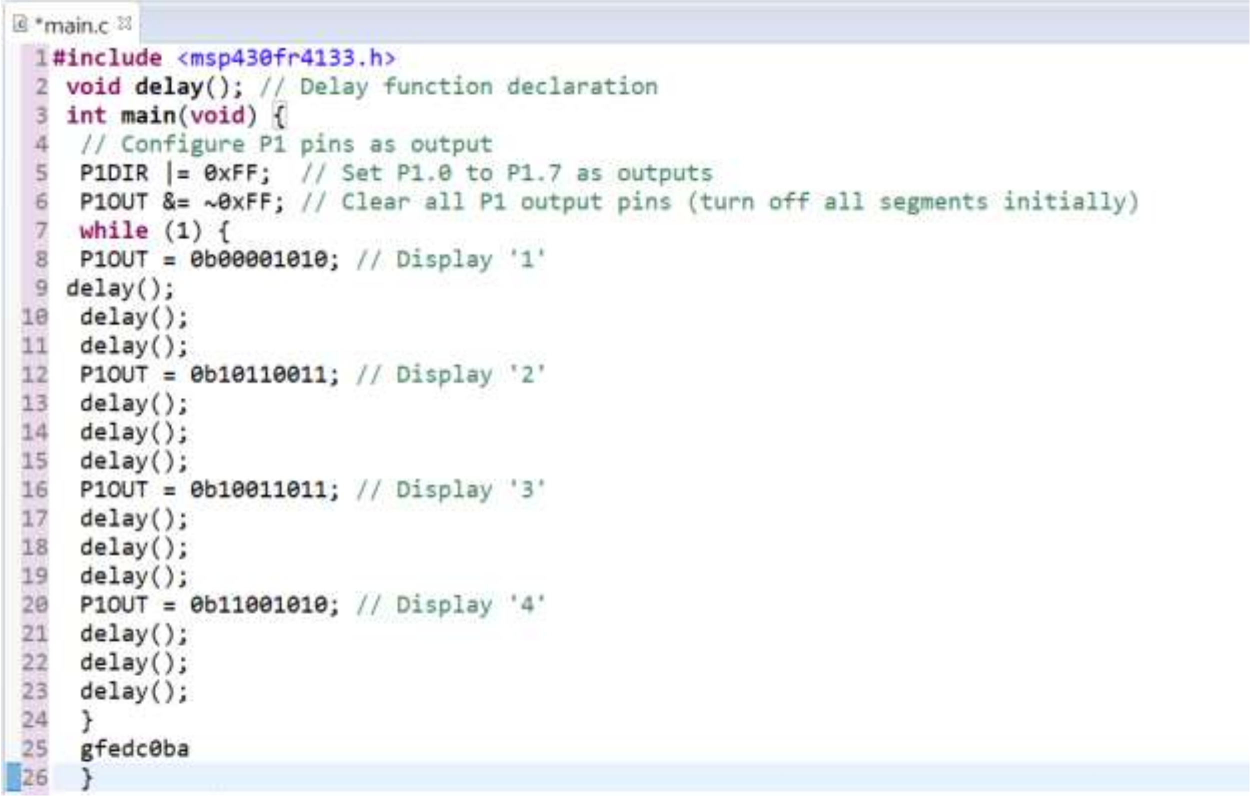


```c
#include <msp430fr4133.h>
 void delay(); // Delay function declaration
int main(void) {
 WDTCTL = WDTPW | WDTHOLD; // Stop the Watchdog Timer
 PM5CTL0 &= ~LOCKLPM5;
// Disable GPIO high-impedance mode
 P1DIR |= 0xFF;  // Set P1.0 to P1.7 as outputs
 P1OUT &= ~0xFF; // Clear all P1 output pins (turn off all segments initially)
 while (1) {
 // Display digits 0-9 in a loop
 P1OUT = 0b00001010; // Display '1'
delay();
 delay();
 delay();
 P1OUT = 0b10011011; // Display '3'
 delay();
 delay();
 delay();
 }
 }
 // Simple delay function
 void delay() {
 volatile unsigned int i;
 for
 gfedc0ba
 (i = 500000; i > 0; i--); // Arbitrary delay
 }
```

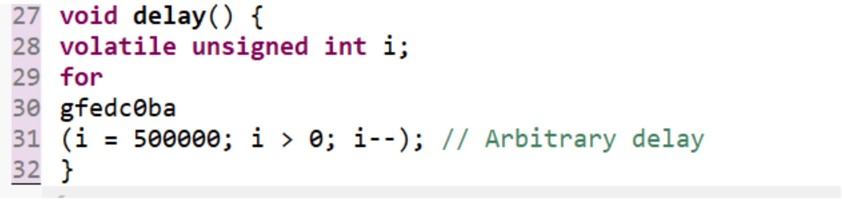
Output:



Display 1 ,2 ,3 ,4:

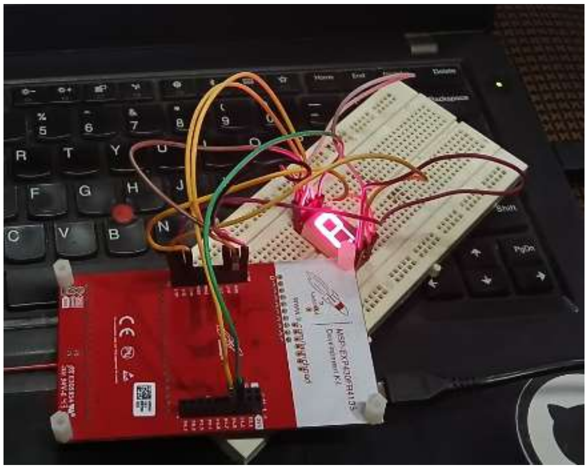


```
#include <msp430fr4133.h>
 void delay(); // Delay function declaration
 int main(void) {
  // Configure P1 pins as output
  P1DIR |= 0xFF;  // Set P1.0 to P1.7 as outputs
  P1OUT &= ~0xFF; // Clear all P1 output pins (turn off all segments initially)
  while (1) {
  P1OUT = 0b00001010; // Display '1'
 delay();
  delay();
  delay();
  P1OUT = 0b10110011; // Display '2'
  delay();
  delay();
  delay();
  P1OUT = 0b10011011; // Display '3'
  delay();
  delay();
  delay();
  P1OUT = 0b11001010; // Display '4'
  delay();
  delay();
  delay();
  }
  gfedc0ba
  }
```



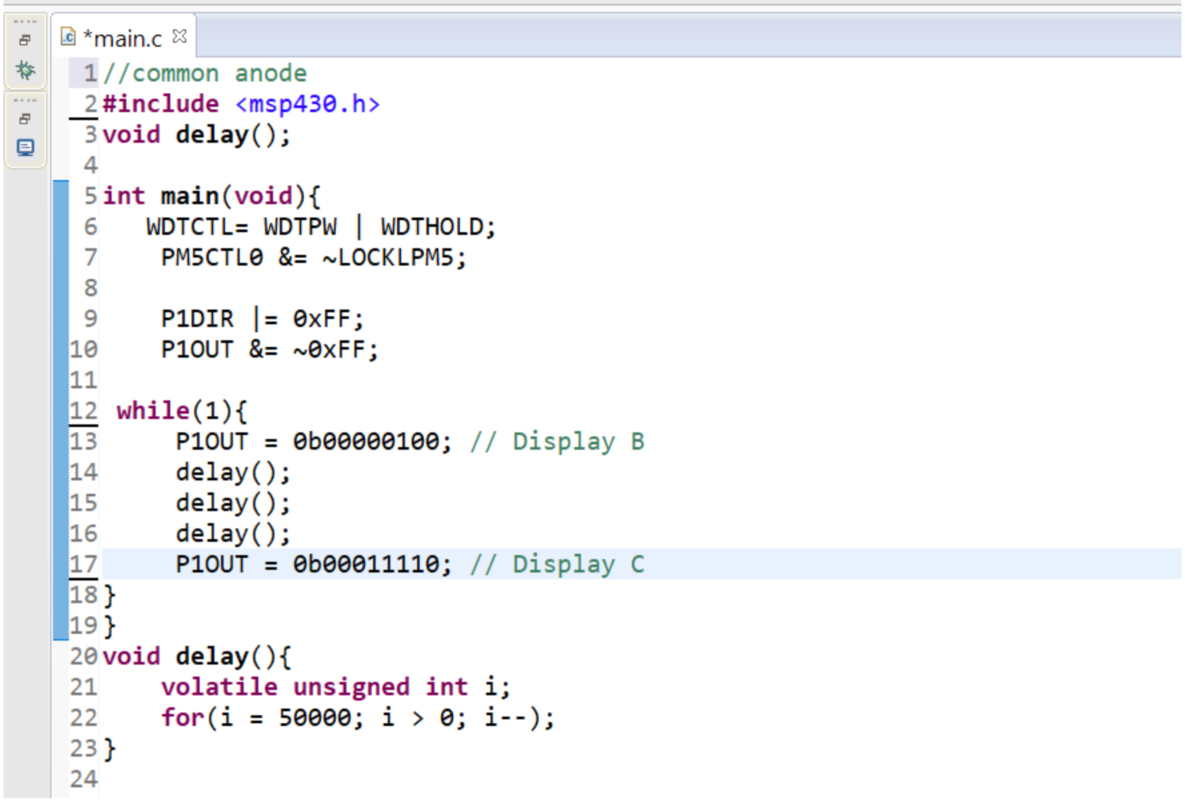
```
void delay() {
volatile unsigned int i;
for
gfedc0ba
(i = 500000; i > 0; i--); // Arbitrary delay
}
```

output:



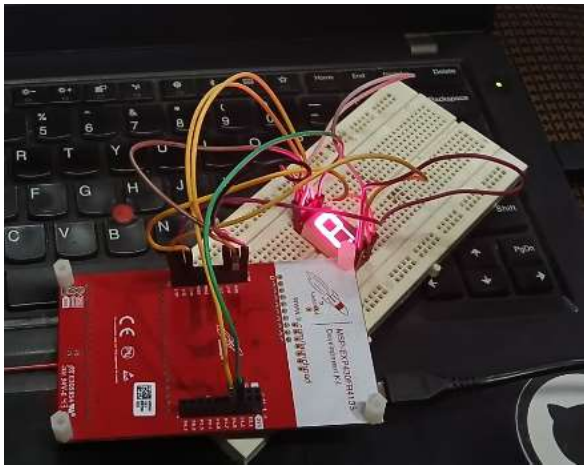
Task 02:

Display different digits on the Seven Segment Display.



```
//common anode
#include <msp430.h>
void delay();

int main(void){
   WDTCTL= WDTPW | WDTHOLD;
    PM5CTL0 &= ~LOCKLPM5;

    P1DIR |= 0xFF;
    P1OUT &= ~0xFF;

 while(1){
     P1OUT = 0b00000100; // Display B
     delay();
     delay();
     delay();
     P1OUT = 0b00011110; // Display C
}
}
void delay(){
    volatile unsigned int i;
    for(i = 50000; i > 0; i--);
}

```

Output:



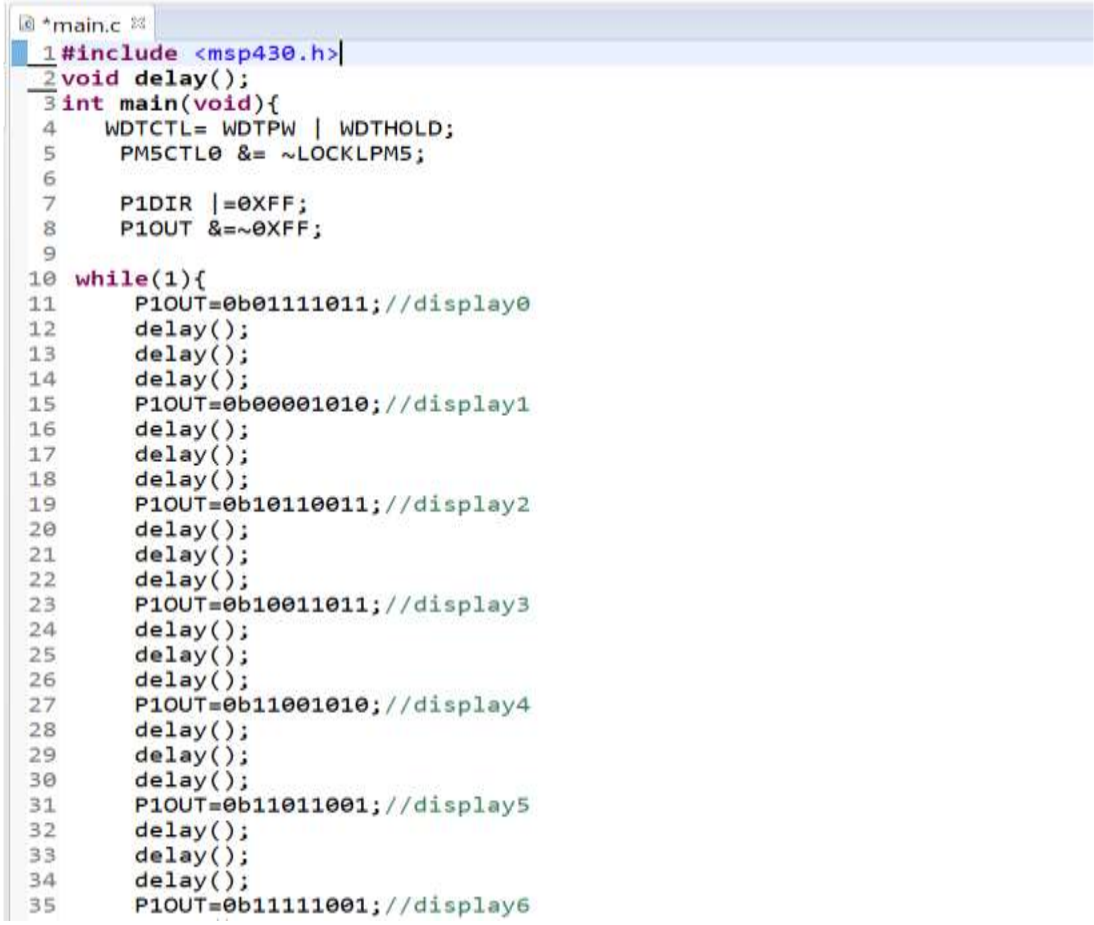
Task 03:

Display digits from 0 to F on Seven Segment Display.

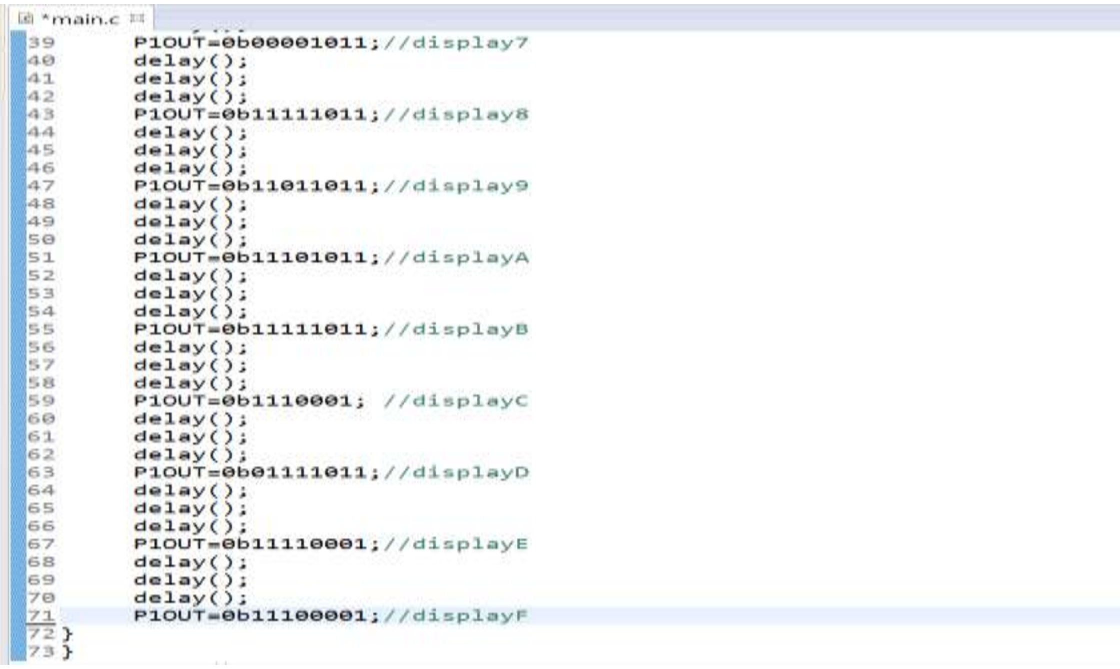
Use Both Common Cathode and Common Anode Seven-Segment Display.

Common Cathode:

Code:

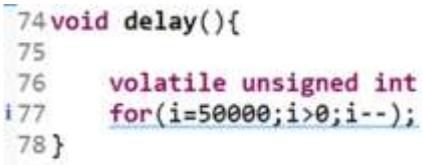


```
#include <msp430.h>
void delay();
int main(void){
   WDTCTL= WDTPW | WDTHOLD;
    PM5CTL0 &= ~LOCKLPM5;

    P1DIR |=0XFF;
    P1OUT &=~0XFF;

 while(1){
     P1OUT=0b01111011;//display0
     delay();
     delay();
     delay();
     P1OUT=0b00001010;//display1
     delay();
     delay();
     delay();
     P1OUT=0b10110011;//display2
     delay();
     delay();
     delay();
     P1OUT=0b10011011;//display3
     delay();
     delay();
     delay();
     P1OUT=0b11001010;//display4
     delay();
     delay();
     delay();
     P1OUT=0b11011001;//display5
     delay();
     delay();
     delay();
     P1OUT=0b11111001;//display6
```



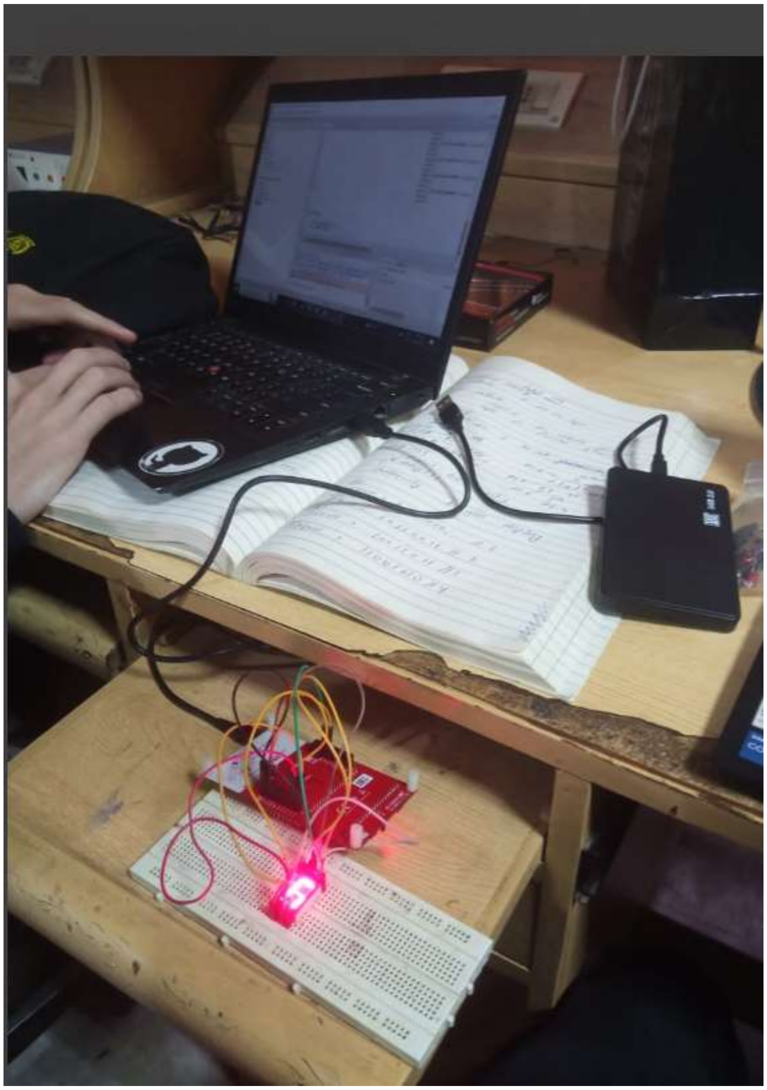
*main.c

```
        P1OUT=0b00001011;//display7
        delay();
        delay();
        delay();
        P1OUT=0b11111011;//display8
        delay();
        delay();
        delay();
        P1OUT=0b11011011;//display9
        delay();
        delay();
        delay();
        P1OUT=0b11101011;//displayA
        delay();
        delay();
        delay();
        P1OUT=0b11111011;//displayB
        delay();
        delay();
        delay();
        P1OUT=0b1110001; //displayC
        delay();
        delay();
        delay();
        P1OUT=0b01111011;//displayD
        delay();
        delay();
        delay();
        P1OUT=0b11110001;//displayE
        delay();
        delay();
        delay();
        P1OUT=0b11100001;//displayF
}
}
```



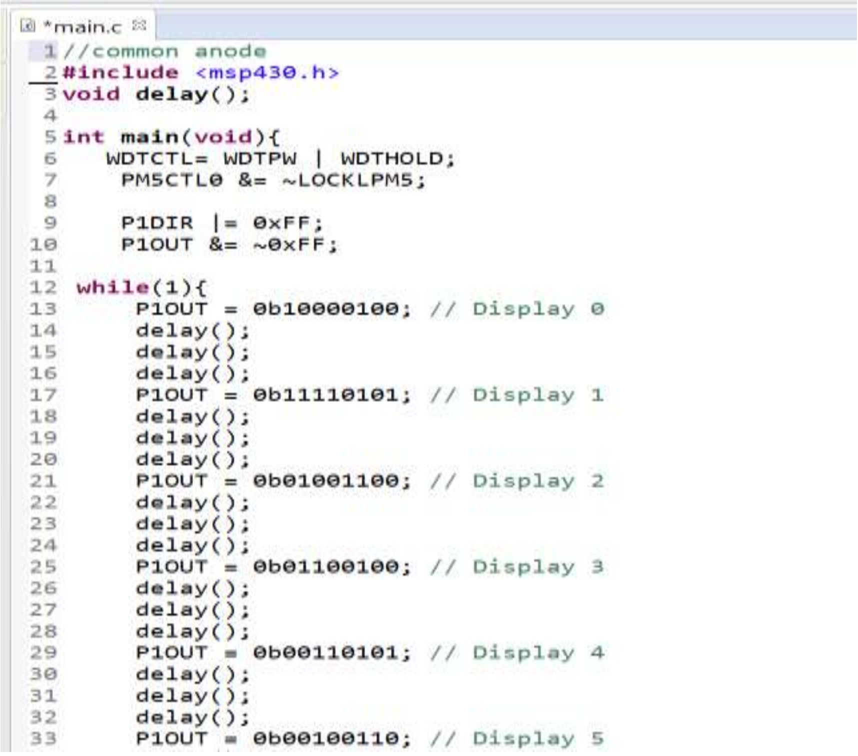
```
void delay(){

    volatile unsigned int i;
    for(i=50000;i>0;i--);
}
```

Output:

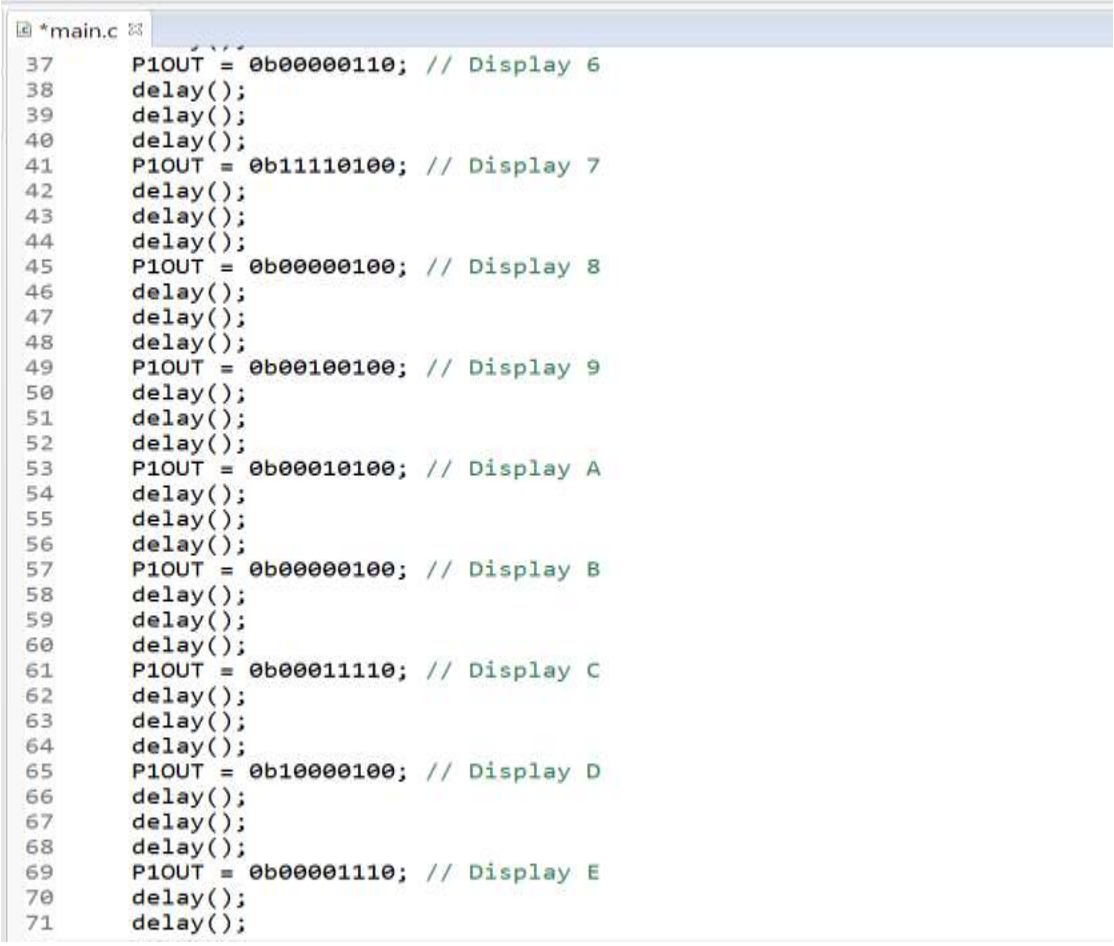


Common Anode:

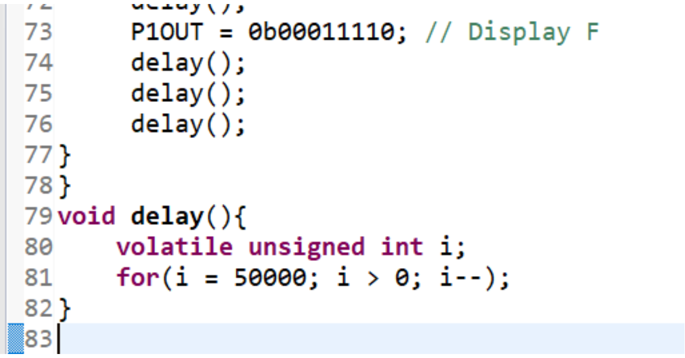
Code:



```
//common anode
#include <msp430.h>
void delay();

int main(void){
    WDTCTL= WDTPW | WDTHOLD;
     PM5CTL0 &= ~LOCKLPM5;

     P1DIR |= 0xFF;
     P1OUT &= ~0xFF;

 while(1){
       P1OUT = 0b10000100; // Display 0
       delay();
       delay();
       delay();
       P1OUT = 0b11110101; // Display 1
       delay();
       delay();
       delay();
       P1OUT = 0b01001100; // Display 2
       delay();
       delay();
       delay();
       P1OUT = 0b01100100; // Display 3
       delay();
       delay();
       delay();
       P1OUT = 0b00110101; // Display 4
       delay();
       delay();
       delay();
       P1OUT = 0b00100110; // Display 5
```



```
       P1OUT = 0b00000110; // Display 6
       delay();
       delay();
       delay();
       P1OUT = 0b11110100; // Display 7
       delay();
       delay();
       delay();
       P1OUT = 0b00000100; // Display 8
       delay();
       delay();
       delay();
       P1OUT = 0b00100100; // Display 9
       delay();
       delay();
       delay();
       P1OUT = 0b00010100; // Display A
       delay();
       delay();
       delay();
       P1OUT = 0b00000100; // Display B
       delay();
       delay();
       delay();
       P1OUT = 0b00011110; // Display C
       delay();
       delay();
       delay();
       P1OUT = 0b10000100; // Display D
       delay();
       delay();
       delay();
       P1OUT = 0b00001110; // Display E
       delay();
       delay();
```



```
     delay();
     P1OUT = 0b00011110; // Display F
     delay();
     delay();
     delay();
}
}
void delay(){
    volatile unsigned int i;
    for(i = 50000; i > 0; i--);
}

```

OUTPUT: