AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Engineering

Lab Report

Experiment # 04

Experiment Title: Study of a Digital Timer using the millis() function of Arduino to avoid problems associated with the delay() function and power measurement in different power reduction modes while the circuit is in sleep modes.

Date of Perform:	16 April 2025	**Date of Submission:**	8 April 2025
Course Title:	Microprocessor and Embedded Systems Lab		
Course Code:	EE4103	**Section:**	G
Semester:	Spring 2024-25	**Degree Program:**	BSc in CSE/EEE
Course Teacher:	**Prof. Dr. Engr. Muhibul Haque Bhuyan**		

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 # 02

Sl No	Name	ID	PROGRAM	SIGNATURE
1	MAHBUB HASAN	22-47419-2	BSc in CSE	
2	MUJTAHID TABASSUM	22-47430-2	BSc in CSE	
3	MD. SALMAN ISLAM	22-47393-2	BSc in CSE	
4	MD. SIAM MEHEDI	22-48342-3	BSc in CSE	
5	MD. TOYEB ALI	23-51819-2	BSc in CSE	

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 application of the millis() function, a built-in Timer of Arduino.
2. Build a digital timer to turn on several LEDs every minute sequentially on an Arduino Microcontroller Board.
3. Implement a sequential LED light pattern control system using an input switch and an Arduino Microcontroller Board.
4. Measure the amount of power and energy saved using power down or power save mode.

Equipment List:

A. For LED Pattern,

1) Arduino IDE (2.0.1 or any recent version)
2) Arduino Microcontroller board
3) LED lights (Red, Green, and Yellow- each 2)
4) Six 100Ohm resistors and one 10 kohm resistor
5) One tilt switch
6) Jumper wires

B. For Power measurement in different modes,

1) Arduino Board
2) Breadboard
3) Resistors
4) Switches
5) LED
6) USB Ammeter
7) Jumper Wires
8) Laptop or PC with Arduino IDE installed in it.

Circuit Diagram:

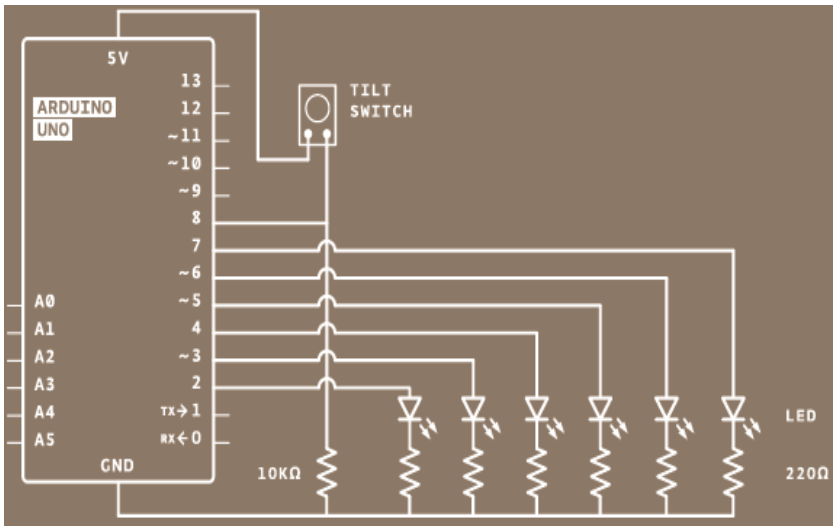


Figure 1: Circuit Diagram of an LED Light Pattern Control System using a Tilde switch and an Arduino Microcontroller Board.

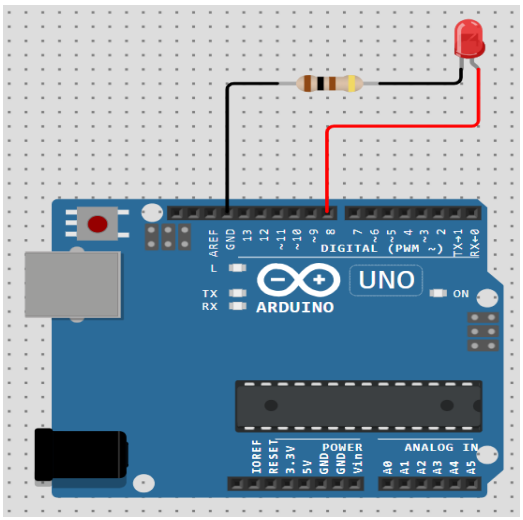
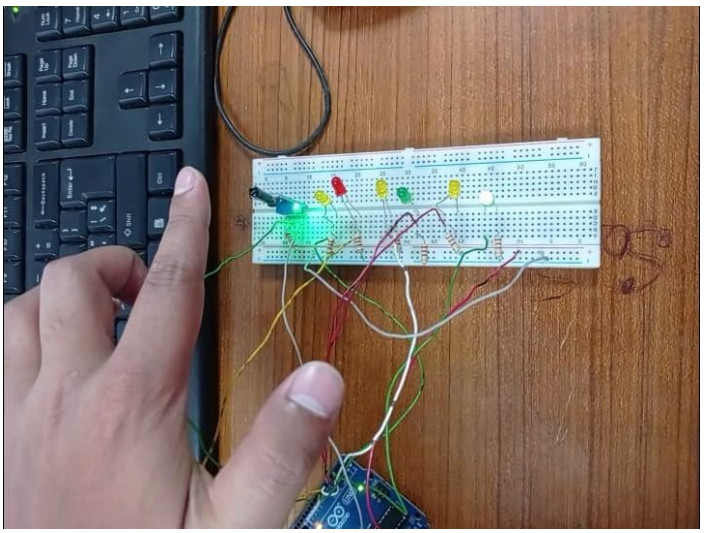


Figure 2: Circuit Diagram for calculating power measurement during different states of LED without any switch.

Experimental Output Results:

For these experiments Arduino board was programmed according to the lab manual where it was programmed to blink the light from PWM pin 2 to 7 and pin 8 was configured as Switch State. If the switch was changing the LED turn off and start blinking from the 1st. For the 2nd experiment a power saving mode was introduced using LowPower.idle() function. Because of the sleep mode, selected part will go for sleep which can save power without unnecessary usage.

A. LED light pattern system using millis() function:



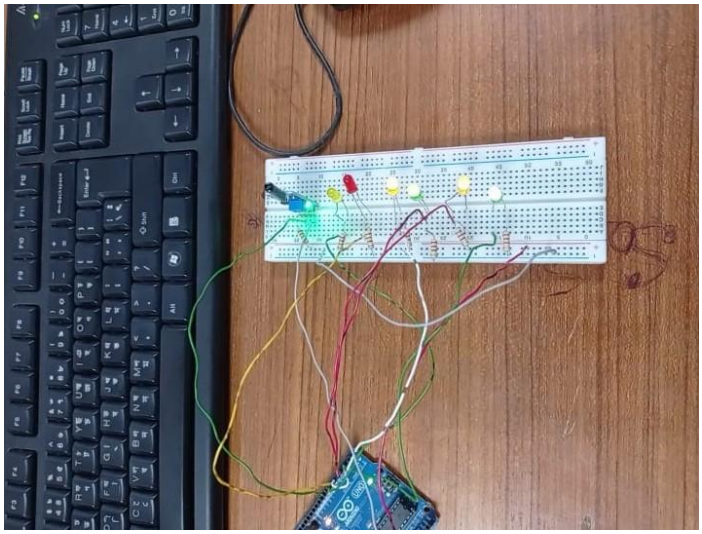


Figure 3: Experimental Result of LED Light Pattern System started to blink.

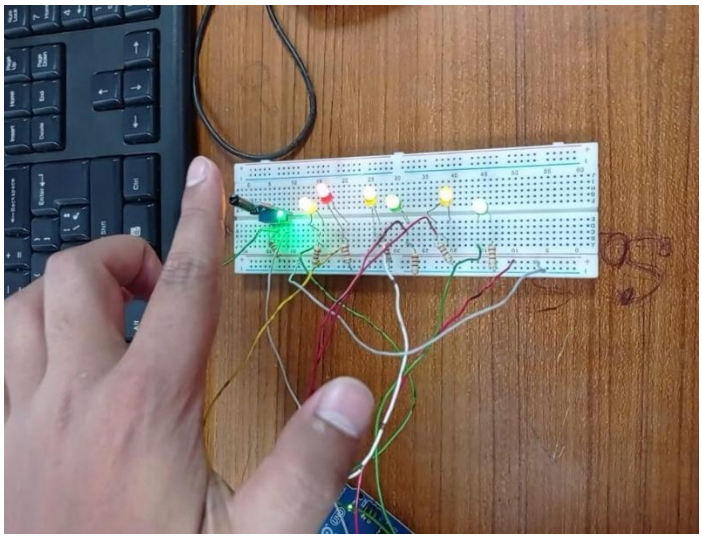


Figure 4: LED Light blinking in a pattern.

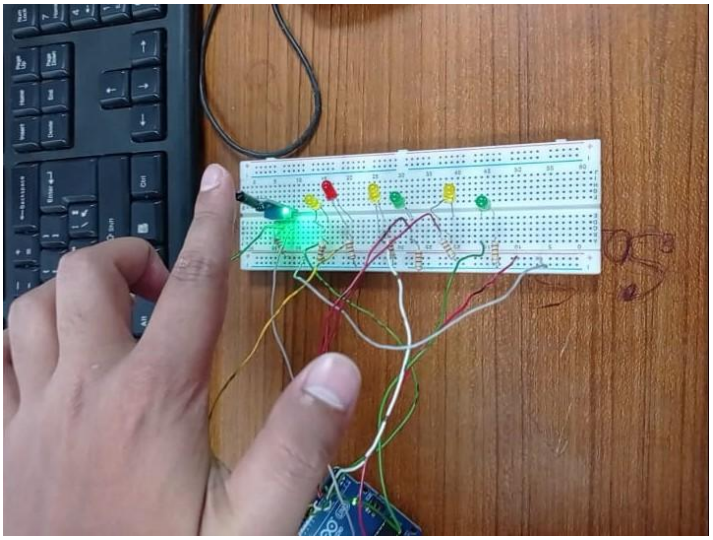


Figure 5: Experimental result of 7 LED lights blinking in a pattern.

Figure 6: Experimental result after touching the tilt switch.

B. Power Measurement in Different Modes:

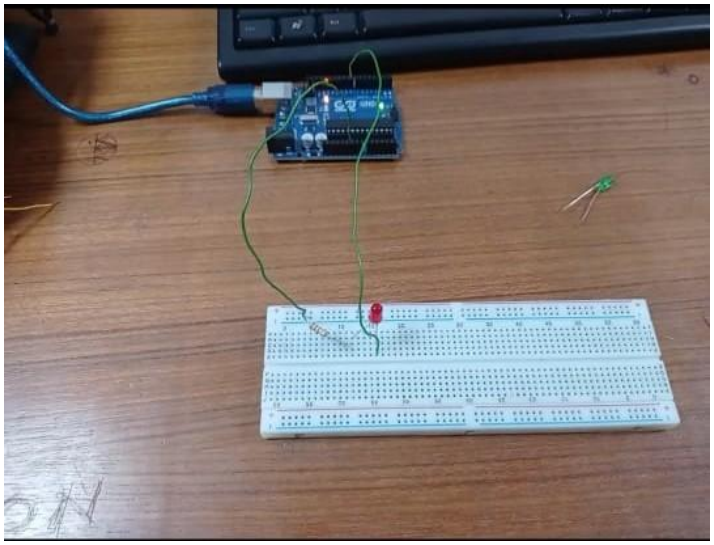


Figure 7: Power measurement in off state.

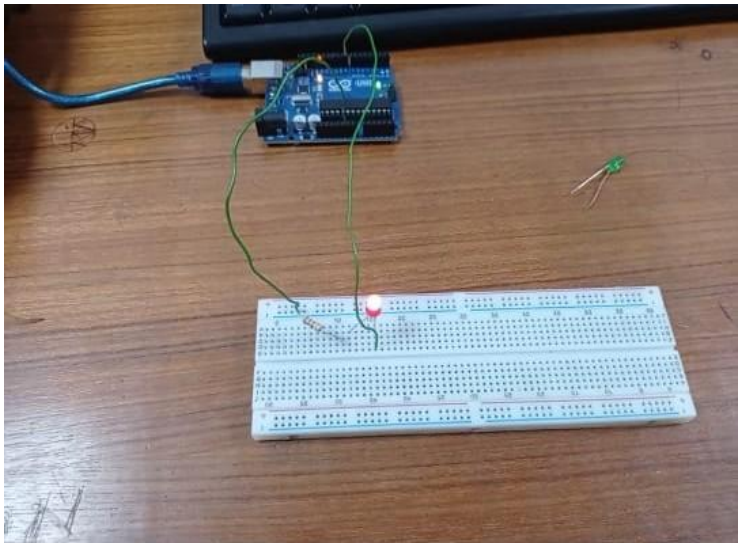


Figure 8: Power measurement in operating state.

Figure 9: Voltage and current dropping while the circuit is in operation mode.

Figure 10: Measurement of current in sleep mode.

Figure 11: Measurement of current in off state.

Simulation Output Results:

In proteus the Arduino board and tilt switch were missing so the necessary extension was downloaded. The circuit was designed according to the circuit diagram provided in the lab manual. Then the Arduino code was complied with Arduino IDE and generated the .hex file. Then the .hex was uploaded to the Arduino uno board. For the 2nd experiment the lowpower.h header file was install in the IDE. Finally, the two programs were run successfully, and the output was shown below.

A. LED light pattern system:

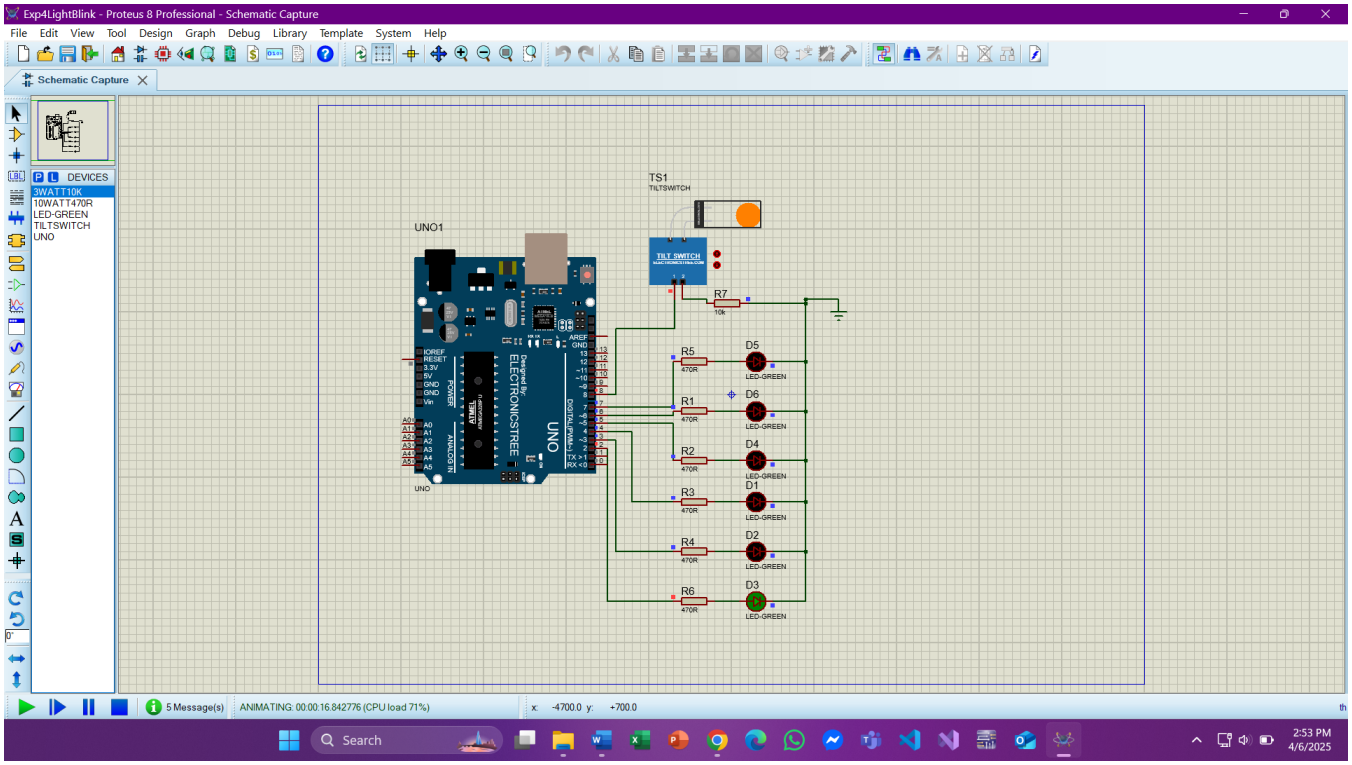


Figure 12: Simulation Result of LED Light Pattern System started to blink.

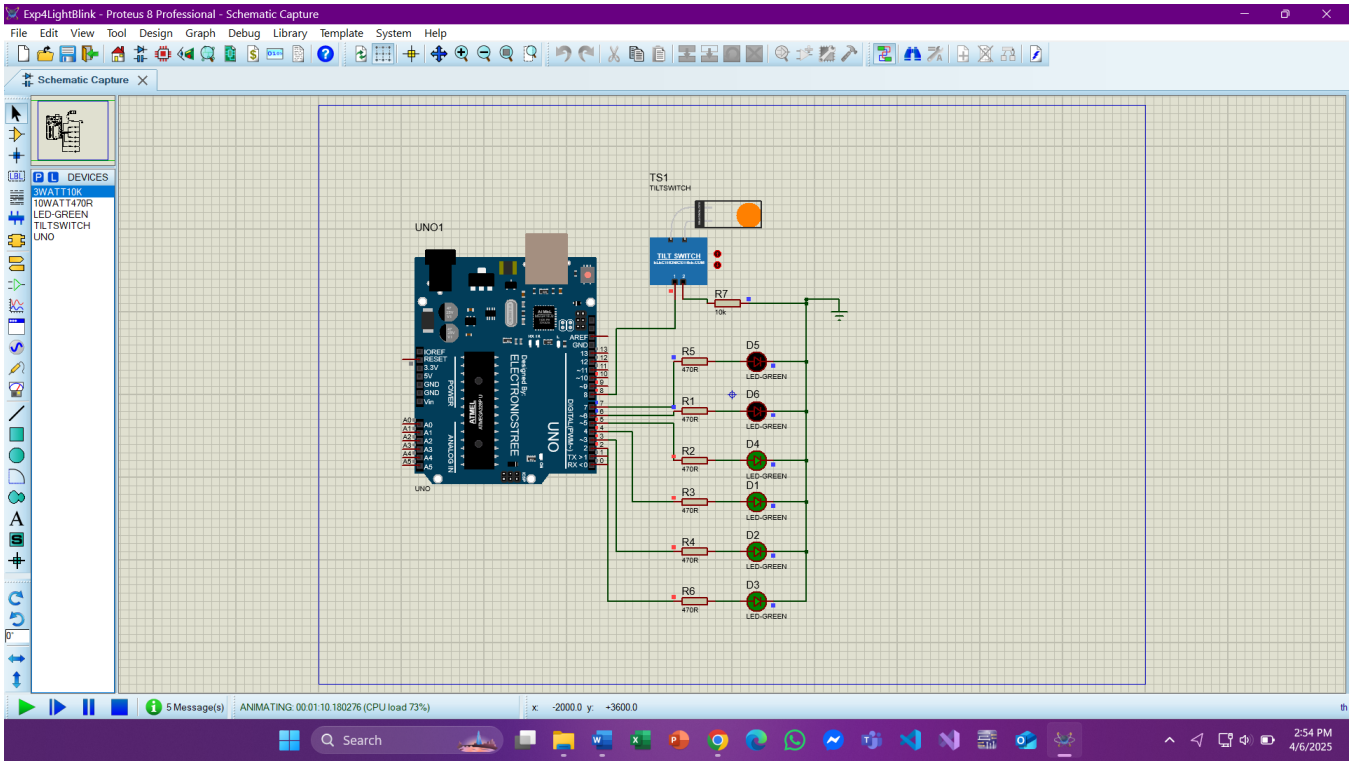


Figure 13: Simulation Result of 7 LED lights blinking in a pattern.

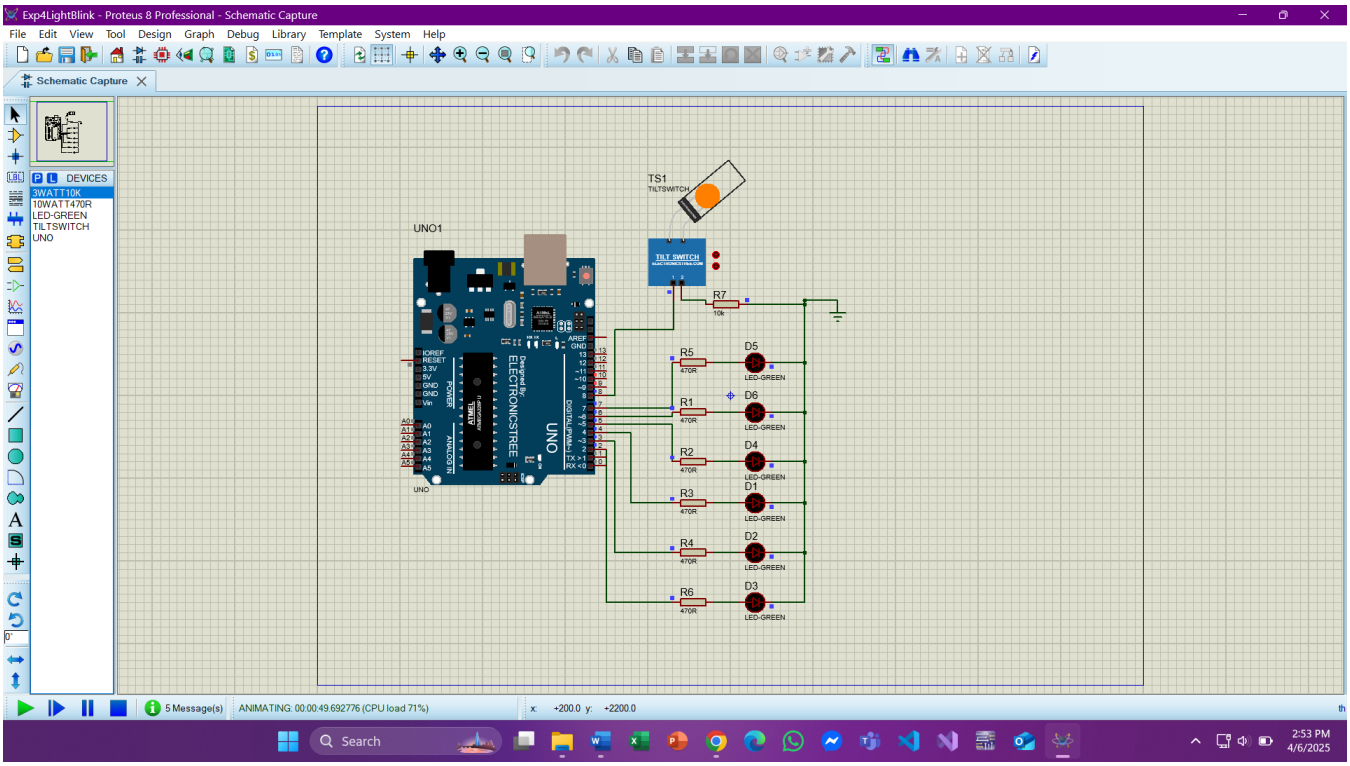


Figure 14: Simulation Result after touching the tilt switch.

B. Power Measurement in Different Modes:

The amount of current is very low so it can't be measured accurately in Ammeter. So, a probe was used to measure the current in the simulation.

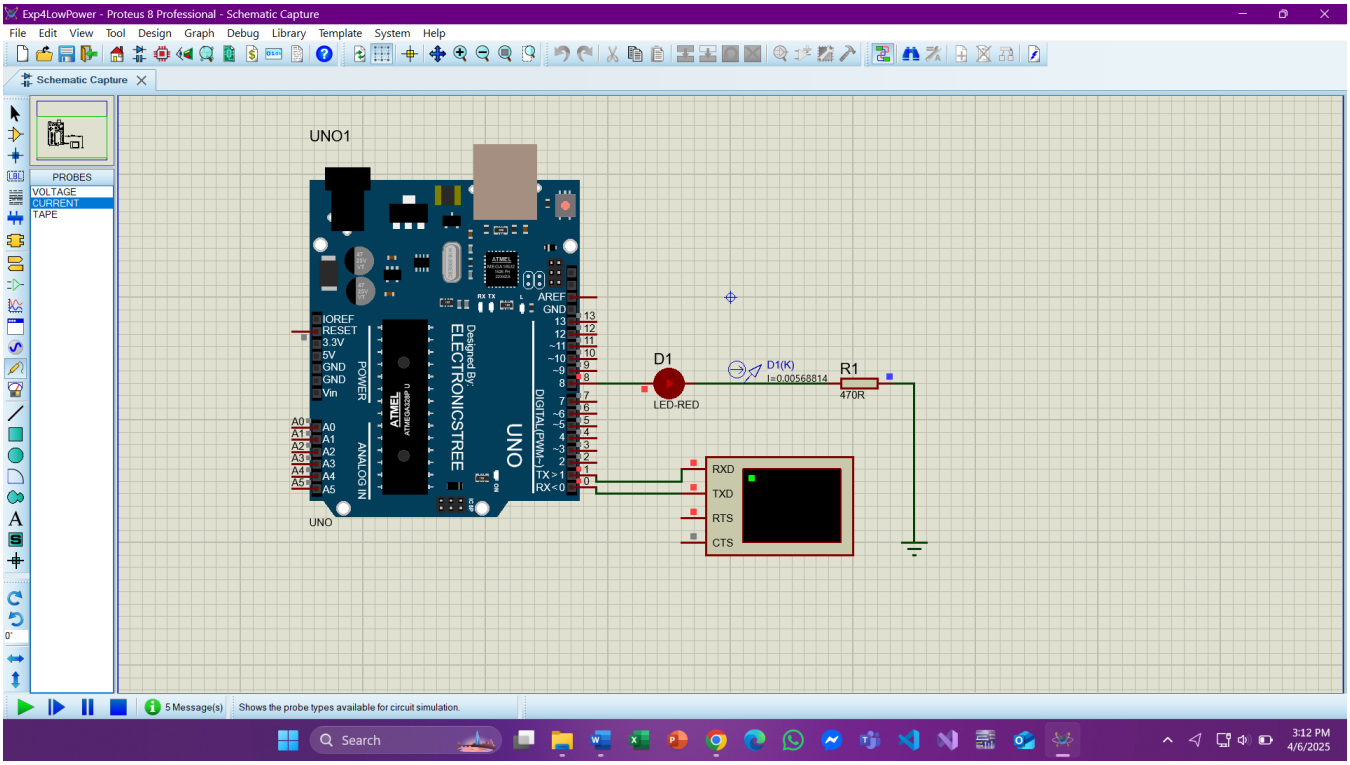


Figure 14: Measurement of current in operating state.

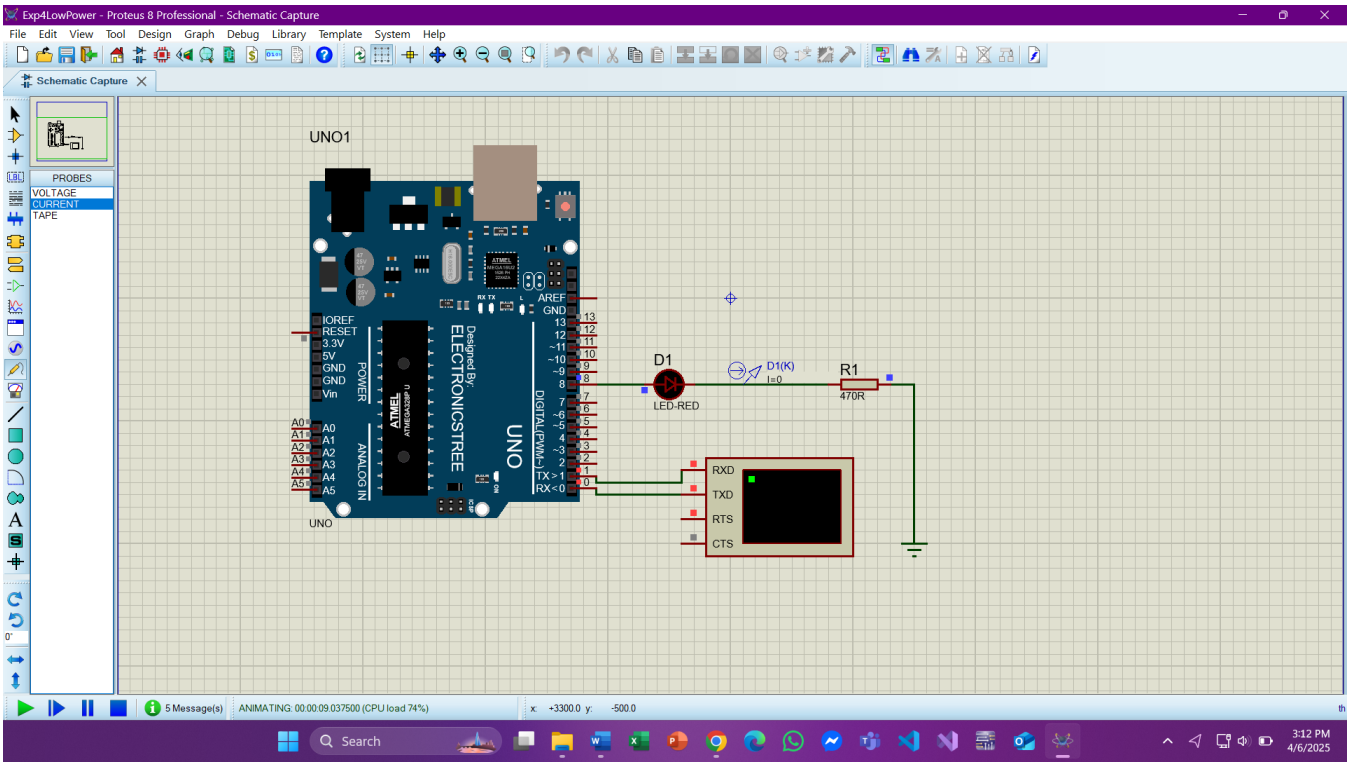


Figure 15: Measurement of current in sleep mode.

Answers to the Questions in the Lab Manual:

1) Code for LED lighting pattern system

```
const int SwitchPin = 8;
unsigned long PreviousTime = 0;
int SwitchState = 0;
int PrevSwitchState = 0;
int led = 2;
long interval = 60000; // for 1 min = 60,000 ms delay

void setup() {
for (int x = 2; x < 8; x++) {
pinMode(x, OUTPUT);
}
pinMode(SwitchPin, INPUT);
}

void loop() {
unsigned long CurrentTime = millis();
if (CurrentTime - PreviousTime > interval) {
PreviousTime = CurrentTime;
```

```
digitalWrite(led,
HIGH); led++;
if (led == 7){
}
}
SwitchState                    =
digitalRead(switchPin);        if
(SwitchState != PrevSwitchState){
for (int x = 2 ; x < 8; x++) {
digitalWrite(x, LOW);
}
led = 2;
PreviousTime = CurrentTime;
}
PrevSwitchState = SwitchState;
}
```

Code for Power saving mode

```
#include <LowPower.h>

#define LED_RED 8

void setup() {
  Serial.begin(9600);
  pinMode(LED_RED, OUTPUT);
  // SMCR = 0b00001101;
}


void loop() {
        digitalWrite(LED_RED, HIGH);
        delay(2000);
        Serial.println("Arduino:- I am going for a Nap");
        digitalWrite(LED_RED,LOW);
        LowPower.idle(SLEEP 4S, ADC OFF, TIMER2 OFF, TIMER1_OFF,
        TIMER0_OFF, SPI_OFF, USART0_OFF, TWI_OFF);
        Serial.println("Arduino:- Hey I just Woke up");
        delay(4000);
        }
```

3) Code for the delays according to my id

```
const int SwitchPin = 8;
unsigned long PreviousTime = 0;
int SwitchState = 0;
int PrevSwitchState = 0;
int led = 2;
long interval[] = {5000,1000,8000,1000,9000,2000}; // for delay according to id 23-51819-2
void setup() {
for (int x = 2; x < 8; x++) {
pinMode(x, OUTPUT);
}
pinMode(SwitchPin, INPUT);
}
void loop() {
unsigned long CurrentTime = millis();
for(int i=0;i<6;i++)
{
if (CurrentTime - PreviousTime > interval[i]) {
PreviousTime = CurrentTime;
digitalWrite(led, HIGH);
led++;
if (led == 7){
}
}
}
SwitchState = digitalRead(SwitchPin);
if (SwitchState != PrevSwitchState){
for (int x = 2 ; x < 8; x++) {
digitalWrite(x, LOW);
}
led = 2;
PreviousTime = CurrentTime;
}
PrevSwitchState = SwitchState;
}
```

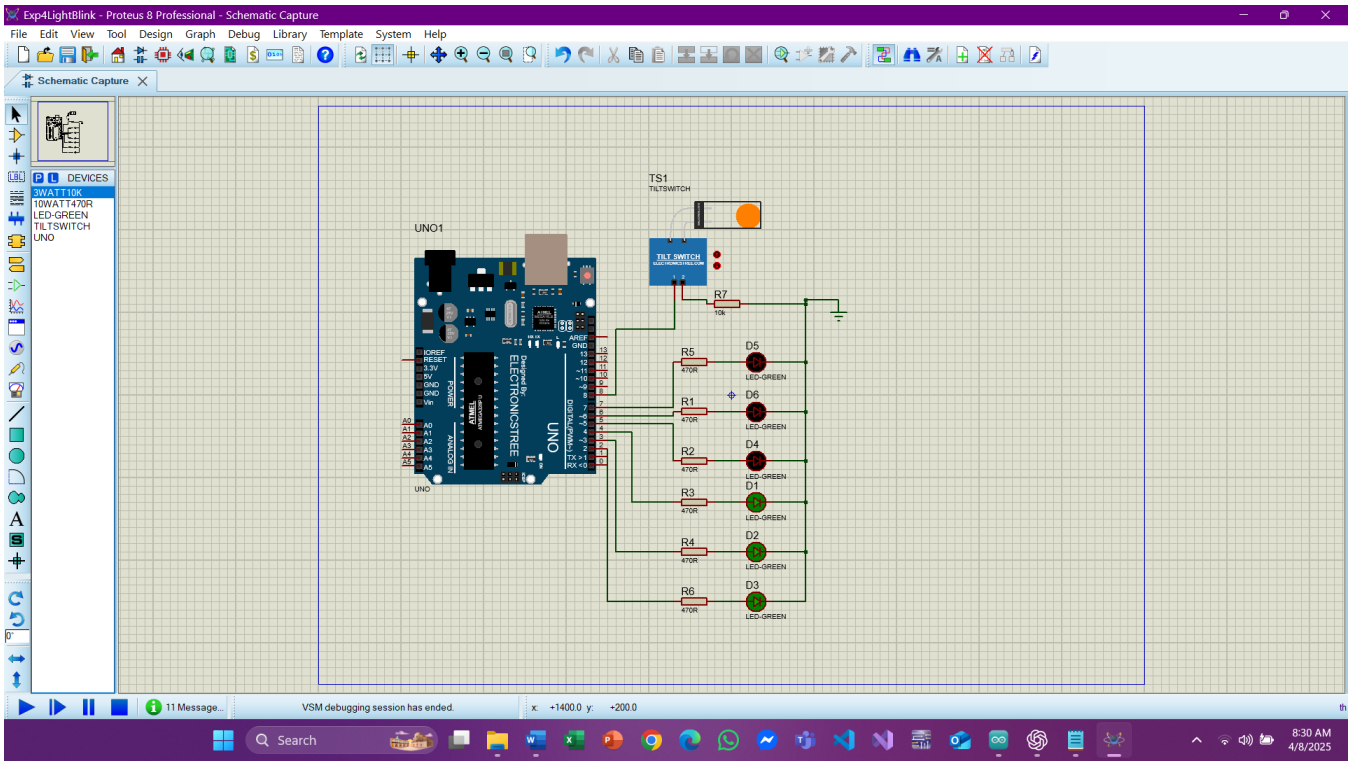


Figure 16: Delay according to my ID LED light blinking in a pattern.

4) Code for the up down direction LED pattern system

```
const int SwitchPin = 8;
unsigned long PreviousTime = 0;
long interval = 3000; // 3s interval

int currentLED = 2;
bool directionUp = true;

void setup() {
  for (int x = 2; x <= 7; x++) {
    pinMode(x, OUTPUT);
    digitalWrite(x, LOW);
  }
  pinMode(SwitchPin, INPUT);
}

void loop() {
  int SwitchState = digitalRead(SwitchPin);

  // Set direction based on switch state
  directionUp = (SwitchState == HIGH);

  unsigned long CurrentTime = millis();
  if (CurrentTime - PreviousTime >= interval) {
    PreviousTime = CurrentTime;
```

```
    // Turn on the current LED (others stay on)
    digitalWrite(currentLED, HIGH);

    // Move to the next LED based on direction
    if (directionUp) {
      currentLED++;
      if (currentLED > 7) currentLED = 7; // stay at 7
    } else {
      currentLED--;
      if (currentLED < 2) currentLED = 2; // stay at 2
    }
  }
}
```

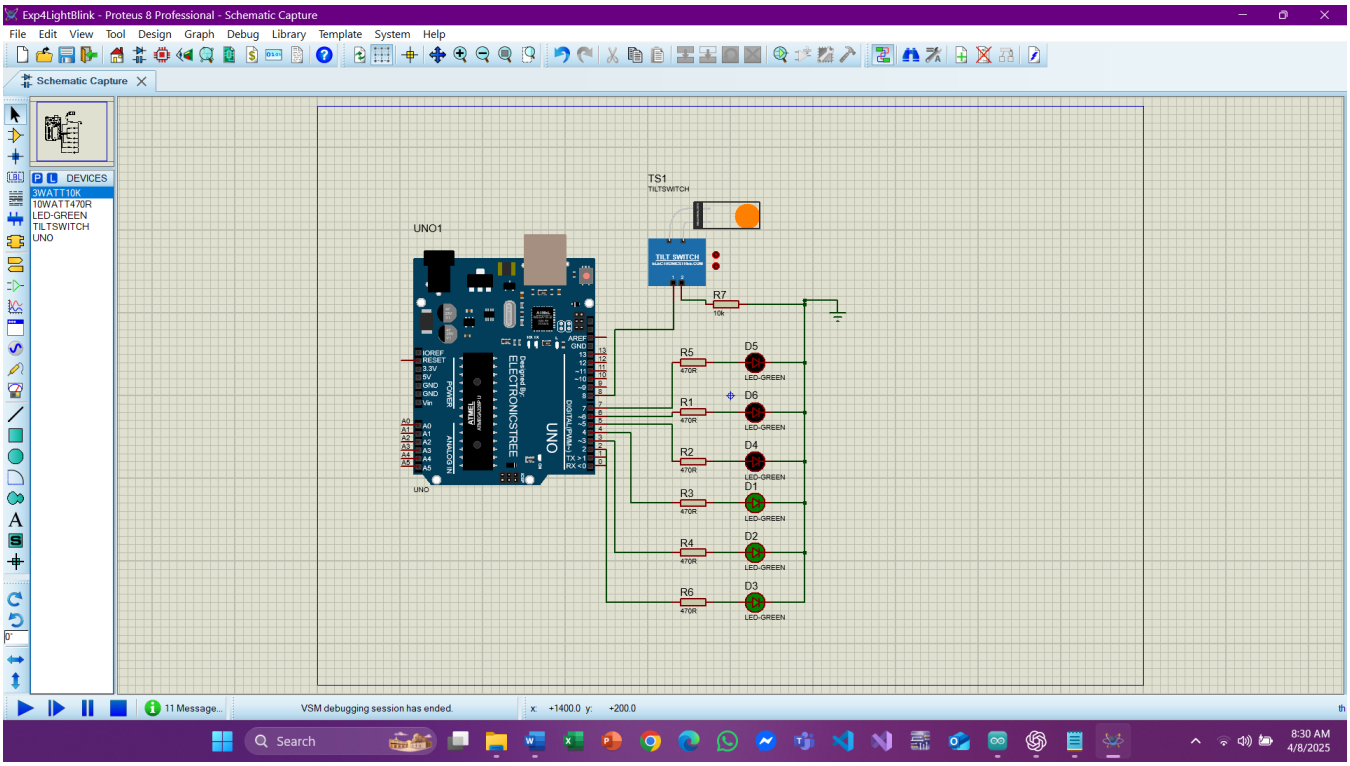


Figure 17: Light blinking in one direction.

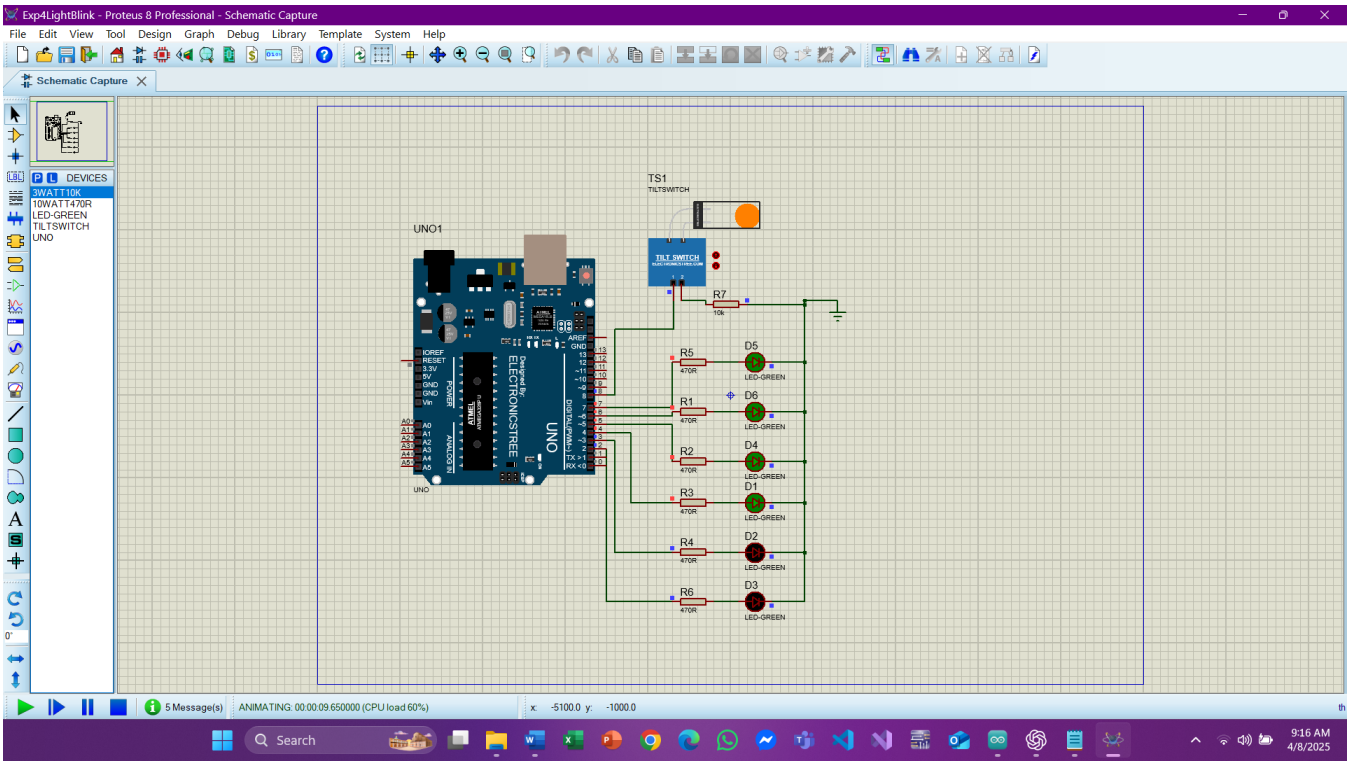


Figure 18: LED blinking in another direction while changing the state.

7) In the operating mode the current drop is 0.18A on average from fig.9

We know,

$$P = VI = 5.048 * 0.18 = 0.91 \text{ W}$$

In the sleep mode the current drop is 0.008A from fig.10

Again,

$$P = VI = 5.055 * 0.008 = 0.04 \text{ W}$$

Therefore, Power Saving = 0.91 – 0.04 = 0.87 W (Almost)

Discussion:

In this experiment, a digital timer was studied using the millis() function of Arduino. The millis() function was utilized to implement non-blocking delays, allowing other processes to run simultaneously, unlike the delay() function, which halts the entire program execution. By using millis(), timing issues and program responsiveness problems commonly associated with delay() were avoided.

Additionally, power consumption was measured while the circuit was placed in various power reduction modes during sleep states. Different sleep modes, such as idle, power-down, and standby, were tested to observe their effects on power efficiency. It was observed that significant power savings could be achieved, and 0.87 W was saved by appropriately selecting lower power modes, especially during inactive periods of the circuit.

The experiment successfully demonstrated how the combination of millis()-based timing and proper sleep mode configuration can lead to efficient timer operation and reduced power consumption in Arduino-based systems.

References:

[1] https://www.arduino.cc/.
[2] ATMega328 manual
[3] https://www.avrfreaks.net/forum/tut-c-newbies-guide-avr-timers
[4] http://maxembedded.com/2011/06/avr-timers-timer0/