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

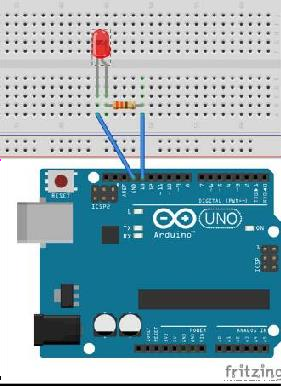
# Introduction:

A low-level programming language known as assembly language is unique to a particular CPU or microcontroller architecture. It represents the machine-level commands that the microcontroller's CPU is capable of understanding via symbolic names and mnemonic codes. While still being very near to the hardware, assembly language offers a mechanism to create code that is more legible by humans than machine code (binary). On a single integrated circuit (IC), a microcontroller is a tiny, independent computer. Its components include a central processing unit (CPU), memory (RAM and ROM), input/output ports, and a number of peripherals including timers, counters, and communication interfaces. Embedded systems, like those in appliances, cars, and Internet of Things (IoT) gadgets, frequently include microcontrollers.

# Equipment List:

1. Arduino Uno Atmega328
2. Arduino IDE
3. One Led
4. One 220 ohm resistor
5. PC having Intel Microprocessor

# Circuit diagram:



***Fig-1:*** *LED Blink to Test Familiarization of assembly language using an Arduino ATMega 328 Microcontroller Boar.*

# Code/program:

**The .ino file:**

extern "C" { void start(); void led(byte); }

// void setup() {

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

**LED Blink**

.equ delayVal, 10000

;initial count value for inner loop ;

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 ;

**The .S file:**

#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 Blink**

myDelay: LDI R20, 100

;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 ;

void setup() {

//ports for connecting LEDs pinMode(RED\_PIN, OUTPUT); pinMode(YELLOW\_PIN, OUTPUT); pinMode(GREEN\_PIN, OUTPUT);

}

void loop() {

//turning on voltage at output red LED digitalWrite(RED\_PIN, HIGH);

//to make red LED on delay(red\_on);

//to turn yellow LED on digitalWrite(YELLOW\_PIN, HIGH); delay(red\_yellow\_on);

//turning off RED\_PIN and YELLOW\_PIN, and turrning on greenLEd digitalWrite(RED\_PIN, LOW);

digitalWrite(YELLOW\_PIN, LOW); digitalWrite(GREEN\_PIN, HIGH); delay(green\_on); digitalWrite(GREEN\_PIN, LOW);

//for turning green Led on and off for 3 timesfor(int i

= 0; i < 3; i = i+1)

{

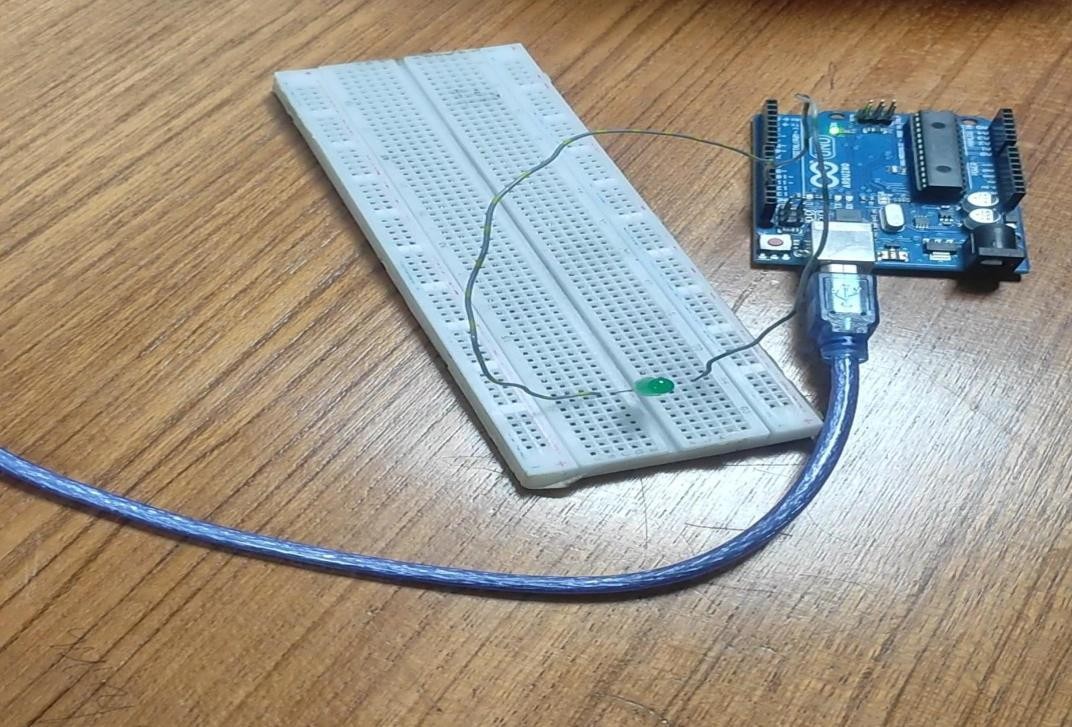
delay(green\_blink); digitalWrite(GREEN\_PIN, HIGH); delay(green\_blink); digitalWrite(GREEN\_PIN, LOW);

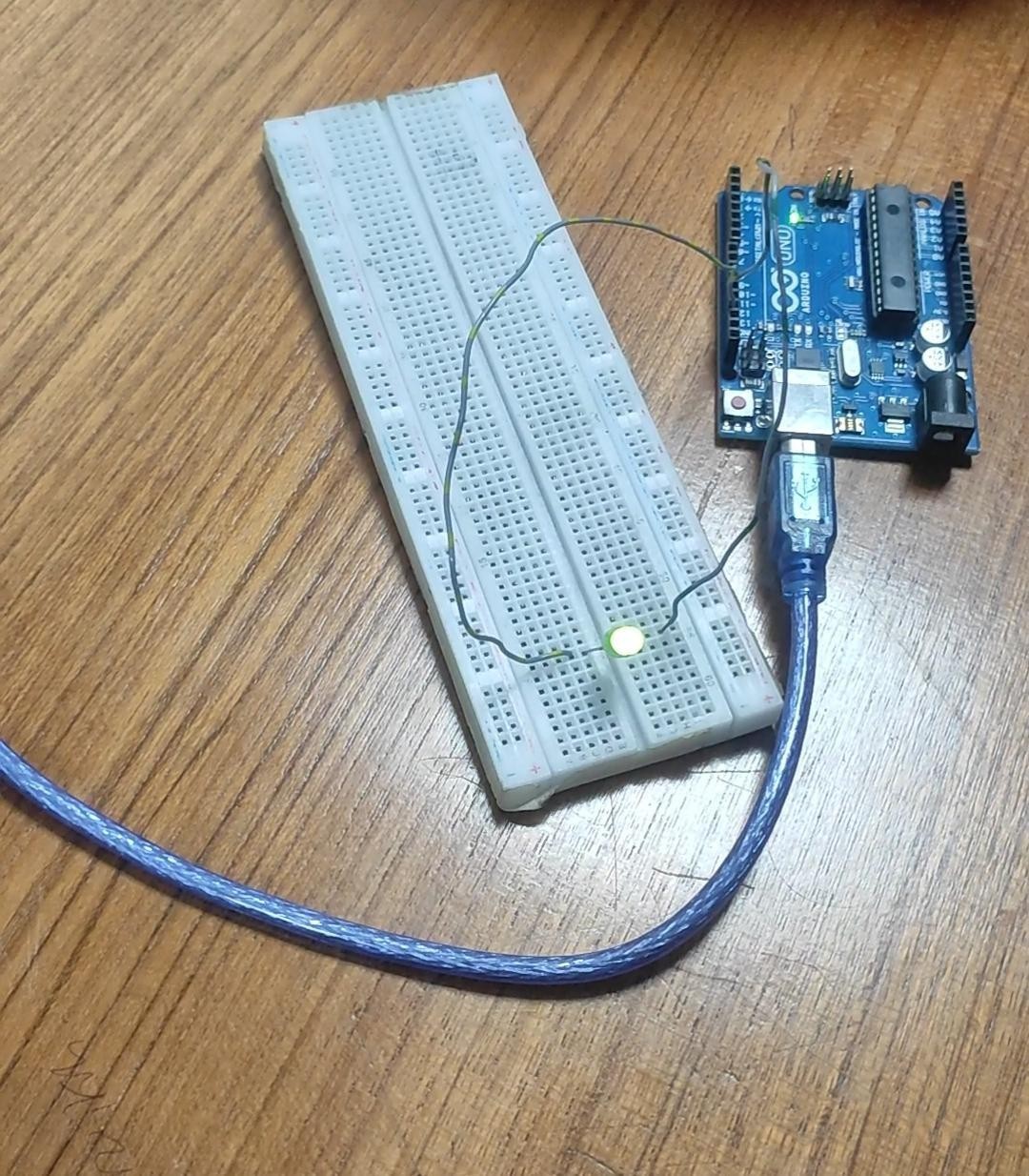
}

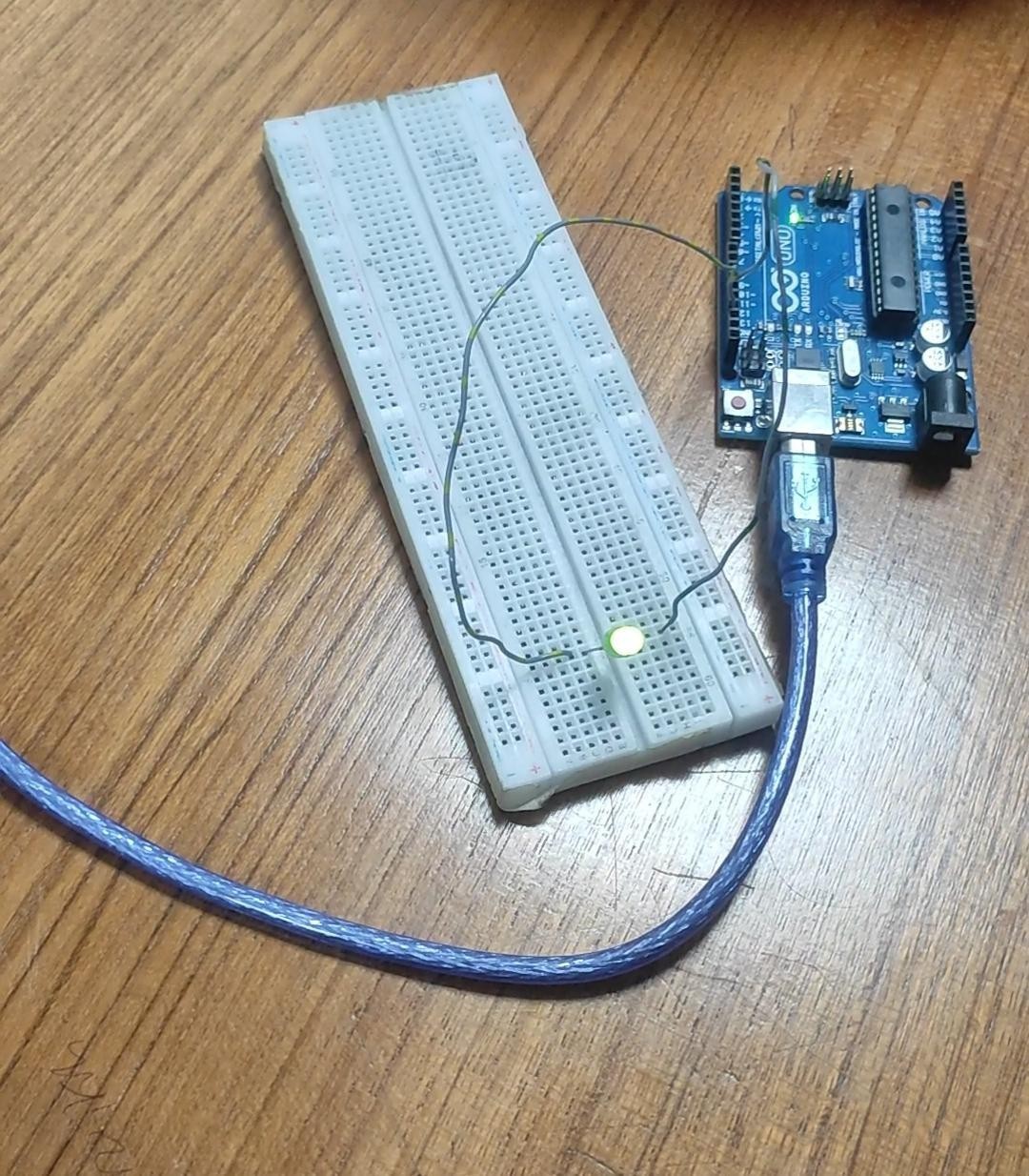
//for turning on yellow LED digitalWrite(YELLOW\_PIN, HIGH); delay(yellow\_on); digitalWrite(YELLOW\_PIN, LOW);

}

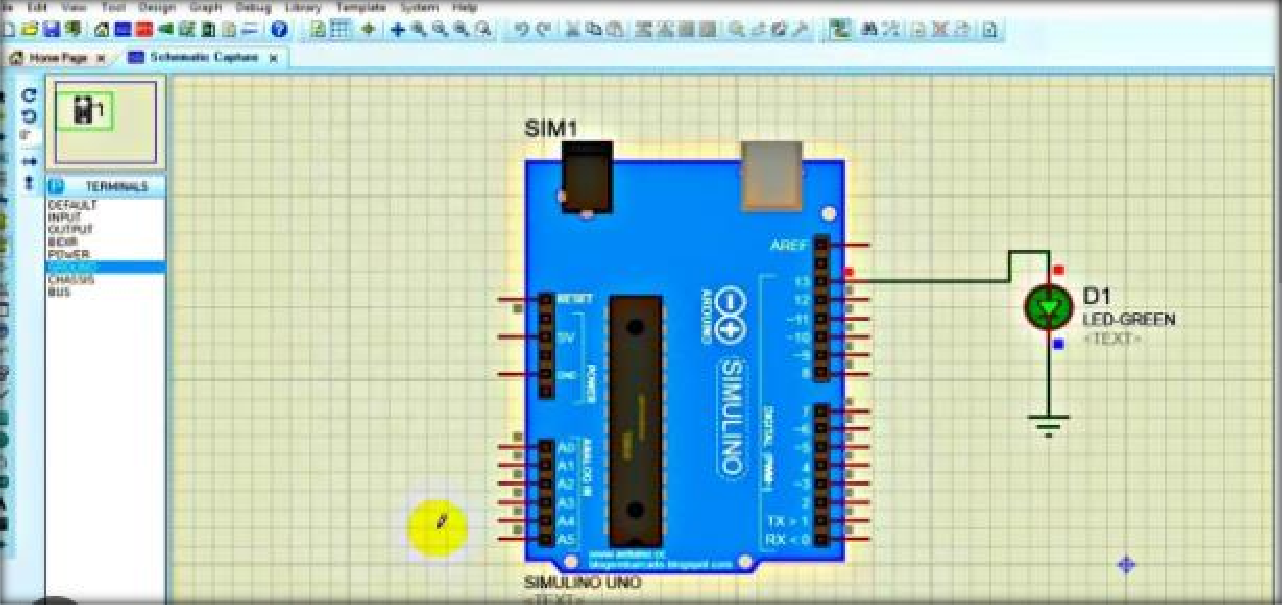
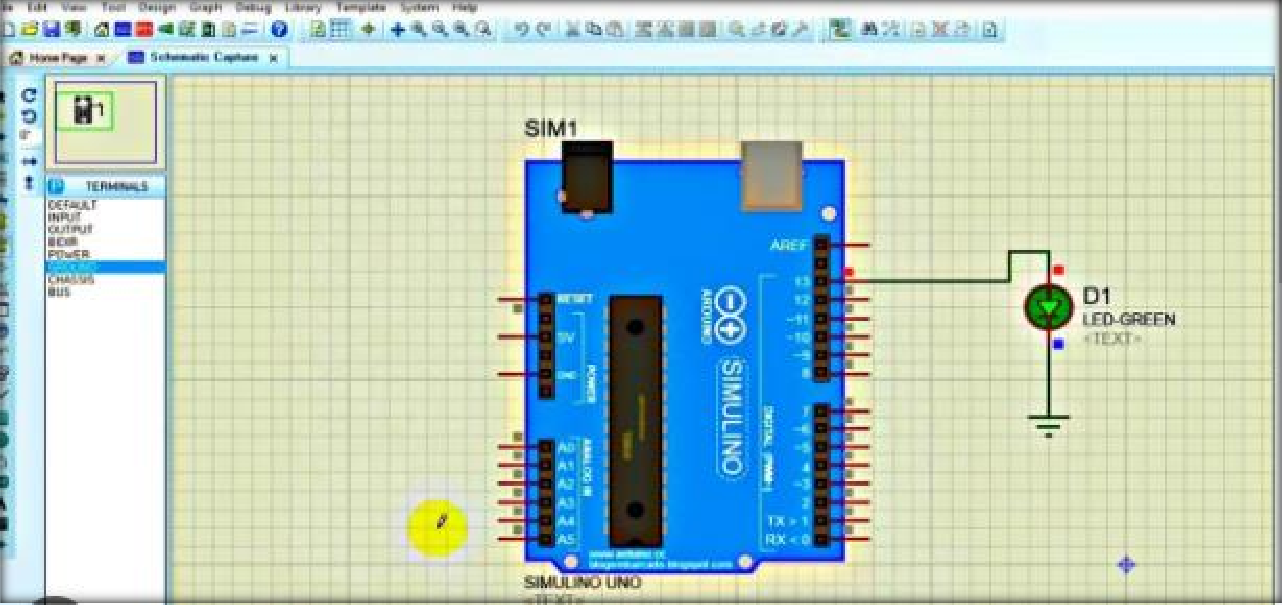
**Hardware Implementation: LED Blink Test**







# Simulation:



**Discussion:**

In this debate, we acknowledged the value of assembly language for microcontroller programming due to its ability to regulate and optimize hardware at the granular level. We gained knowledge of the particular microcontroller architecture and overcome early difficulties with comprehension assembly syntax. With the aid of the given development tools, we were able to advance from simple applications to more difficult tasks.We were able to see how our programs were used in the real world by loading and running assembly code on the microcontroller. Our programming abilities were boosted through debugging and testing, which were essential for finding and fixing mistakes.We talked about the real-world uses of assembly language in various embedded systems and offered some practice exercises and advised more readings for ongoing development. Overall, this lab exercise was successful in advancing our comprehension of assembly language and preparing us for upcoming microcontroller projects.

# Reference(s):

1. https://[www.arduino.cc/.](http://www.arduino.cc/)
2. https:/[/www.course](http://www.coursera.org/learn/arduino/lecture/ei4ni/1-10-first-glance-at-a-program)r[a.org/learn/arduino/lecture/ei4ni/1-10-first-glance-at-a-program](http://www.coursera.org/learn/arduino/lecture/ei4ni/1-10-first-glance-at-a-program)
3. Jeremy Blue; Exploring Arduino: Tools and Techniques for Engineering Wizardry