CPE 301 - 1001

DESIGN ASSIGNMENT 3

The goal of the assignment is use GPIO and delays using Timers and Interrupts:

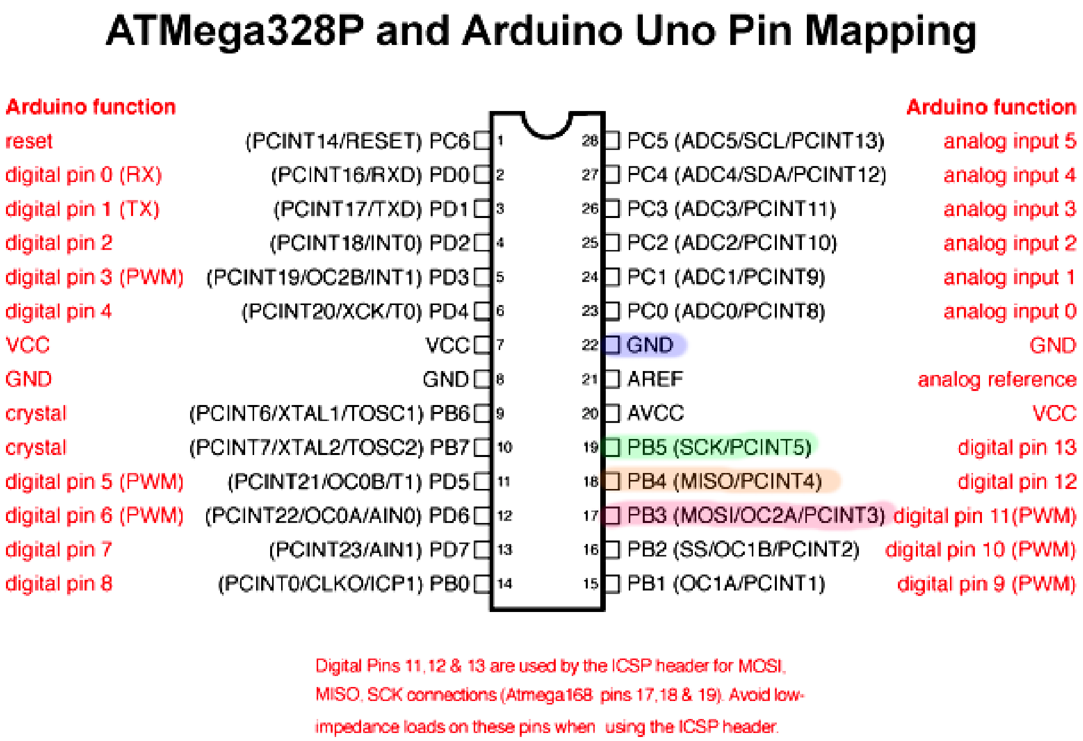
1. Generate three delays using three timers T0, T3, and T4.

    a. Implement a delay of 0.125ms using Timer 0 in normal mode. Count OVF occurrence if needed. Do not use interrupts. Turn 'on' PB5 LED (also monitor and verify using logic analyzer) for approx. 1.5 sec and 'off' for 1.5 sec.

    b. Implement a delay of 0.250ms using Timer 3 TIMER3_COMPA_vect interrupt mechanism in CTC mode. Count OVF occurrence if needed in the IRQ subroutine. Turn 'on' PB4 LED (also monitor and verify using logic analyzer) for approx. 2 sec and 'off' for 2 sec.

    c. Implement a delay of 0.100ms using Timer 4 TIMER4_OVF_vect interrupt mechanism in normal mode. Count OVF occurrence if needed in the IRQ subroutine. Turn 'on' PB3 LED (also monitor and verify using logic analyzer) for approx. 1 sec and 'off' for 1 sec.

Components Used/Connected



ATMega328P and Arduino Uno Pin Mapping

Arduino function					Arduino function
reset	(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)	analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)	analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)	analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)	analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)	analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)	analog input 0
VCC	VCC	7	22	GND	GND
GND	GND	8	21	AREF	analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC	VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)	digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)	digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)	digital pin 11(PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)	digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)	digital pin 9 (PWM)

Digital Pins 11,12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17,18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

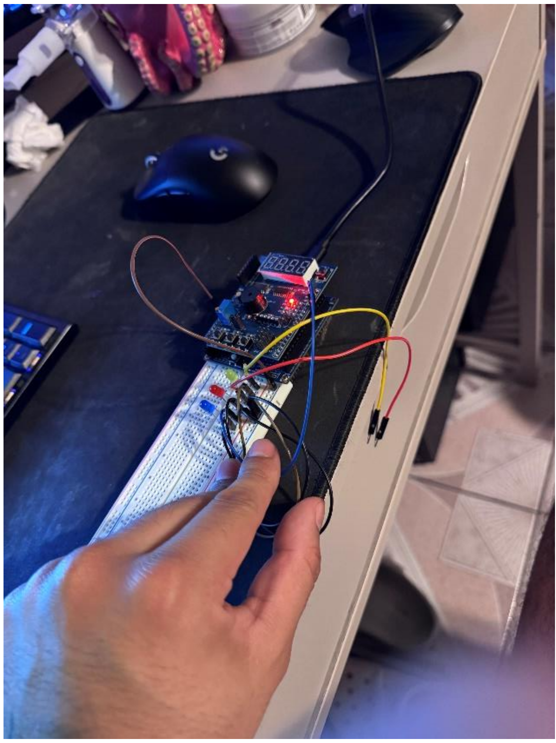
DA2

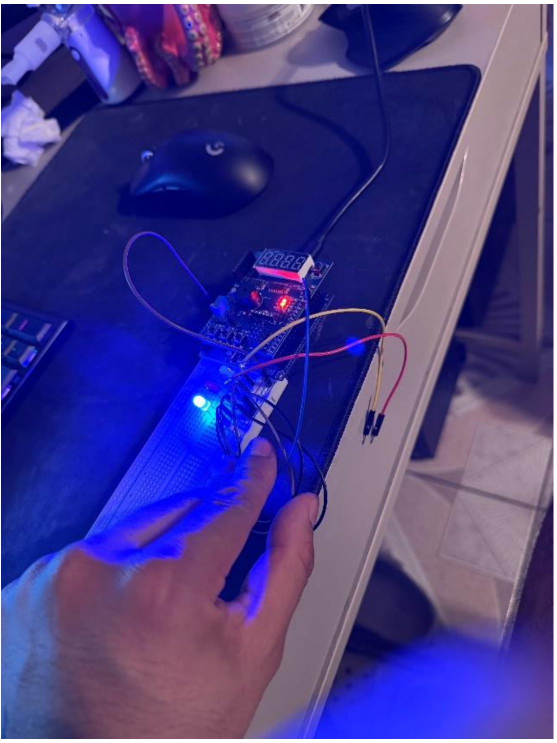
a. Timer0

```
// Timer0 polling
void Normal_Timer0(void)
{
        TCNT0 = 225;                                      // start count from 225
        TIFR0 |= (1 << TOV0);                             // clear overflow flag
        TCCR0B = (1 << CS01) | (1 << CS00);               // prescaler 64

        while (!(TIFR0 & (1 << TOV0)))
        {
                // wait
        }
        // stop timer
        TCCR0B = 0x00;
}
```

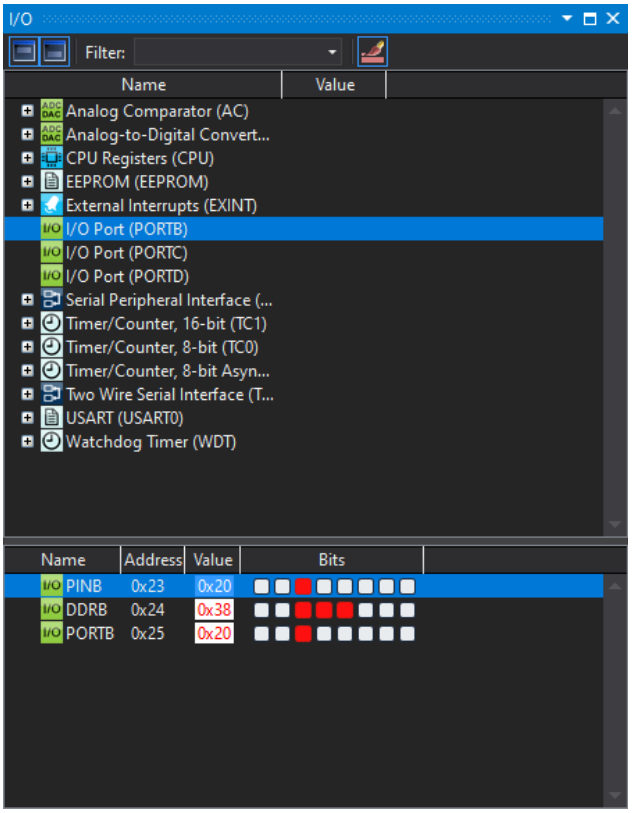
```
int main(void)
{
        // Set PB5 as output for Timer0 toggling
DDRB |= (1 << PB5);

// Timer0 poll loop for PB5 => 1.5s on/off
while (1)
{
        // PB5 on
        PORTB |= (1 << PB5);
        for (uint16_t i = 0; i < 12000; i++)
        {
                Normal_Timer0();
        }

        // PB5 off
        PORTB &= ~(1 << PB5);
        for (uint16_t i = 0; i < 12000; i++)
        {
                Normal_Timer0();
        }
}
```

DA2



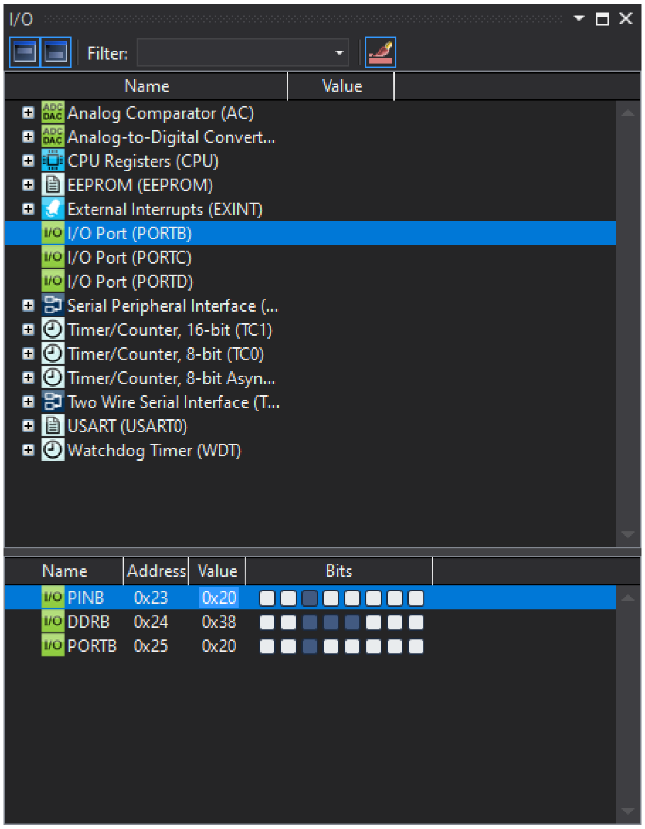


DA2



Name	Value
Analog Comparator (AC)	
Analog-to-Digital Convert...	
CPU Registers (CPU)	
EEPROM (EEPROM)	
External Interrupts (EXINT)	
I/O Port (PORTB)	
I/O Port (PORTC)	
I/O Port (PORTD)	
Serial Peripheral Interface (...	
Timer/Counter, 16-bit (TC1)	
Timer/Counter, 8-bit (TC0)	
Timer/Counter, 8-bit Asyn...	
Two Wire Serial Interface (T...	
USART (USART0)	
Watchdog Timer (WDT)	

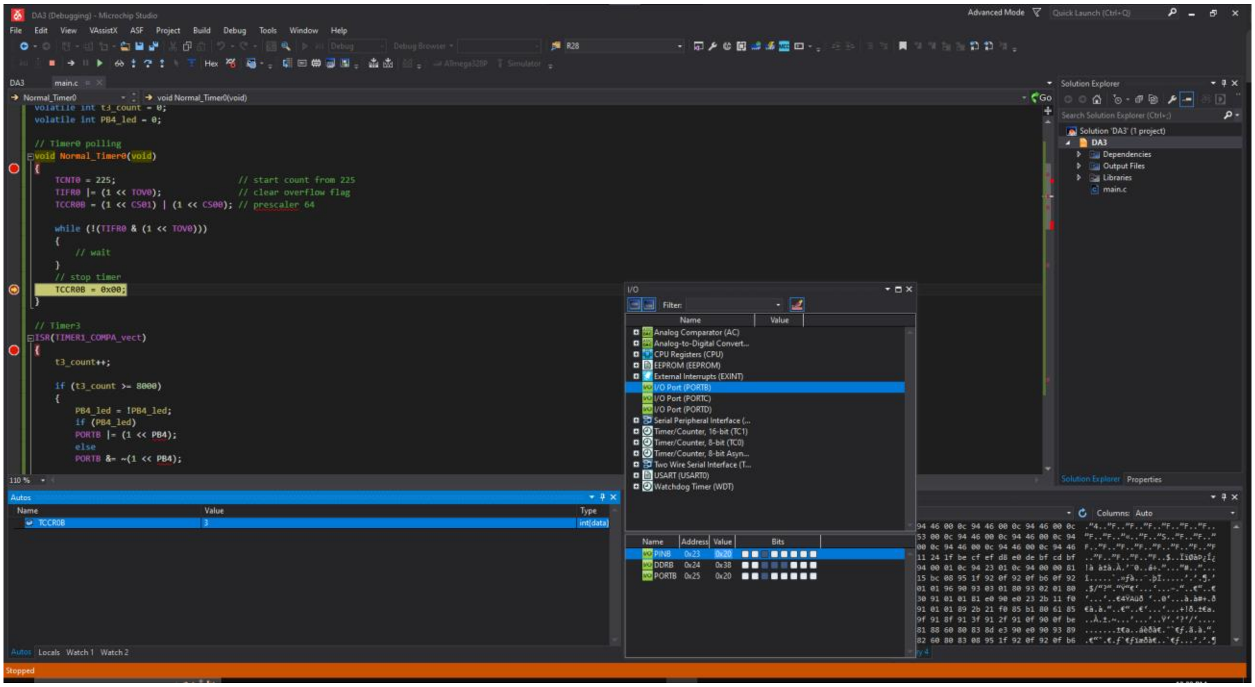
Name	Address	Value	Bits
PINB	0x23	0x20	□□■□□□□□
DDRB	0x24	0x38	□□■■■□□□
PORTB	0x25	0x20	□□■□□□□□



Name	Value
Analog Comparator (AC)	
Analog-to-Digital Convert...	
CPU Registers (CPU)	
EEPROM (EEPROM)	
External Interrupts (EXINT)	
I/O Port (PORTB)	
I/O Port (PORTC)	
I/O Port (PORTD)	
Serial Peripheral Interface (...	
Timer/Counter, 16-bit (TC1)	
Timer/Counter, 8-bit (TC0)	
Timer/Counter, 8-bit Asyn...	
Two Wire Serial Interface (T...	
USART (USART0)	
Watchdog Timer (WDT)	

Name	Address	Value	Bits
PINB	0x23	0x20	□□■□□□□□
DDRB	0x24	0x38	□□■■■□□□
PORTB	0x25	0x20	□□■□□□□□

DA2



DA3 (Debugging) - Microchip Studio
File Edit View VAssistX ASF Project Build Debug Tools Window Help
main.c
Normal_Timer0
void Normal_Timer0(void)
I/O
Solution Explorer
Solution 'DA3' (1 project)
DA3
Dependencies
Output Files
Libraries
main.c
Stopped

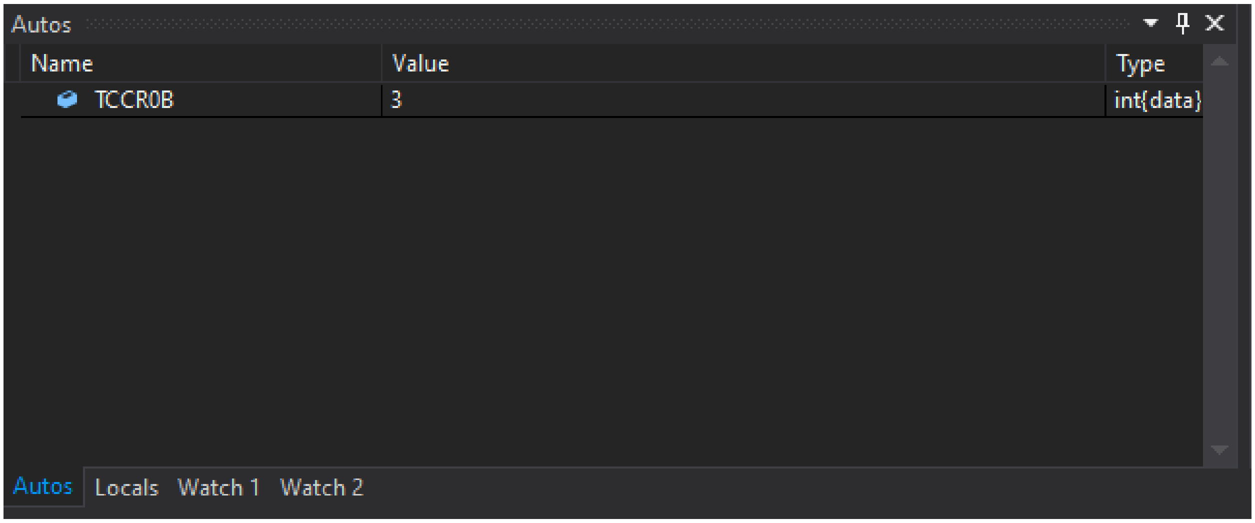
```c
volatile int t3_count = 0;
volatile int PB4_led = 0;

// Timer0 polling
void Normal_Timer0(void)
{
    TCNT0 = 225;                         // start count from 225
    TIFR0 |= (1 << TOV0);                // clear overflow flag
    TCCR0B = (1 << CS01) | (1 << CS00); // prescaler 64

    while (!(TIFR0 & (1 << TOV0)))
    {
        // wait
    }
    // stop timer
    TCCR0B = 0x00;
}

// Timer3
ISR(TIMER1_COMPA_vect)
{
    t3_count++;

    if (t3_count >= 8000)
    {
        PB4_led = !PB4_led;
        if (PB4_led)
        PORTB |= (1 << PB4);
        else
        PORTB &= ~(1 << PB4);
```

Name	Value	Type
TCCR0B	3	int{data}

I/O:

- Analog Comparator (AC)
- Analog-to-Digital Convert...
- CPU Registers (CPU)
- EEPROM (EEPROM)
- External Interrupts (EXINT)
- I/O Port (PORTB)
- I/O Port (PORTC)
- I/O Port (PORTD)
- Serial Peripheral Interface (...
- Timer/Counter, 16-bit (TC1)
- Timer/Counter, 8-bit (TC0)
- Timer/Counter, 8-bit Asyn...
- Two Wire Serial Interface (T...
- USART (USART0)
- Watchdog Timer (WDT)

Name	Address	Value
PINB	0x23	0x20
DDRB	0x24	0x38
PORTB	0x25	0x20



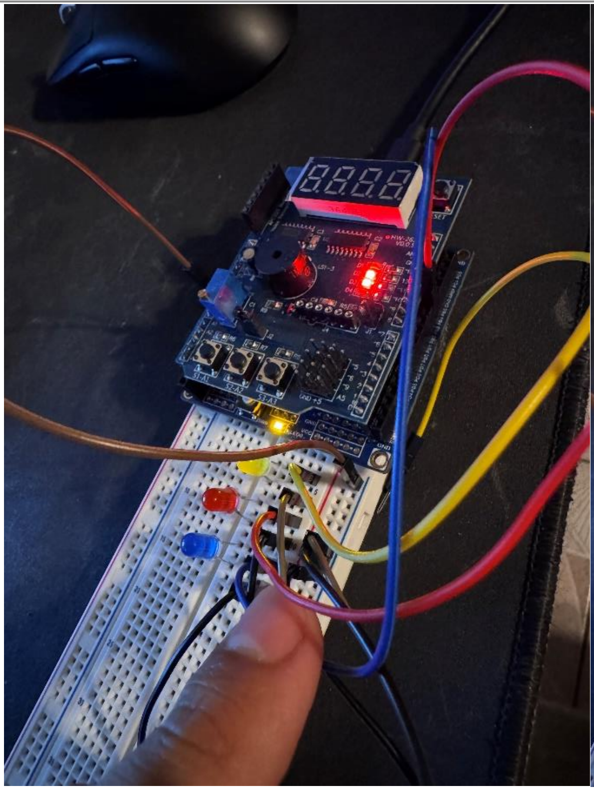
Name	Value	Type
TCCR0B	3	int{data}

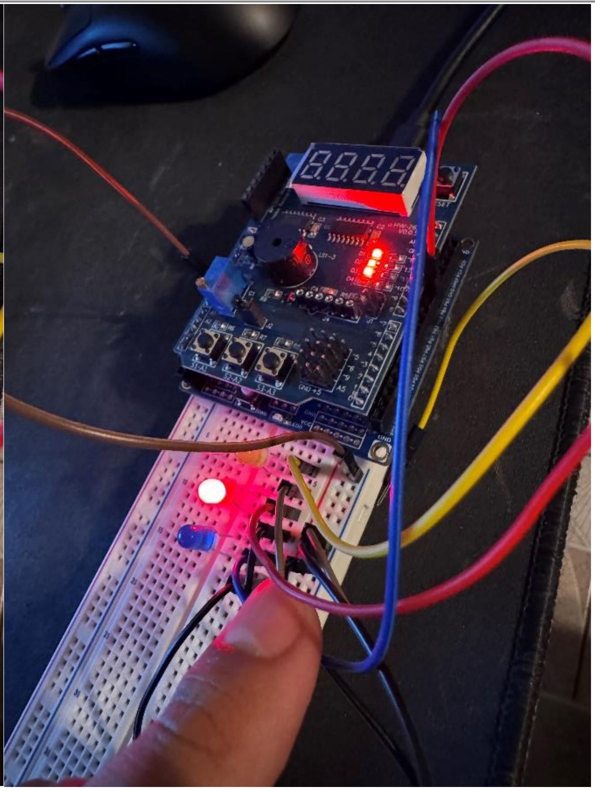
3 = 011

Timer0 is running in normal mode

b. Timer3

```
// Timer3
ISR(TIMER1_COMPA_vect)
{
        t3_count++;

        if (t3_count >= 8000)
        {
                PB4_led = !PB4_led;
                if (PB4_led)
                        PORTB |= (1 << PB4);
                else
                        PORTB &= ~(1 << PB4);

                t3_count = 0;
        }
}

void CTC_Timer3(void)
{
        DDRB |= (1 << PB4);                       // PB4 output
        TCCR1B |= (1 << WGM12);                   // CTC mode => WGM12 = 1
        OCR1A = 61;                               // OCR1A = 61 => ~248us
        TCCR1B |= (1 << CS11) | (1 << CS10);      // Prescaler=64 => CS11=1, CS10=1
        TIMSK1 |= (1 << OCIE1A);                  // Enable interrupt
}
```



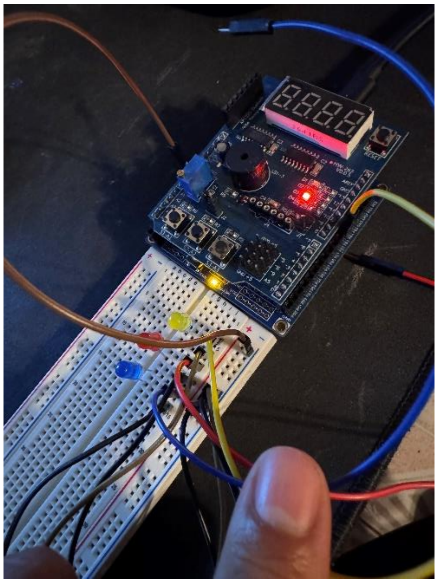


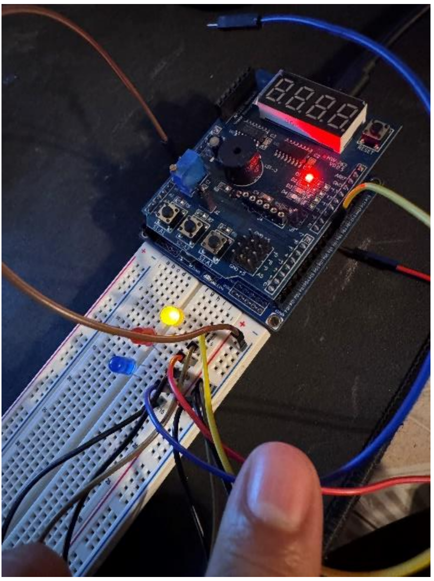
DA2

c. Timer4

```
// Timer4 ISR
ISR(TIMER2_OVF_vect)
{
        // Reload for 100us
        TCNT2 = 231;

        t4_count++;
        if (t4_count >= 10000) // ~1 second
        {
                PB3_led = !PB3_led;
                if (PB3_led)
                        PORTB |= (1 << PB3);
                else
                        PORTB &= ~(1 << PB3);

                t4_count = 0;
        }
}

void Normal_Timer4(void)
{
        DDRB |= (1 << PB3);    // PB3 output
        TCNT2 = 231;           // preload for 100us

        // Normal mode, prescaler=64 => TCCR2B = 0b100
        TCCR2A = 0x00;
        TCCR2B = (1 << CS22);

        // Enable OVF interrupt
        TIMSK2 |= (1 << TOIE2);
}
```

DA2



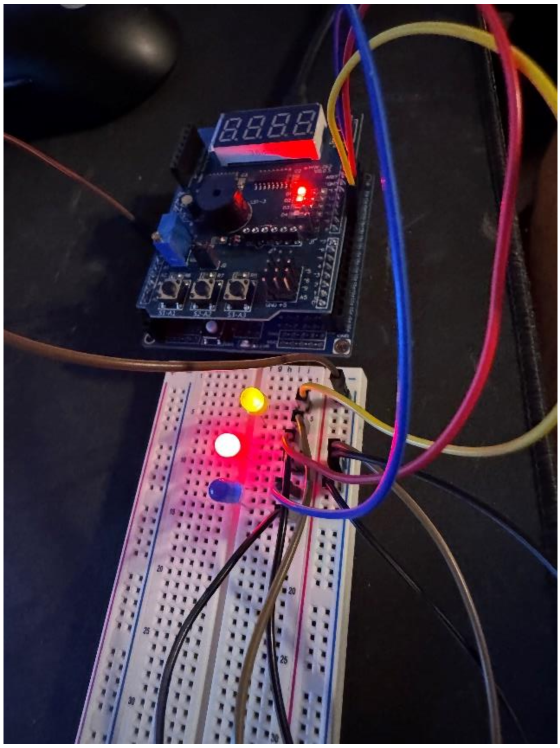


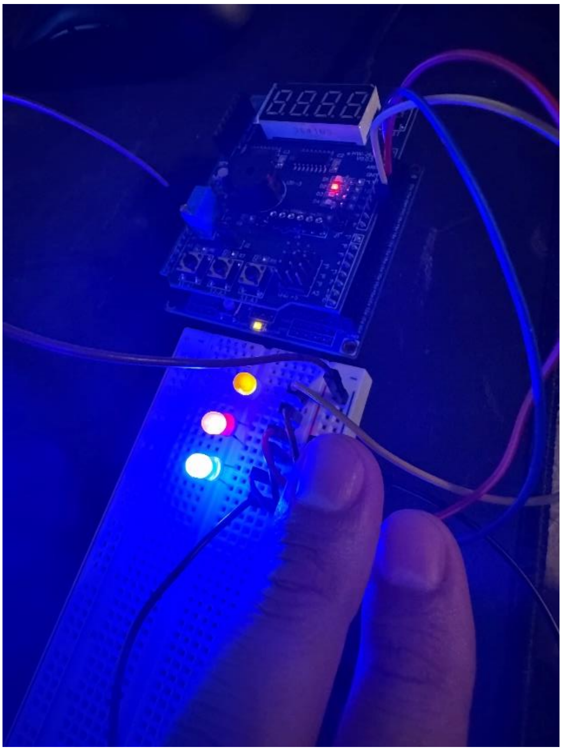
Final Code

```
/*
 * DA3.c
 *
 * Created: 3/28/2025 5:03:16 PM
 * Author : enriq
 */

#define F_CPU 16000000UL
#include <avr/io.h>
#include <avr/interrupt.h>

// Timer4 Variables
volatile int t4_count = 0;
volatile int PB3_led = 0;

// Timer3 Variables
volatile int t3_count = 0;
volatile int PB4_led = 0;

// Timer0 polling
void Normal_Timer0(void)
{
        TCNT0 = 225;                                      // start count from 225
        TIFR0 |= (1 << TOV0);                             // clear overflow flag
        TCCR0B = (1 << CS01) | (1 << CS00); // prescaler 64

        while (!(TIFR0 & (1 << TOV0)))
        {
                // wait
        }
        // stop timer
        TCCR0B = 0x00;
}
```

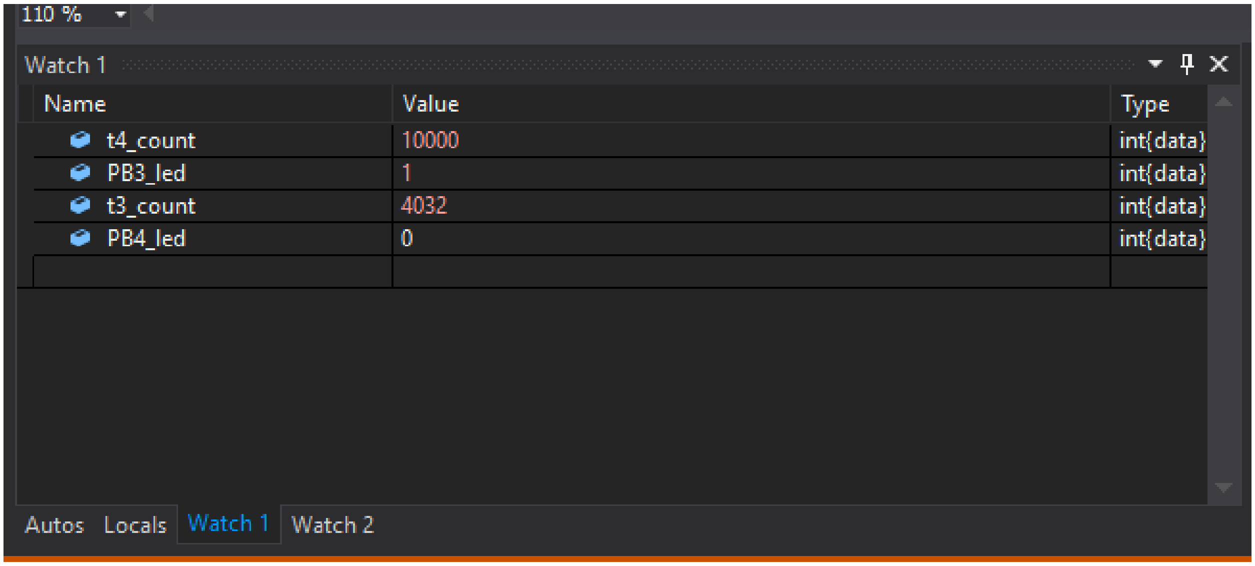
```
// Timer3
ISR(TIMER1_COMPA_vect)
{
        t3_count++;

        if (t3_count >= 8000)
        {
                PB4_led = !PB4_led;
                if (PB4_led)
                PORTB |= (1 << PB4);
                else
                PORTB &= ~(1 << PB4);

                t3_count = 0;
        }
}

void CTC_Timer3(void)
{
        DDRB |= (1 << PB4);                                        // PB4 output
        TCCR1B |= (1 << WGM12);                           // CTC mode => WGM12 = 1
        OCR1A = 61;                                                        // OCR1A =
61 => ~248us
        TCCR1B |= (1 << CS11) | (1 << CS10);      // Prescaler=64 => CS11=1, CS10=1
        TIMSK1 |= (1 << OCIE1A);                                   // Enable
interrupt
}

// Timer4 ISR
ISR(TIMER2_OVF_vect)
{
        // Reload for 100us
        TCNT2 = 231;

        t4_count++;
        if (t4_count >= 10000) // ~1 second
        {
                PB3_led = !PB3_led;
                if (PB3_led)
                PORTB |= (1 << PB3);
                else
                PORTB &= ~(1 << PB3);

                t4_count = 0;
        }
}

void Normal_Timer4(void)
{
        DDRB |= (1 << PB3);    // PB3 output
        TCNT2 = 231;           // preload for 100us

        // Normal mode, prescaler=64 => TCCR2B = 0b100
        TCCR2A = 0x00;
        TCCR2B = (1 << CS22);

        // Enable OVF interrupt
        TIMSK2 |= (1 << TOIE2);
}

int main(void)
{
        CTC_Timer3();           // Initialize Timer3
        Normal_Timer4();        // Initialize Timer4

        // Set PB5 as output for Timer0 toggling
        DDRB |= (1 << PB5);

        // Enable global interrupts
        sei();

        // Timer0 poll loop for PB5 => 1.5s on/off
        while (1)
        {
                // PB5 on
                PORTB |= (1 << PB5);
                for (uint16_t i = 0; i < 12000; i++)
                {
                        Normal_Timer0();
                }

                // PB5 off
                PORTB &= ~(1 << PB5);
                for (uint16_t i = 0; i < 12000; i++)
                {
                        Normal_Timer0();
                }
        }
}
```

DA2

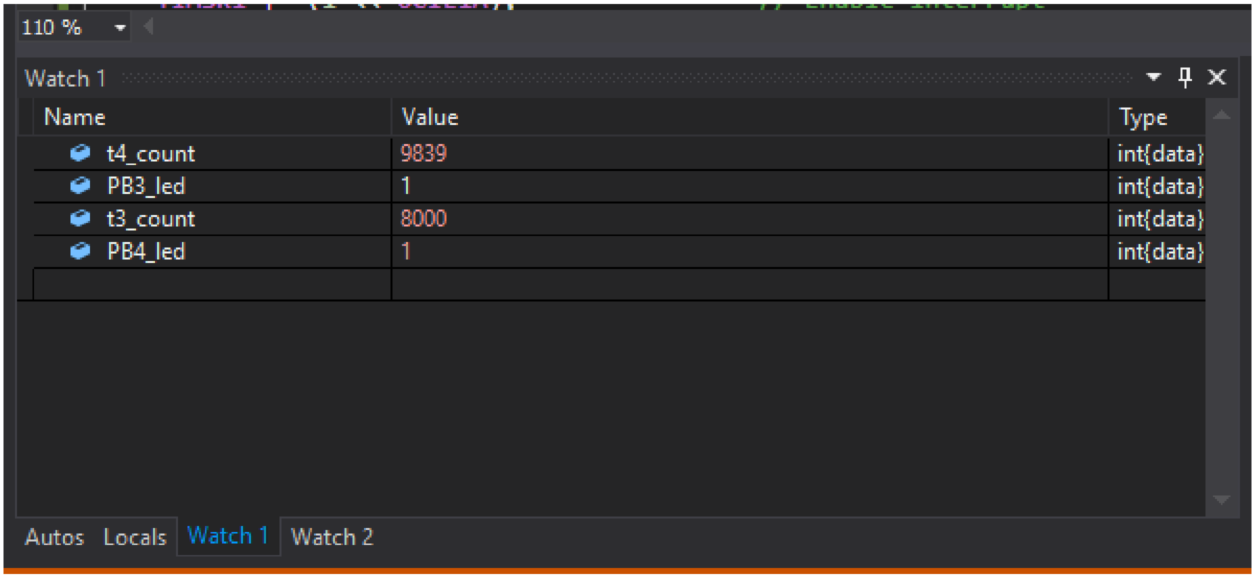




DA2



Name	Value	Type
t4_count	10000	int{data}
PB3_led	1	int{data}
t3_count	4032	int{data}
PB4_led	0	int{data}



Name	Value	Type
t4_count	9839	int{data}
PB3_led	1	int{data}
t3_count	8000	int{data}
PB4_led	1	int{data}

At 10000, PB3 is triggered so the LED turns on. T4_count is incremented every 100 us, so after 10000 we have a 1 second delay.

At 8000, PB4 is triggered to the LED turns on. T3_count is incremented every 250 us, so after 8000 we have a 2 second delay.