AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Engineering

Lab Report

Experiment # 04

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

Course Title:	MICROPROCESSOR AND EMBEDDED SYSTEMS LAB		
Course Code:	COE3104	**Section:**	A
Semester:	Spring 2022-23	**Degree Program:**	BSc in CSE/BSc in EEE
Course Teacher:	**Md. Ali Noor**		

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 and no part of it has been copied from any other student's work or from 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 clearly acknowledged in the assignment.
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 detecting plagiarism.
6. I/we give permission for 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 # 06

Sl No	Name	ID	Program
1	Syed Aftab Uddin	19-41522-3	EEE
2	Syeda Aynul Karim	19-41829-3	CSE
3	Md. Sumon	20-42556-1	CSE
4	Maimona Rahman Farjana	20-42954-1	CSE
5	Md. Rashedul Islam	20-43301-1	CSE
6	Medha Chowdhury	20-41930-1	CSE
7	Md. Sajidul Haque Shohan	20-42022-1	CSE

Faculty use only		
FACULTY COMMENTS	**Marks Obtained**	
	Total Marks	

Abstract:

As a prelusive experiment, knowledge about digital timers and usage of the Millis () function of Arduino the experiment was done to avoid problems associated with the delay () function. With the help of the Arduino Uno's built-in timer, this experiment aims to teach participants how to configure an Arduino board, use a digital timer, turn on an LED once every minute, and use delay functions. For simulation purposes, this experiment required a few key pieces of hardware, including an Arduino Uno board, six LED lights, jumper wires, one tilt switch, and an Arduino IDE. A timer or counter is a distinctive piece of hardware found inside many microcontrollers. They are used simply for counting. The Millis () function helps to solve problems with the delay () function. It keeps track of the time Arduino has been running in milliseconds.

Objective:

In this experiment, a digital timer will build that turns on an LED every minute. From this, it will know how long the projects are working on by using Arduino built-in Timer.

Apparatus:

- Arduino Uno/Arduino Mega Microcontroller Board
- Tilt sensor (One)
- LEDs (Six)

Results:

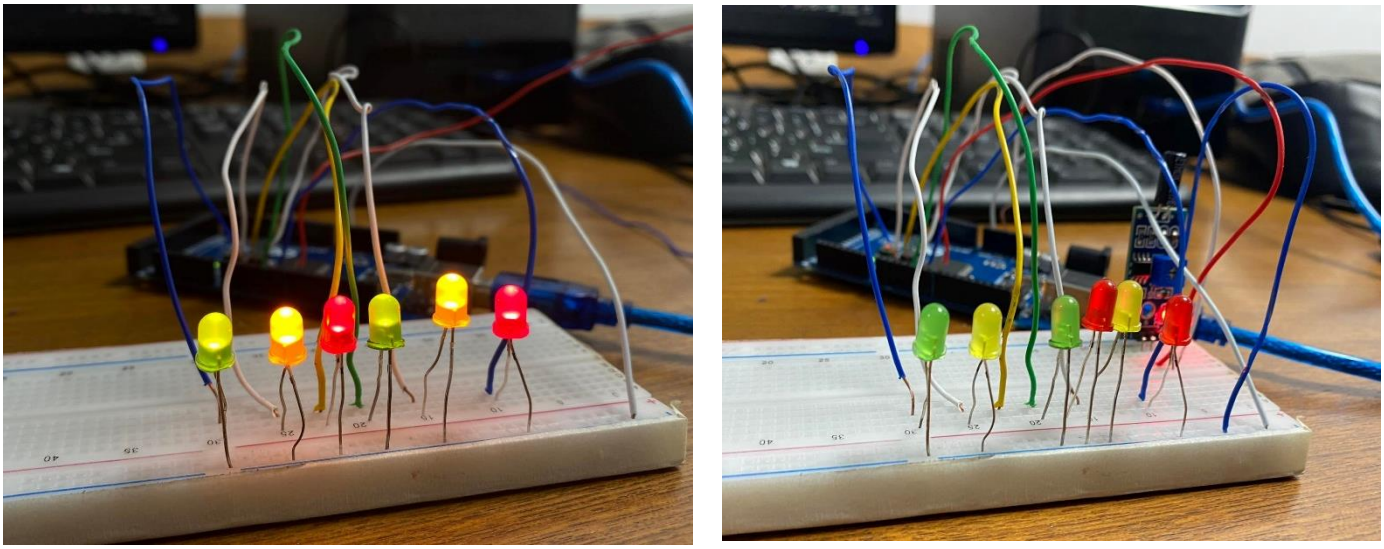

Fig 1: Practical circuit diagram

Simulation Results:

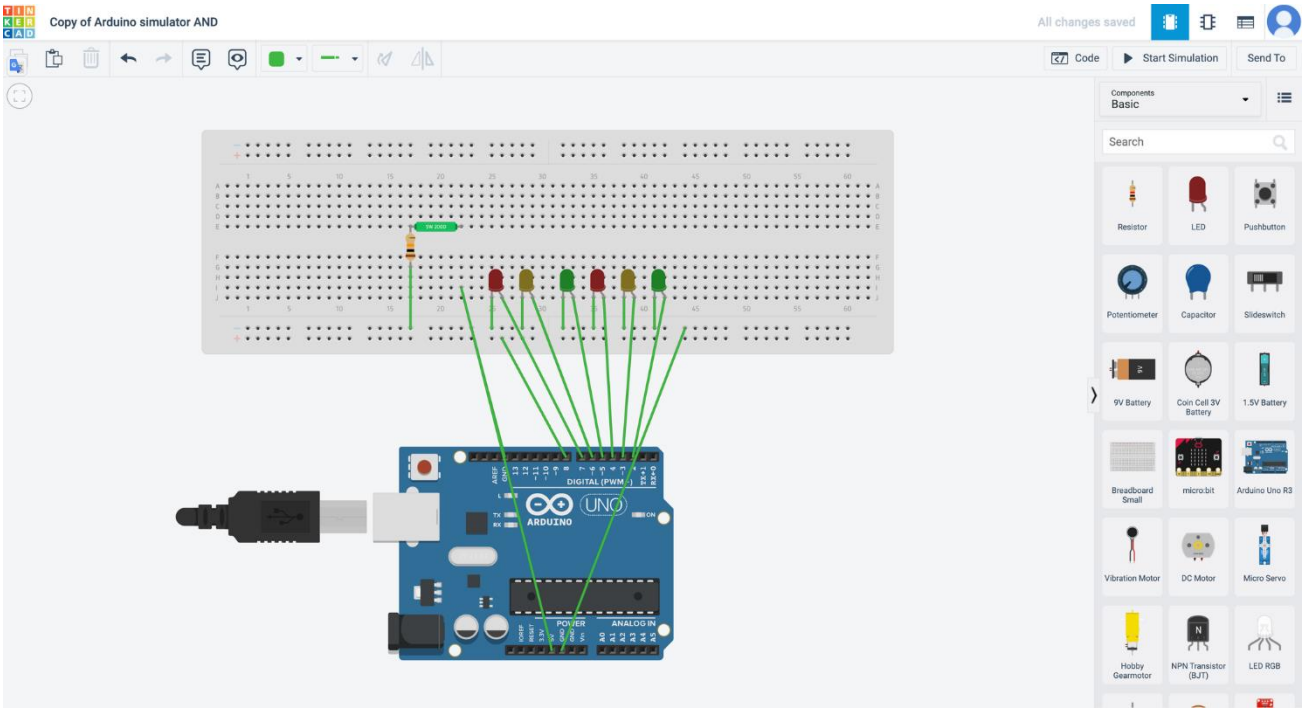

Fig 2: Simulated circuit diagram with lights off

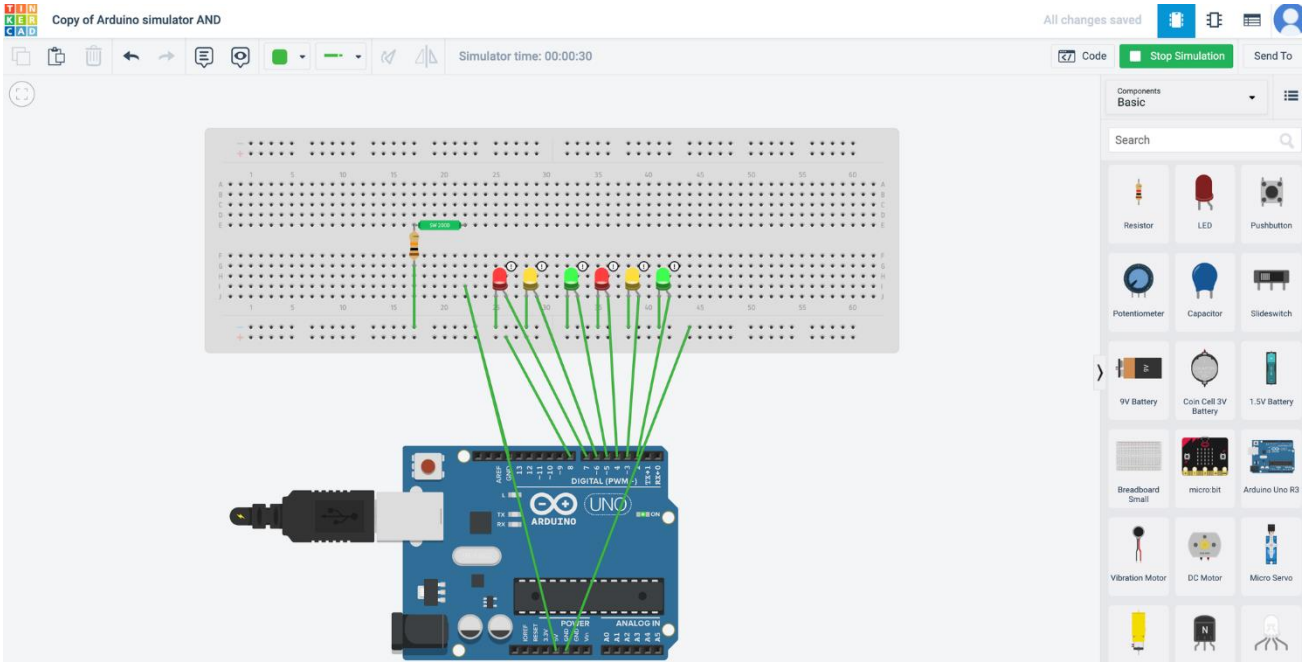

Fig 3: Simulated circuit diagram with lights on

Code:

millies_function | Arduino 1.8.19

File Edit Sketch Tools Help

millies_function

```
const int switchPin = 8;
unsigned long previousTime = 0;
int switchState = 0;
int prevSwitchState = 0;
int led = 2;
long interval = 2000;
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;
}
```

Done uploading.

```
Sketch uses 1768 bytes (0%) of program storage space. Maximum is 253952 bytes.
Global variables use 19 bytes (0%) of dynamic memory, leaving 8173 bytes for local variables. Maximum is 8192 bytes.
```

millies_function | Arduino 1.8.19

File Edit Sketch Tools Help

millies_function

```
int switchState = 0;
int prevSwitchState = 0;
int led = 2;
long interval = 2000;
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;
}
```

Done uploading.

Sketch uses 1768 bytes (0%) of program storage space. Maximum is 253952 bytes.
Global variables use 19 bytes (0%) of dynamic memory, leaving 8173 bytes for local variables. Maximum is 8192 bytes.

Discussion and Conclusion:

The implementation of a digital timer using Arduino's millis() function to avoid issues with the delay() function demonstrated that millis() can offer a more precise and reliable timer. During the delay () function can impede program execution and result in issues in some applications, the Millis () function allows for more flexible program flow and multitasking. Therefore, when implementing timing-dependent tasks in Arduino programs, it is advised to use the millis() function rather than delay().

The experiment's goal was achieved, and a digital timer was built. Our course teacher and lab instructor walked us through the procedure and demonstrated how to run this code correctly. We have learned about the millis() function and how to use the tilt sensor to build the digital timer. The circuit has been meticulously designed, and Arduino programming has been used to create it. LEDs were turned on every minute, and whenever the tilt function was touched or moved, lights started to turn on from the beginning. The tinkercad software was used to simulate and verify the experiment's outcome.

References:

1. AIUB LAB MANUAL
2. https://www.arduino.cc/.
3. ATMega328 manual
4. https://www.avrfreaks.net/forum/tut-c-newbies-guide-avr-timers
5. http://maxembedded.com/2011/06/avr-timers-timer0
6. Jeremy Blum; Exploring Arduino: Tools and Techniques for Engineering Wizardry.