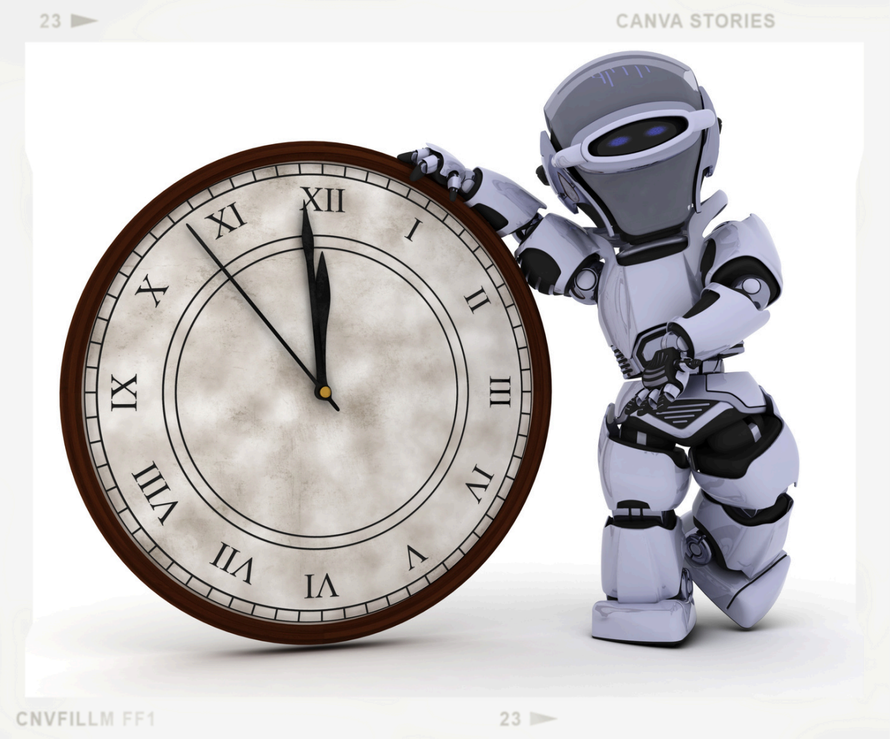
PassionBots

STUDENT'S GUIDE



Tick Tock Tech →

© PASSIONBOTS

WWW.PASSIONBOTS.IN

Tick Tock Tech

© PASSIONBOTS

About The Class

AIM

To design and develop a simple alarm system and programming it to ring after a set timer.

Topics Covered

Introduction to Arduino Uno and its programming environment, concept of timing, incorporating an audible alarm system

Time

120 minutes

Tools Used

Arduino IDE

© PASSIONBOTS

Benefits Of The Project

Hands-on experience in integrating sensors

Understanding of basic concept of timers

Introduction to Alarm System

Creativity enhancement through designing

© PASSIONBOTS

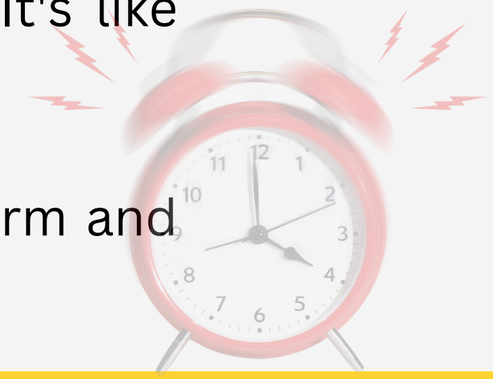
Introduction

What is Tick Tock Tech?

- In this project, we will be building an **Alarm system** in which will ring when the set time is up just as any normal alarm

- Imagine a knob like the volume control on a radio. With our project, you turn this knob (**potentiometer**) to **pick the time** when you want the alarm to ring. It's like picking the hour on a clock.

- We will be using a **button** which the user will have to press **to start** the alarm and when the set time arrives teh buzzer goes off.



© PASSIONBOTS

Components

COMPONENT LIST	QUANTITY
Arduino Uno	1
USB cable	1
Rotary potentiometer	1
Jumper wire	2
buzzer	1
LCD display	1

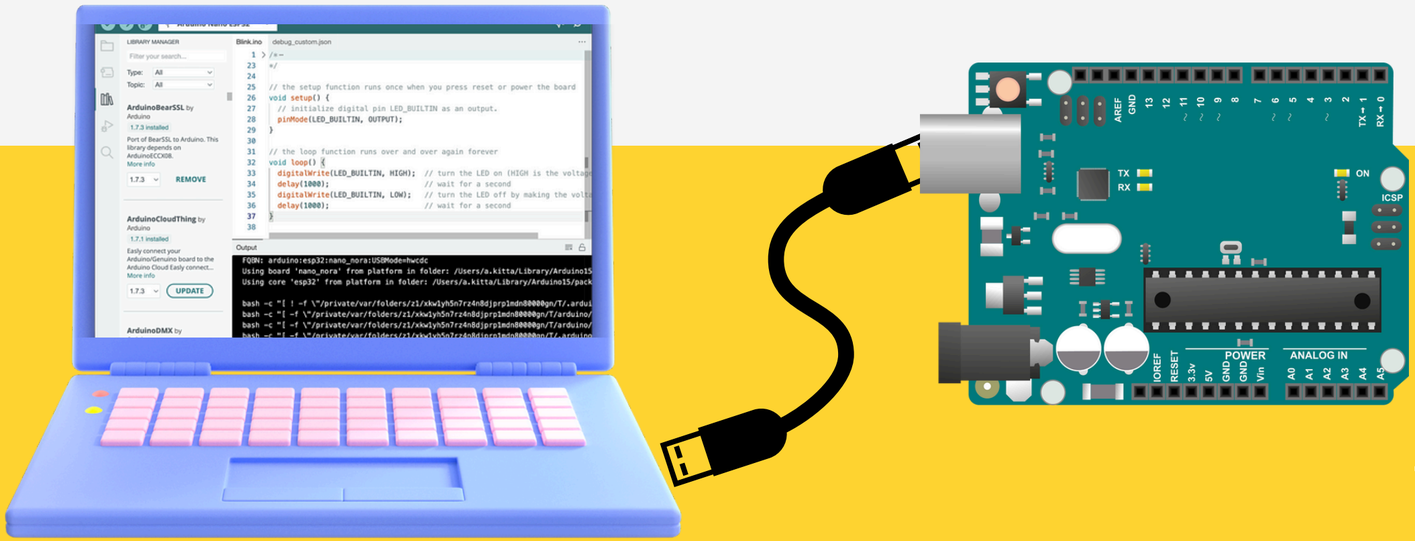
© PASSIONBOTS

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.



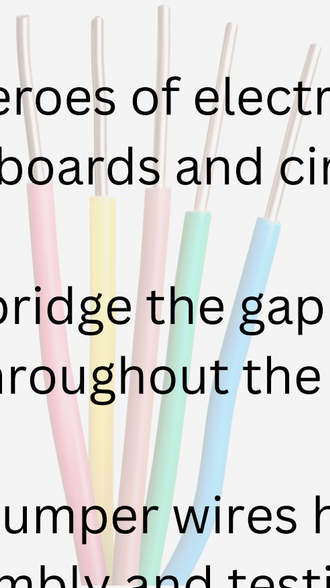
© PASSIONBOTS

USB cable

- **Description:** A standard USB cable used to connect the Arduino Uno to a computer for programming and power.
- **Function:** Provides power to the Arduino Uno and allows data transfer between the board and the computer for programming.
- **Usage**: Connects the Arduino Uno board to a computer's USB port for programming and power supply.

© PASSIONBOTS

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.

© PASSIONBOTS

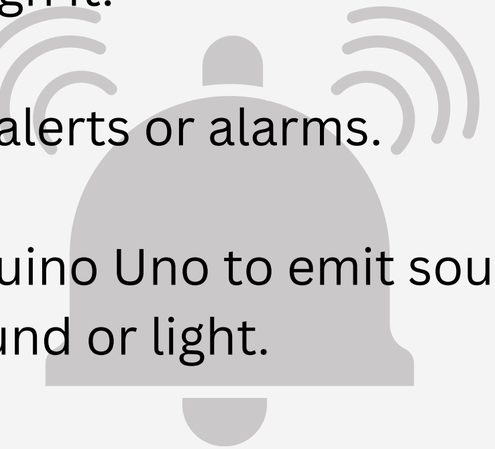
Rotary Potentiometer

- **Description:** A variable resistor with a knob that can be turned to adjust its resistance.
- **Function:** Used to control parameters such as volume, sensitivity, or threshold in electronic circuits.
- **Usage:** In the "Sound the Alarm" project, it can be used to adjust the sensitivity of the sound sensor or the threshold for triggering the alarm.

© PASSIONBOTS

Buzzer

- **Description:** An electromechanical component that produces sound when an electric current passes through it.
- **Function:** Generates audible alerts or alarms.
- **Usage:** Connected to the Arduino Uno to emit sound when triggered by input signals, such as detecting sound or light.



© PASSIONBOTS

How it Works?

- In our project, we're creating an Alarm system that functions just like any standard alarm clock.

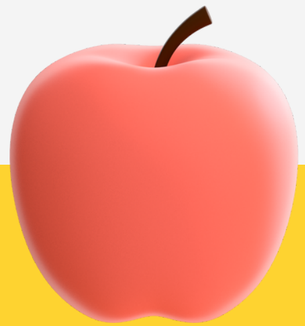
- We will be using a knob like component, called a potentiometer,it allows you to select the time you want the alarm to ring. It's as simple as setting the hour on a clock.

- Once you've set the desired time using the knob, we press a button which initiates the countdown. When the set time arrives, a buzzer will sound off, just like any regular alarm clock.

© PASSIONBOTS

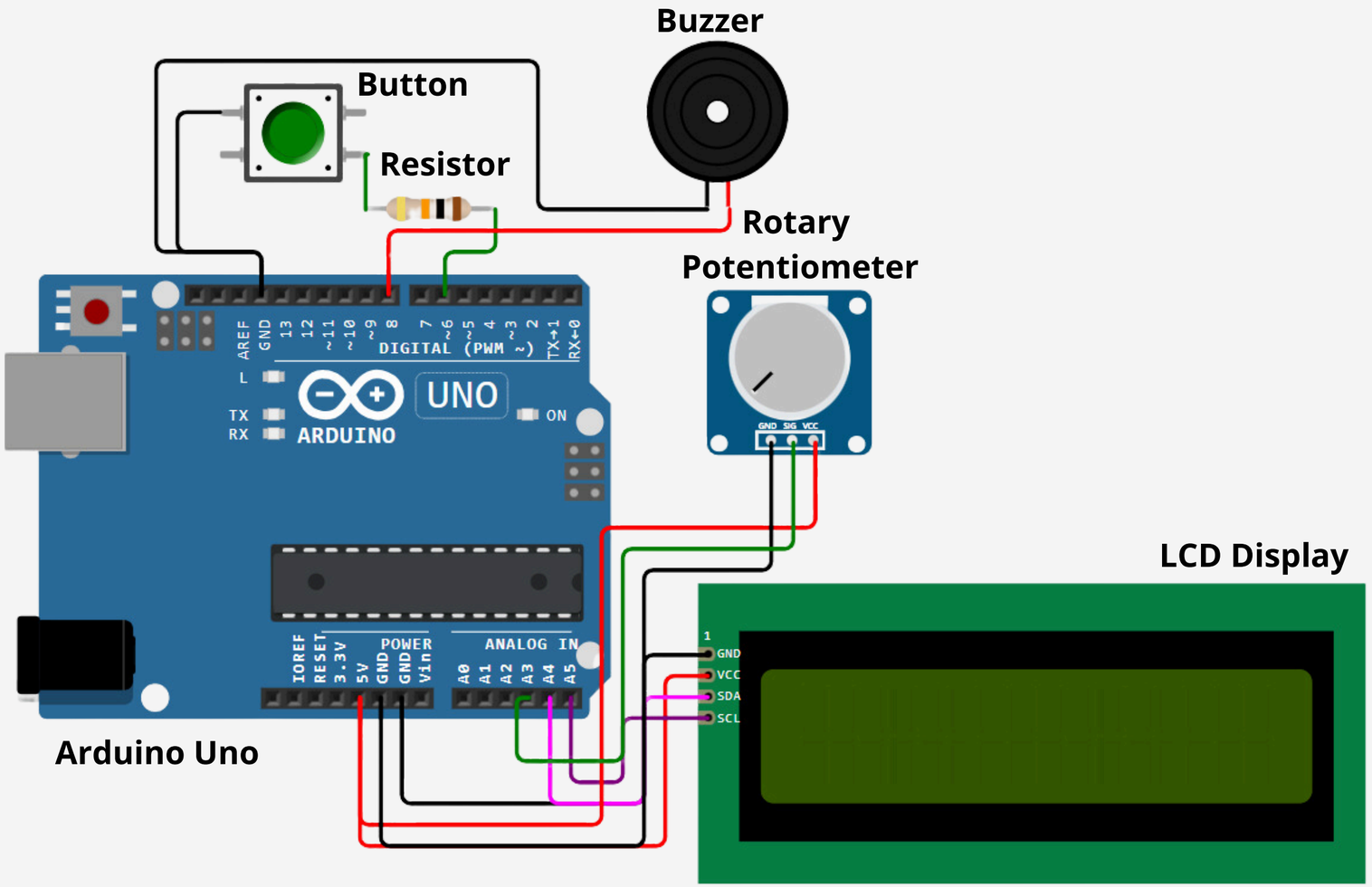
Student Activity Link

Activity reference video	**Student Activity**



© PASSIONBOTS

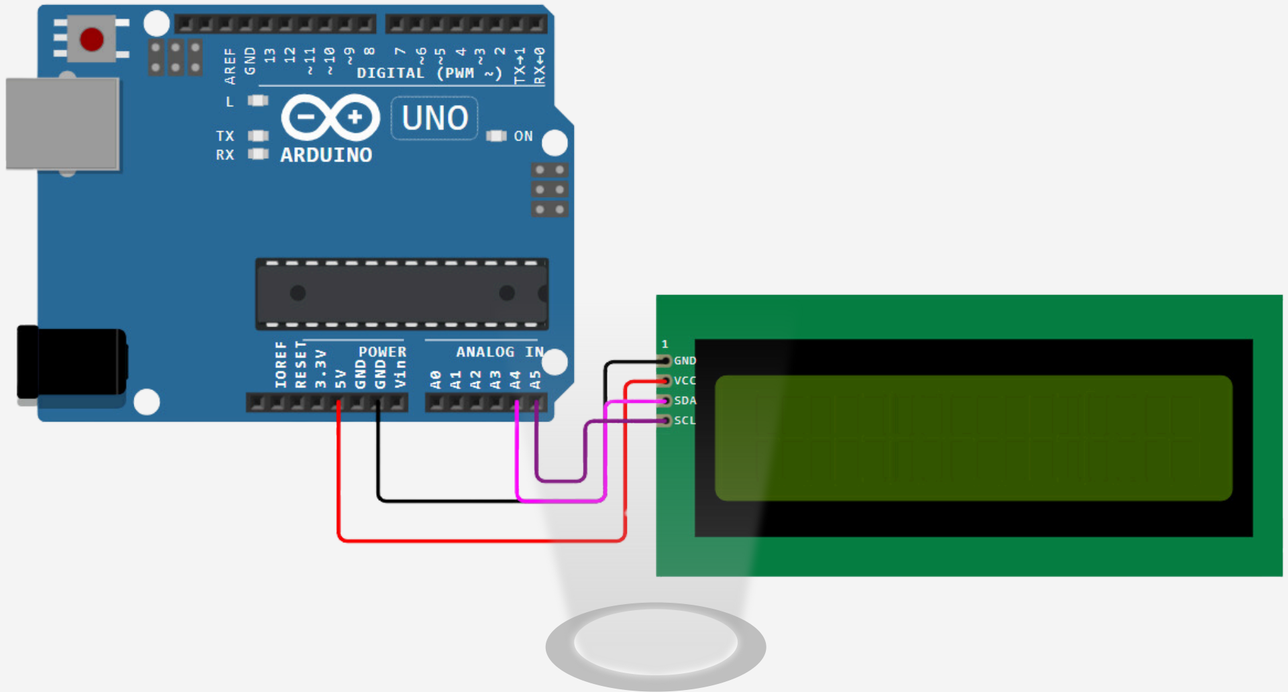
Circuit Connection



© PASSIONBOTS

Step 1

- We will first connect the LCD display to the Arduino Uno.
- VCC pin to 5V of the Arduino and GND pin to GND of the Arduino.
- SDA and SCL pin to A4 and A5 of the Arduino (for I2C communication).



© PASSIONBOTS

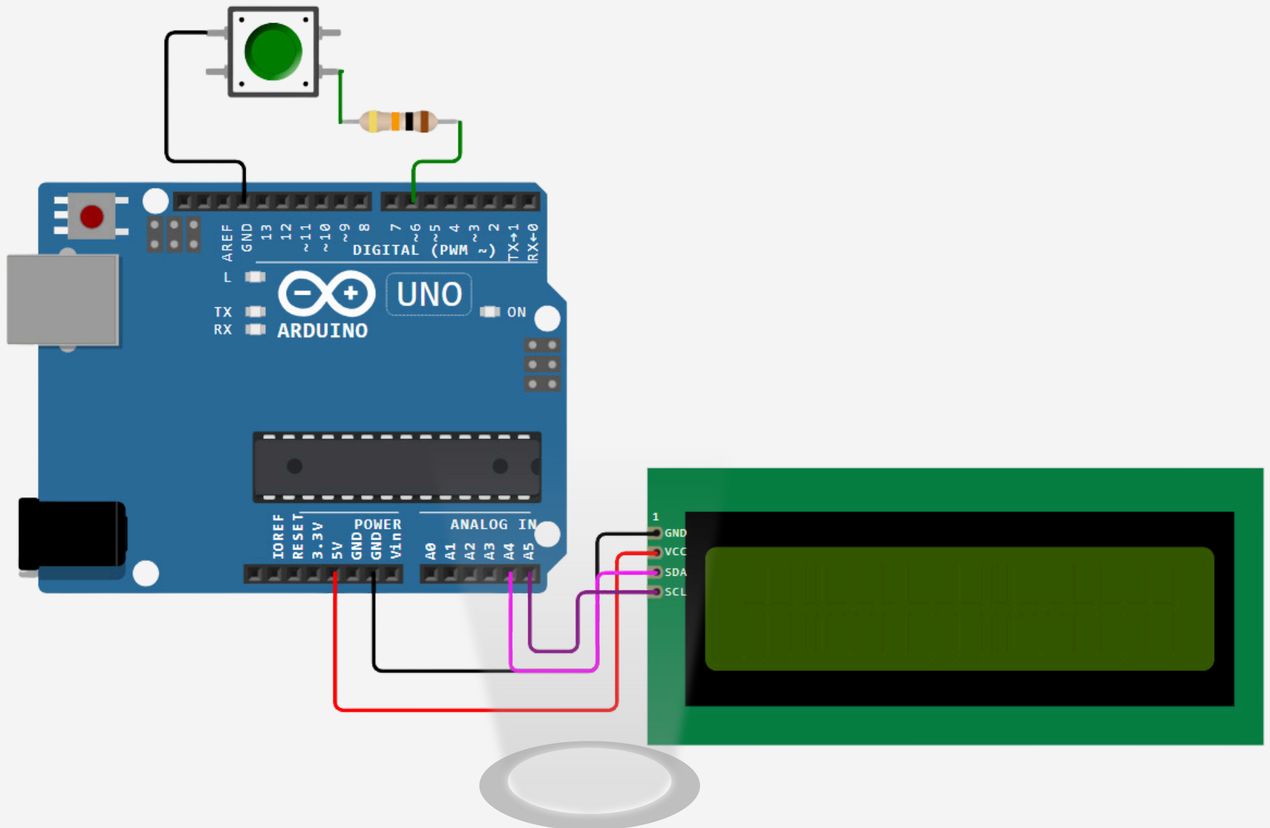
LCD display

- **Description:** A liquid crystal display module that can display characters and graphics.
- **Function:** Provides visual feedback or information to the user.
- **Usage:** In the "Sound the Alarm" project, it can be used to display status messages, sensor readings, or alarm activation alerts, enhancing the user interface and interaction with the system.

© PASSIONBOTS

Step 2

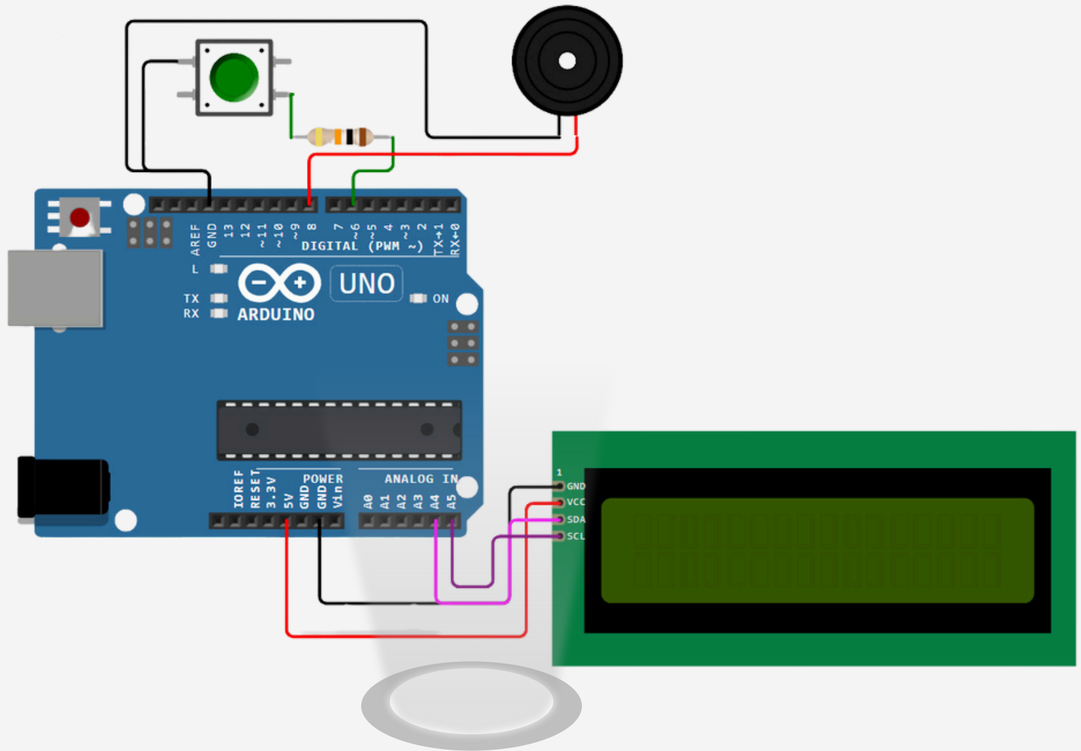
- Connect one terminal of the button to the digital pin 6 of the Arduino Uno.
- Connect other terminal of the button to the ground (GND) of the Arduino.



© PASSIONBOTS

Step 3

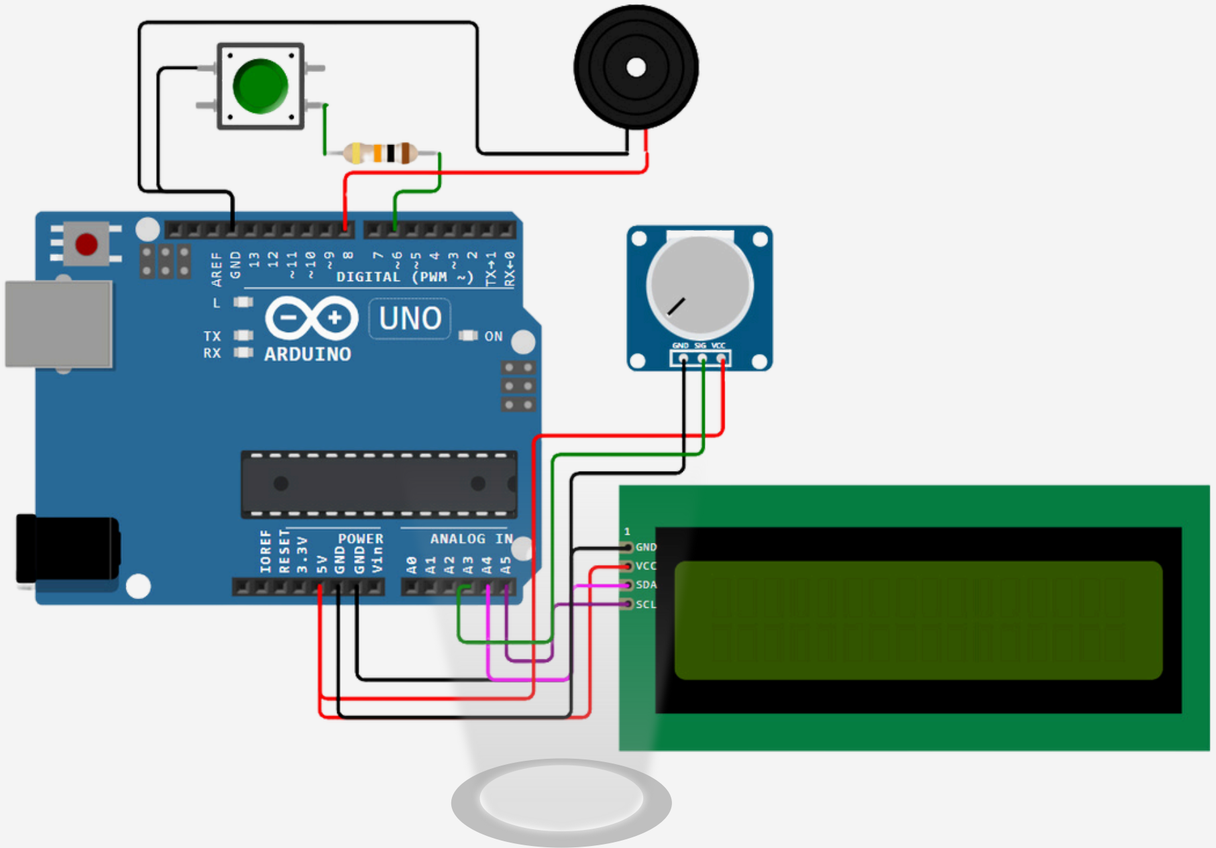
- Connect Positive (longer) leg to digital pin 8 of the Arduino.
- Connect Negative (shorter) leg to GND of the Arduino.



© PASSIONBOTS

Step 4

- Connect positive pin to 5V and negative pin to GND of the Arduino Uno.
- Then connect center to analog pin A3 of the Arduino.



© PASSIONBOTS

Step 5

- Now, we begin with the coding part of our project.
- Include Wire.h and LiquidCrystal_I2C.h libraries.
- Define pin connections: buttonSocket, buzzerSocket, dialSocket.
- Initialize variables: Time, timerStarted, buttonPressed, timerCompleted.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

//Change here if you're using a different socket
#define buttonSocket 6
#define buzzerSocket 8
#define dialSocket A3

int Time;
bool timerStarted = false;
bool buttonPressed = false;
bool timerCompleted = false;

LiquidCrystal_I2C lcd(0x27, 16, 2);
```

© PASSIONBOTS

Step 6

- Now, In void setup, we need to initialize the INPUT and OUTPUT.
- Initialize the LCD display with 16 columns and 2 rows.
- Set the button pin as an input.
- Set the buzzer pin as an output.

```
void setup()
{
  lcd.begin(16, 2);
  pinMode(buttonSocket, INPUT);
  pinMode(buzzerSocket, OUTPUT);
}
```

© PASSIONBOTS

Step 7

- First check if the timer hasn't started or completed.
- Display the dial value on the LCD and wait for the button press.
- When the button is pressed, it starts the timer with the current dial value.

```
void loop()
{
  if (!timerStarted && !timerCompleted) {
    digitalWrite(buzzerSocket, LOW);
    lcd.setCursor(0, 0);
    lcd.print("Timer Amount:   ");
    lcd.setCursor(0, 1);
    lcd.print("       ");
    lcd.setCursor(0, 1);
    lcd.print(map(analogRead(dialSocket), 0, 1024, 0, 3600));
    delay(100);
    if (digitalRead(buttonSocket) && !buttonPressed) {
      buttonPressed = true;
      timerStarted = true;
      Time = (map(analogRead(dialSocket), 0, 1024, 0, 3600));
    }
  }
```

© PASSIONBOTS

Step 8

- Now the if statement first checks if the timer has started.
- If the timer has started, it counts down the time and displays "Time remaining" on the LCD.
- When the time reaches 0, it clears the LCD, turns on the buzzer, and displays "Time is up!" on the LCD.

```
if (timerStarted) {
  while (Time > 0) {
    lcd.setCursor(0, 0);
    lcd.print("Time remaining:");
    lcd.setCursor(0, 1);
    lcd.print("       ");
    lcd.setCursor(0, 1);
    lcd.print(Time);
    delay(1000);
    Time = Time - 1;
  }
  lcd.clear();
  digitalWrite(buzzerSocket, HIGH);
  lcd.setCursor(0, 0);
  lcd.print("Time is up!");
  lcd.setCursor(0, 1);
  lcd.print("       ");
  timerCompleted = true;
}
```

© PASSIONBOTS

Step 9

- Then we will include another if statement that checks if the button is pressed and the timer has completed.
- If both conditions are true, it turns off the buzzer.
- It also resets the button press state and timer completion state to false.

```
  if (digitalRead(buttonSocket) && timerCompleted) {
    digitalWrite(buzzerSocket, LOW);
    buttonPressed = false;
    timerCompleted = false;
  }
}
```

© PASSIONBOTS

Final OutPut

Link	**Output**

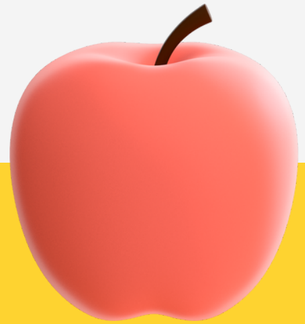
© PASSIONBOTS

Congratulations

Hurray!

We have completed the

Tick Tock Tech Project



© PASSIONBOTS

Learning outcomes:

- Students understand how to seamlessly integrate components into a cohesive alarm system.
- Enhancing problem-solving and project management skills.
- Gain experience in designing user-friendly interfaces by providing both auditory and visual feedback, fostering creativity and improving user experience.

© PASSIONBOTS

Notes

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

© PASSIONBOTS

Thank you!

Do you have any questions for me?

© PASSIONBOTS