American International University- Bangladesh (AIUB)
Faculty of Engineering (EEE)

Course Name:	MICROPROCESSOR AND EMBEDDED SYSTEMS	**Course Code:**	EEE 4103
Semester:	2023-2024, Spring	**Sec:**	F
Lab Instructor:	Md Sajid Hossain	**Group:**	06

Experiment No:	04
Experiment Name:	Part-2: Digital Timer project using millis() function to avoid the problems associated with delay().

Submitted by (NAME):	**TRIDIB SARKAR**	**Student ID:**	**22-46444-1**

Group Members	**ID**	**Name**
1.	22-46588-1	AZMINUR RAHMAN
2.	22-46444-1	TRIDIB SARKAR
3.	19-41203-1	MD. IMTIAZ HOSSAIN
4.	19-40315-1	S.M. FAYSAL MAHMUD
5.	21-45397-3	SOWRABH CHANDRA DAS
6.	21-45489-3	AVISHEK CHANDA PRATYAY
7.	21-45408-3	KOWSHIK HALDER

Performance Date:	**February 14, 2024**	**Due Date:**	

Marking Rubrics (to be filled by Lab Instructor)

Category	Proficient [6]	Good [4]	Acceptable [2]	Unacceptable [1]	Secured Marks
Theoretical Background, Methods & procedures sections	All information, measures and variables are provided and explained.	All Information provided that is sufficient, but more explanation is needed.	Most information correct, but some information may be missing or inaccurate.	Much information missing and/or inaccurate.	
Results	All of the criteria are met; results are described clearly and accurately;	Most criteria are met, but there may be some lack of clarity and/or incorrect information.	Experimental results don't match exactly with the theoretical values and/or analysis is unclear.	Experimental results are missing or incorrect;	
Discussion	Demonstrates thorough and sophisticated understanding. Conclusions drawn are appropriate for analyses;	Hypotheses are clearly stated, but some concluding statements not supported by data or data not well integrated.	Some hypotheses missing or misstated; conclusions not supported by data.	Conclusions don't match hypotheses, not supported by data; no integration of data from different sources.	
General formatting	Title page, placement of figures and figure captions, and other formatting issues all correct.	Minor errors in formatting.	Major errors and/or missing information.	Not proper style in text.	
Writing & organization	Writing is strong and easy to understand; ideas are fully elaborated and connected; effective transitions between sentences; no typographic, spelling, or grammatical errors.	Writing is clear and easy to understand; ideas are connected; effective transitions between sentences; minor typographic, spelling, or grammatical errors.	Most of the required criteria are met, but some lack of clarity, typographic, spelling, or grammatical errors are present.	Very unclear, many errors.	
Comments:				Total Marks (Out of):	

Title: Part-2: Digital Timer project using millis() function to avoid the problems associated with delay().

Abstract:

This experiment aims to build a digital timer using an Arduino board and its built-in Timer library. The timer will be set to turn on an LED every minute, with the additional function of keeping track of the time spent on a particular task. The experiment involves connecting the LED to the Arduino board, coding the timer functionality using the millis() function, and incorporating a tilt switch for user input. The experiment demonstrates the usefulness and practical application of Arduino technology for time management and task tracking, while also providing hands-on experience in electronics and coding. By the end of the experiment, the digital timer will enable better time management and planning by measuring the duration of work on different projects. This experiment offers an engaging way to learn about electronics and coding, while also providing a practical tool for daily use.

Introduction:

The objectives of this experiment are as follows:

1. To familiarize with the application of millis() function of Arduino.
2. To develop a digital timer to turn on several LEDs every minute.
3. To be able to know the working time of the project by using the Arduino's built-in Timer.

Theory and Methodology:

Until now, when something needed to happen at a specific time interval with the Arduino, delay() was used. This was handy but a little confining. When delay() was called by the Arduino, its current state was frozen for the duration of the delay. That meant there could be no other input or output while it was waiting. Delays were also not very helpful for keeping track of time. If something needed to be done every 10 seconds, a 10- second delay would have been cumbersome. These problems were solved by the millis() function. The time the Arduino had been running in milliseconds was kept track of by this function. Variables have been declared as int so far. An int (integer) is a 16-bit number that holds values between -32,768 and 32,767. While these may be large numbers, if the Arduino was counting 1000 times a second with millis(), the available space would run out in less than a minute. The long datatype, on the other hand, holds a 32-bit number (between -2,147,483,648 and 2,147,483,647). Since time cannot run backwards to get negative numbers, the variable used to store millis() time is called an unsigned long. When a datatype is called unsigned, it only represents positive numbers. This allows counting even higher. An unsigned long can count up to 4,294,967,295. That provides enough space for millis() to store time for almost 50 days. By comparing the current millis() to a specific value, it can be determined if a certain amount of time has passed. When the Digital Timer was turned over, its state was changed by a tilt switch, setting off another cycle of LEDs turning on. The tilt switch functioned just like a regular switch in that it was an on/off sensor. It was used as a digital input. What made tilt switches unique was their ability to detect orientation. Typically, they had a small cavity inside the housing that contained a metal ball. When tilted in the proper way, the ball rolled to one side of the cavity and connected the two leads on the breadboard, closing the switch. With six LEDs, the timer ran for six minutes **[2].**

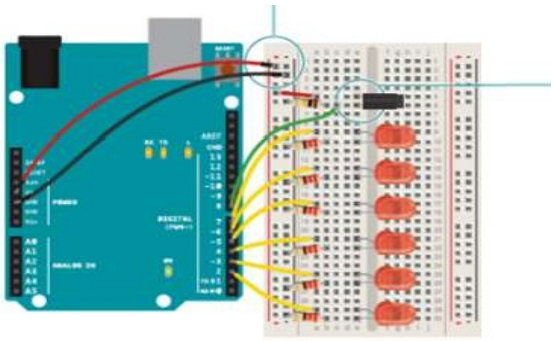

Figure 1: Experimental setup for the digital timer project circuit **[4].**

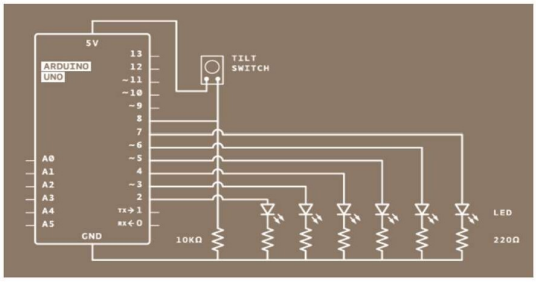

Figure 2: Simulation setup for the digital timer project circuit **[4].**

Apparatus:

1. Arduino Uno/ Arduino Mega
2. Tilt sensor
3. LED lights (Six)
4. Resistors

Precautions:

1. Safety precautions were observed during the experiment, including the proper handling of electronic components and materials to prevent accidents or damage.
2. Wiring and connections were carefully attended to prevent short circuits or incorrect wiring.
3. Simulations were run in Proteus software to validate the circuits and code before physical implementation to reduce the risk of errors.
4. Power sources were disconnected when changes were made to the circuit.

Experimental Procedure:

1. The circuit was constructed for the digital timer project with LEDs, a pushbutton, and an Arduino board. Connect the Arduino board to a PC using a USB cable.
2. The timer count required to achieve a 1ms delay was calculated using the millis() function.
3. The code was written for the Digital Timer project. Ensured that the code used the millis() function for the specified time.
4. In the Arduino IDE, the Arduino board was set up, and the code was written.
5. Register initialization was carried out, and the highest prescaling was selected for safer operation.
6. The code was uploaded to the Arduino board, and the expected output was achieved.
7. The hardware setup was verified to ensure that the expected output, in terms of LED sequencing and timer functionality, was achieved.

Simulation and Measurement:

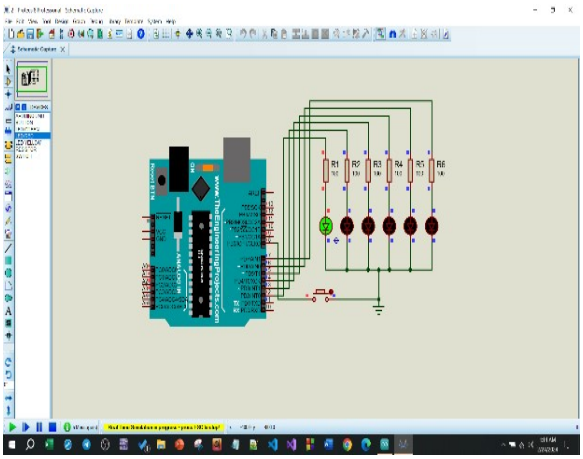

Figure 3: Simulation of One Green LED is On.

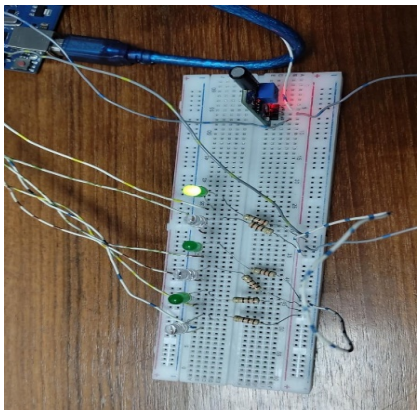

Figure 4: Image of One Green LED is On.

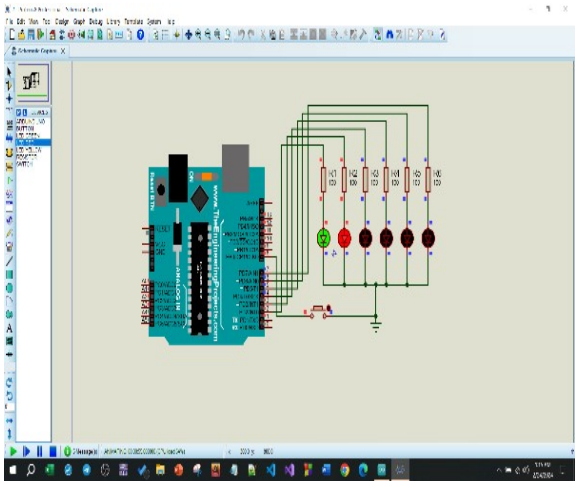

Figure 5: Simulation of 1 Green and 1 Red LED is On.

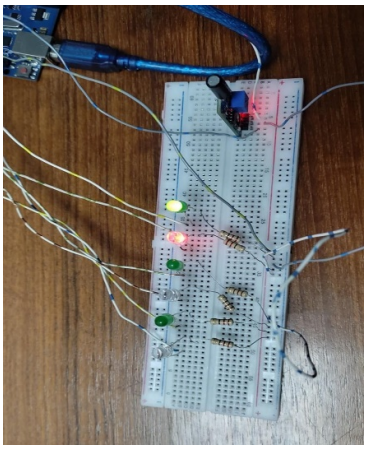

Figure 6: Image of 1 Green and 1 Red LED is On.

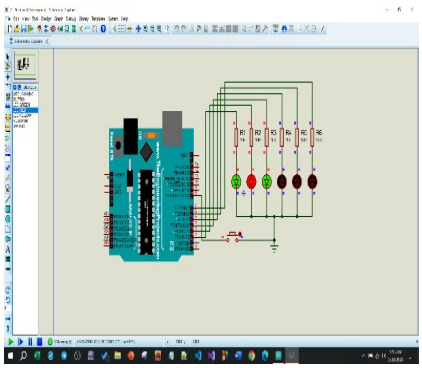

Figure 7: Simulation of 2 Green and 1 Red LED is On.

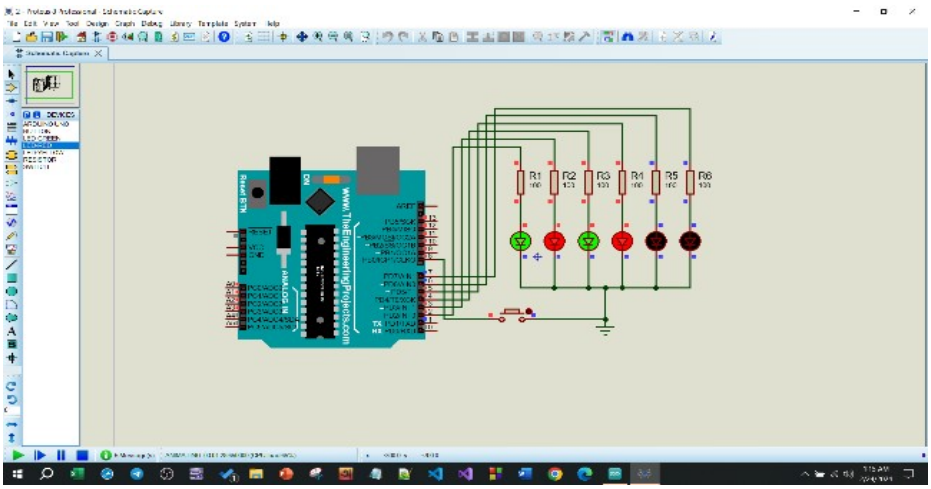

Figure 8: Simulation of **2** Green & **2** Red LED is On.

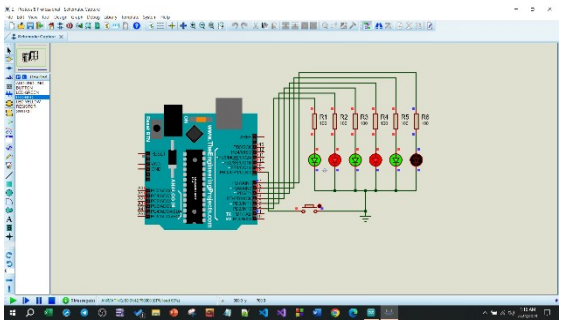

Figure 9: Simulation of **3** Green and **2** Red LED is On.

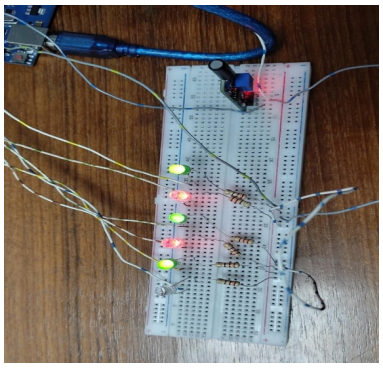

Figure 10: Simulation of **3** Green and **2** Red LED is On.

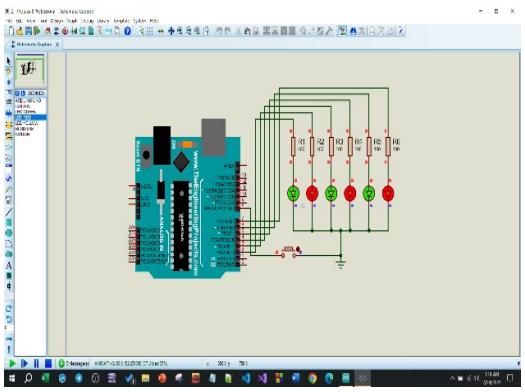

Figure 11: Simulation of When Six LED is On.

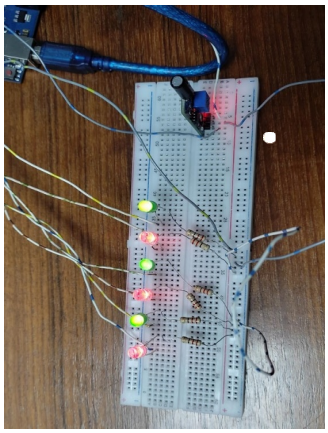

Figure 12: Image of When Six LED is On.

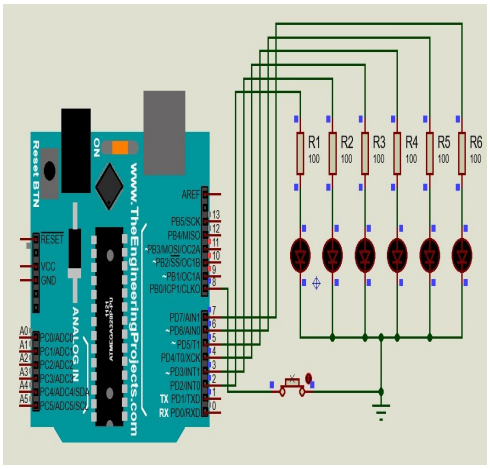

Figure 13: Simulation of When Reset Button is Pressed.

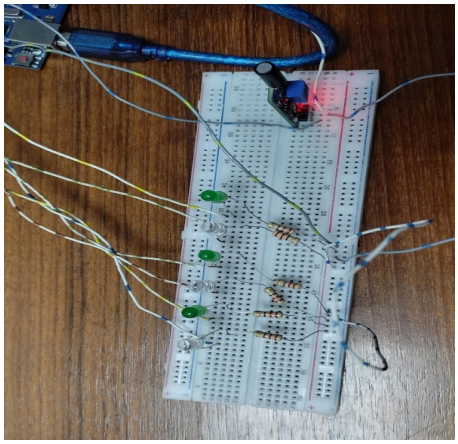

Figure 14: Image of When Reset Button is Pressed.

Code snippet for the System:

```
const int SwitchPin = 8;
unsigned long PreviousTime = 0;
int SwitchState = 0;
int PrevSwitchState = 0;
int led = 2;
long interval = 1000; // for 1 min = 60,000 ms delay

void setup()
{
  Serial.begin(9600);
  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);
  Serial.println(SwitchState);
  if (SwitchState != PrevSwitchState)
  {
    for (int x = 2 ; x < 8; x++)
    {
      digitalWrite(x, LOW);
    }
    led = 2;
    PreviousTime = CurrentTime;
  }
  PrevSwitchState = SwitchState;
}
```

Figure 15: Code snippet for the System.

Results/Findings:

1. All code snippets were provided.
2. Proteus simulation was included in the simulation, complete with detailed views.

Discussion:

The digital timer was built using an Arduino board and its built-in Timer library. The timer turns on an LED every minute and also keeps track of the time spent on a particular task. The millis() function was used instead of the delay() function to avoid freezing the Arduino's state during the delay and to allow for more precise time tracking. The tilt switch was incorporated for user input, and the experiment offers a practical tool for time management and task tracking. The digital timer offers an engaging way to learn about electronics and coding while also providing hands-on experience in both areas. The functionality of the digital timer was successfully tested, and the experiment demonstrated the practical application of Arduino technology for time management and planning.

Conclusion:

In conclusion, this experiment successfully used an Arduino board and its built-in Timer library. By utilizing timers, the timer turns on an LED every minute and also keeps track of the time spent on a particular task. This experiment provided valuable hands-on experience and expanded the knowledge of digital timers.

Reference:

[1] Arduino, "Arduino - Home," Available: https://www.arduino.cc , Accessed On: 23-Feb-2024. [Online].

[2] Introduction to AVR Timers. Available: https://www.avrfreaks.net/forum/tut-c-newbies-guide-avr-timers , Accessed On: 23-Feb-2024. [Online].

[3] AVR Freaks. Available: https://www.avrfreaks.net/forum/tut-c-newbies-guide-avr-timers , Accessed On: 23-Feb-2024. [Online].

[4] AIUB Lab Manual.