AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH
Faculty of Engineering

Lab Report

Experiment # 05

Experiment Title: Familiarization of assembly language program and Interrupts in a microcontroller.

Date of Perform:	08 April 2025	**Date of Submission:**	15 April 2025
Course Title:	Microprocessor and Embedded Systems Lab		
Course Code:	EEE4103	**Section:**	Q
Semester:	Spring 2024-25	**Degree Program:**	BSc in CSE
Course Teacher:	**PROTIK PARVEZ SHEIKH**		

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case Study is my/our original work; no part has been copied from any other student's work or any other source except where due acknowledgment is made.
3. No part of this Assignment/Case Study has been written for me/us by any other person except where such collaboration has been authorized. by the concerned teacher and is acknowledged in the assignment.
4. I/we have not previously submitted or am submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared, and archived to detect plagiarism.
6. I/we permit a copy of my/our marked work to be retained by the Faculty Member for review by any internal/external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea, or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offense that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic, and visual forms, including electronic data, and oral presentations. Plagiarism occurs when the origin of the source is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or copy my/our work.

** Student(s) must complete all details except the faculty use part.*
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group # 08

Sl No	Name	ID	PROGRAM	SIGNATURE
1	**MD. ABRAR RAFID SHARIAR**	**22-48055-2**	**BSc in CSE**	
2	MD. AL-IMRAN SAYEM	22-48023-2	BSc in CSE	
3	MD TANJIL TASHRIK ZIM	22-48021-2	BSc in CSE	
4	HUMAIRA ZANNAT	22-47789-2	BSc in CSE	
5	SANJUKTA DAS	22-47801-2	BSc in CSE	
6				

Faculty use only

FACULTY COMMENTS	**Marks Obtained**	
	Total Marks	

Table of Contents

Marking Rubrics (to be filled by Faculty):

Level Category	Excellent [5]	Proficient [4]	Good [3]	Acceptable [2]	Unacceptable [1]	No Response [0]
Title and Objectives	Able to clarify the understanding of the lab, no issues are missing and formatting is good.	Able to clarify the understanding of the lab experiment, no issues are missing but its formatting is not good.	Able to clarify the understanding of the lab experiment, but a few issues are wrong, and its formatting is bad.	Able to clarify the understanding of the lab experiment, but it lacks a few important issues of the experiment without maintaining the format.	Unable to clarify the understanding of the lab experiment.	No Response/ copied from others/ identical submissions with gross errors/image file printed
Codes and Methods	Able to explain the experimental codes and simulation methods using Proteus very well.	Able to explain the experimental codes and simulation methods using Proteus but is not formatted well.	Able to explain the experimental codes but simulation method using Proteus is not explained well.	Presents the experimental codes but didn't explain simulation methods using Proteus clearly.	Presents the experimental codes but didn't explain simulation methods using Proteus.	
Results	Key results and images are there. Figures/Tables have all identifications and refer to them properly in the texts.	Key results and images are there. Figures/Tables have all identifications, such as the axis labels, numbers, and captions with a few minor errors; the texts refer them.	Key results and images are there. Figures/Tables lack a few identifications, such as the axis labels, numbers, and captions; the texts refer them.	Misses several key results and images. Figures/Tables lack identification, such as the axis labels, numbers, and captions; the texts don't refer them.	Major results, such as experimental and simulation results' images are not included. Figures and tables are poorly constructed or not presented.	
Discussion and Conclusion	Proper interpretation of results and summarizes the results to draw a conclusion, discusses its applications in real-life situations to connect with the report's conclusion.	Proper interpretation of results and summarizes the results to draw a conclusion but didn't discuss its applications in real-life situations to connect with the conclusion of the report.	Interpretation of results is presented. However, there is a disconnect between the results and discussion.	Misses the interpretation of key results. There is little connection between the results and discussion.	Very poor interpretation of the results. No connection between results and discussions.	
Question and Answer	Able to produce all questions' answers correctly maintaining the lab report format.	Able to produce all questions' answers but didn't maintain the lab report format.	Able to produce all questions' answers but wrong answers to a few questions.	Able to produce all questions' answers but wrong/missing answers to multiple questions.	Unable to produce all questions' answers and completely wrong answers.	
Comments						**Total Marks (25)**

Objectives:

The objectives of this experiment are to

1. Study the assembly language program of an Arduino.
2. Write assembly language programming code for an Arduino.
3. Build a circuit to turn on and off an LED on an Arduino Microcontroller Board connected to an I/O port of the microcontroller.
4. Study of the external interrupt using its digital I/O port and Timer interrupt of an Arduino.
5. Build a circuit to turn on and off an LED on an Arduino Microcontroller Board connected to an I/O port of the microcontroller due to the external interrupt and Timer interrupt.

Equipment List:

1) Arduino IDE (2.0.1 or any recent version)
2) Arduino Microcontroller board
3) PC having an Intel processor
4) LED lights (Red, Green, Yellow, 1 pc each)
5) One 100 □ resistor
6) Three push switches
7) Jumper wires

Circuit Diagram:

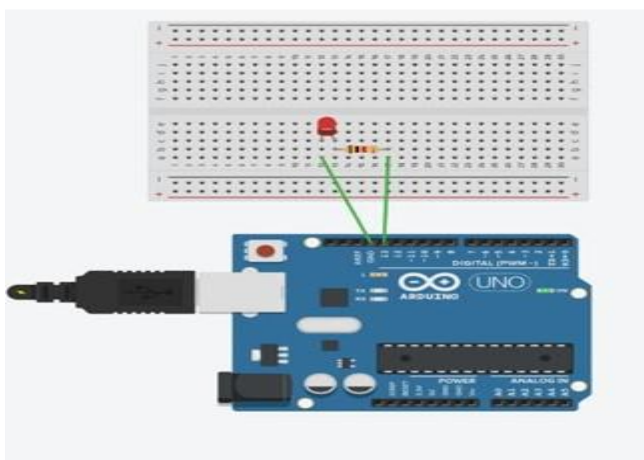


Figure 1: Experimental setup of an LED blink system using an Arduino Microcontroller Board

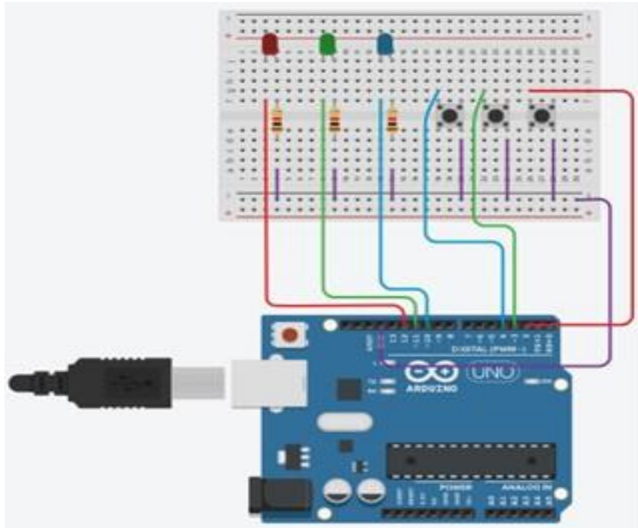


Figure 2: Experimental setup of an RGB LED ON/OFF using push button on an Arduino Microcontroller Board

Experimental Output Results:

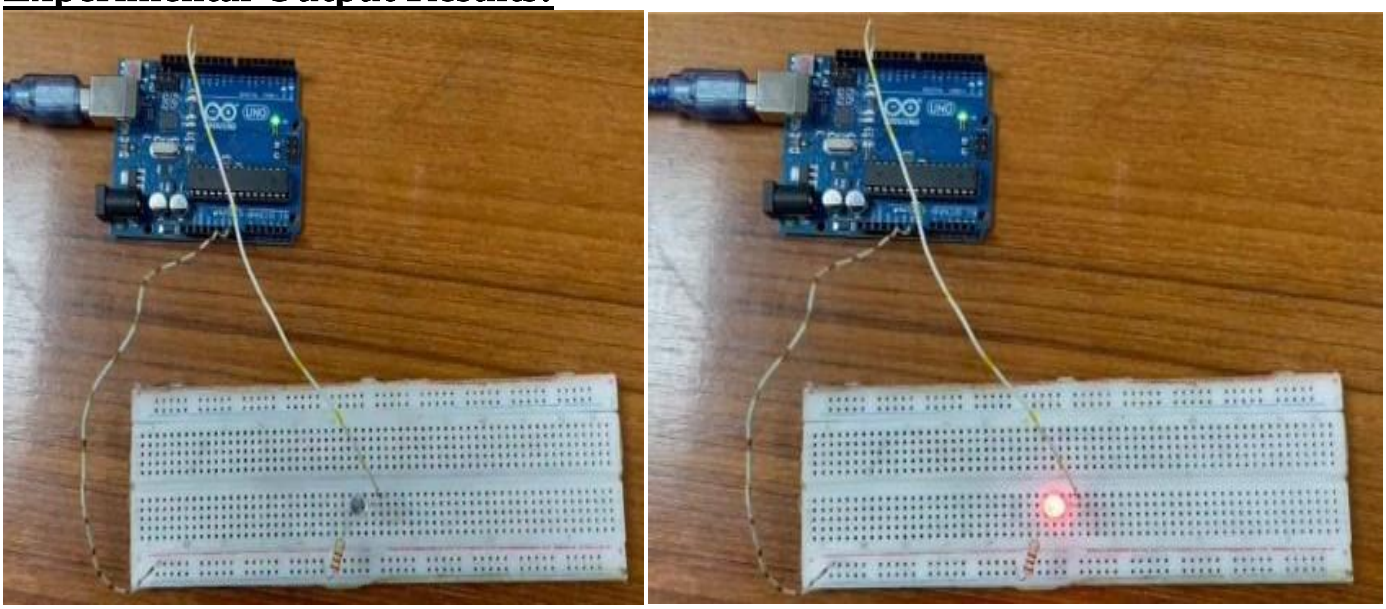
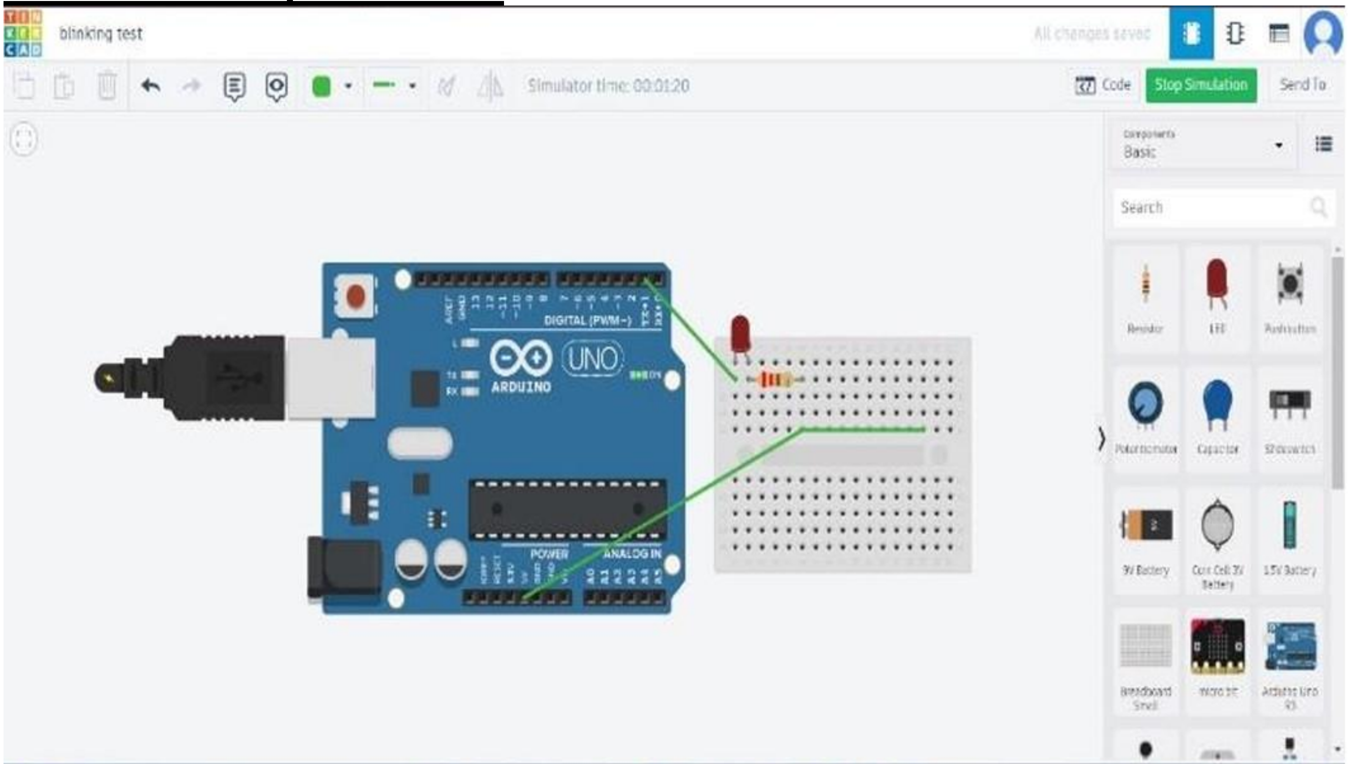
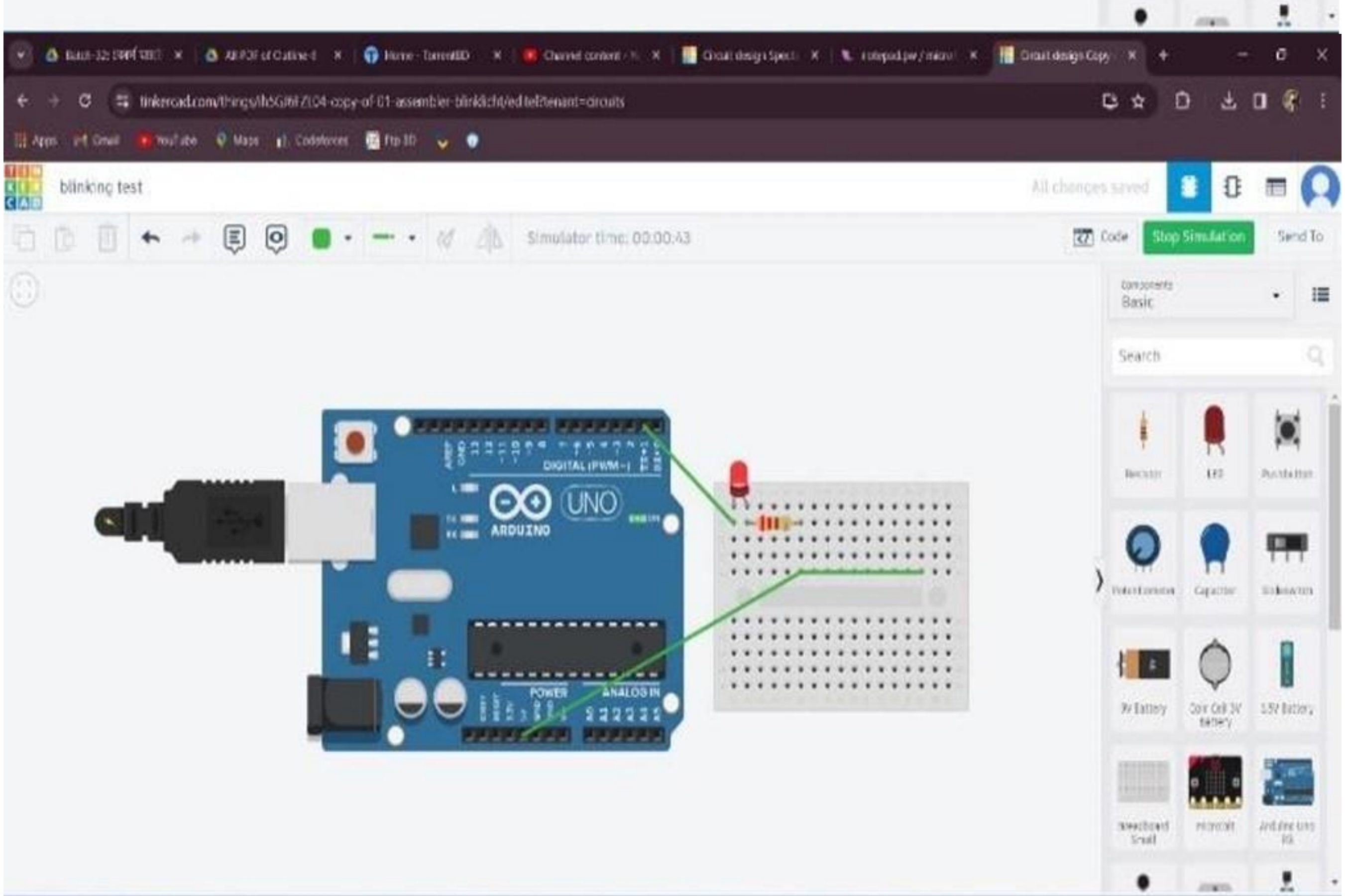


Figure 1: Picture for LED blink system using an Arduino Microcontroller

Simulation Output Results:





Answers to the Questions in the Lab Manual:

My ID is 22-48055-2

4)Ans: Configure Port numbers

ID Breakdown:
Last six digits: 480552
Port B bits: 4 8 0 5 5 (5 digits, so we use the first 5 bits of Port B)
Port D bits: 2 (1 digit, so we use the first bit of Port D)
Binary Representation:
4 8 0 5 5 → Binary:
4 = 0100 (take last 4 bits: 0100)
8 = 1000
0 = 0000
5 = 0101
5 = 0101
2 → Binary: 0010 (take last bit: 0)

Port Configuration:
Port B (PB0-PB4):
PB0: 0 (from 4 = 0100)
PB1: 0
PB2: 1
PB3: 0
PB4: 0 (from 8 = 1000)

Usage: Configure PB2 as output (LED) and PB4 as input (switch).
Port D (PD0):
PD0: 0 (from 2 = 0010)
Usage: Configure PD0 as input (switch).

Code:

```
#define __SFR_OFFSET 0x00
#include "avr/io.h"
.global start
.global controlLEDs
start:
    SBI DDRB, 2    ; Set PB2 (output for LED)
    CBI DDRB, 4    ; Set PB4 (input for switch)
    CBI DDRD, 0    ; Set PD0 (input for switch)
    RET
controlLEDs:
    SBIS PINB, 4    ; Skip if PB4 (switch 1) is pressed
    RJMP switch1_not_pressed
    SBI PORTB, 2    ; Turn on PB2 LED if switch 1 pressed
switch1_not_pressed:
    SBIS PIND, 0    ; Skip if PD0 (switch 2) is pressed
    RET
```

```
    CBI PORTB, 2    ; Turn off PB2 LED if switch 2 pressed
    RET
```

5) Ans: Control Three LEDs Using Two Switches

Circuit Setup:

Switches:
Switch 1: PD2 (INT0, external interrupt)
Switch 2: PD3 (INT1, external interrupt)

LEDs:
LED1: PB0
LED2: PB1
LED3: PB2

Code:

```
#define __SFR_OFFSET 0x00
#include "avr/io.h"
.global start
.global main
start:
    SBI DDRB, 0    ; PB0 = output (LED1)
    SBI DDRB, 1    ; PB1 = output (LED2)
    SBI DDRB, 2    ; PB2 = output (LED3)
    CBI DDRD, 2    ; PD2 = input (Switch 1)
    CBI DDRD, 3    ; PD3 = input (Switch 2)
    RET
main:
    SBIS PIND, 2    ; Check Switch 1 (PD2)
    RJMP switch1_pressed
    SBIS PIND, 3    ; Check Switch 2 (PD3)
    RJMP switch2_pressed
    CBI PORTB, 0    ; Default: LED1 off
    CBI PORTB, 1    ; LED2 off
    CBI PORTB, 2    ; LED3 off
    RET
switch1_pressed:
    SBI PORTB, 0    ; LED1 on
    CBI PORTB, 1    ; LED2 off
    CBI PORTB, 2    ; LED3 off
    RET
switch2_pressed:
    CBI PORTB, 0    ; LED1 off
    SBI PORTB, 1    ; LED2 on
    SBI PORTB, 2    ; LED3 on
    RET
```

6) Ans: LED Blink Using Timer2 (1-Second Delay with OCR2B)

Calculations:
Clock frequency = 16 MHz
Prescaler = 1024 (Timer2 max prescaler)
Timer clock = 16 MHz / 1024 = 15.625 kHz
Timer period = 1 / 15.625 kHz = 64 μs
Desired delay = 1 second → Counts = 1 / (64 μs) = 15625
OCR2B value = 15625 - 1 = 15624

Code:

```
#include <avr/interrupt.h>
bool LED_State = true;
void setup() {
   pinMode(13, OUTPUT);
   cli();                  // Disable interrupts
   TCCR2A = 0;             // Reset Timer2 registers
   TCCR2B = 0;
   TCCR2B |= (1 << CS22) | (1 << CS21) | (1 << CS20); // Prescaler = 1024
   TIMSK2 |= (1 << OCIE2B); // Enable Timer2 compare match B interrupt
   OCR2B = 15624;          // Set compare value for 1-second delay
   sei();                  // Enable interrupts
}
void loop() {
   // Main loop (empty)
}
ISR(TIMER2_COMPB_vect) {
   TCNT2 = 0;              // Reset Timer2
   LED_State = !LED_State;  // Toggle LED state
   digitalWrite(13, LED_State);
}
```

Discussion:

In this lab experiment the assembly language program executed on the microcontroller allowed us a deeper
understanding of low-level programming concepts. Through the direct manipulation of registers and memory
addresses, we gained insight into the fundamental operations of the microcontroller. Writing assembly code
enabled us to control the hardware at a granular level, facilitating efficient utilization of resources and precise
execution of instructions. By dissecting the assembly code, we could comprehend the intricacies of instruction
execution, including fetching, decoding, and executing operations. This hands-on experience enhanced our
comprehension of processor architecture and instruction set architecture (ISA), which are essential for
optimizing code performance and troubleshooting potential issues. Moreover, the assembly language

program
provided valuable insights into the real-time behavior of the microcontroller. We observed how different instructions affected program flow and data manipulation, contributing to a holistic understanding of embedded systems programming.
In conclusion, the familiarity gained with assembly language programming in a microcontroller environment
was invaluable. This experience not only reinforced theoretical concepts but also instilled practical skills essential for embedded systems development. By bridging the gap between hardware and software, we acquired a deeper appreciation for the underlying mechanisms driving microcontroller functionality. Moving
forward, the knowledge and skills acquired from this lab experiment will serve as a solid foundation for tackling more complex embedded systems projects and optimizing code efficiency in real-world applications.

References:

- Circuit Digest website https://circuitdigest.com/article/everything-you-need-to-know-about-arduino-uno-board-hardware

- MicroPi, Program the Arduino Uno in Assembly Language [Online: March 23, 2021], [Cited: June 24, 2023] Available: https://micropi.wordpress.com/2021/03/23/program-the-arduino-uno-in-assembly-language/
- Arduino CC Website, [Online] [Cited: June 10, 2023] Available: https://docs.arduino.cc/hardware/uno-rev3
- AIUB Microprocessor and Embedded Systems Lab Manual 5