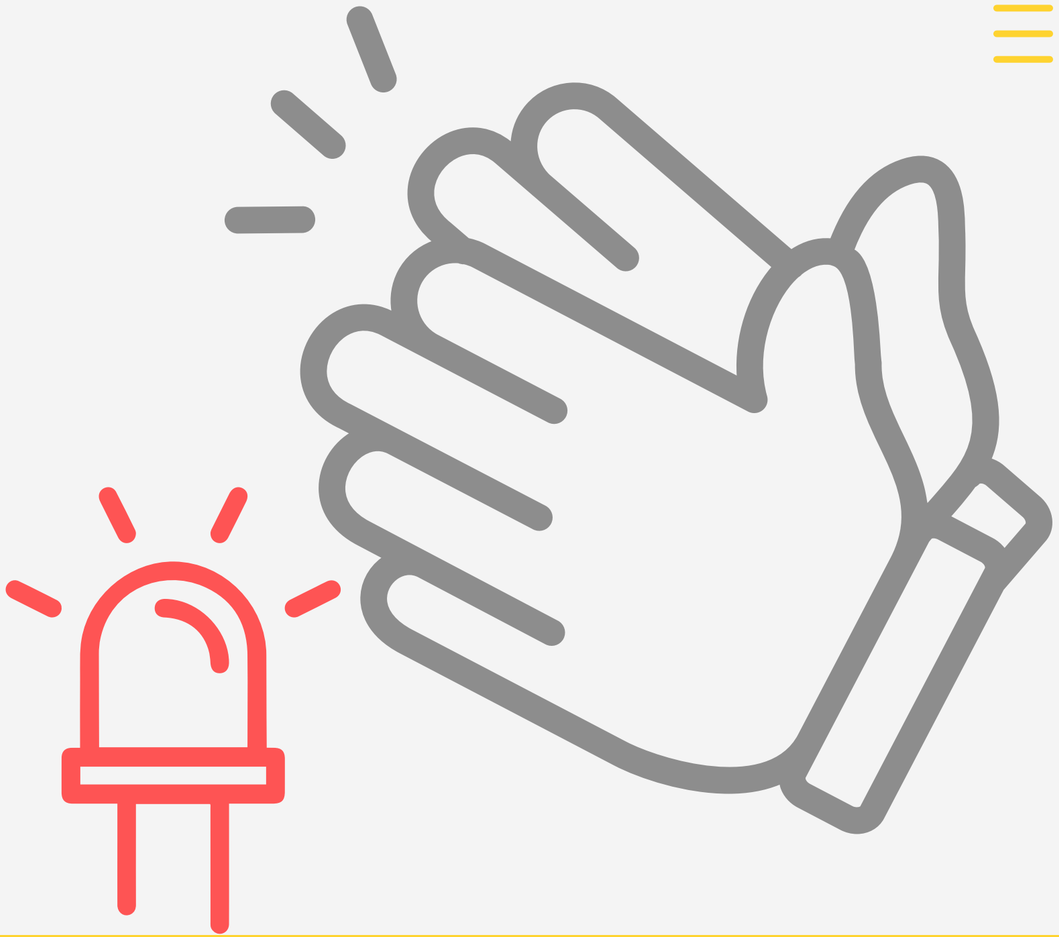
PassionBots

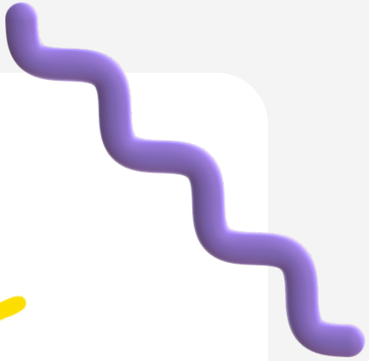
STUDENT'S GUIDE



Clap on clap off →

WWW.PASSIONBOTS.IN

Clap on Clap off



About The Class

AIM

Design and implement a Clap On Clap Off system using a sound sensor and LED on an Arduino Uno.

Topics Covered

Introduction to Arduino Uno, basics of electronics, understanding how a sound sensor works

Time

120 minutes

Tools Used

Arduino IDE

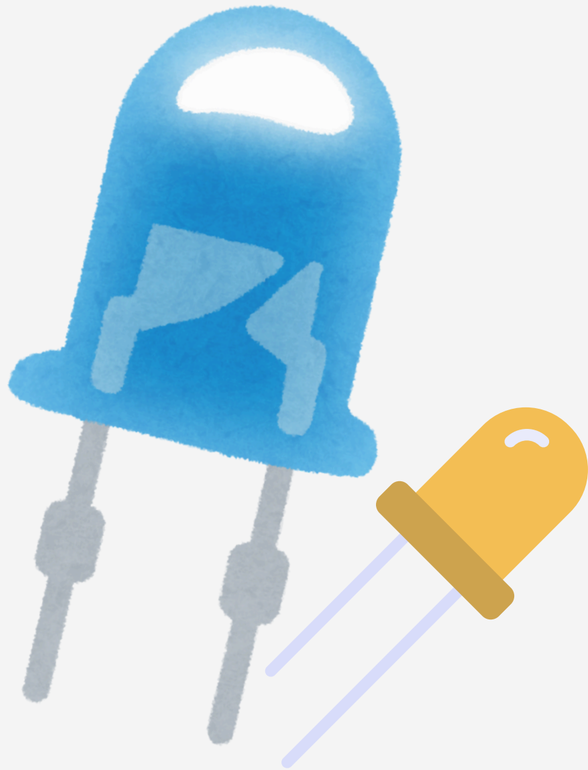
Benefits Of The Project

Hands-on learning

Understanding sound sensor principles

Practical application of home automation

Enhancing problem-solving skills



Introduction

What is Clap on Clap off?

- The Clap On Clap Off system is a classic example of **simple home automation** by detecting clapping sounds.

- It enables users to **control** electrical compoenets such as **LED** with just the **sound of a clap.**

- We will make use of a sound sensor which will detect the sound of a clap and accordingly switch the LED on or off

Components

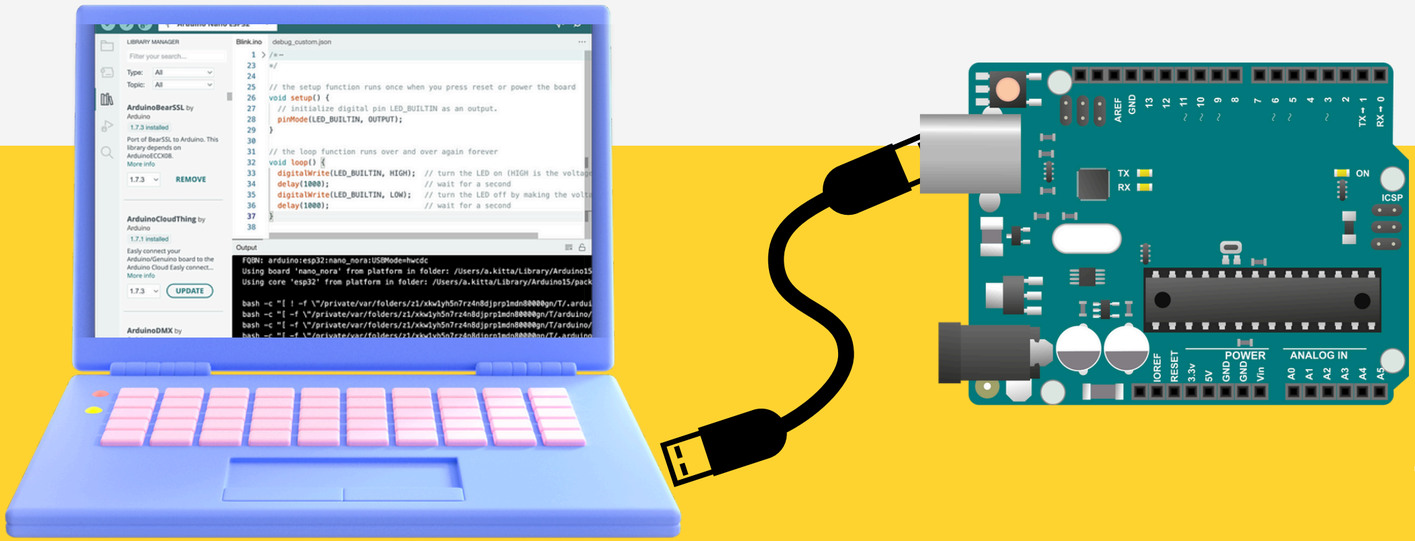
COMPONENT LIST	QUANTITY
Arduino Uno	1
Sound Sensor	1
Jumper Wire	2
LED	1

Arduino UNO

- Think of the Arduino Uno as a smart brain for your projects, helping you easily connect and control different electronic parts.

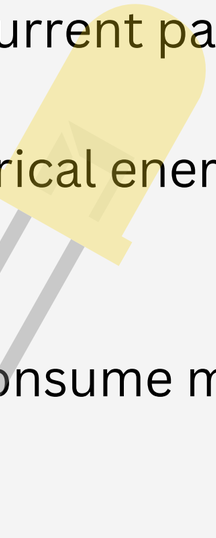
- It's like being part of a big, friendly club where everyone shares ideas and helps each other - that's the open-source magic of Arduino Uno.

- With its superpowers to understand and communicate with various gadgets, the Arduino Uno turns your ideas into reality, whether it's lighting up LEDs or making robots move.



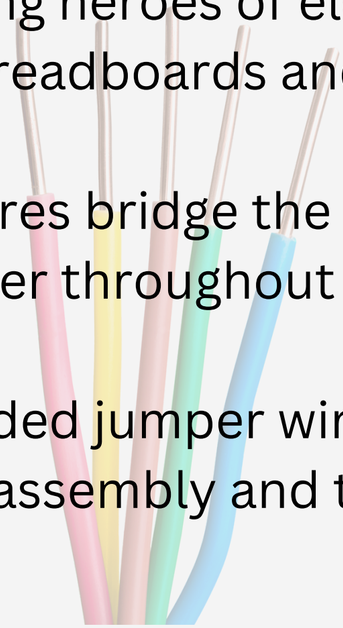
LEDs

- **LEDs (Light-Emitting Diodes)** serve as visual indicators in electronic projects. They emit light when an electric current passes through them.
- **Functionality:** LEDs convert electrical energy into light energy efficiently, making them ideal for signaling.
- **Low Power Consumption:** LEDs consume minimal power compared to traditional incandescent bulbs



Jumper Wires

- **Jumper wires** are the unsung heroes of electronics, facilitating connections between components on breadboards and circuit boards.
- **Bridge Builders:** Jumper wires bridge the gap between components, enabling the flow of signals and power throughout the circuit.
- **Organization Aid:** Color-coded jumper wires help organize connections and reducing confusion during assembly and testing.



Sound Sensor

- The **Sound sensor** converts vibrations from **sound waves** into **electrical signals**.
- This conversion process allows it to detect sound intensity changes, like claps, by measuring these electrical variations.
- With adjustable sensitivity levels, the sound sensor can be **fine-tuned to respond accurately** to different sound intensities.
- In this project, the **Arduino interprets signals** from the sensor to **detect claps,** facilitating hands-free control of electrical appliances.

How it Works?

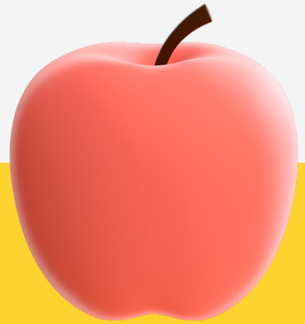
- A **Sound sensor** captures sound signals, which the Arduino Uno processes. It analyzes signal intensity and pattern, recognizing sharp spikes like claps.

- When a **clap is detected**, the Arduino **triggers a relay module** acting as a switch. This controls the power supply to an electrical appliance, either completing or interrupting the circuit.

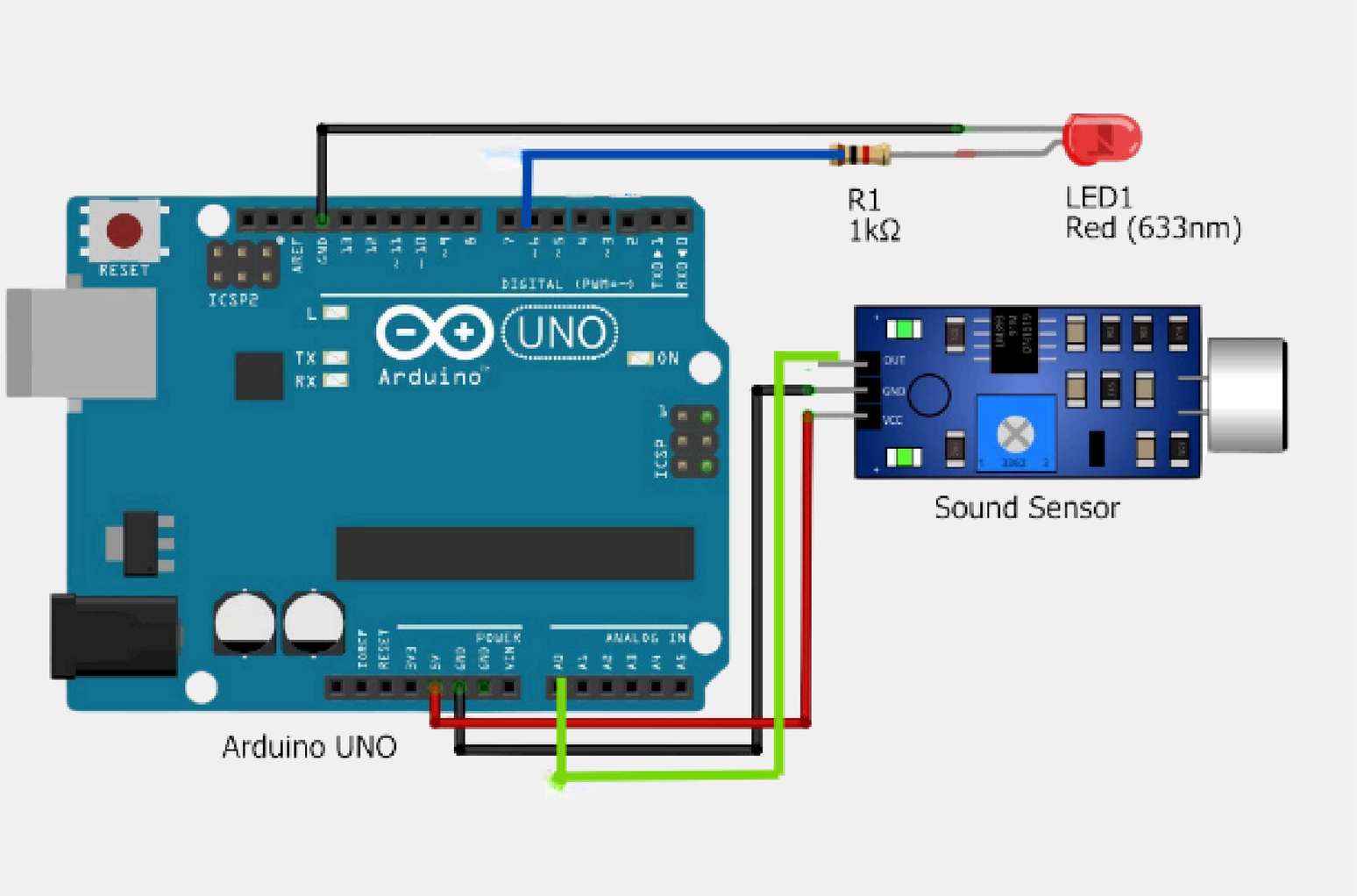
- Users can turn the appliance **on or off with a clap gesture**. Depending on its initial state, the relay module adjusts the circuit, enabling easy control of the appliance.

Student Activity Link

Activity reference video	**https://shorturl.at/cwFGV**



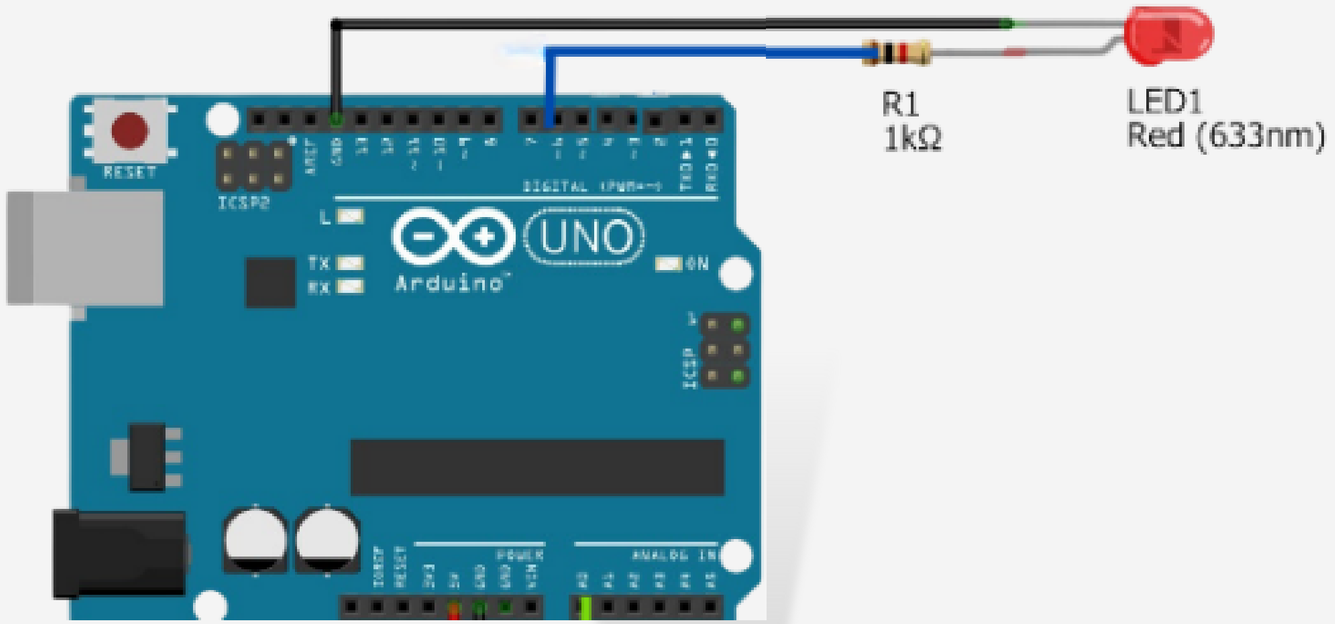
Circuit Connection



Step 1

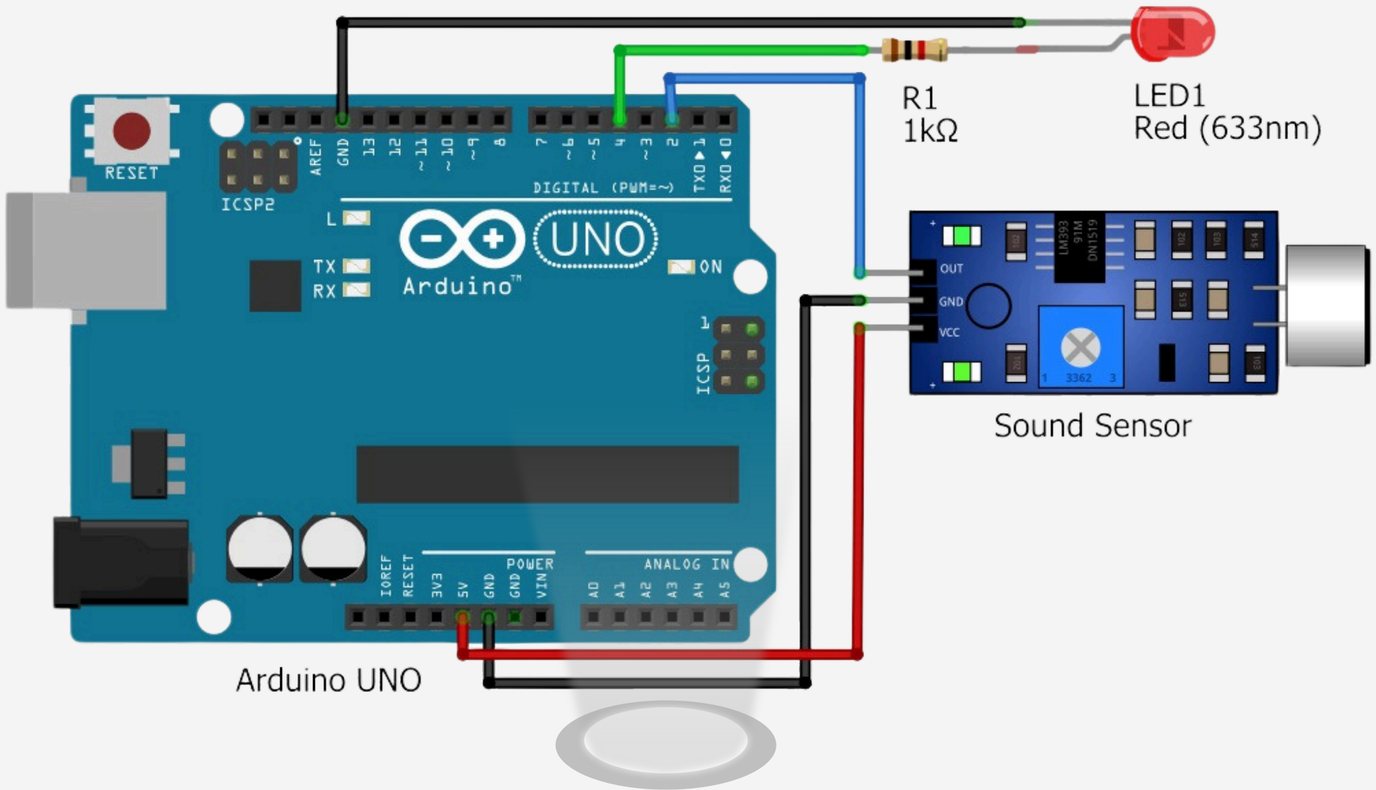
- Connect the LED to the Arduino Uno.

- Here, we connect the Anode (+ve longer end) to pin 4 of the Arduino Uno and the Cathode (-ve Shorterend) to the GND pin of the Arduino Uno.



Step 2

- Connect the Sound Sensor OUT pin to Arduino Pin 2.
- Connect Vcc and Gnd of Sound Sensor with Arduino **5V** and **GND** respectively.



Step 3

- Let’s write the Code for our System.
- First of all, we need to define the Sound Senosr and LED pins, here we define them at pins A0 and digital pin 6 respectively.
- Then we declares integer variables for sound and LED states, alongside a trigger threshold (600) determining LED activation based on sound sensor readings.

```
//If you aren't using A0 as your sound sensor socket change it below
#define soundSensorSocket A0

//If you aren't using pin 6 for your led socket change it below
#define ledSocket 6

int soundState;
int ledState;
int trigger = 600;
```

Step 4

- Now we will write the setup code.
- For the LED to respond to clapping, the pinMode function should first set the pin as an input to read the sensor.
- After that we will set the pinMode function as an output to control the LED.

```
void setup()
{
  pinMode(ledSocket, INPUT);
  pinMode(ledSocket, OUTPUT);
}
```

Step 5

- Here, we continuously read the analog value from the sound sensor.
- If the value exceeds a predefined trigger level, it sets the soundState variable to 1, otherwise to 0.
- Then, it toggles the LED state based on sound detection: turning it off if it's on and a sound is detected, and vice versa, with a short delay.

```
void loop()
{
  if (analogRead(soundSensorSocket) > trigger) {
    soundState = 1;
  } else {
    soundState = 0;
  }

  ledState = digitalRead(ledSocket);
  if (ledState == 1 && soundState == 1) {
    delay(400);
    digitalWrite(ledSocket, LOW);
  }
  if (ledState == 0 && soundState == 1) {
    delay(400);
    digitalWrite(ledSocket, HIGH);
  }
}
```

Final OutPut

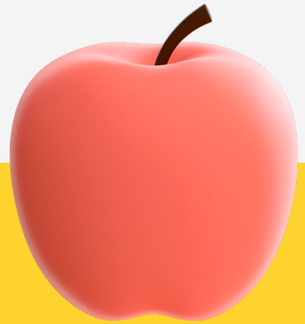
Link	**https://shorturl.at/cwFGV**

Congratulations

Hurray!

We have completed the

Clap on Clap off Project



Learning outcomes:

- **Electronics Basics**: Learn how circuits and switches function.
- **Problem-Solving:** Develop troubleshooting skills to fix issues.
- **Problem-Solving:** Develop troubleshooting skills to fix issues.

.

Notes

- Verify all connections with respected teachers.
- Safely use the electronics components.
- Check the Project is working properly.

Thank you!

Do you have any questions for me?