Review

Lesson 6

Topics

1. Led
2. Led Button Controlled blinks
3. Led Dimmer with Modes
4. Sounds and Light Alarm
5. Smart Doorbell with Led & Buzzer
6. Automatic doorbell

HW Points

Those who did:

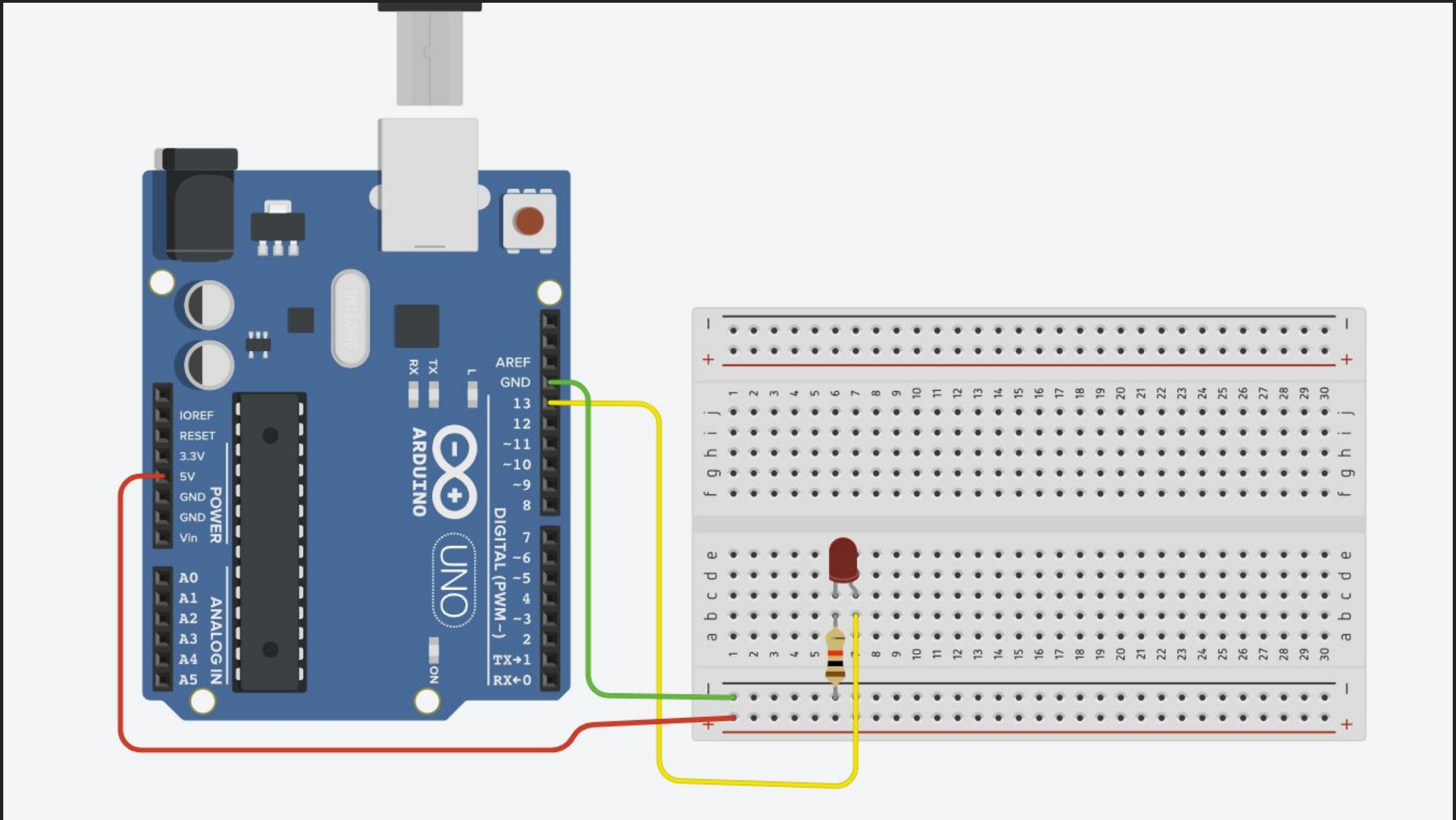
submit on time get **+12 points**

not and will by friday midnight will get **+6 points**

did not **-12 points**

Check your githubs usernames again

L1: LED Blink and Button Control



Step-by-step

`/* … */` — comment block: notes for humans; ignored by the compiler.

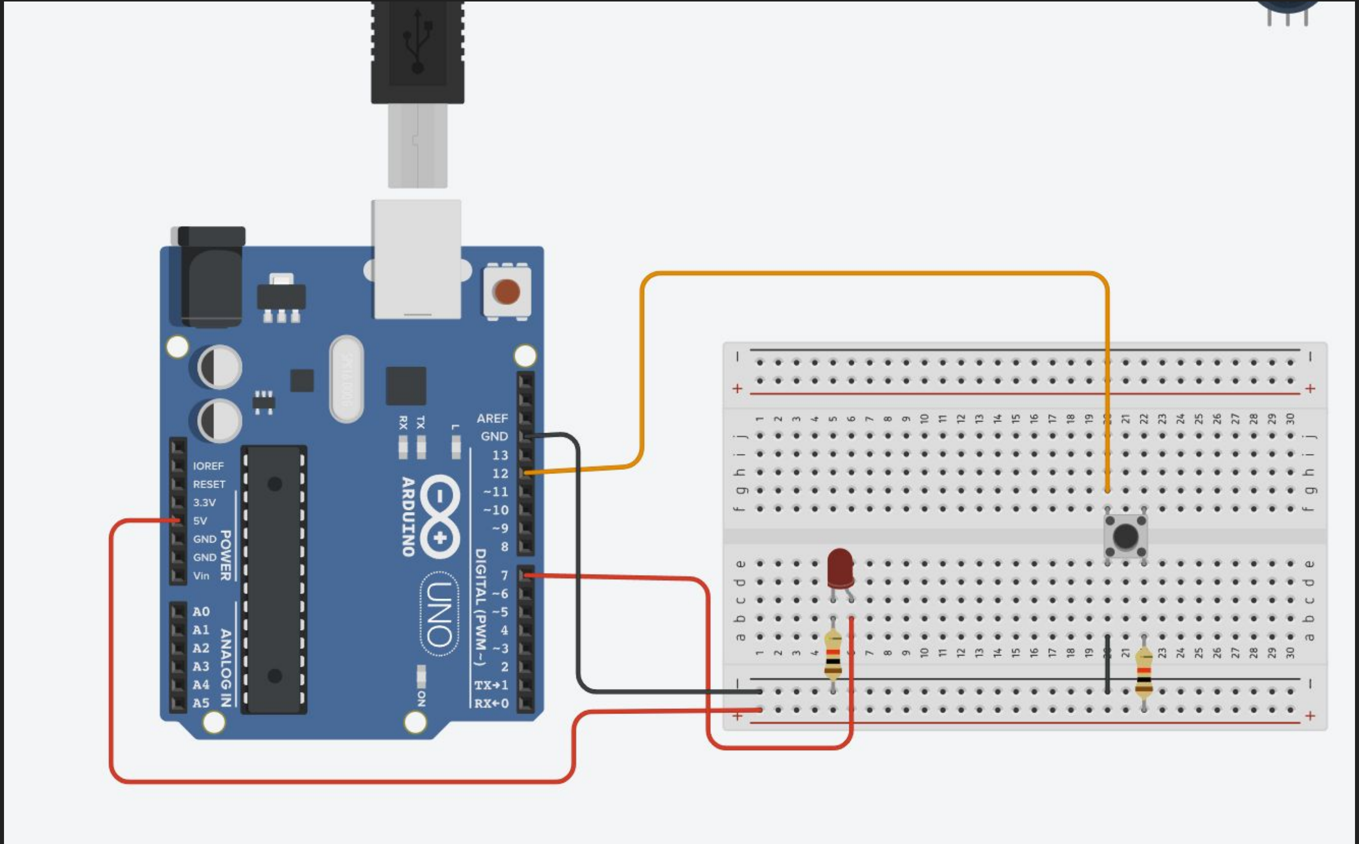
`void setup() { … }` — runs **once** after power-up or reset.

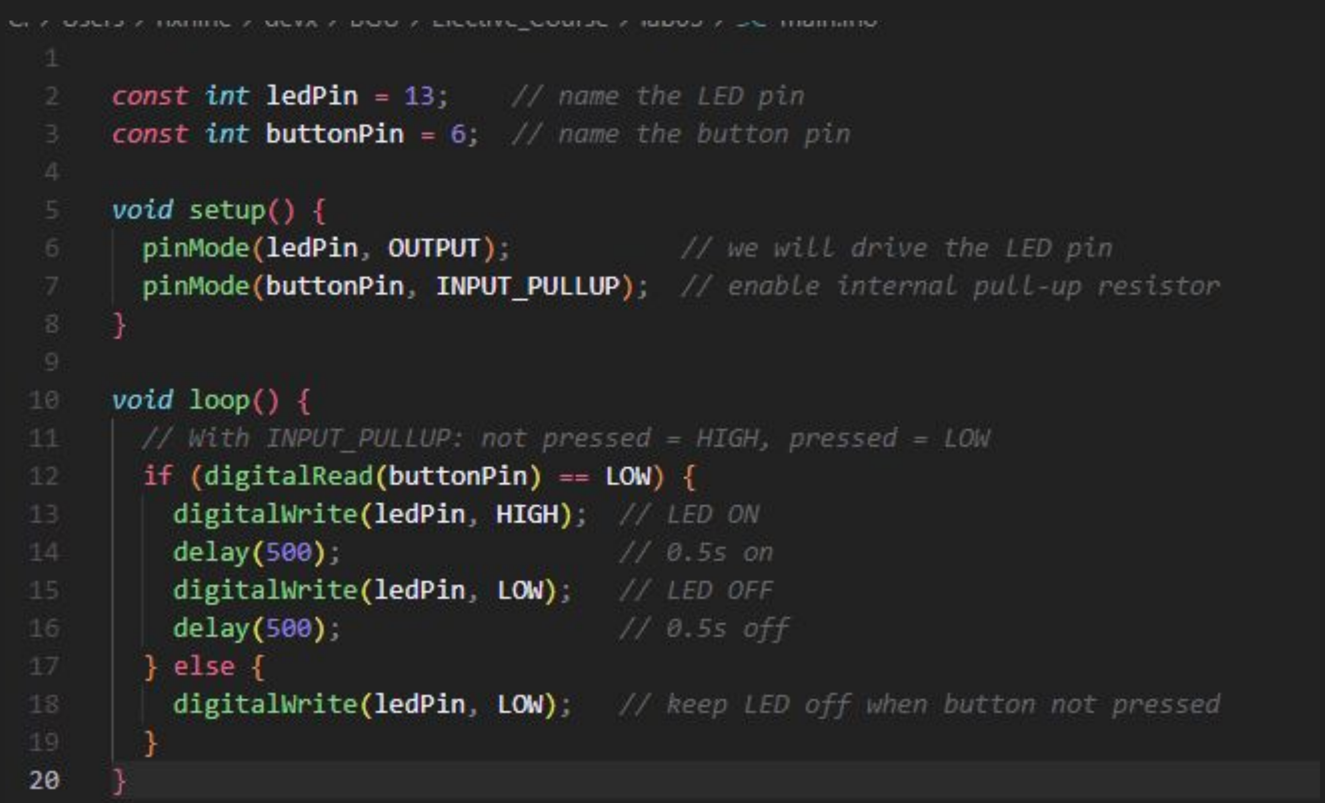
- `pinMode(13, OUTPUT);` — tells Arduino you'll **drive** pin 13 (not read it).

`void loop() { … }` — runs **forever**.

- `digitalWrite(13, HIGH);` — sets pin 13 to ~5V → current flows through the LED → **ON**.
- `delay(1000);` — pauses the program for 1 second (the LED stays on).
- `digitalWrite(13, LOW);` — sets pin 13 to 0V → **OFF**.
- `delay(1000);` — 1 second off before repeating

Button-controlled blink





```

const int ledPin = 13;    // name the LED pin
const int buttonPin = 6;  // name the button pin

void setup() {
  pinMode(ledPin, OUTPUT);           // we will drive the LED pin
  pinMode(buttonPin, INPUT_PULLUP);  // enable internal pull-up resistor
}

void loop() {
  // With INPUT_PULLUP: not pressed = HIGH, pressed = LOW
  if (digitalRead(buttonPin) == LOW) {
    digitalWrite(ledPin, HIGH);  // LED ON
    delay(500);                  // 0.5s on
    digitalWrite(ledPin, LOW);   // LED OFF
    delay(500);                  // 0.5s off
  } else {
    digitalWrite(ledPin, LOW);   // keep LED off when button not pressed
  }
}
```

Step-by-step

`const int ledPin = 13;` / `const int buttonPin = 6;`

Constants with readable names so you don’t hard-code numbers everywhere.

`void setup()`:

`pinMode(ledPin, OUTPUT);` — we’ll **output** voltage on the LED pin.

`pinMode(buttonPin, INPUT_PULLUP);` — turns on the **internal pull-up resistor** (~20–50 kΩ to +5V).

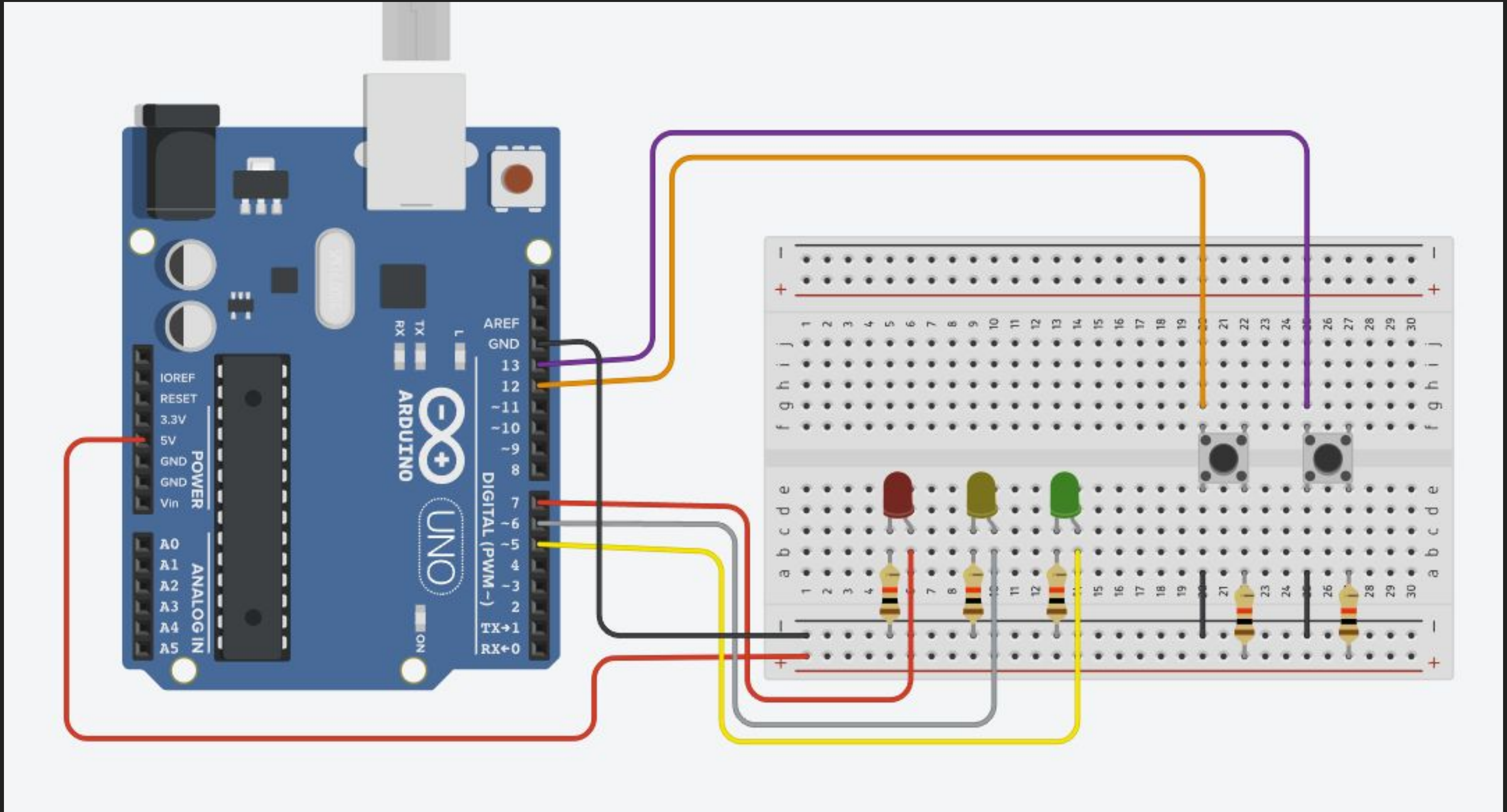
- This makes the input **stable** without an external resistor.
- Logic becomes **inverted**:
  - Button **released** → pin reads **HIGH**.
  - Button **pressed** (pin shorted to GND) → reads **LOW**.

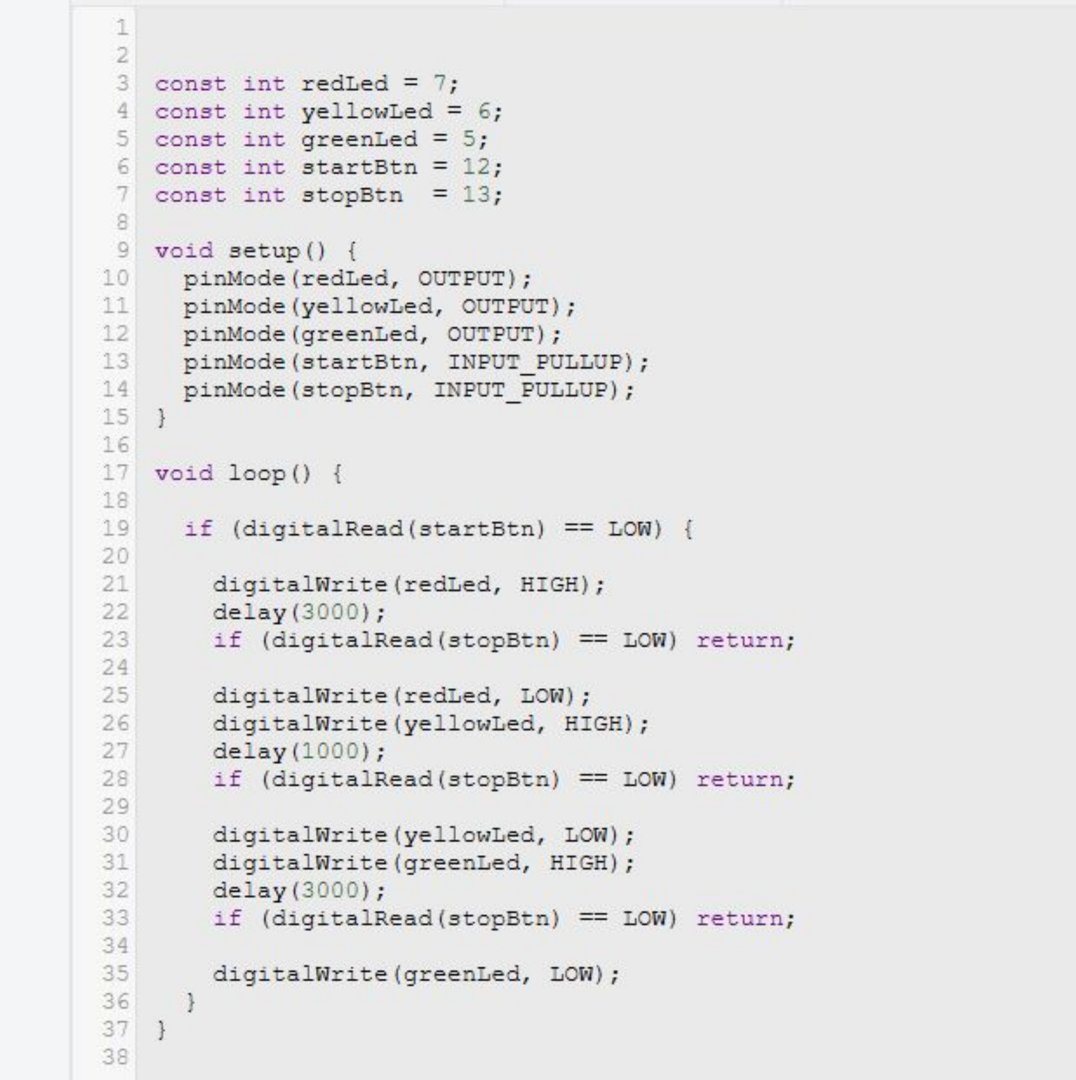
`void loop()`:

`if (digitalRead(buttonPin) == LOW)` — check if the button is **pressed**.

- If pressed: blink the LED (on 500 ms, off 500 ms).
- If not pressed: force LED **LOW** so it stays off.

Traffic Light with Led



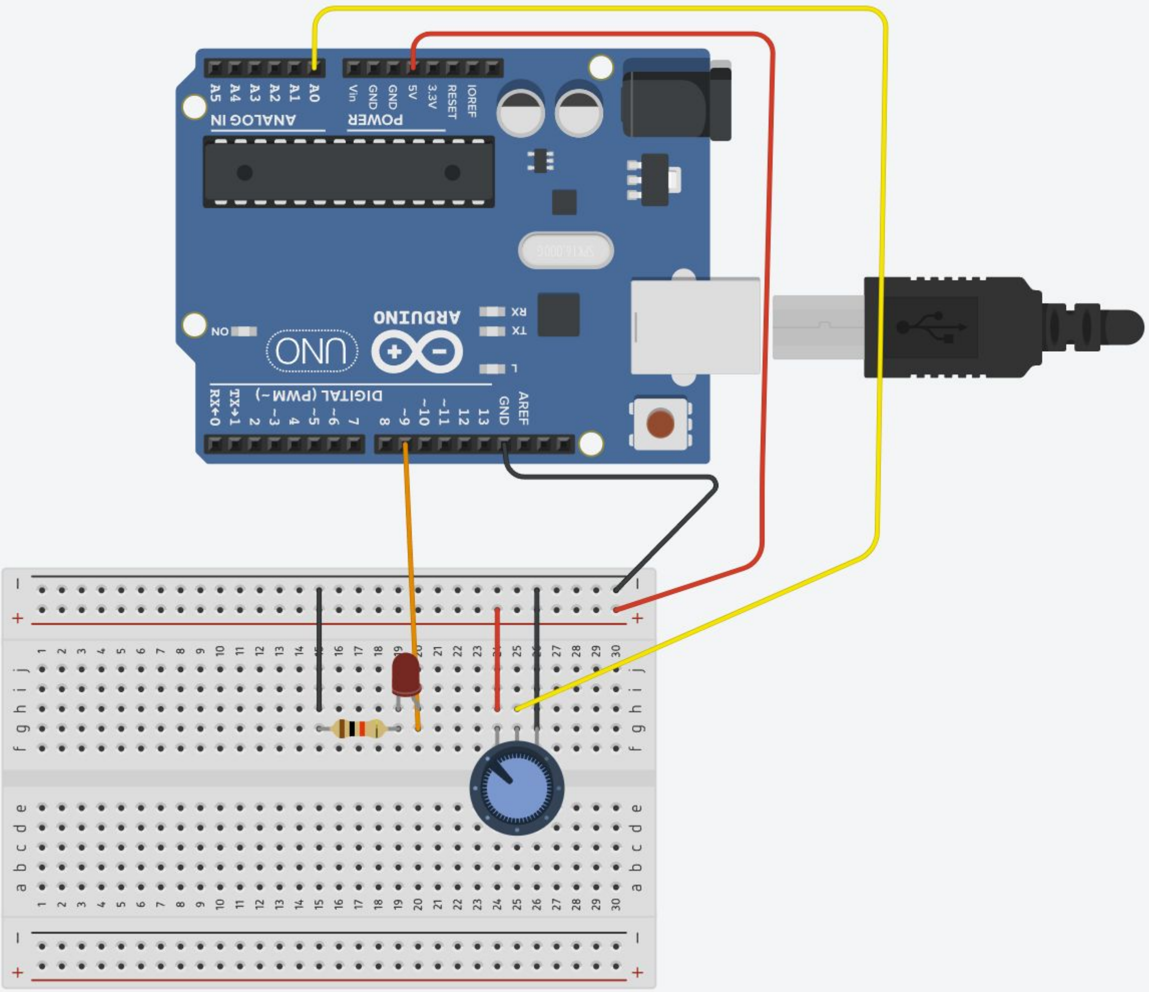


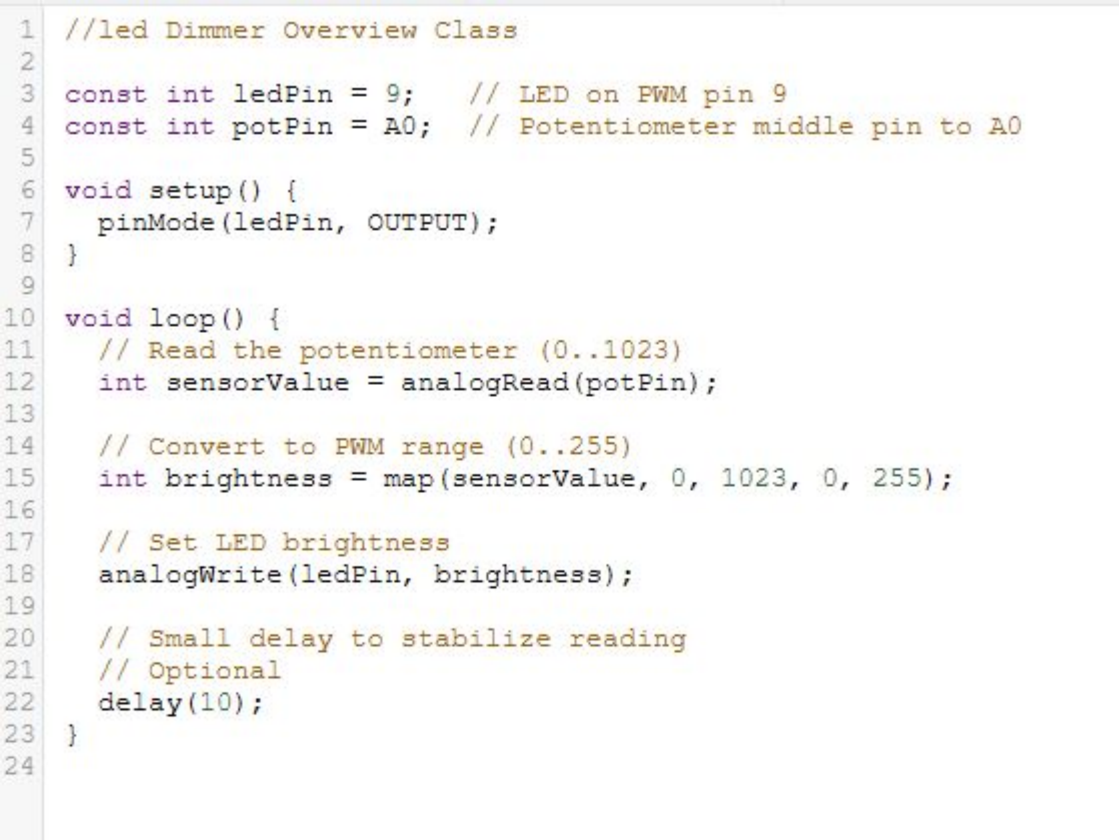
```
const int redLed = 7;
const int yellowLed = 6;
const int greenLed = 5;
const int startBtn = 12;
const int stopBtn  = 13;

void setup() {
  pinMode(redLed, OUTPUT);
  pinMode(yellowLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
  pinMode(startBtn, INPUT_PULLUP);
  pinMode(stopBtn, INPUT_PULLUP);
}

void loop() {

  if (digitalRead(startBtn) == LOW) {

    digitalWrite(redLed, HIGH);
    delay(3000);
    if (digitalRead(stopBtn) == LOW) return;

    digitalWrite(redLed, LOW);
    digitalWrite(yellowLed, HIGH);
    delay(1000);
    if (digitalRead(stopBtn) == LOW) return;

    digitalWrite(yellowLed, LOW);
    digitalWrite(greenLed, HIGH);
    delay(3000);
    if (digitalRead(stopBtn) == LOW) return;

    digitalWrite(greenLed, LOW);
  }
}
```

LED Dimmer with Modes

Цель:

Сделать управляемую лампу: поворот потенциометра меняет яркость LED (через PWM), а кнопка переключает режимы работы.


ANALOG IN
A0
A1
A2
A3
A4
A5
POWER
Vin
GND
GND
5V
3.3V
RESET
IOREF
ARDUINO
UNO
ON
RX
TX
L
DIGITAL (PWM~)
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
+
−
a
b
c
d
e
f
g
h
i
j
1
5
10
15
20
25
30



```
//led Dimmer Overview Class

const int ledPin = 9;   // LED on PWM pin 9
const int potPin = A0;  // Potentiometer middle pin to A0

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // Read the potentiometer (0..1023)
  int sensorValue = analogRead(potPin);

  // Convert to PWM range (0..255)
  int brightness = map(sensorValue, 0, 1023, 0, 255);

  // Set LED brightness
  analogWrite(ledPin, brightness);

  // Small delay to stabilize reading
  // Optional
  delay(10);
}

```

Step-by-step

```
int sensorValue = analogRead(A0);   // 0..1023

int brightness  = sensorValue / 4;  // 0..255 (integer division)

analogWrite(9, brightness);
```

Module Sample Questions