AMERICAN INTERNATIONAL UNIVERSITY- BANGLADESH

Faculty of Engineering Laboratory

Report Cover Sheet

Students must complete all details except the faculty use part.

Please submit all reports to your subject supervisor or the office of the concerned faculty.

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

Experiment Number: 05 Due Date: 27-02-2024 Semester: Spring 23-24

Subject Code: EEE 4103 Subject Name: Microprocessor and Embedded Systems

Section: E Course Instructor: Protik Parvez Sheikh Degree Program: BSc. CSE

Declaration and Statement of Authorship:

1. I/we hold a copy of this report, which can be produced if the original is lost/ damaged.
2. This report is my/our original work and no part of it has been copied from any other student's workor fromany other source except where due acknowledgement is made.
3. No part of this report has been written for me/us by any other person except where such collaboration hasbeen authorized by the lecturer/teacher concerned and is clearly acknowledged in thereport.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detectingplagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by theSchool forreview andcomparison, including review by external examiners.

I/we understand that

7. 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 offence that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, includingelectronic data, and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited.
8. Enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy your work

Group Number (if applicable): ☐ Individual Submission ☑ Group Submission

No.	Student Name	Student Number	Student Signature	Date
Submitted by:				
1	Md. Jobaer Hossain	22-47116-1		
Group Members:				
2	Shayan Abrar	22-47156-1		
3	Md. Samin Yeasar	22-47139-1		
4	Rifah Sanzida	22-47154-1		
5	Samia Sharmin Dola	22-47126-1		

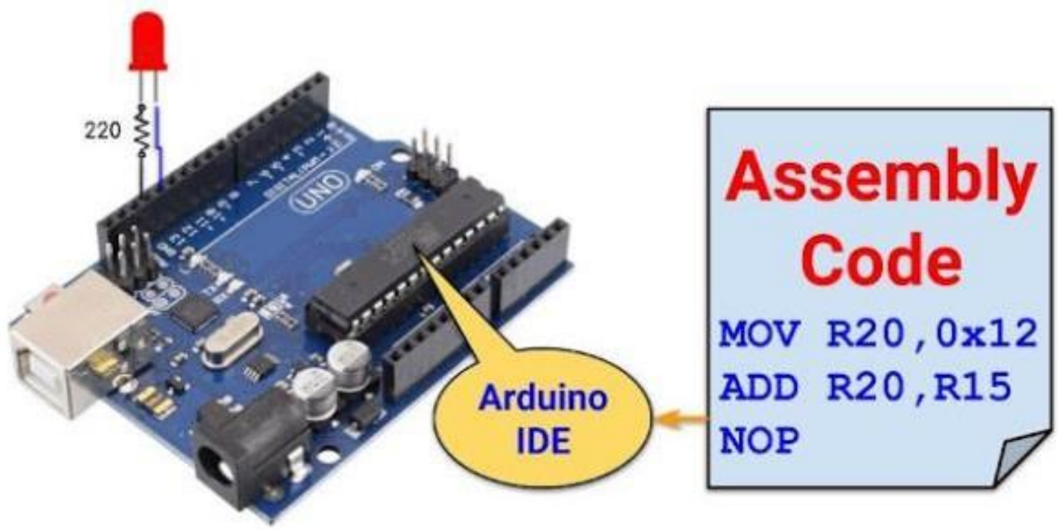
For faculty use only: **Total Marks:** **Marks Obtained:**

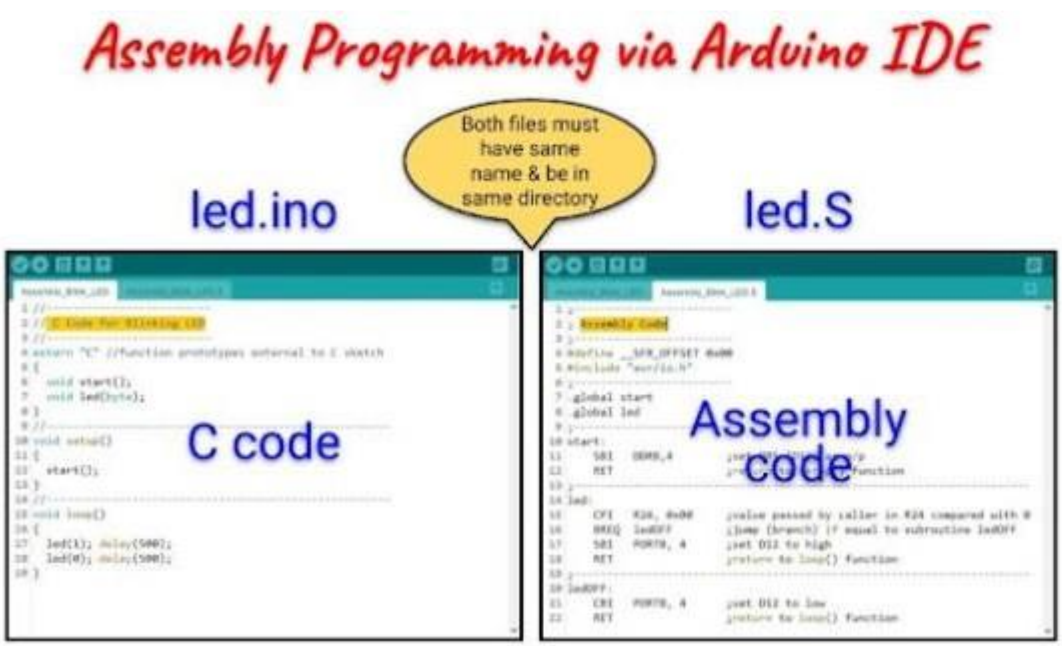
Faculty comments____________________

Title: Familiarization of assembly language program in a microcontroller.

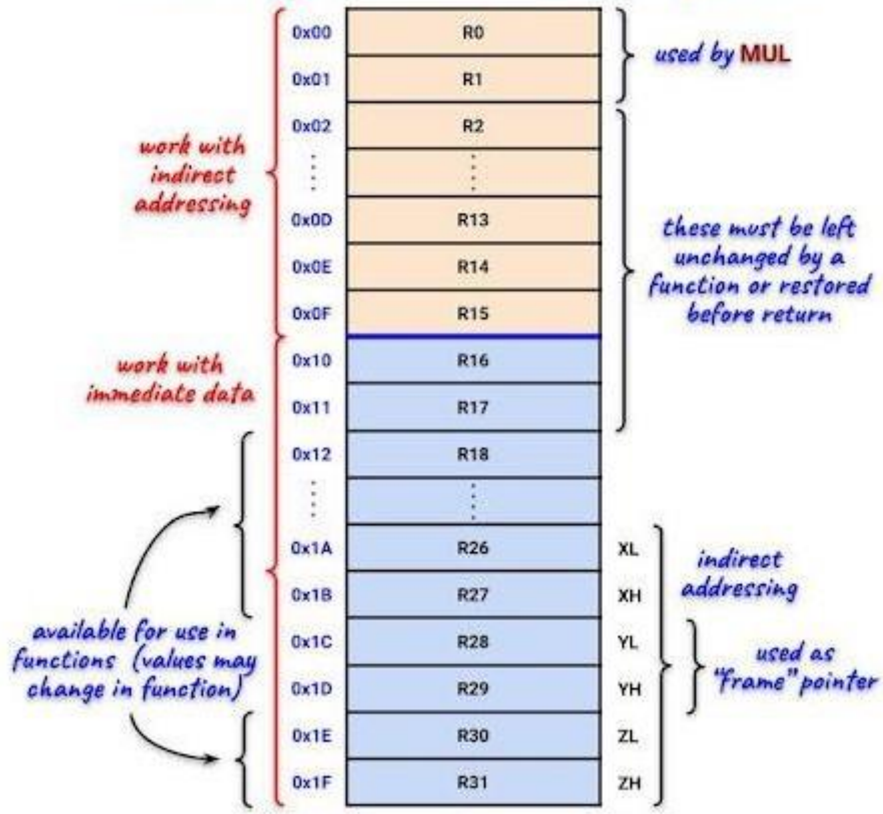
Introduction: In this experiment, the main objective is to learn how to write an assembly program for ablink LED program in a microcontroller.

Theory and Methodology: Assembly language programming using Arduino IDE.

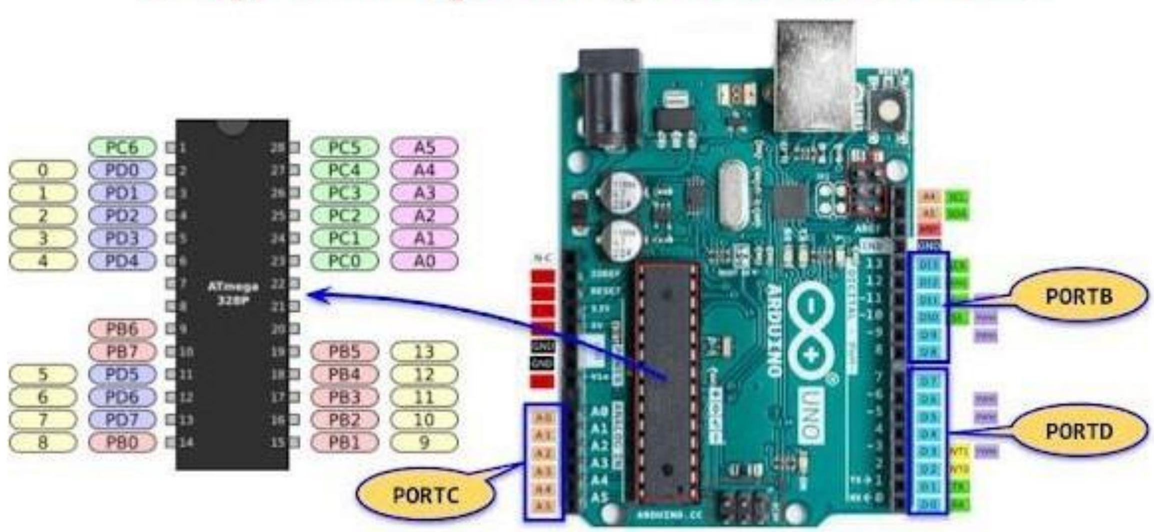


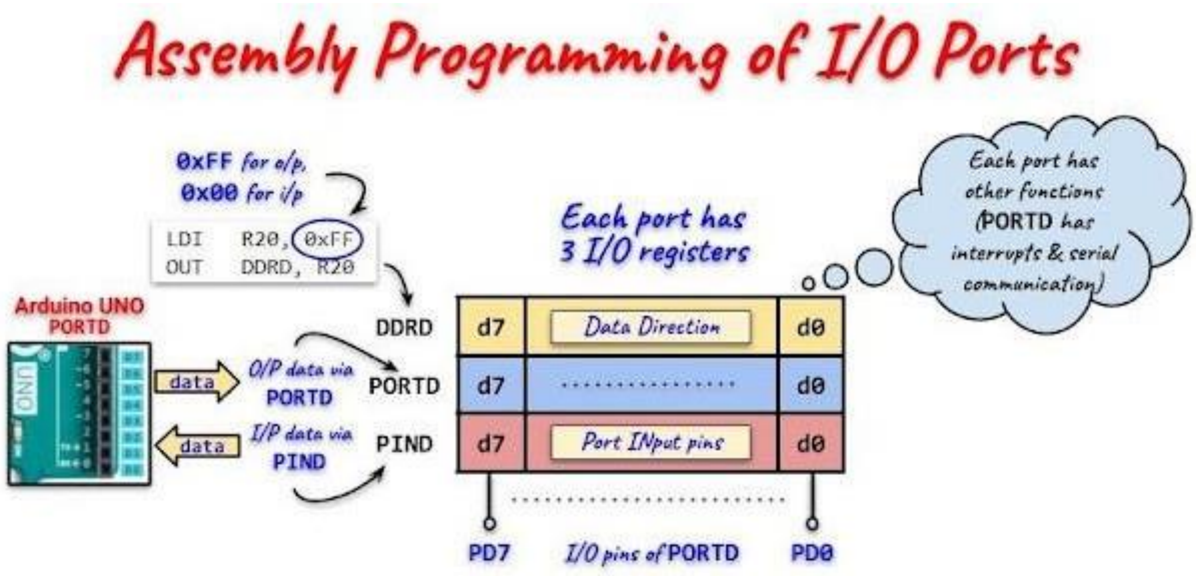


ATMega328P *MCU Registers*



Programming ATmega328 I/O Ports





<u>PART 1:</u> Blink a LED after 2s

The .ino file:

```
//-------------------------
// C Code for Blinking LED
 //-------------------------
extern "C" { void start(); void led(byte); }
 //---------------------------------------------------
 void setup() { start(); }
 //---------------------------------------------------
void loop()
{ led(1); led(0); }
```

The .S file:

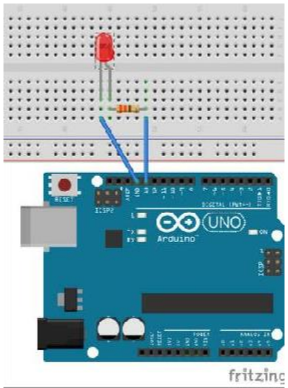
```
 ;---------------
; Assembly Code
;---------------
#define __SFR_OFFSET 0x00
#include "avr/io.h"
 ;-----------------------
 .global start .global led
 ;-----------------------
start:
 SBI DDRB, 5 ;set PB5 (D13) as o/p
```

```
 RET   ;return to setup() function
 ;--------------------------------------------------------------------------
 led:
CPI R24, 0x00 ;value in R24 passed by caller compared with 0
 BREQ ledOFF ;jump (branch) if equal to subroutine ledOFF
 SBI PORTB, 5 ;set D13 to high
 RCALL myDelay
 RET ;return to loop() function
;--------------------------------------------------------------------------
 ledOFF:
 CBI PORTB, 5 ;set D13 to low
RCALL myDelay
 RET ;return to loop() function
 ;--------------------------------------------------------------------------
 .equ delayVal, 160000 ;initial count value for inner loop
 ;--------------------------------------------------------------------------
 myDelay:
 LDI R20, 200 ;initial count value for outer loop
outerLoop: LDI R30, lo8(delayVal) ;low byte of delayVal in R30
LDI R31, hi8(delayVal) ;high byte of delayVal in R31
innerLoop:
SBIW R30, 1 ;subtract 1 from 16-bit value in R31, R30
BRNE innerLoop ;jump if countVal not equal to 0
 ;--------------
 SUBI R20, 1 ;subtract 1 from R20
BRNE outerLoop ;jump if R20 not equal to 0
 RET
 ;--------------------------------------------------------------------------
   .
```

Equipment

1) Arduino Uno
2) Arduino IDE
3) One Led
4) One 220 ohm resistor
5) PC having Intel Microprocessor

Experimental Setup:



Experimental procedure:

1) Create led.ino and led.S files using code given above.
2) Create a folder named led and place the above two files in the led folder.
3) Open led.ino using Arduino IDE.
4) Compile and upload to the hardware.
5) Modify the program to blink a led at digital PIN 12 with a different delay.

Experimental Setup and Result:

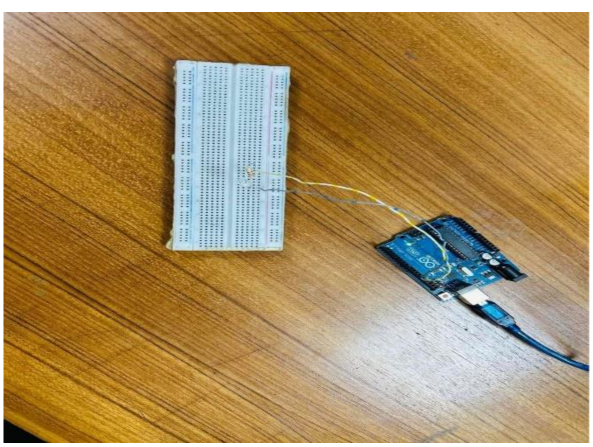


Fig.1: LED Off

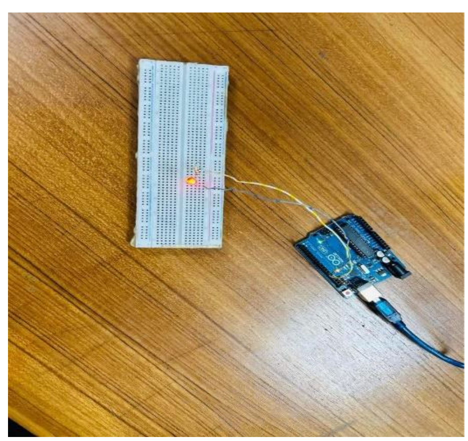
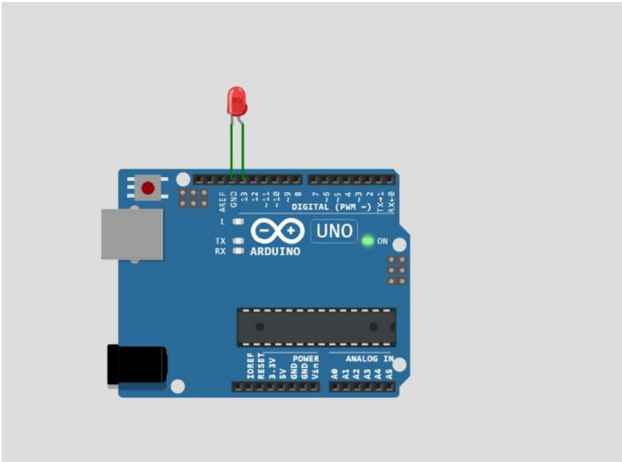


Fig.2: LED On

Simulation Setup:



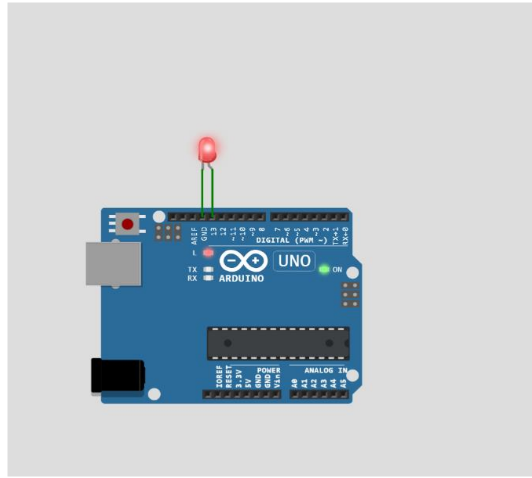
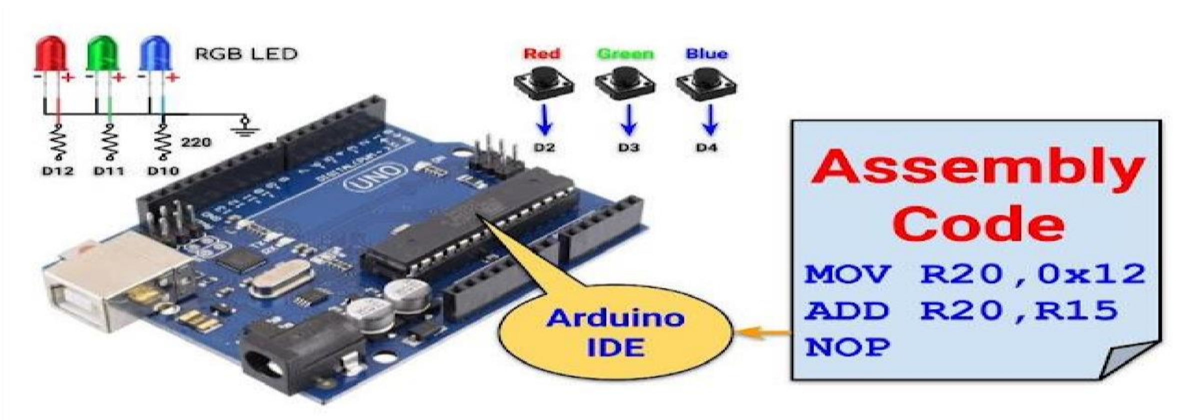


Fig.1: LED Off

Fig.2: LED On

<u>PART 2</u>: Push button LED control



Code:

```
//-------------------------
// C Code for traffic light
//-------------------------
extern "C"
{
void start();
void led(byte);
void led1(byte);
void led2(byte);
}
```

```
//------------------------------------------------------
void setup()
{
start();
}
//------------------------------------------------------
void loop()
{
led(1);
led(0);

led1(1);
led1(0);

led2(1);
led2(0);

}

;---------------
; Assembly Code
;---------------
#define __SFR_OFFSET 0x00
#include "avr/io.h"
;------------------------
.global start
.global led
.global led1
.global led2
;------------------------
start:
SBI DDRB, 5 ; Set PB5 (D13) as output for LED1
SBI DDRB, 4 ; Set PB4 (D12) as output for LED2
SBI DDRB, 3 ; Set PB3 (D11) as output for LED3
RET ; Return to setup() function
;---------------------------------------------------------------------------
led:
CPI R24, 0x00 ; Value in R24 passed by caller compared with 0
BREQ ledOFF ; Jump (branch) if equal to subroutine ledOFF
SBI PORTB, 5 ; Set LED1 to high
RCALL myDelay
RET ; Return to loop() function
;---------------------------------------------------------------------------
ledOFF:
CBI PORTB, 5 ; Set LED1 to low
```

```
RCALL myDelay
RET ; Return to loop() function
;---------------------------------------------------------------------------
led1:
CPI R24, 0x00 ; Value in R24 passed by caller compared with 0
BREQ led1OFF ; Jump (branch) if equal to subroutine led1OFF
SBI PORTB, 4 ; Set LED2 to high
RCALL myDelay
RET ; Return to loop() function
;---------------------------------------------------------------------------
led1OFF:
CBI PORTB, 4 ; Set LED2 to low
RCALL myDelay
RET ; Return to loop() function
;---------------------------------------------------------------------------
led2:
CPI R24, 0x00 ; Value in R24 passed by caller compared with 0
BREQ led2OFF ; Jump (branch) if equal to subroutine led2OFF
SBI PORTB, 3 ; Set LED3 to high
RCALL myDelay
RET ; Return to loop() function
;---------------------------------------------------------------------------
led2OFF:
CBI PORTB, 3 ; Set LED3 to low
RCALL myDelay
RET ; Return to loop() function
;---------------------------------------------------------------------------
.equ delayVal, 160000 ; Initial count value for inner loop
;---------------------------------------------------------------------------
myDelay:
LDI R20, 200 ; Initial count value for outer loop
outerLoop:
LDI R30, lo8(delayVal) ; Low byte of delayVal in R30
LDI R31, hi8(delayVal) ; High byte of delayVal in R31
innerLoop:
SBIW R30, 1 ; Subtract 1 from 16-bit value in R31, R30
BRNE innerLoop ; Jump if countVal not equal to 0
;--------------
SUBI R20, 1 ; Subtract 1 from R20
BRNE outerLoop ; Jump if R20 not equal to 0
RET
;---------------------------------------------------------------------------
```

Experimental Setup and Result:

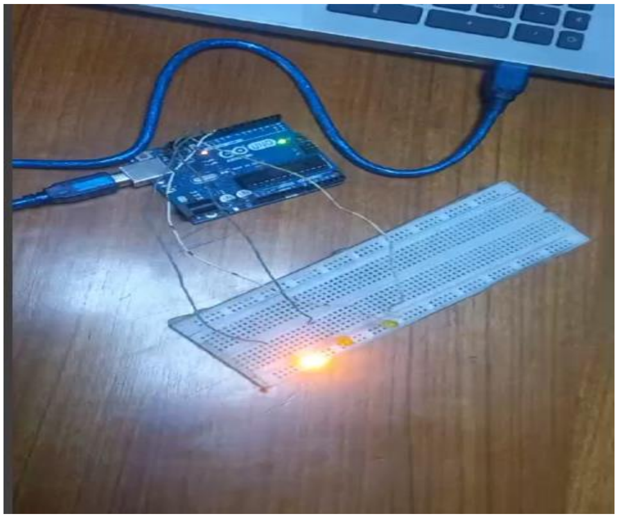


Fig.1: 1st LED On

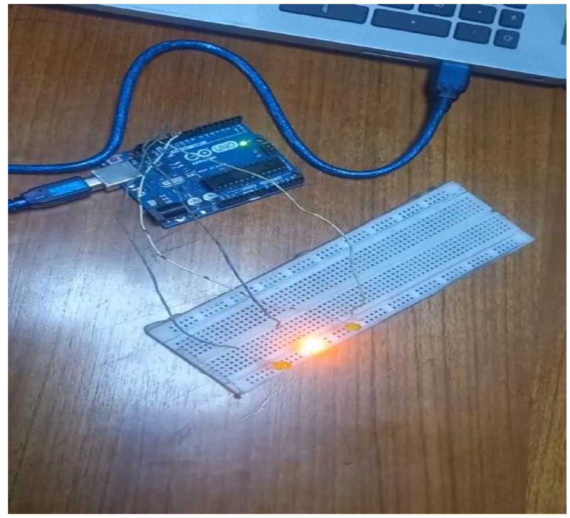


Fig.2: 2nd LED On

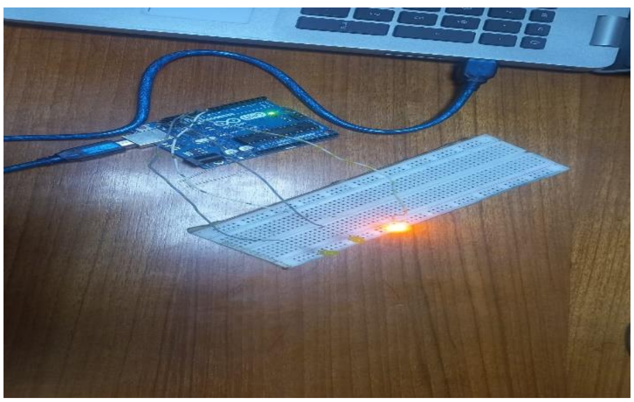


Fig.3: 3rd LED On

Simulation Setup:

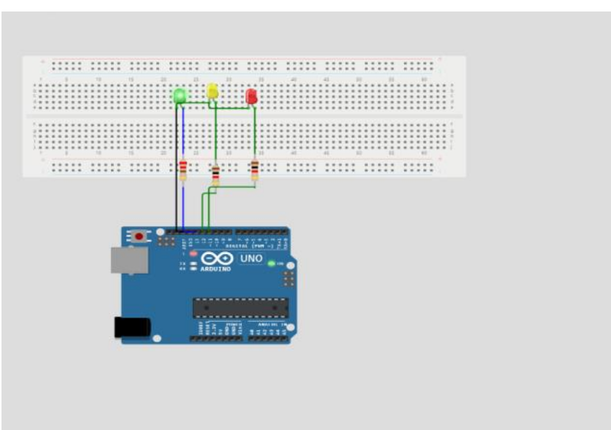


Fig.1: Green LED On

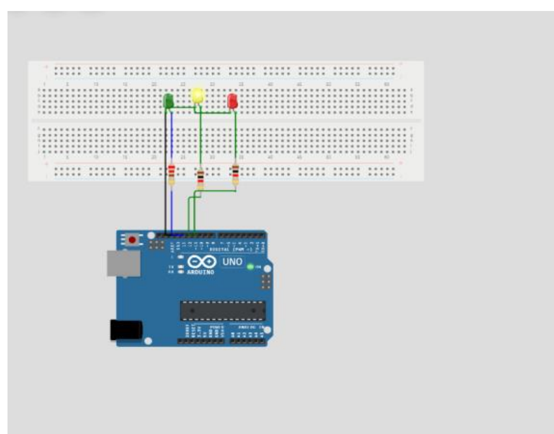


Fig.2: Yellow LED On

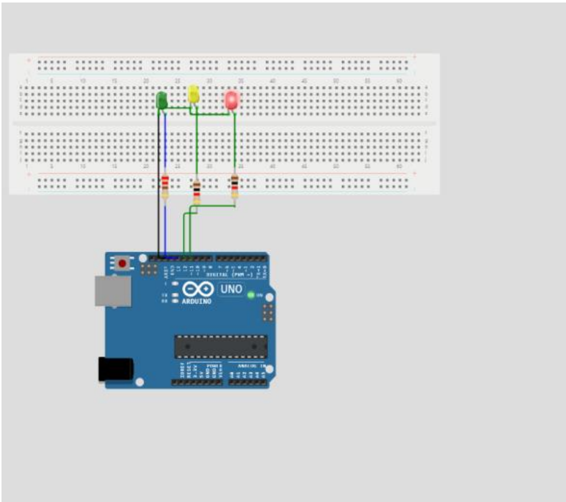


Fig.3: Red LED On

Discussion:

In this lab, we familiarized ourselves with AVR assembly language programming for microcontrollers. Assembly language provides precise control over hardware resources, making it ideal for direct microcontroller programming. We began by understanding the basic structure of an AVR assembly program, defining global labels and LED control subroutines. These subroutines set pin directions and manipulate port states to control LED activation. Additionally, we implemented a delay subroutine using nested loops to control LED blink speed. This hands-on experience equipped us with essential skills for developing embedded systems applications, including subroutine writing, pin configuration, and time delay creation. Overall, this lab provided a solid foundation in AVR assembly language programming for microcontroller-based projects.

Conclusion:

In conclusion, this lab provided valuable insights into AVR assembly language programming for microcontrollers, enhancing our understanding of hardware control and resource utilization. Through hands-on analysis and modification of assembly code, we acquired practical skills for developing embedded systems applications. Assembly language's precision in resource management makes it indispensable for projects with stringent timing or resource constraints.

References:

1) https://www.arduino.cc/.
2) ATMega328 manual
3) https://www.avrfreaks.net/forum/tut-c-newbies-guide-avr-timers
4) Microprocessor and Embedded Systems Lab manual (AIUB)