AMERICAN INTERNATIONAL UNIVERSITY – BANGLADESH (AIUB)
Faculty of Engineering
Department of Electrical and Electronic Engineering

Course/Lab Name: EEE4103 Microprocessor and Embedded Systems

Semester: Spring 2023-24 **Term**: Mid **Quiz**: 02M **Total Marks**: 10 **Time**: 20 Minutes

Question Mapping with Course Outcomes:

Item	COs	POIs	K	P	A	Marks	Obtained Marks
Q1-2	CO1	P.a.4.C.3	K4			2×5	
					Total:	10	

Student Information:

Student Name:	Solve Sheet			Section:	B
Student ID #:	Solve Sheet	Date:	28.02.2024	Department:	

1. Write an Arduino microcontroller code for an 800 ms interval LED blinking system using Timer_n Interrupts and a Pre-scalar value of 256. LED is connected to pin 2 of the board. Compute the value to be loaded into the OCRnA register. The clock frequency is 16 MHz. Timer ISR vectors are TIMERn_COMPx_vect. Fill up the gaps for the above problem. [5]

Answer:

```
bool LED_State = 'True';

void setup() {
        pinMode(2, OUTPUT);
        cli();
        TCCR_1_A = ob_00000000_;      |COMnA1|COMnA0| COMnB1| COMnB0| - | - |WGMn1|WGMn0|
        TCCR_1_B = ob_00000000_;      |FOCnA|FOCnB| - | - |WGMn2|CSn2| CSn1| CSn0|
        TCCR_1_B = ob_00000100_;
        TIMSK_1_ = ob_00000010_;    | - | - | - | - | - |OCIEnB|OCIEnA|TOIEn|
        OCR_1_A = _50000_;    | - | - | - | - | - | - | - | - |
        sei();
}

void loop() {
        // main code here, to run repeatedly.
}

ISR(TIMER__1__COMP_A_vect){
        TCNT_1_ = ox__0000___;
        LED_State = !LED_State;
        DigitalWrite(_2__, __ LED_State __);
}
```

Required delay = 800 ms = 800000 µs
Given system frequency = 16 MHz
So, clock period = 1/frequency = 1/16 MHz = 0.0625 µs
As such, Timer Count = Required delay/(clock period×pre-scaler value) – 1
= 800000/(0.0625×256) – 1 = 50000 – 1 = 49999
But Timer 0 can count up to 255 and Timer 1 can count up to 65,535, so Timer 1 is suitable for this application. So, the Timer Count value (50000) should be loaded into the OCR1A register.

The required program is corrected/filled up above (red marked).

2. Determine the output of the program if the signal at pin 2 becomes 1 from 0. **[5]**

```
volatile boolean var_pin;

void isr_pin() {
 flag = true; }

void setup () {
  attachInterrupt (digitalPinToInterrupt (2), isr_pin, CHANGE);  }

void loop () {
  if (var_pin) {
    // interrupt has occurred
    }
}
```

Answer:

If the signal at pin 2 becomes 1 from 0, that means there is a CHANGE at the digital pin, 2. So, the Interrupt Service Routine (ISR) Function, isr_pin() is called. This function will just change the value of the volatile Boolean type of variable 'flag' to 'true'.