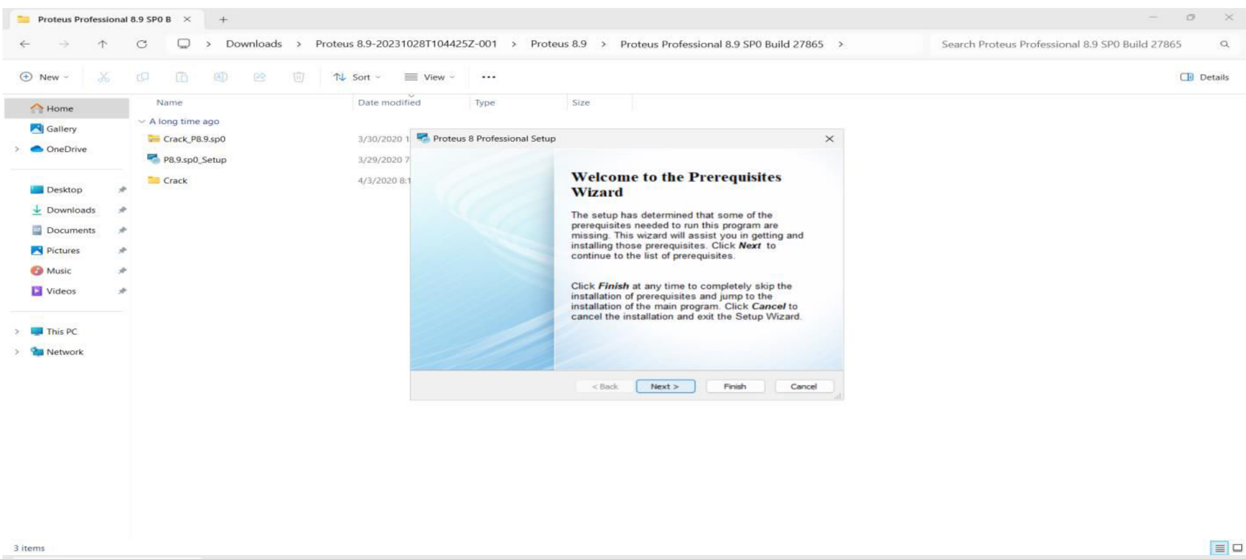
Index

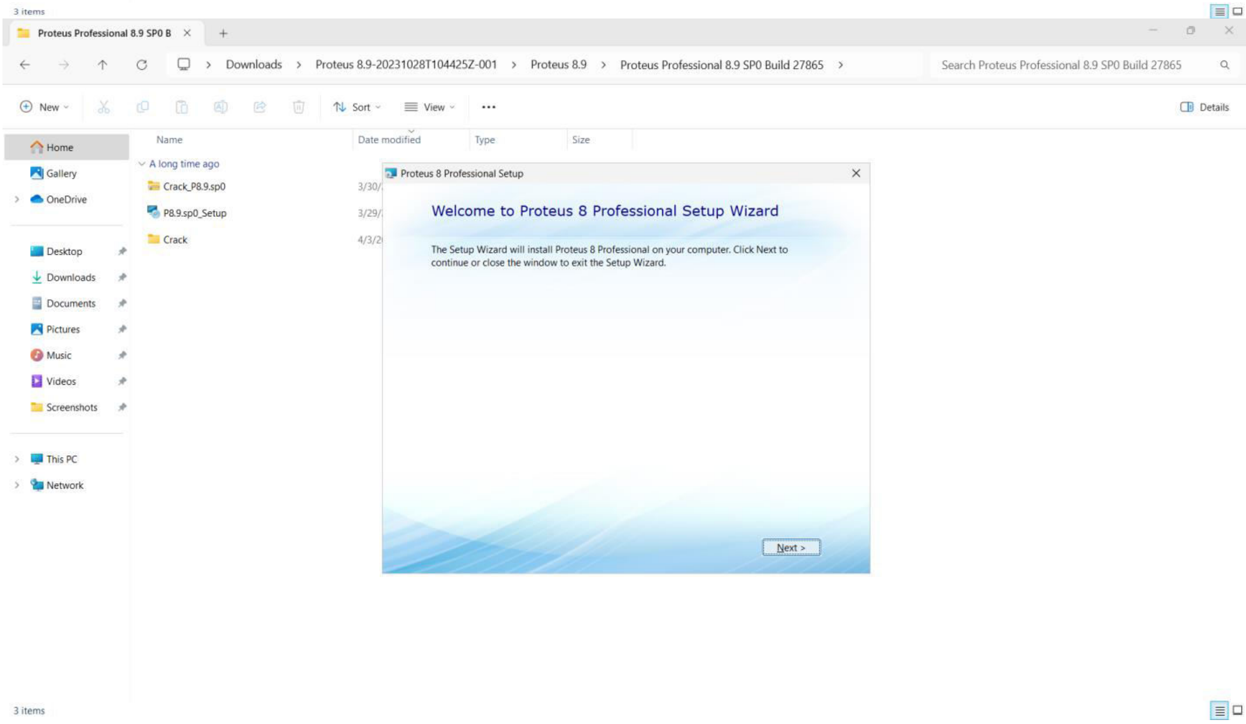
SI	Name of Experiment	Date of Issue	Date of Submission	Signature
1	Microcontroller and Arduino Environment setup.			
2	Experiment of LED blinking and fading with Atmega8 microcontroller			
3	Automating LED with LDR			
4	7 segment numerical value display using ATMEGA328P microcontroller.			
5	Experiment of motion sensor using Arduino UNO.			
6	Microcontroller Interrupt Implementation.			
7	Distance measurement using sonar sensor and ATMEGA328P microcontroller.			
8	Home Automation using Arduino UNO.			

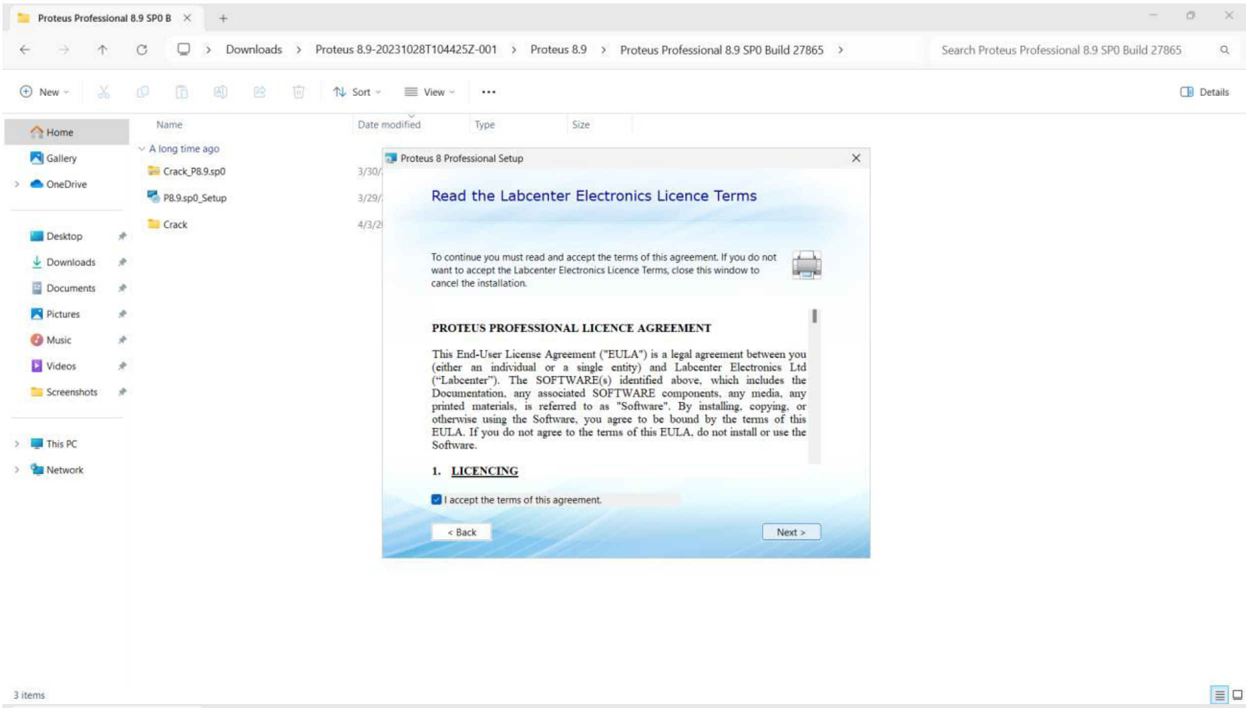
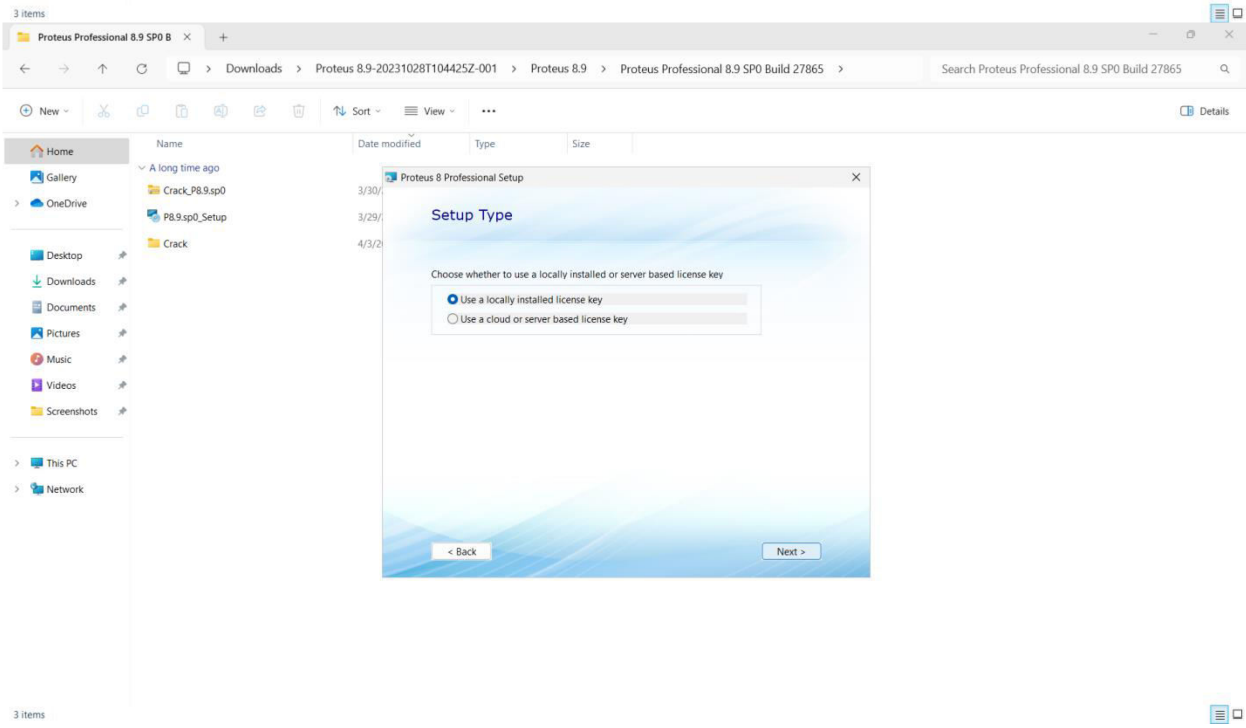
Experiment Number: 01
Experiment Name: Microcontroller and Arduino Environment setup.

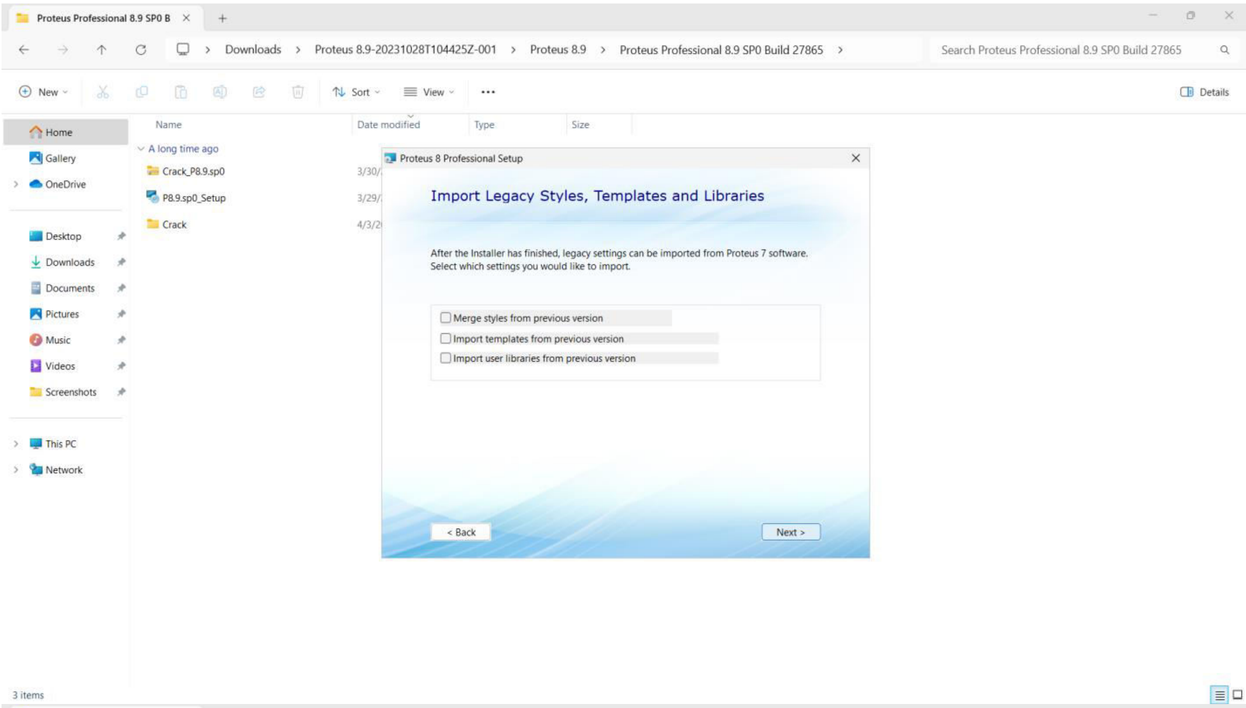
Theory:The installation of Proteus software is a critical precursor to engaging in electronics simulation and microcontroller-based projects within a laboratory environment. This theoretical framework outlines the key steps and considerations involved in the installation process.

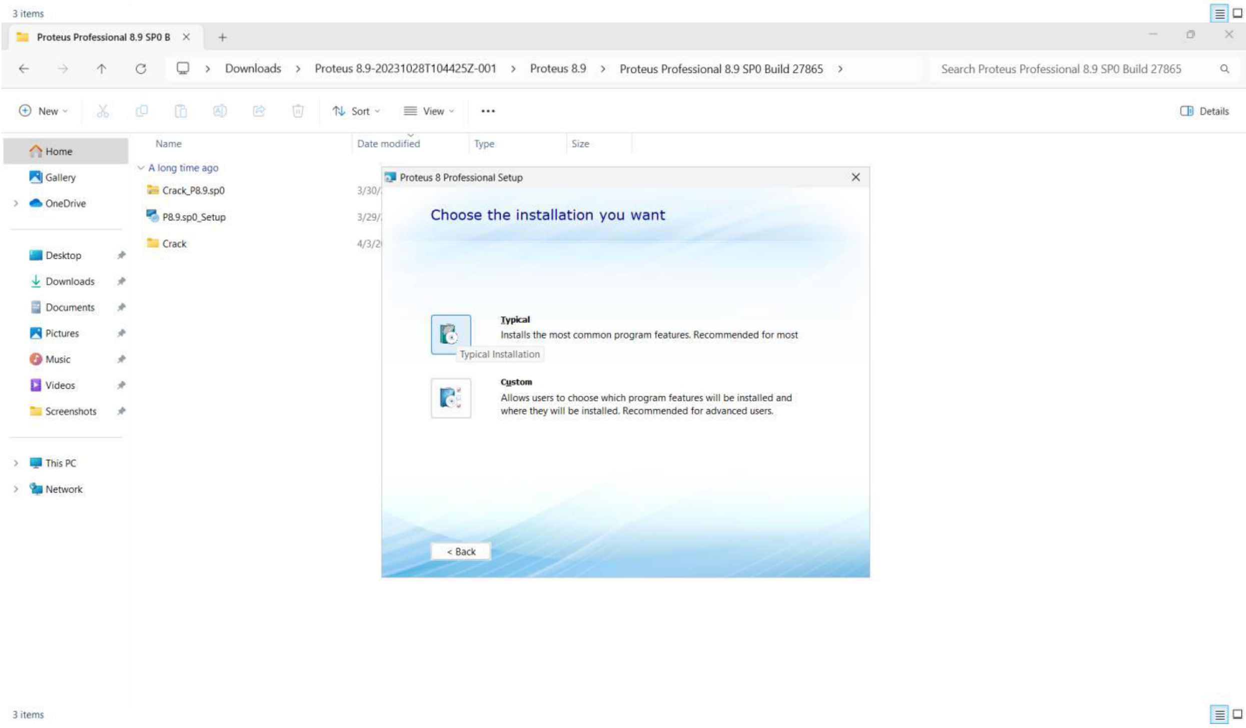
Result:

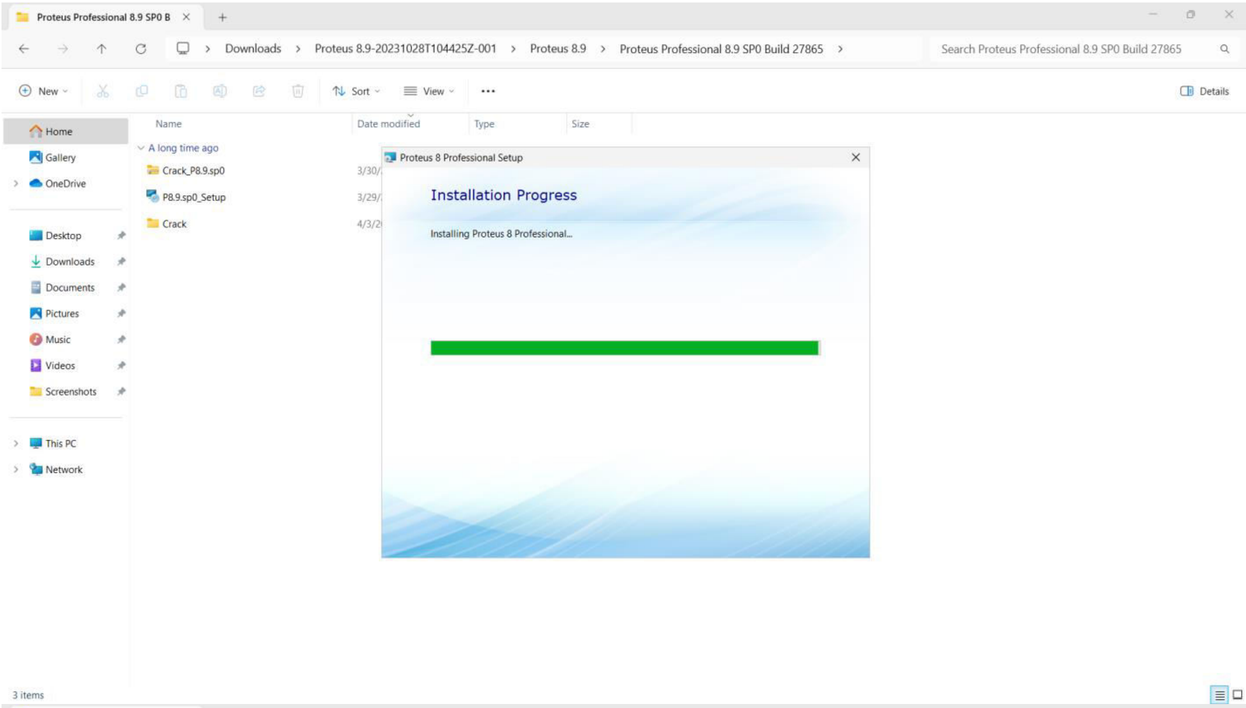
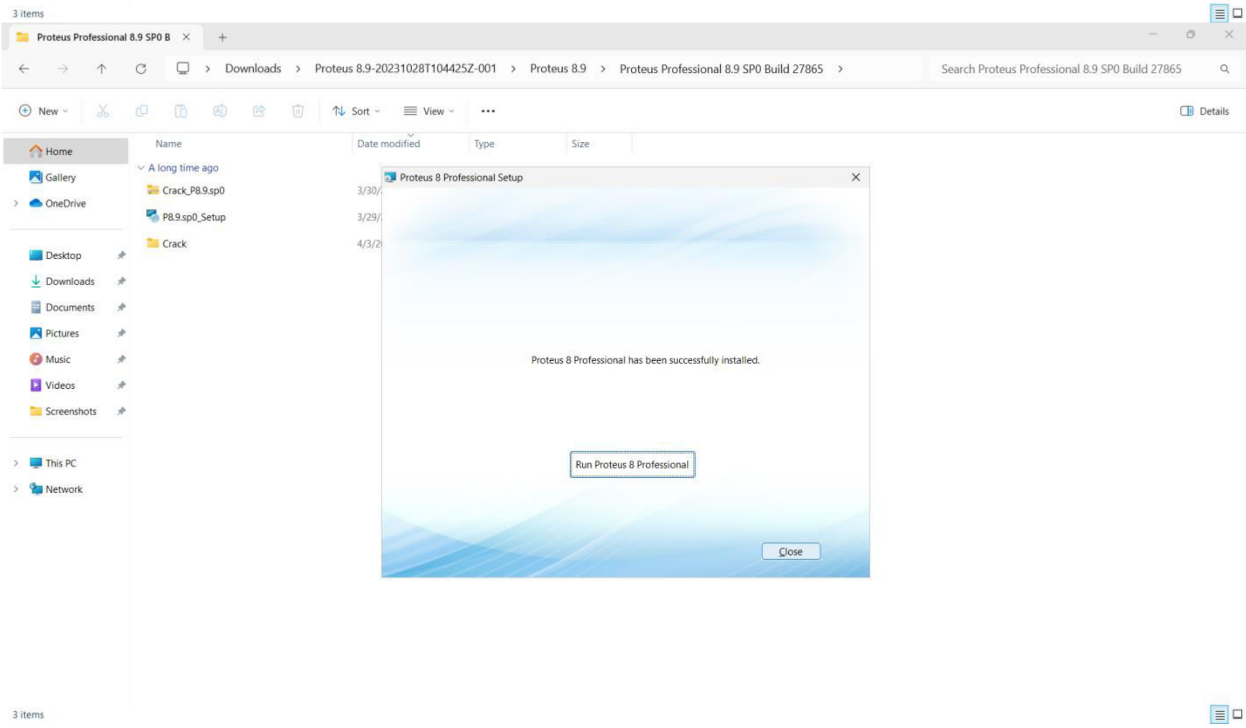


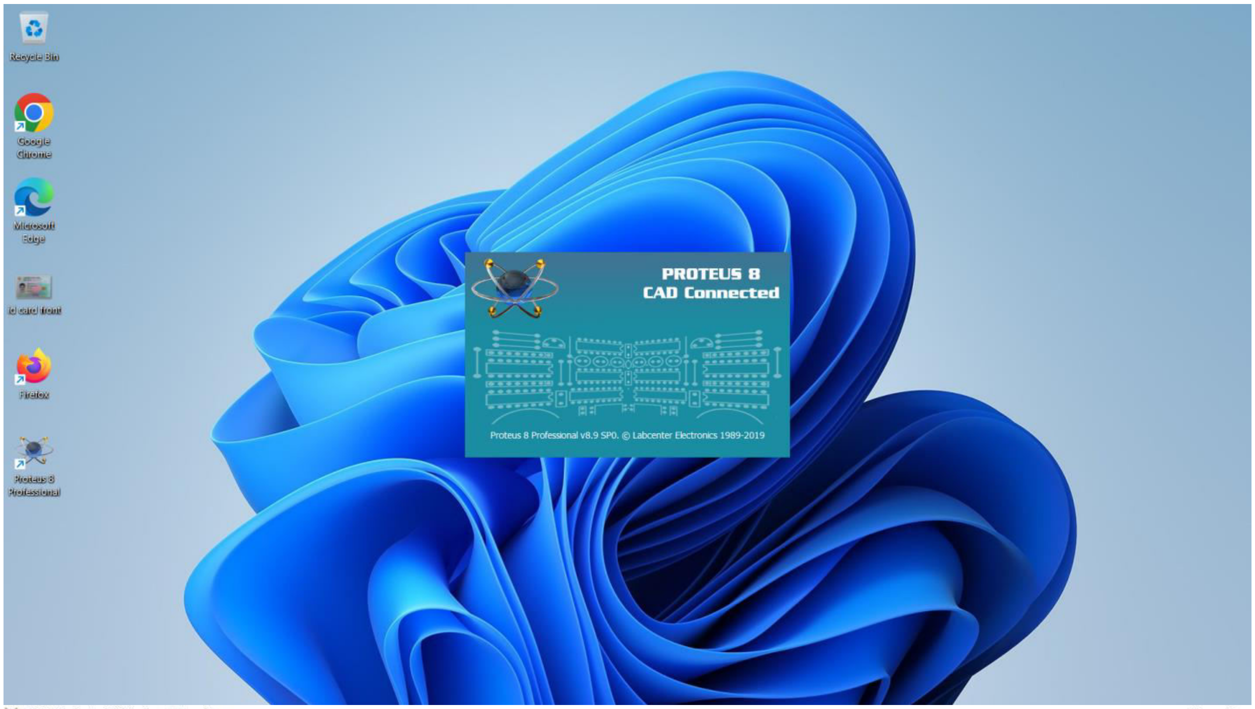



Proteus Professional 8.9 SP0 B
Downloads > Proteus 8.9-20231028T104425Z-001 > Proteus 8.9 > Proteus Professional 8.9 SP0 Build 27865
Search Proteus Professional 8.9 SP0 Build 27865
New
Sort
View
Details
Home
Gallery
OneDrive
Desktop
Downloads
Documents
Pictures
Music
Videos
Screenshots
This PC
Network
Name
Date modified
Type
Size
A long time ago
Crack_P8.9.sp0
P8.9.sp0_Setup
Crack
Proteus 8 Professional Setup
Read the Labcenter Electronics Licence Terms
To continue you must read and accept the terms of this agreement. If you do not want to accept the Labcenter Electronics Licence Terms, close this window to cancel the installation.
PROTEUS PROFESSIONAL LICENCE AGREEMENT
This End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and Labcenter Electronics Ltd ("Labcenter"). The SOFTWARE(s) identified above, which includes the Documentation, any associated SOFTWARE components, any media, any printed materials, is referred to as "Software". By installing, copying, or otherwise using the Software, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this EULA, do not install or use the Software.
1. LICENCING
I accept the terms of this agreement.
< Back
Next >
3 items

Proteus Professional 8.9 SP0 B
Downloads > Proteus 8.9-20231028T104425Z-001 > Proteus 8.9 > Proteus Professional 8.9 SP0 Build 27865
Search Proteus Professional 8.9 SP0 Build 27865
New
Sort
View
Details
Home
Gallery
OneDrive
Desktop
Downloads
Documents
Pictures
Music
Videos
Screenshots
This PC
Network
Name
Date modified
Type
Size
A long time ago
Crack_P8.9.sp0
P8.9.sp0_Setup
Crack
Proteus 8 Professional Setup
Setup Type
Choose whether to use a locally installed or server based license key
Use a locally installed license key
Use a cloud or server based license key
< Back
Next >
3 items

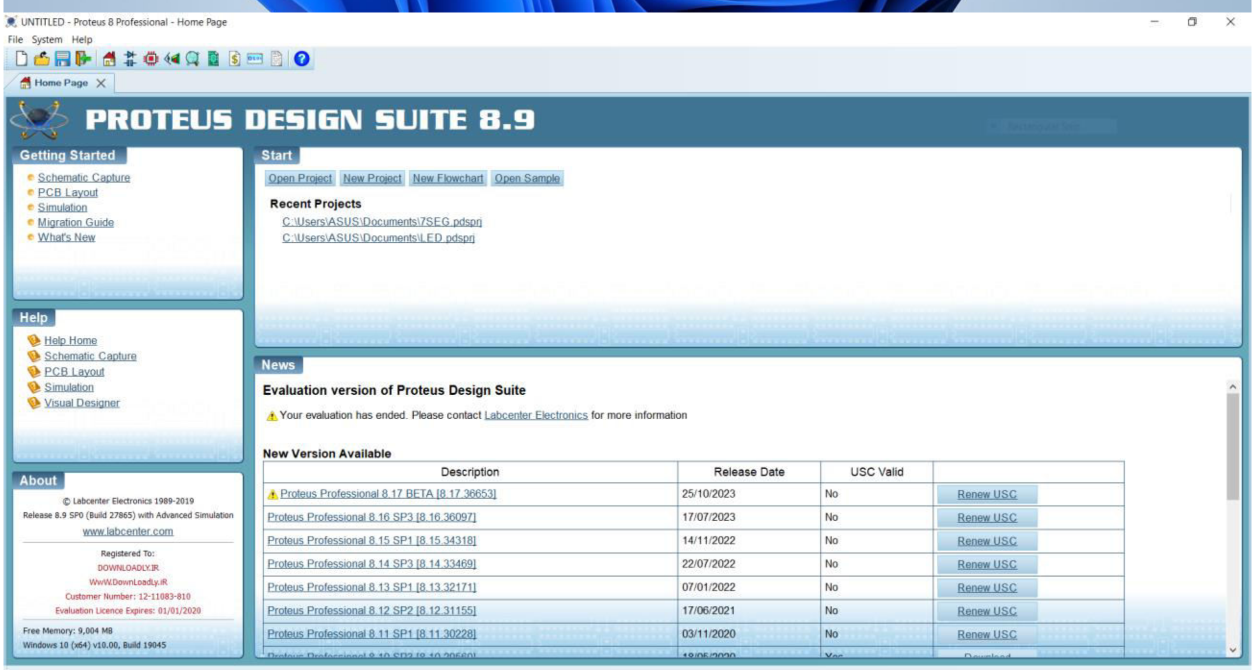

Proteus Professional 8.9 SP0 B
Downloads
Proteus 8.9-20231028T104425Z-001
Proteus 8.9
Proteus Professional 8.9 SP0 Build 27865
Search Proteus Professional 8.9 SP0 Build 27865
New
Sort
View
Details
Home
Gallery
OneDrive
Desktop
Downloads
Documents
Pictures
Music
Videos
Screenshots
This PC
Network
Name
Date modified
Type
Size
A long time ago
Crack_P8.9.sp0
P8.9.sp0_Setup
Crack
Proteus 8 Professional Setup
Import Legacy Styles, Templates and Libraries
After the Installer has finished, legacy settings can be imported from Proteus 7 software.
Select which settings you would like to import.
Merge styles from previous version
Import templates from previous version
Import user libraries from previous version
< Back
Next >
3 items


Proteus Professional 8.9 SP0 B
Downloads
Proteus 8.9-20231028T104425Z-001
Proteus 8.9
Proteus Professional 8.9 SP0 Build 27865
Search Proteus Professional 8.9 SP0 Build 27865
New
Sort
View
Details
Home
Gallery
OneDrive
Desktop
Downloads
Documents
Pictures
Music
Videos
Screenshots
This PC
Network
Name
Date modified
Type
Size
A long time ago
Crack_P8.9.sp0
P8.9.sp0_Setup
Crack
Proteus 8 Professional Setup
Choose the installation you want
Typical
Installs the most common program features. Recommended for most
Typical Installation
Custom
Allows users to choose which program features will be installed and
where they will be installed. Recommended for advanced users.
< Back
3 items

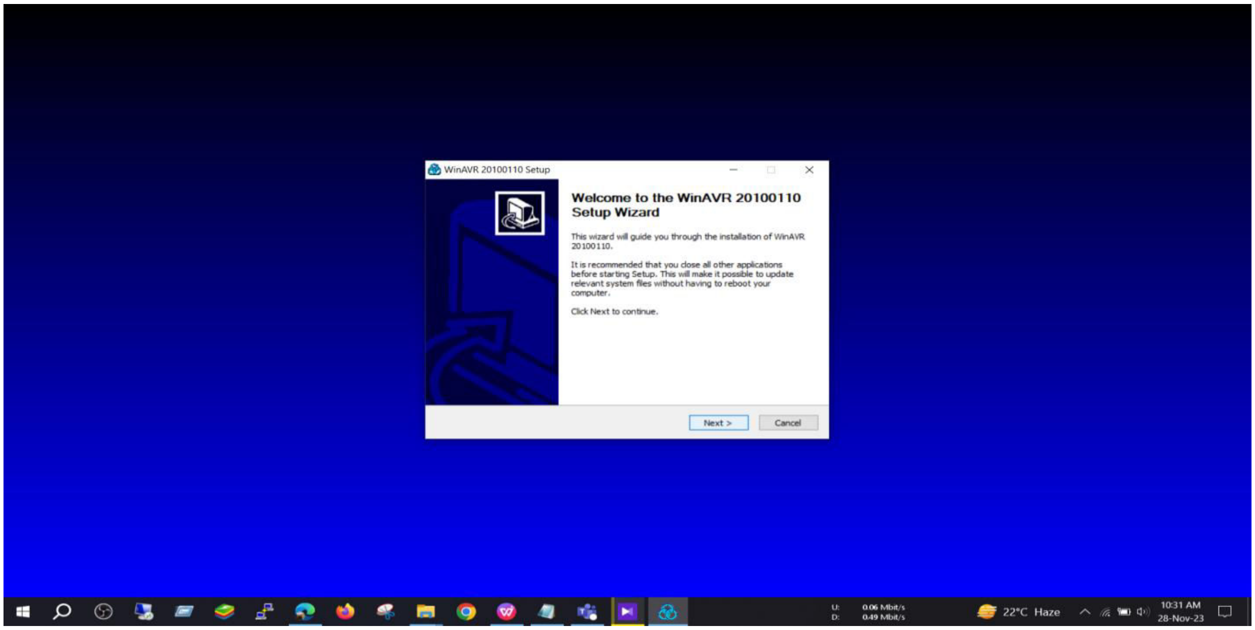

Proteus Professional 8.9 SP0 B
Downloads
Proteus 8.9-20231028T104425Z-001
Proteus 8.9
Proteus Professional 8.9 SP0 Build 27865
Search Proteus Professional 8.9 SP0 Build 27865
New
Sort
View
Details
Home
Gallery
OneDrive
Desktop
Downloads
Documents
Pictures
Music
Videos
Screenshots
This PC
Network
Name
Date modified
Type
Size
A long time ago
Crack_P8.9.sp0
P8.9.sp0_Setup
Crack
Proteus 8 Professional Setup
Installation Progress
Installing Proteus 8 Professional...
3 items

Proteus Professional 8.9 SP0 B
Downloads
Proteus 8.9-20231028T104425Z-001
Proteus 8.9
Proteus Professional 8.9 SP0 Build 27865
Search Proteus Professional 8.9 SP0 Build 27865
New
Sort
View
Details
Home
Gallery
OneDrive
Desktop
Downloads
Documents
Pictures
Music
Videos
Screenshots
This PC
Network
Name
Date modified
Type
Size
A long time ago
Crack_P8.9.sp0
P8.9.sp0_Setup
Crack
Proteus 8 Professional Setup
Proteus 8 Professional has been successfully installed.
Run Proteus 8 Professional
Close
3 items

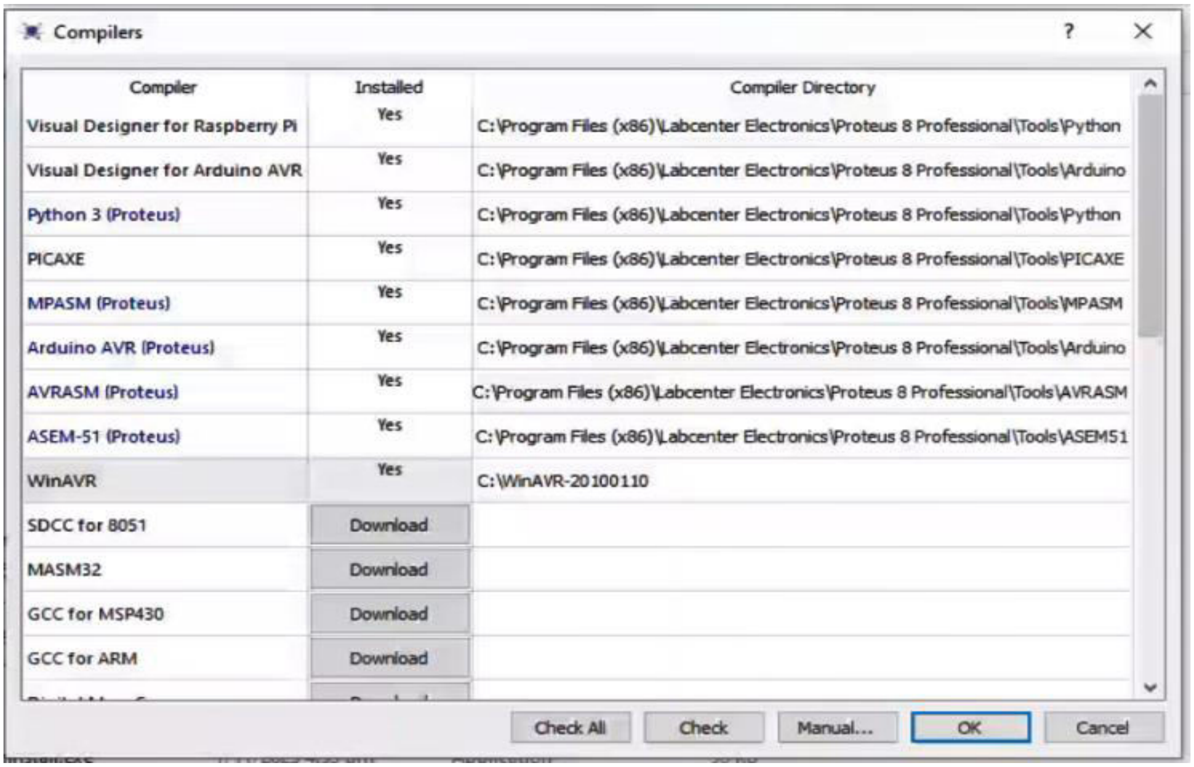


Recycle Bin
Google Chrome
Microsoft Edge
id card front
Firefox
Proteus 8 Professional
PROTEUS 8
CAD Connected
Proteus 8 Professional v8.9 SP0. © Labcenter Electronics 1989-2019



UNTITLED - Proteus 8 Professional - Home Page
File System Help
Home Page
PROTEUS DESIGN SUITE 8.9
Getting Started
Schematic Capture
PCB Layout
Simulation
Migration Guide
What's New
Start
Open Project
New Project
New Flowchart
Open Sample
Recent Projects
C:\Users\ASUS\Documents\7SEG.pdsprj
C:\Users\ASUS\Documents\LED.pdsprj
Help
Help Home
Schematic Capture
PCB Layout
Simulation
Visual Designer
News
Evaluation version of Proteus Design Suite
Your evaluation has ended. Please contact Labcenter Electronics for more information
New Version Available
Description
Release Date
USC Valid
Proteus Professional 8.17 BETA [8.17.36653] 25/10/2023 No Renew USC
Proteus Professional 8.16 SP3 [8.16.36097] 17/07/2023 No Renew USC
Proteus Professional 8.15 SP1 [8.15.34318] 14/11/2022 No Renew USC
Proteus Professional 8.14 SP3 [8.14.33469] 22/07/2022 No Renew USC
Proteus Professional 8.13 SP1 [8.13.32171] 07/01/2022 No Renew USC
Proteus Professional 8.12 SP2 [8.12.31155] 17/06/2021 No Renew USC
Proteus Professional 8.11 SP1 [8.11.30228] 03/11/2020 No Renew USC
About
© Labcenter Electronics 1989-2019
Release 8.9 SP0 (Build 27865) with Advanced Simulation
www.labcenter.com
Registered To:
DOWNLOADLY.IR
WwW.DownLoadLy.iR
Customer Number: 12-11083-810
Evaluation Licence Expires: 01/01/2020
Free Memory: 9,004 MB
Windows 10 (x64) v10.00, Build 19045



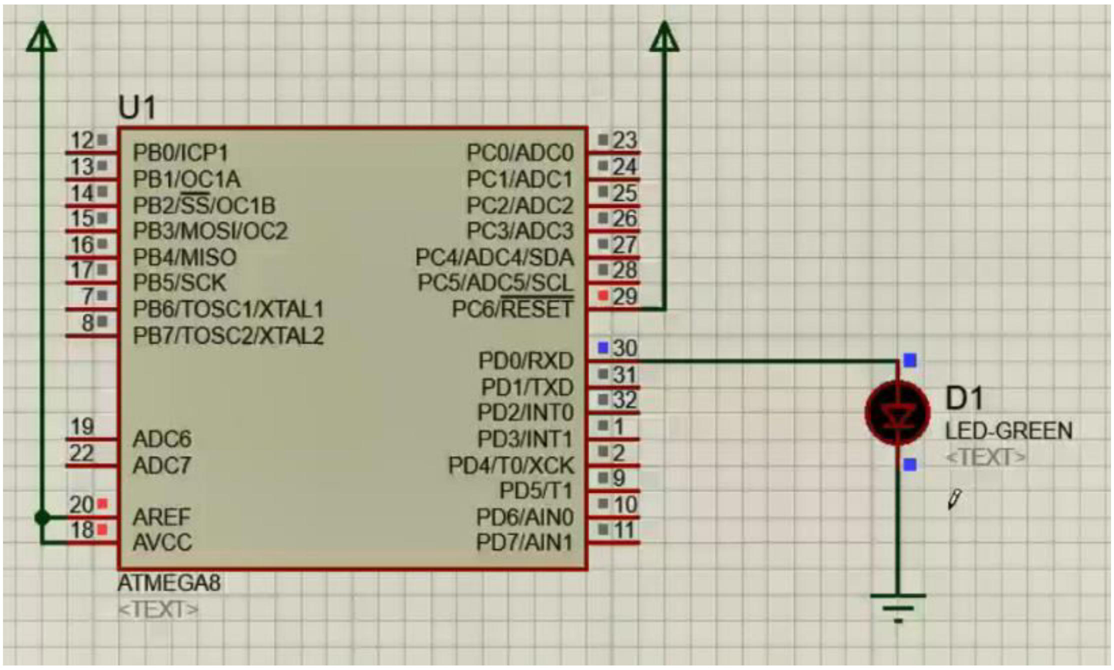


Conclusion: The theoretical installation framework outlined here serves as a foundational guide for setting up Proteus in a laboratory setting. A meticulous approach to system requirements, official documentation, and step-by-step installation procedures is paramount to ensuring a seamless and productive experience with Proteus in the realm of electronics simulation.

Experiment Number: 02
Experiment Name: Experiment of LED blinking and fading with Atmega8 microcontroller.

Theory:This experiment utilizes the Atmega8 microcontroller to control LED blinking and fading. LED blinking involves toggling digital output pins at specific intervals, while LED fading employs PWM signals for gradual brightness changes. The Atmega8's capabilities enable precise control over these lighting effects.

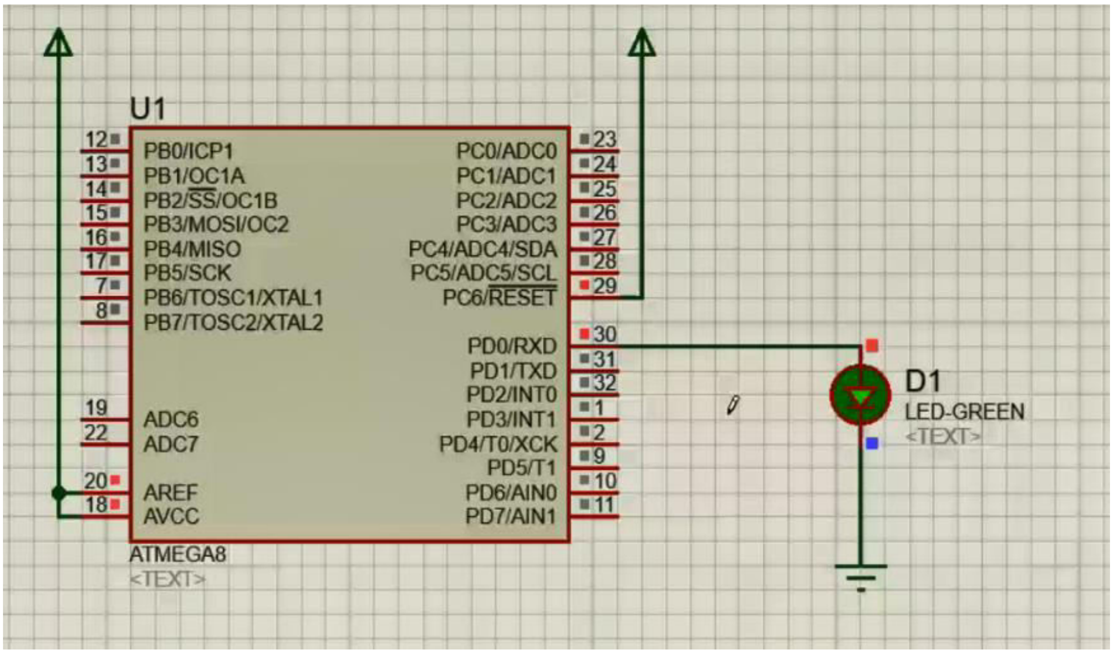
Circuit:



Code:

```c
#include <avr/io.h>
#include <util/delay.h>
int main()
{
DDRD = 0b00000001;
while(1)
{
PORTD = 0b0000001;
_delay_ms(1000);
PORTD = 0b0000000;
_delay_ms(1000);
}
}
```

Result:



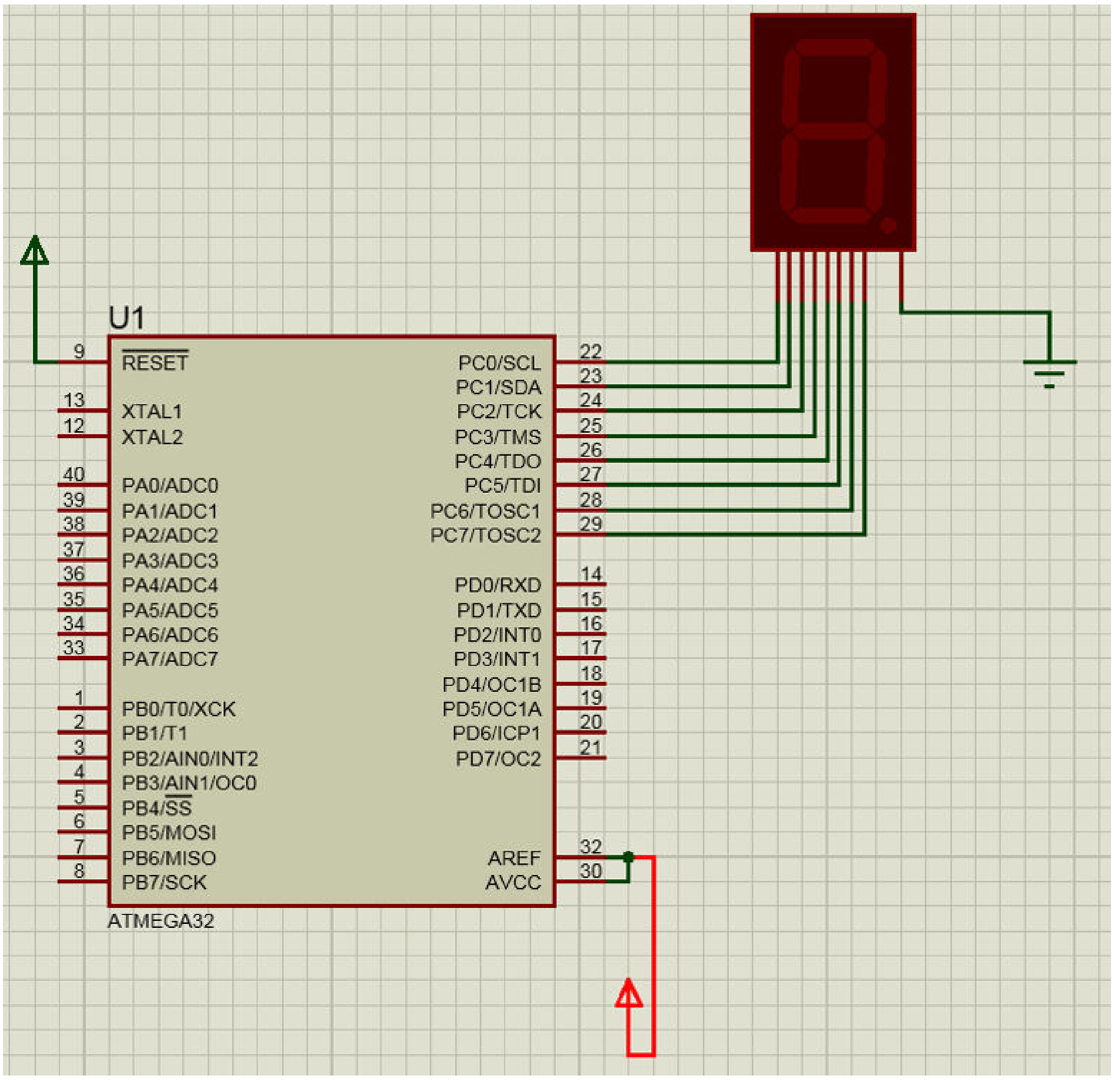
Conclution:

The Atmega8 microcontroller effectively manages LED blinking and fading, showcasing its versatility in lighting applications. The experiment illustrates the practical use of microcontrollers for dynamic visual effects.

Experiment Number: 04
Experiment Name: 7 segment numerical value display using ATMEGA328P microcontroller.

Theory:
The 7-segment display is a common numeric representation method using seven LEDs. The ATMEGA328P microcontroller is employed to control this display. Through programming, the microcontroller activates specific LEDs to display desired numerical values on the 7-segment setup. This lab explores the interaction between the microcontroller and the display to convey numeric information.

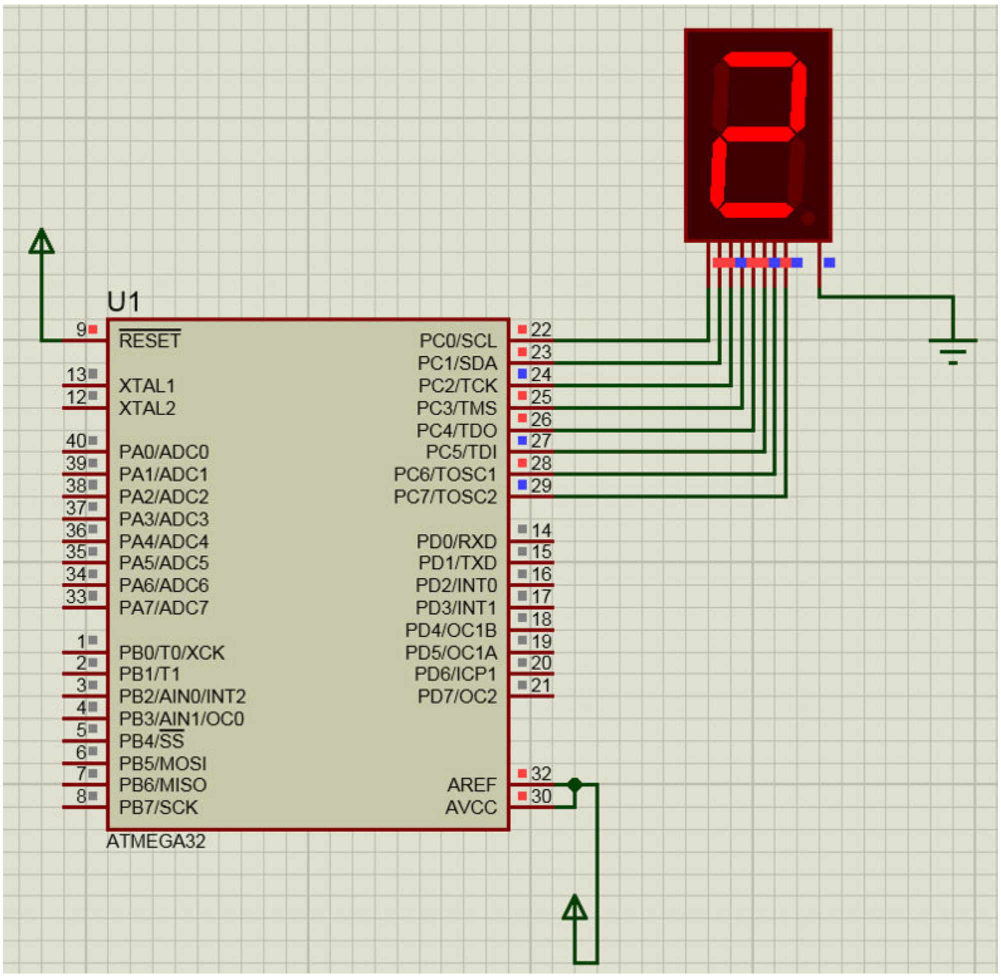
Circuit:



Code:

```
#include <inttypes.h>
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/sleep.h>

int main(void)
{

    DDRC=0xff;

    while (1)
    {
      PORTC=0x5b;

    }
    return 0;
}
```

Result:

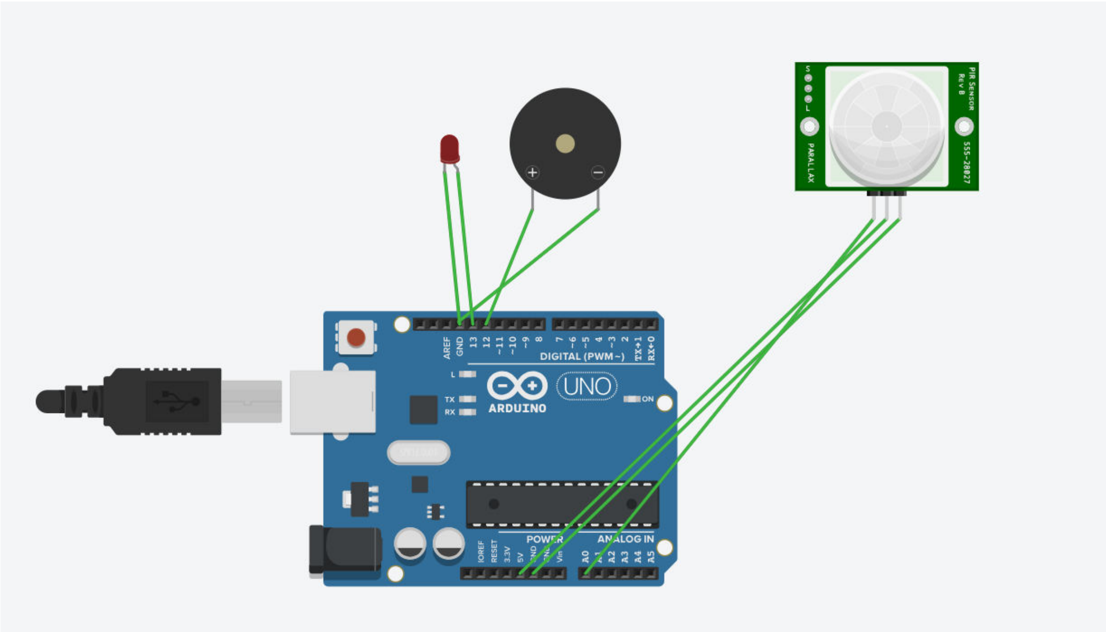


Conclution:
The successful implementation of the 7-segment numerical display using the ATMEGA328P microcontroller highlights the practical application of digital display interfacing. This project enhances understanding of microcontroller programming, GPIO pin control, and dynamic numeric representation.

Experiment Number: 05
Experiment Name: Experiment of motion sensor using Arduino UNO.

Theory: The motion sensor experiment using Arduino UNO involves integrating a Passive Infrared (PIR) sensor with an Arduino board to detect motion in its vicinity. The PIR sensor is designed to detect changes in infrared radiation, which typically occur when there is movement within its detection range. The Arduino UNO serves as the control unit, processing the signals from the PIR sensor and triggering actions, such as lighting up an LED and activating a piezo buzzer.The PIR sensor consists of a pyroelectric sensor that generates a voltage when exposed to infrared radiation. The Arduino UNO reads this voltage and interprets it as a motion event. When motion is detected, the Arduino sends a signal to illuminate the LED and produce a sound through the piezo buzzer, providing a visual and audible indication of the detected motion.

Circuit:



Code:

```
int pirSeg=A0;
int Mot=0; //motion
int buz = 12;
int LED =13;
void setup()
{
  Serial.begin(9600);
  pinMode(LED, OUTPUT);
  pinMode(buz , OUTPUT);
  pinMode(pirSeg,INPUT);
}
void loop()
{
  pirSeg = analogRead(A0);
  Mot = map(pirSeg,0,1023,0,255);
  Serial.println(Mot);
  if(Mot > 150){
    Serial.println("Motion detected");
    delay(5000);
    digitalWrite(buz,HIGH);
    delay(5000);
    digitalWrite(LED,HIGH);
    delay(5000);
  }
  else{
    Serial.println("No motion dected");
    delay(5000);
    digitalWrite(buz,HIGH);
    delay(5000);
    digitalWrite(LED,HIGH);
    delay(5000);
  }
  digitalWrite(LED_BUILTIN, HIGH);
  delay(5000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(5000);
}
```

Result:

Serial Monitor

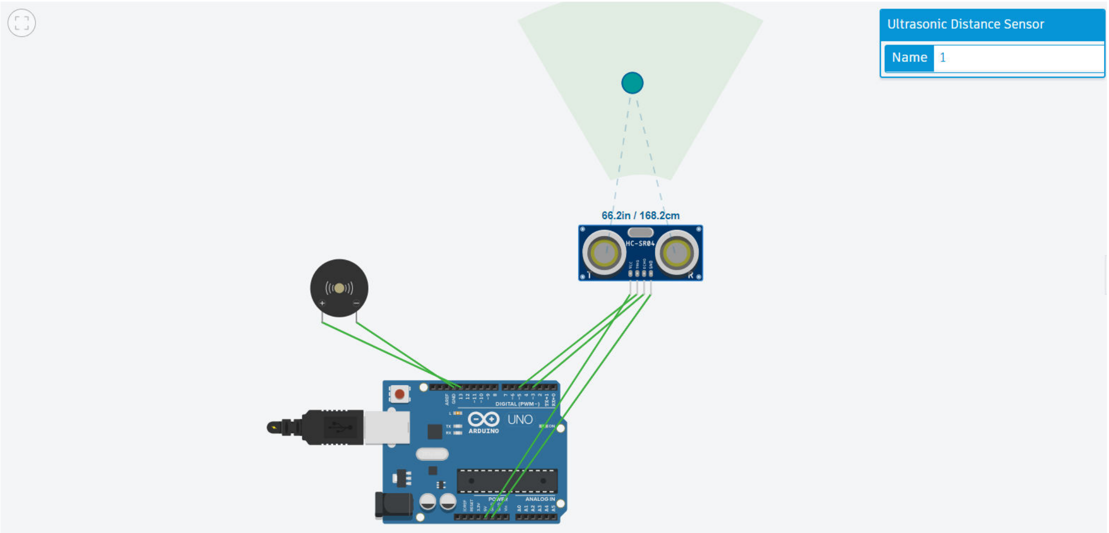
```
0
No motion dected
253
Motion detected
0
No motion dected
```

Conclution:
In conclusion, the motion sensor experiment using Arduino UNO successfully demonstrated the integration of a PIR sensor for motion detection. The Arduino's ability to interface with external sensors was highlighted, showcasing its versatility in creating responsive electronic systems. The combination of the PIR sensor, LED, and piezo buzzer provided a straightforward yet effective motion detection setup, offering practical insights for future projects in security systems or interactive applications.

Experiment Number: 07
Experiment Name: Distance measurement using sonar sensor and ATMEGA328P microcontroller.

Theory: The project involves using an Arduino microcontroller, a piezo buzzer, and an ultrasonic distance sensor. The ultrasonic sensor works on the principle of sending ultrasonic waves and measuring the time it takes for the waves to bounce back after hitting an object. This time is then used to calculate the distance between the sensor and the object. The Arduino processes this information and triggers the piezo buzzer to produce sound based on the distance.

Circuit:



Code:

```
int tr = 6;
int echo = 3;
int buz = 13;
int time;
int distance;

void setup() {
  Serial.begin(9600);
  pinMode(tr, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(buz, OUTPUT);
}

void loop() {
  digitalWrite(tr, HIGH);
  delayMicroseconds(10);
  digitalWrite(tr, LOW);
  time = pulseIn(echo, HIGH);
  distance = (time * 0.0343) / 2;  // Corrected the calculation

  if (distance <= 40) {
    Serial.println(time);
    Serial.println(distance);
    digitalWrite(buz, HIGH);
    delay(500);
  } else {
    Serial.println(distance);
    digitalWrite(buz, LOW);
    delay(500);
  }
}
```

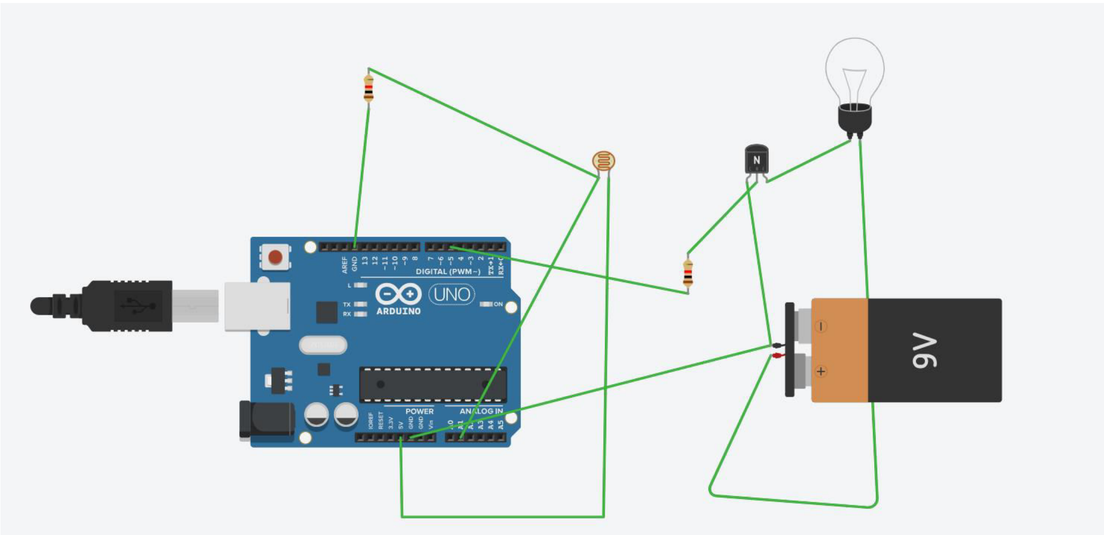
Result:

Conclution: The project successfully demonstrates the integration of an ultrasonic distance sensor with an Arduino microcontroller to create a distance-sensitive alarm using a piezo buzzer. By measuring the distance to an object, the system provides a practical application for alerting users to proximity events.

Experiment Number: 8
Experiment Name: Home Automation using Arduino UNO.

Theory: The Home Automation System using Arduino UNO is designed to enhance the efficiency and convenience of home management through the integration of microcontroller technology. This project leverages the capabilities of Arduino UNO to control and automate various household devices, creating a smart and interconnected living space. The system utilizes sensors, actuators, and communication modules to enable remote monitoring and control of home appliances.

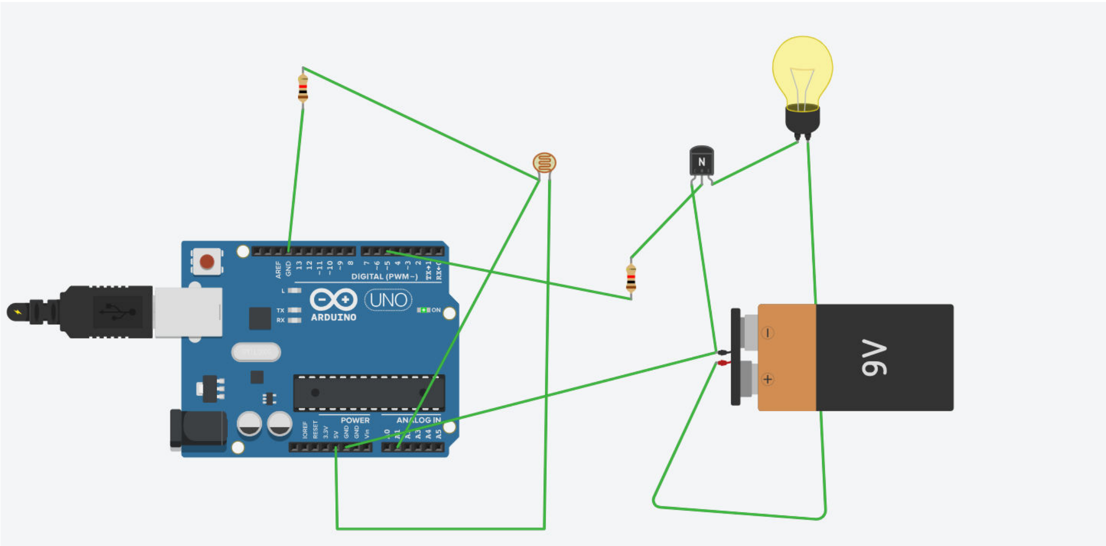
Circuit:



Code:

```
int LDR = A1;
int value_of_LDR;
int bulb=5;

void setup()
{
  pinMode(bulb, OUTPUT);
  pinMode(LDR,INPUT);

}

void loop()
{
  value_of_LDR=analogRead(LDR);
  if(value_of_LDR > 512)
    digitalWrite(bulb,LOW);
  else{
      digitalWrite(bulb,HIGH);
  }

}
```

Result:



Conclusion: My project highlights the feasibility of utilizing Arduino in combination with simple electronic components for an automated lighting system. The photoresistor acted as a light sensor, influencing the NPN transistor's conductivity and subsequently the bulb's brightness. This low-cost and energy-efficient solution offers a foundation for further exploration in smart lighting applications.