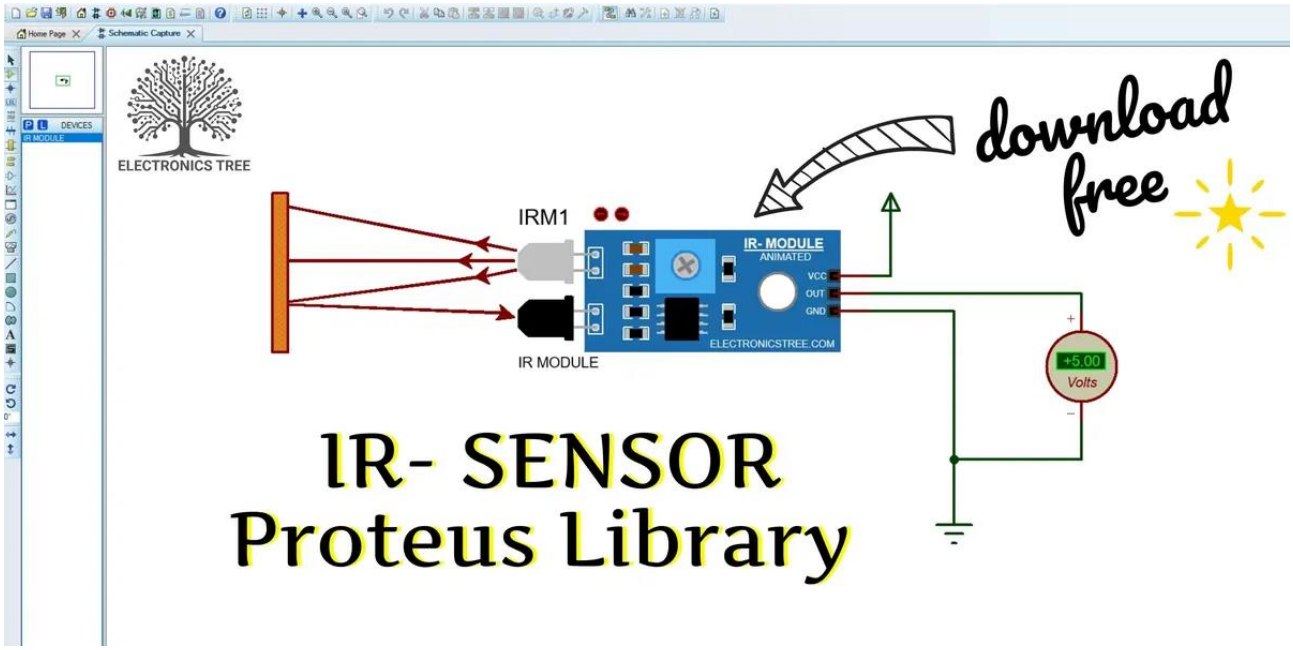
Lab-07 RF Sensors

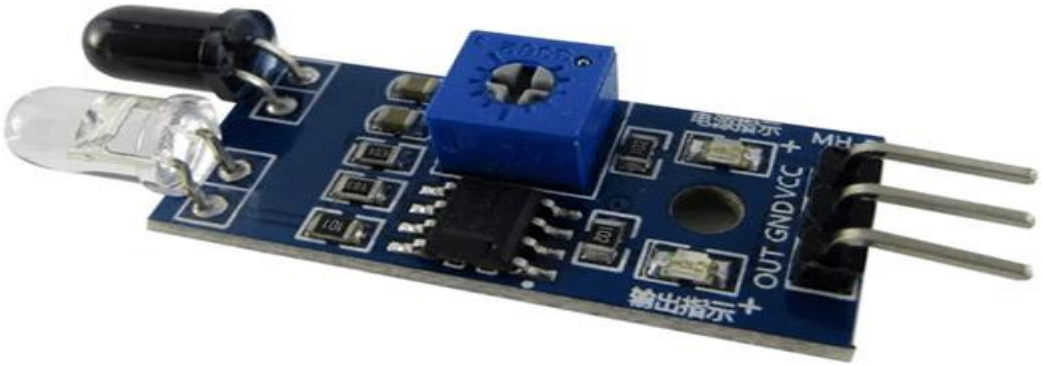
1. **Objectives: After this lab, the learner will be able to:**
   - Interface SD module with Arduino board
   - Interface
2. **Facilities**
   - Proteus software
   - Arduino IDE
3. **Prerequest**
   - Basic electronic
   - C programming
4. **Introduction**

In today's tutorial, we unveil an exciting addition to the world of Proteus simulations—the Infrared Sensor Library for Proteus. This new IR sensor is a breakthrough, especially for those engaged in Embedded Systems Projects, particularly in robotics and automation. Join me as we delve into the intricacies of this library, designed exclusively for Proteus.



4.1. IR sensors

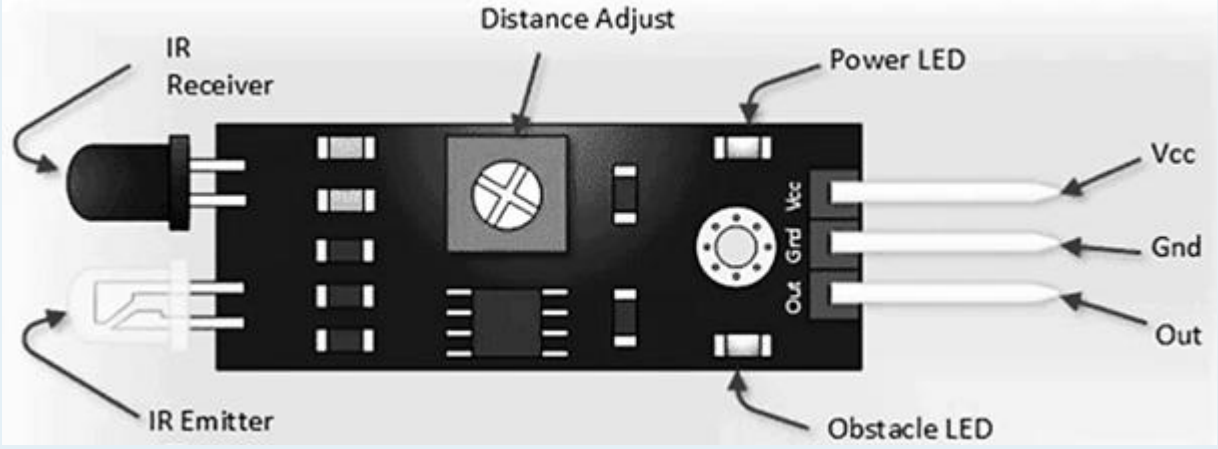
In the realm of electronic devices IR Sensors, **IR (Infrared) Proximity Sensors** are indispensable components that operate within a specific frequency range, c**onverting IR signals into electric signals on their digital output** pins (usually referred to as OUT pins). These versatile sensors find applications in touchscreen phones and various other devices. During phone calls, for instance, these sensors deactivate the display, ensuring that even when the user's cheek comes into contact with the touchscreen, it remains unaffected



This Easy Electronics' multipurpose infrared sensor(IR Proximity Sensor) can also be used as **Barrier sensor, Line Sensing Robotics and Encoding sensor**. When an object is placed in front of the sensor with a logical zero (0V) output, it provides a digital output of **1 or 0**. Whether the device is getting enough power with inbuilt LED indicators and start experimenting with your logic.

Other important IR Sensors references.

- Introduction to the Arduino – What is Arduino?
- Arduino software Installation
- Temperature Sensor and Types
- How to Interface LM35 Sensor with Arduino UNO
- Digital Temperature Sensor for Arduino
- How to Interface DHT11 Sensor with Arduino Board
- IR (Infrared) Obstacle Detection Temperature Sensor



Types of IR Sensors sensor

4.2. Understanding Infrared Sensors

Types and Applications

Before we proceed, let's briefly understand the types of Infrared Sensors available in the market. Some modules feature separate transmitter and receiver chips, triggered by light interruptions (See the below table for this). This library, however, integrates both components on a single chip. This innovation allows the IR signal to transmit, bounce back upon encountering an obstacle, and be received by the IR sensor.

TYPE	APPLICATIONS
Separate Transmitter and Receiver Modules	Commonly used for light interruption detection in various applications.
Integrated Transmitter and Receiver on a Single Chip	Designed for efficient obstacle detection, ideal for robotics and automation projects.
Long-Range Infrared Sensors	Suitable for applications requiring detection over extended distances, such as security systems.
Short-Range Infrared Proximity Sensors	Perfect for close-range object detection, often used in touchless interfaces.
Multi-Channel Infrared Sensors	Offering the capability to detect multiple sources simultaneously, suitable for advanced applications.

Benefits of Infrared Sensor

BENEFITS	DETAILS
Efficient Obstacle Detection	The integrated design allows for effective detection of obstacles in front of the sensor.
Compact Single-Chip Design	The sensor's single-chip configuration enhances simplicity and ease of integration into projects.
Versatility	Adaptable to various applications, from robotics to automation, due to its versatile functionality.
Simulation Capability	Designed for seamless integration with Proteus software, facilitating realistic simulations.
Easy Integration with Microcontrollers	Future tutorials will guide users on interfacing the sensor with popular Microcontrollers like Arduino and PIC.

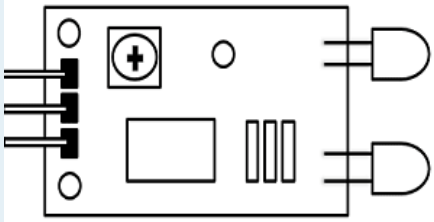
Infrared Sensor Pin Configuration

PIN	DESCRIPTION
Vcc	Provide +5V to this pin.
GND	Ground this pin.
OUT	Output pin; goes HIGH when an obstacle is detected and remains LOW in normal conditions.

PIN	DESCRIPTION
TestPin	Used in Proteus simulation; LOW indicates normal conditions, HIGH simulates the presence of an obstacle.

Pin Configuration

The following figure is a Schematic view of the IR Sensors-sensor module.



Pin Name	Description
VCC	Power Supply Input
GND	Power Supply Ground
OUT	Active High Output

1. OUT (output pin): Digital Output (High or Low)

2. VCC(+5v): Connected to circuit supply

3. GND(-ve): Ground Connected to circuit ground

Features

The key features of an IR Proximity Sensor include:

- IR transmitter
- Ambient light-protected IR receiver
- 3-pin easy interface connectors
- Indicator LED & Power LED
- Detection range from 2cm to 30cm
- Ability to differentiate between dark and light colors
- Active low on object detection
- Operates on a voltage range of 3.3V to 5V

Proximity Sensor Applications

These sensors find extensive applications across various domains:

- **Object Detection**: Proximity sensors excel at identifying the presence of objects without any physical contact.
- **Environmental Monitoring**: They can measure and detect changes in the surrounding environment.
- **Home Automation**: Proximity sensors are used in home automation systems to automatically control lighting based on occupancy in a room.
- **Conveyor Belts**: They measure objects on conveyor belts, ensuring smooth operations.
- **Motor Control**: Proximity sensors are employed to measure speed and guide the rotation of electrical motors.
- **Vehicle Monitoring**: In the automotive industry, these sensors are used to measure the speed and distance of vehicles.
- **Industrial Automation**: Proximity sensors are integral in rolling mills, machine tools, and other automation systems.
- **Mobile Phones**: Mobile phones utilize proximity sensors for automatic screen dimming during calls, enhanced gaming experiences, and touchscreen functionality.

4.3. Circuit Integration and Simulation

Now, let's dive into the practical aspect. Place the IR Obstacle Sensor in your Proteus workspace. If all goes well, you'll witness a simulated output that mirrors real-world scenarios.

Understanding Pins

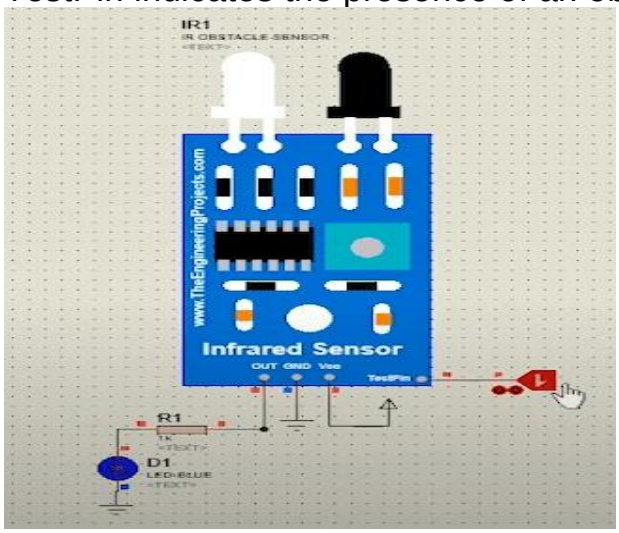
As shown in the simulation, the Infrared sensor has four pins: Vcc, GND, OUT, and TestPin. Learn how to provide the necessary power, ground the sensor, and interpret output from the TestPin.

Properties Panel Configuration

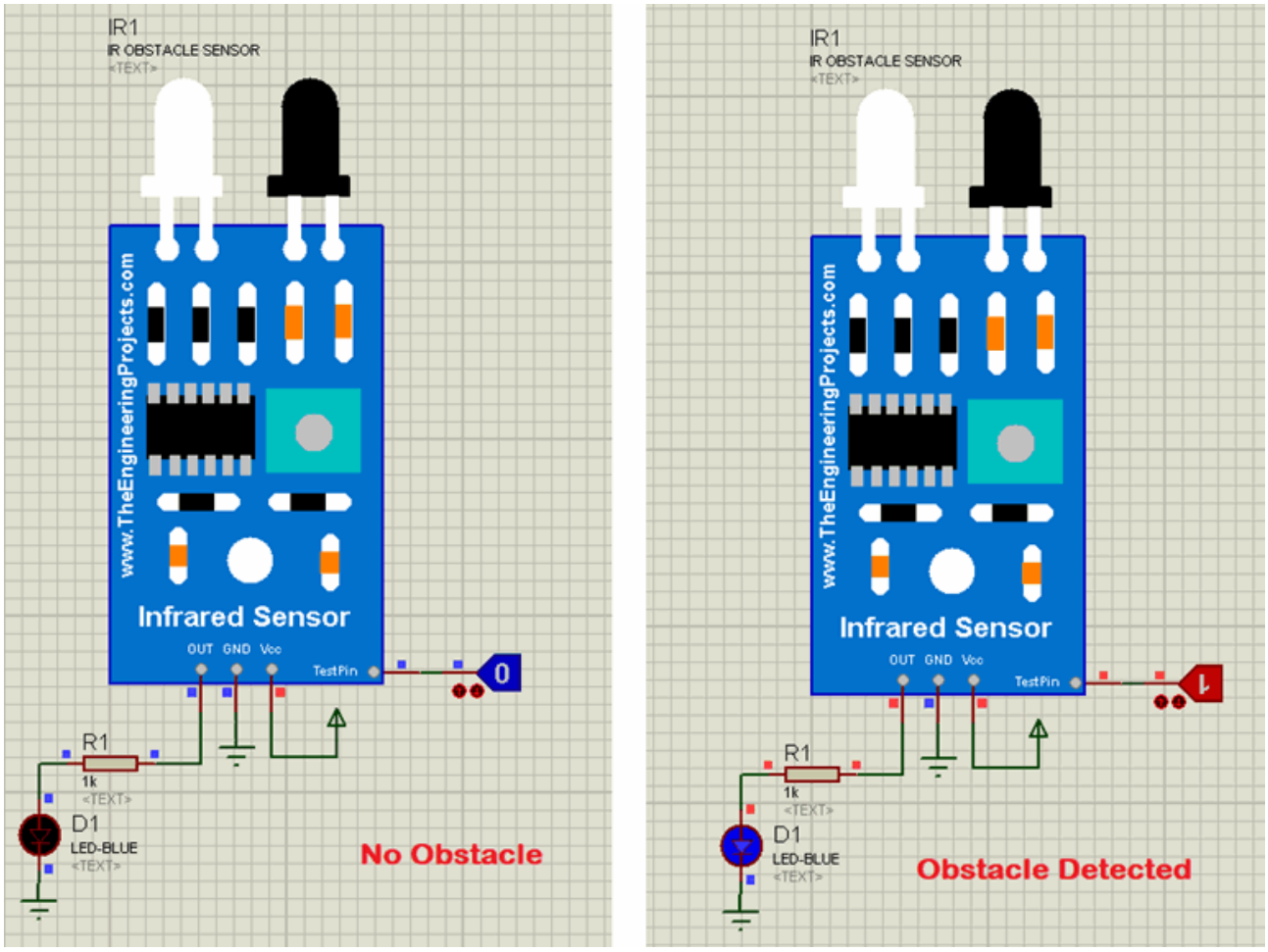
To maximize the functionality of the Infrared Sensor, configure its Properties Panel by selecting the InfraredSensorsTEP.HEX file.

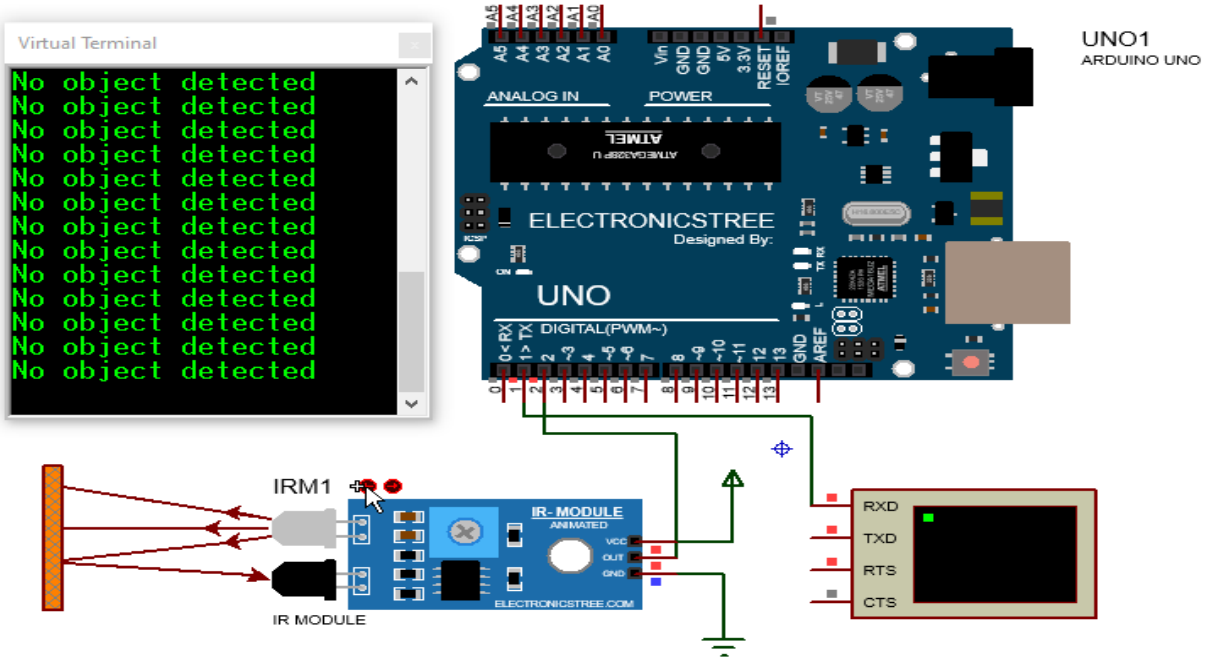
Simulation Results

As you run your Proteus simulation, observe the results. When the TestPin is LOW, the OUT Pin remains LOW, signifying no obstacle. Conversely, a HIGH TestPin indicates the presence of an obstacle.



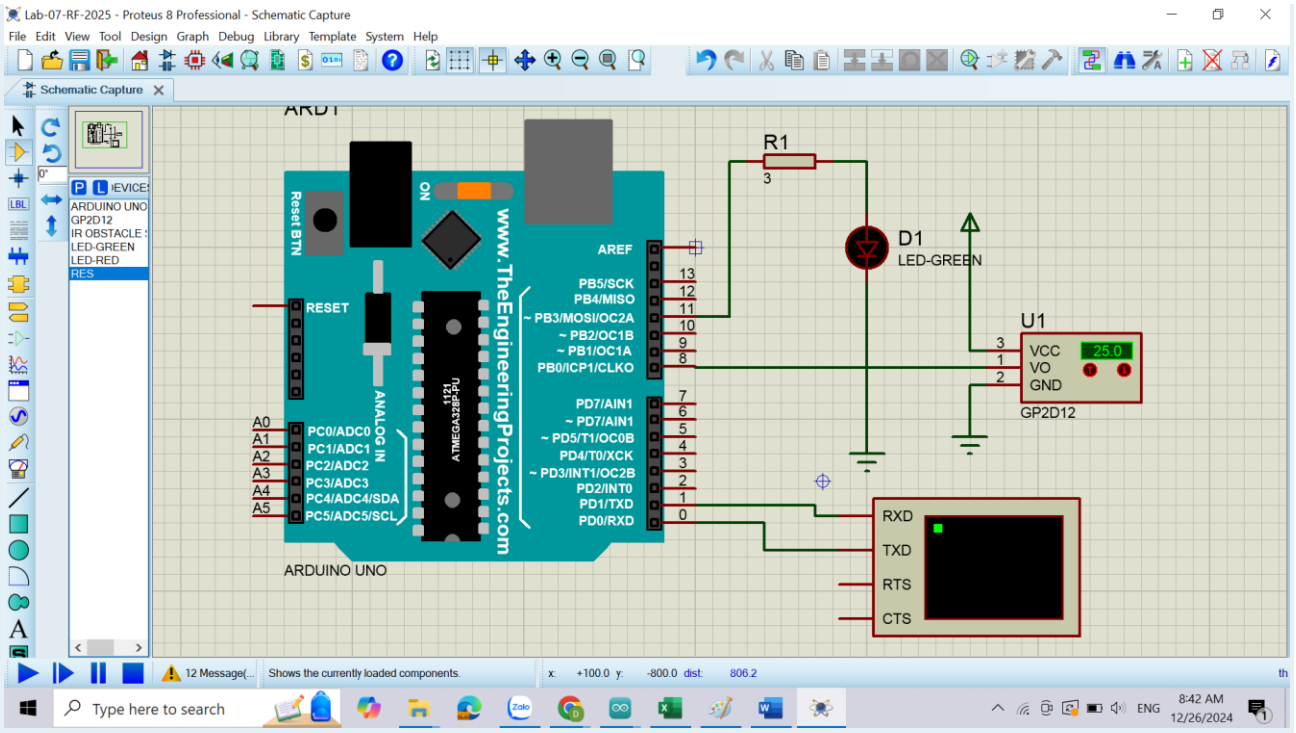
If right sensor have more distance than other two then it moves right. If all the sensors have less than max range then it checks which is greater in them.





5. How to Simulate the IR (Infrared) Sensor with Arduino in Proteus?

In this lab, we will delve into the fascinating world of **Simulating IR (infrared) Sensor using Arduino within the Proteus software environment**. The IR sensor is a pivotal electronic device capable of emitting light to comprehend its surroundings. Functioning as a **temperature sensor**, it can measure an object's heat and even **detect its speed**, making it a versatile tool with numerous applications.



In the previous tutorial, we learned the basic knowledge of temperature sensors and their various types and **IR (Infrared) Temperature Sensor**. Let us now start with the simple project of **IR sensors** in **Proteus** window.

Components Required:

Before we dive into the simulation process, let's gather the necessary components:

- Proteus Software
- Arduino Software(IDE)
- Arduino Uno board
- Sensor module
- Connecting terminals
- LED with 220ohm Resistor
- Virtual terminal

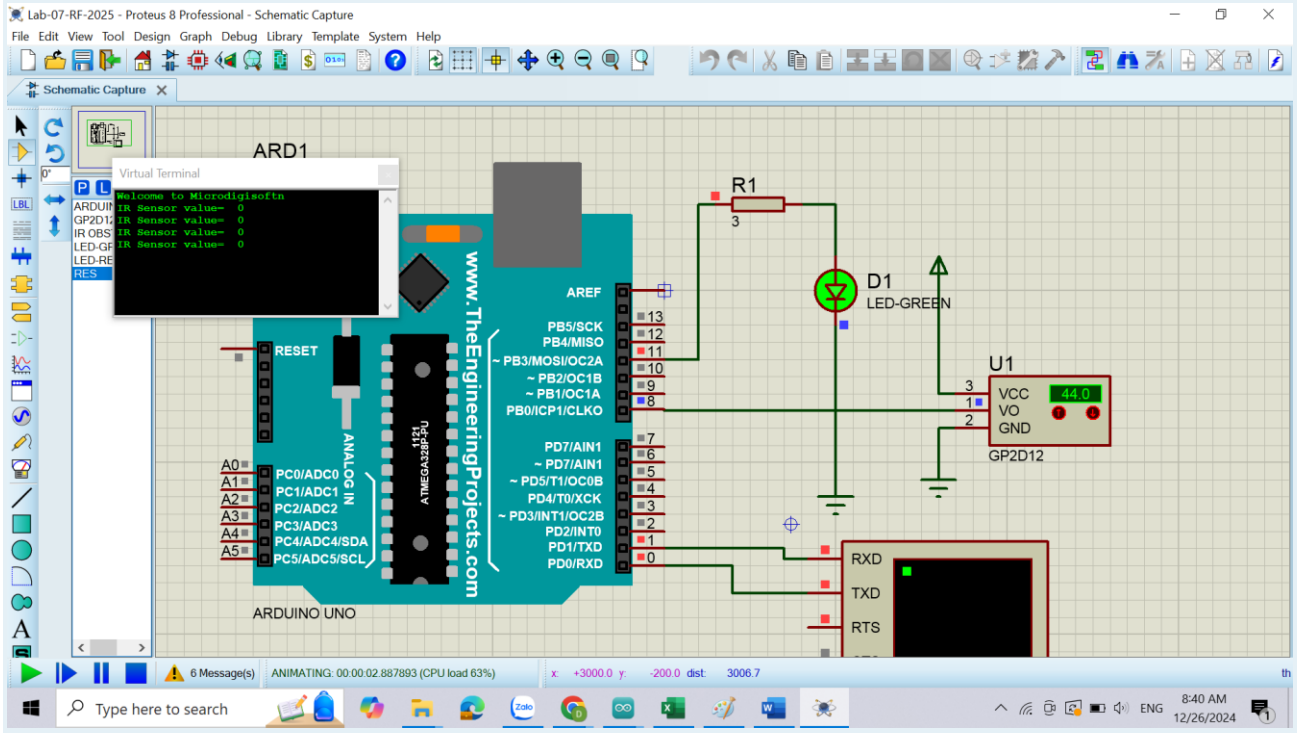
Connections:

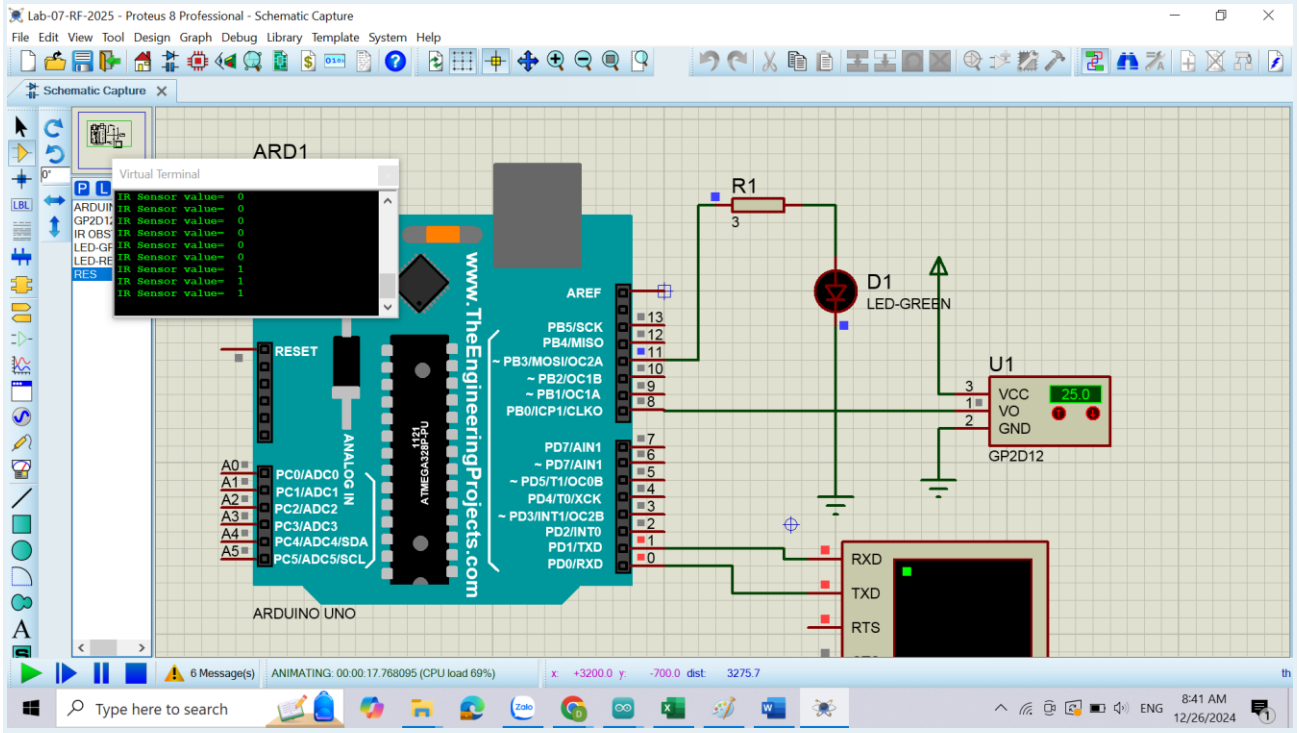
The success of our simulation hinges on making the right connections:

- **VCC** connects to the **5V** power terminal.
- **OUT** is linked to **Arduino's digital PIN (PIN 8)**.

- **GND** from the sensor module is connected to the ground terminal.
- An LED, coupled with a **220-ohm resistor**, is attached to **Arduino board PIN 11** and the ground terminal.
- Additional virtual terminals are established to display the simulation results effectively.

Circuit Diagram





Arduino Sketch

The simulation model of IR (Infrared) Sensor must be present in the library of Proteus. But the simulation model is not in the library of the Proteus, so you must download the library first and then place files containing the simulation model in the **Proteas library**.

```
int IRSensor = 8; // connect ir sensor to arduino pin 2
int LED = 11; // connect Led to arduino pin 13
void setup()
{
  pinMode (IRSensor, INPUT); // sensor pin INPUT
  pinMode (LED, OUTPUT);     // Led pin OUTPUT
  Serial.begin(9600);
  delay(500);
  Serial.println("Welcome to Microdigisoftn");
}
void loop()
{
  delay(500);
  Serial.print("IR Sensor value=  ");
  Serial.println(digitalRead(IRSensor));
  int statusSensor = digitalRead (IRSensor);
  if(statusSensor == 1)
      digitalWrite(LED, LOW); // LED LOW
  else
  {
      digitalWrite(LED, HIGH); // LED High
  }
}
```

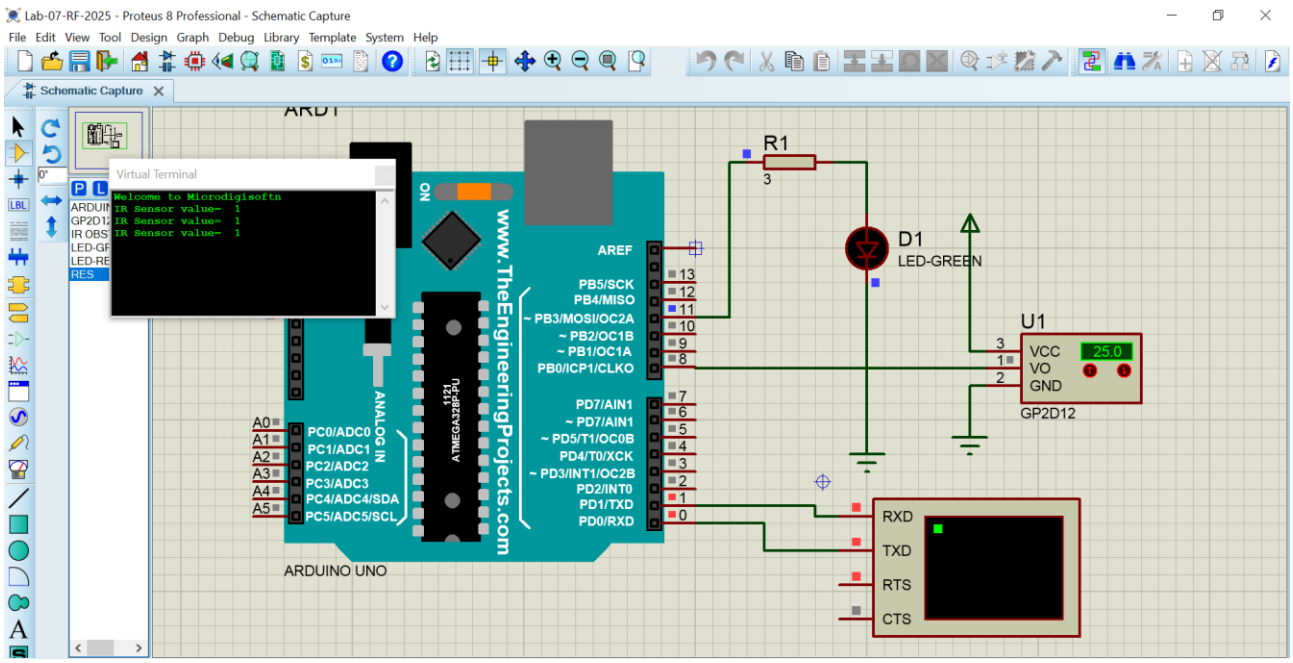
How the Code Works?

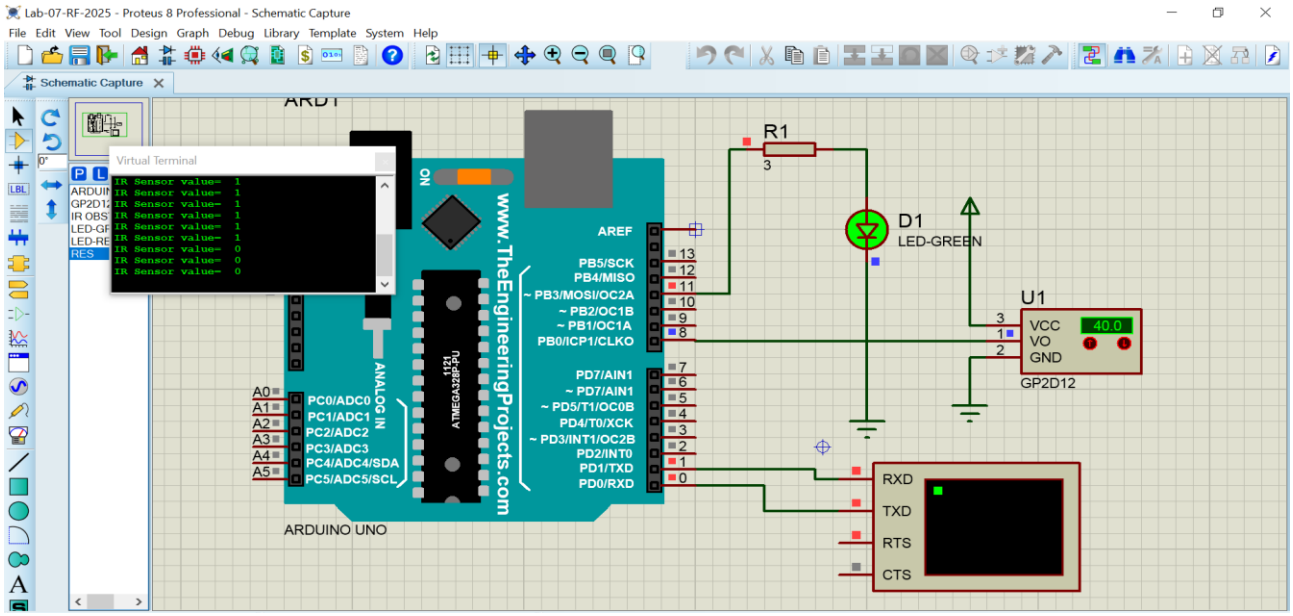
The code begins by defining two important variables: `IRSensor` and `LED`, corresponding to the Arduino PINs where the IR sensor and LED are connected, respectively. In the `setup()` function, we set the sensor PIN as INPUT and the LED PIN as OUTPUT. We also initialize serial communication for debugging purposes.

In the `loop()` function, the code reads the value from the IR sensor and prints it to the serial monitor. Based on the sensor's status, the code either turns the LED off or on, providing a visual indication of the IR sensor's output.

Compile the above code, create the **hex file** and upload it in Proteus. Then double click **"ARDUINO UNO R3"**, we will see Edit Component window after that Browse it in Program File/ Put file path which selecting from the arduino uploading window, then **click OK**.

Result

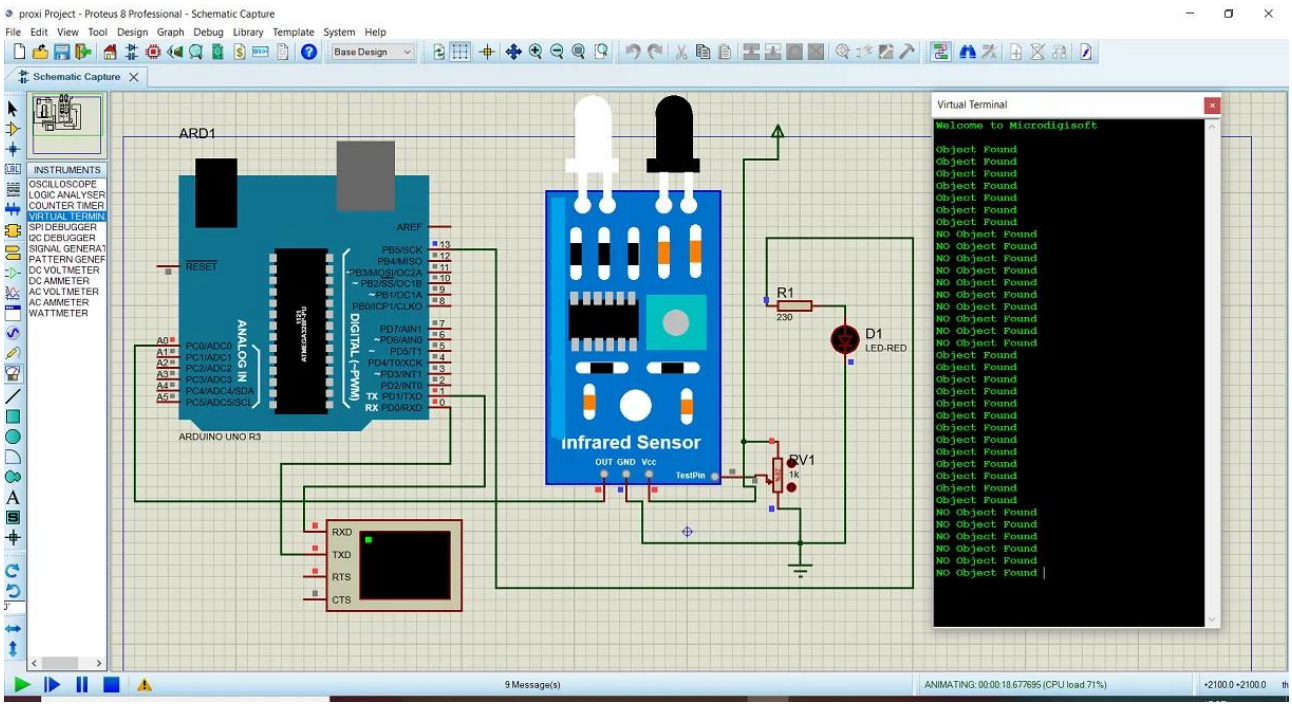




When you put the object in front of the IR (Infrared) Sensor and observe, the LED should be turned on, which is attached to the Arduino board. Otherwise, when you remove the object, you will see the LED automatically off.

6. Arduino IR Proximity Sensor Simulation in Proteus Software

In this comprehensive tutorial, we will guide you through the process of simulating and interfacing an **IR proximity sensor with an Arduino** in the Proteus Software environment. This project is made possible by using an Arduino board, an **IR Proximity Sensor module**, an LED, a Virtual Terminal, and the Arduino IDE software. If you're new to working with sensors, you might want to check out our previous tutorial, "How to Simulate the IR (Infrared) Sensor with Arduino in Proteus," which covers the basics of temperature sensors and IR (Infrared) Temperature Sensor.



IR Proximity Sensors are used to **detect obstacles** in their way. They are usually used on **robots** to avoid path routing and **Avoiding obstacles**. **Infrared(IR)** Proximity Sensors have **IR LED** and light detectors(**Photodiode**) to detect reflections.

6.1. How It Works

IR Proximity Sensors recognizes the existence of an object by emitting **infrared light** rays. It works like **ultrasonic sensors**. It provides a **digital output** of 1 or 0. Whether the device is getting enough power with inbuilt **LED indicators** and start experimenting with your logic. You will visit my last tutorial to better understand **IR Proximity Sensor with Arduino Board**.

6.2. Component required:

Before diving into the simulation, ensure you have the following components ready:

- Arduino Uno
- IR Proximity sensor module
- LED with 220ohm resistor
- Connecting terminals
- virtual monitor

Note:=>Proximity sensors are not available in the Proteus window. You must download and simulate the **IR Proximity Sensor Library** for Proteus.

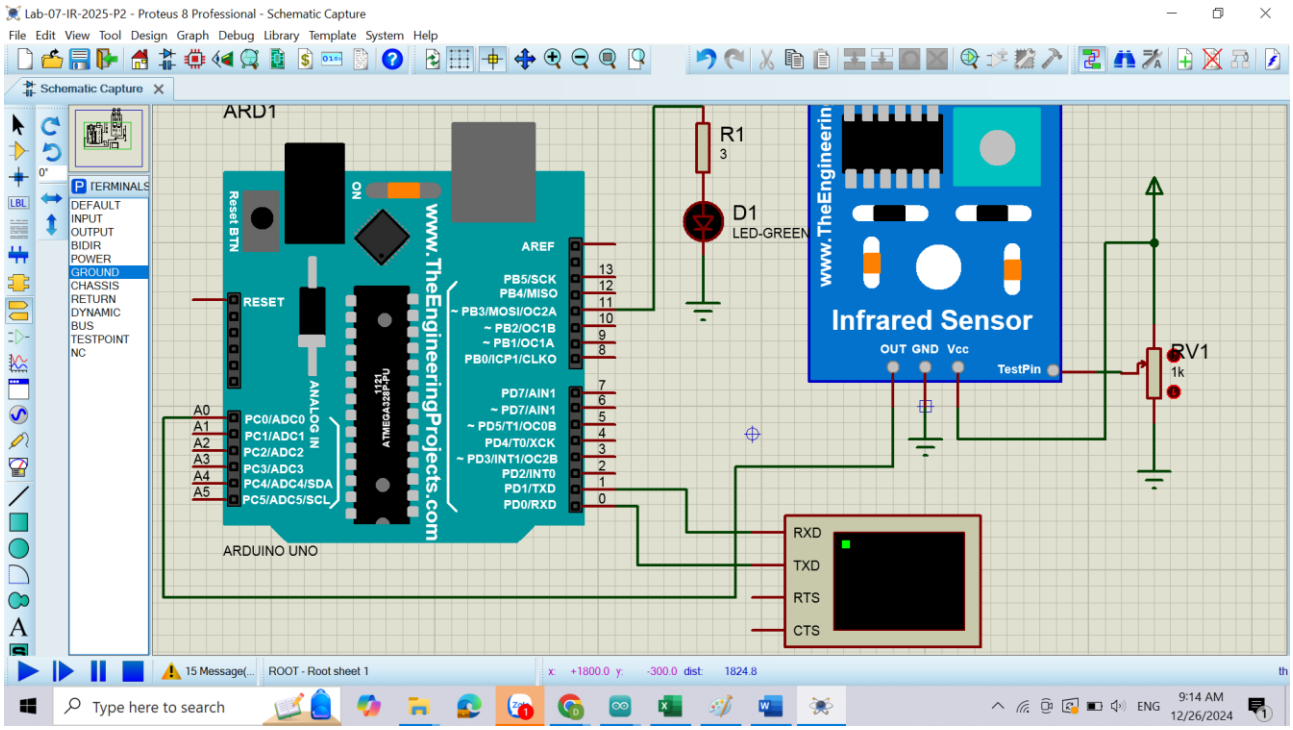
6.3. Connections:

Make the following connections to set up your simulation:

- **VCC** of sensor module connected to 5v of power terminal.
- **OUT** connects to Arduino's Analog PIN (A0)
- **GND** of sensor module are connected to ground terminal.
- **TestPin** (Proteus is a simulation software, so we can't put something in front of this sensor and therefore he has used the text PIN.) connected with 10k potentiometer.
- **LED** attachments with 220ohm resistor to arduino board pin 13 and ground terminal .

And connect additional **virtual terminals** to show results, as shown below.

6.4. Circuit Diagram



6.5. Arduino Sketch

Here's the code that makes this simulation possible:

```
//Setup function initialize Serial monitor and Read the
Analog Data
void setup() {
  Serial.begin(9600);
  delay(500);
  Serial.println("Welcome to Microdigisoft \n");
  pinMode(A0, INPUT);// Define pin A0 as a input
  Serial.begin(9600);
  pinMode(13, OUTPUT);// Define pin 13 as a output
}
void loop() {
  delay(500);
  Serial.println("");
  //Serial.println(digitalRead(A0));
  if (digitalRead(A0) == 0)
  {
    digitalWrite(13, HIGH);
    Serial.print("Object Found ");
  }
  else {
    digitalWrite(13, LOW);
    Serial.print("NO Object Found ");
  }
}
```

6.6. How the Code Works

The setup function initializes the Serial monitor and sets up pins for input and output. The loop function constantly checks the digital input from A0, which corresponds to the IR Proximity Sensor's output. If it detects an object, it turns on the LED and prints "Object Found" to the Serial monitor. Otherwise, it turns off the LED and prints "NO Object Found."

ir_sensor_2 | Arduino 1.8.12

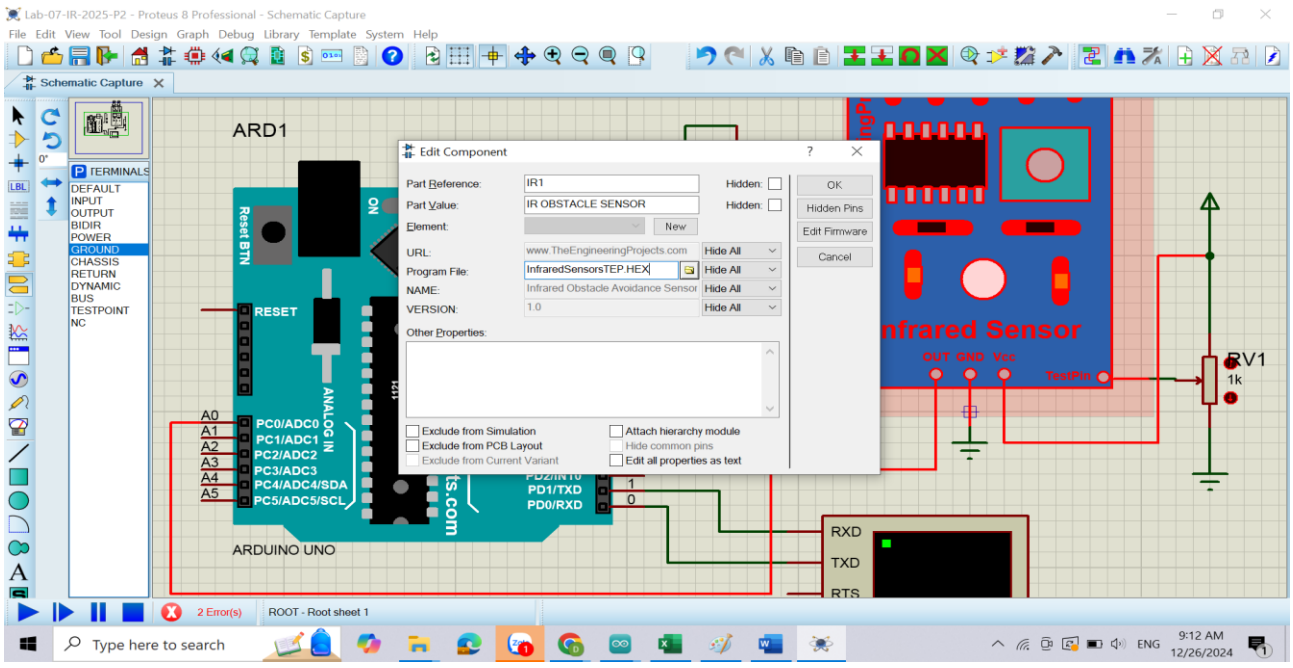
File Edit Sketch Tools Help

ir_sensor_2

```
void setup() {
    Serial.begin(9600);
delay(500);
  Serial.println("Welcome to Microdigisoft\n");
  pinMode(A0,INPUT);
  Serial.begin(9600);
  pinMode(13,OUTPUT);
}

void loop() {
  delay(500);
Serial.println("");
//Serial.println(digitalRead(A0));

if(digitalRead(A0)==0)
{
  digitalWrite(13,HIGH);
  Serial.print("Object Found ");
  }
 else{
    digitalWrite(13,LOW);
    Serial.print("NO Object Found ");
    }

}
```

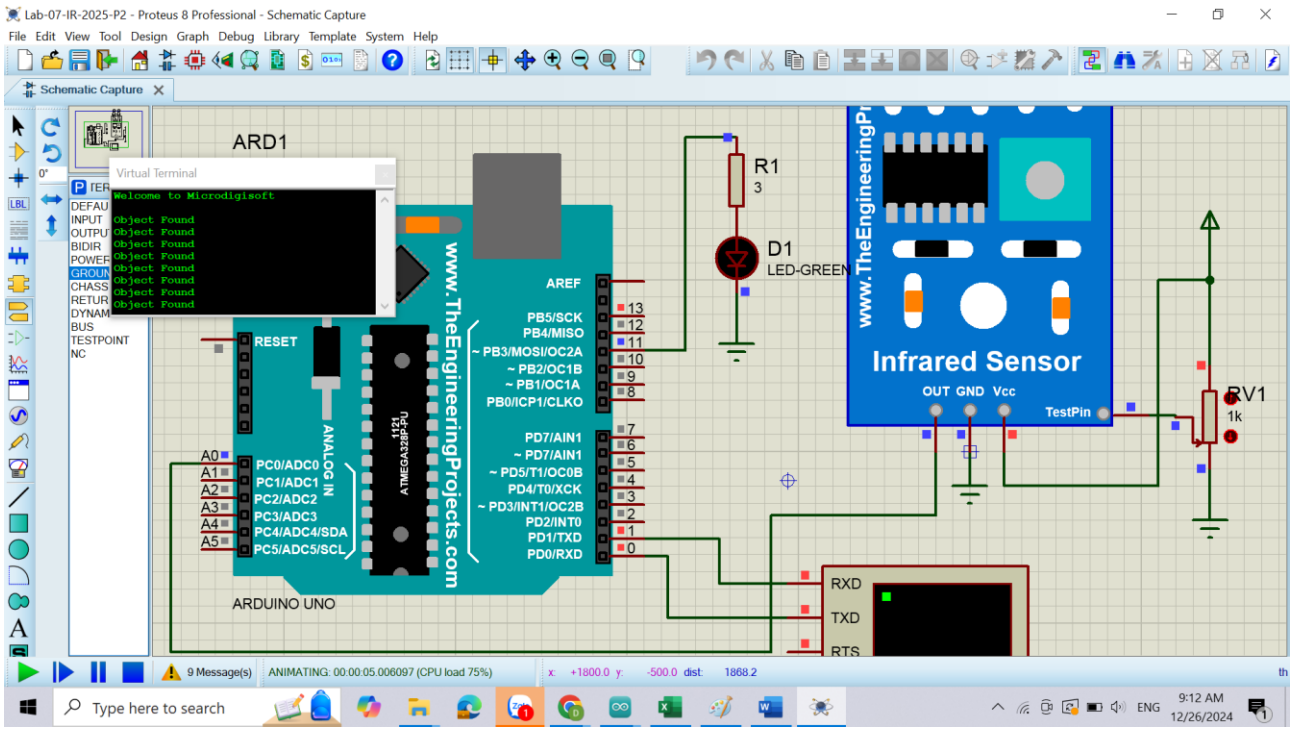
Done compiling.

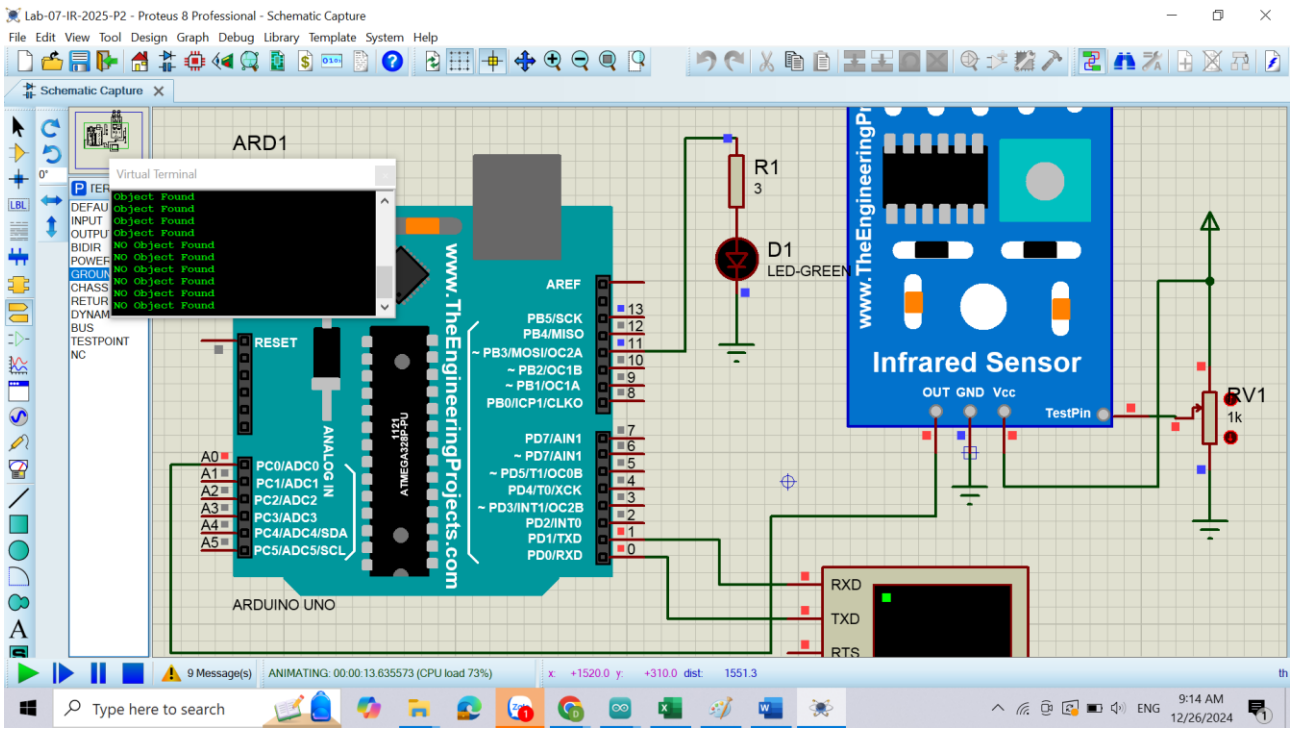
Compile the above code, create the **hex file** and upload it in Proteus. Then double click **"ARDUINO UNO R3"**, we will see Edit Component window after that Browse it in Program File/ Put file path which selecting from the arduino uploading window, then **click OK**.



6.7. Result:

When you put the object in **front of the IR sensor**(Adjust the potentiometer upwards) and observe, the **LED** should be turned **ON**, which is attached to the Arduino board. Otherwise, when you **remove the object**(Adjust the potentiometer downward), you will see the **LED** automatically **OFF**.





7. Read More

Frequently Asked Questions (FAQs)

1. **Where can I find the Infrared Sensor Library for Proteus on GitHub?**

- You can easily access the Infrared Sensor Library for Proteus on GitHub by visiting the official repository. Additionally, you can visit this website for the Infrared Sensor Library: goalmdcat.com/infrared-sensor-library/.

2. **Is the Infrared Sensor Library for Proteus available for free download?**
    - Yes, the library is available for free download, ensuring that all Proteus users can easily obtain and use it. For direct download, visit goalmdcat.com/infrared-sensor-library/.
3. **Can I use the Infrared Sensor Library for Proteus with Arduino?**
    - Absolutely! The library is fully compatible with Arduino, offering seamless integration for a wide range of projects.
4. **How do I download the Infrared Sensor Library for Proteus?**
    - Downloading the Infrared Sensor Library for Proteus is a breeze. Simply visit our official website and click the download button at goalmdcat.com/infrared-sensor-library/.
5. **What is the best Infrared Sensor Library for Proteus?**
    - The library provided on our website is widely considered one of the best, offering reliable performance for your Proteus simulations.
6. **Where can I download the IR Sensor Library for Proteus 8?**
    - You can find the download link for the IR Sensor Library compatible with Proteus 8 on our official website at goalmdcat.com/infrared-sensor-library/.
7. **Is there a proximity sensor library for Proteus available for download?**
    - Certainly! You can easily download the proximity sensor library for Proteus from our website at goalmdcat.com/infrared-sensor-library/.
8. **How can I get the IR sensor in Proteus?**
    - Follow our step-by-step guide available on our website to smoothly incorporate the IR sensor into your Proteus simulations. Visit goalmdcat.com/infrared-sensor-library/ for more details.