LED Dimmer with Modes

Lesson 2

LED Dimmer with Modes

Цель:

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

Режим 1 (ручной диммер):

Положение потенциометра определяет яркость LED (0–255) в реальном времени.

Режим 2 (автопульс):

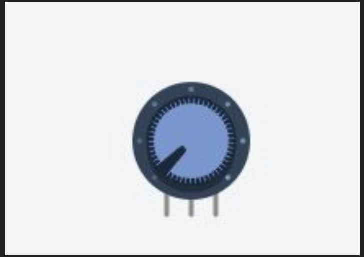
LED мягко «дышит» (плавно увеличивает и уменьшает яркость).

Переключение режимов:

Короткое нажатие кнопки D6 переключает Режим 1 ↔ Режим 2.

Используйте INPUT_PULLUP: не нажата = HIGH, нажата = LOW.

Потенциометр/ Potentiometer



Inside the Potentiometer

Inside is a resistive track shaped like a ring or arc.

The two outer terminals connect to the ends of this resistive track.

The middle terminal (the wiper) is a sliding contact that moves along the track as you turn the knob.

If you connect the two outer pins to 5 V and 0 V (GND), there’s a continuous 5 V drop along the track.

Think of the track as one long resistor split into two variable pieces:

[5 V] ----[R1]----(wiper)----[R2]---- [GND]

R1 = resistance from 5 V to the wiper.

R2 = resistance from the wiper to GND.

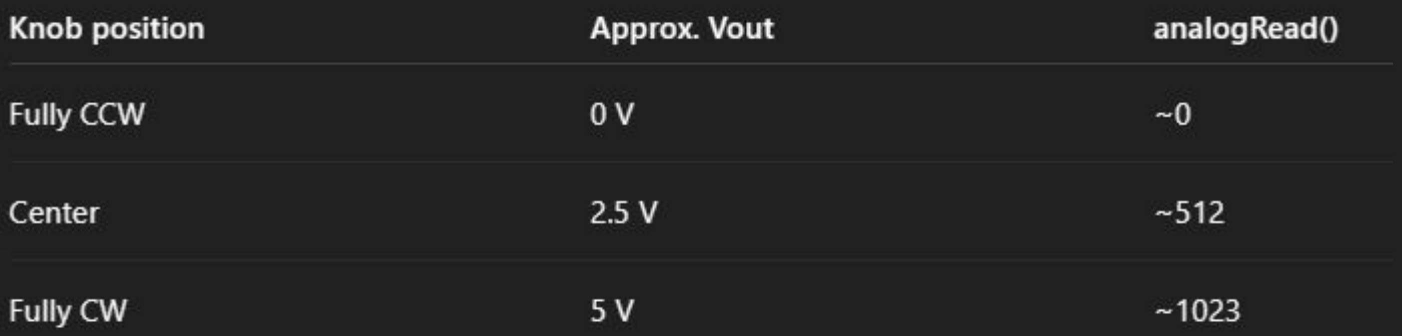
Total resistance = R1 + R2 (fixed by the potentiometer’s rating, e.g., 10 kΩ).

The voltage at the wiper (Vout) is given by the divider formula:

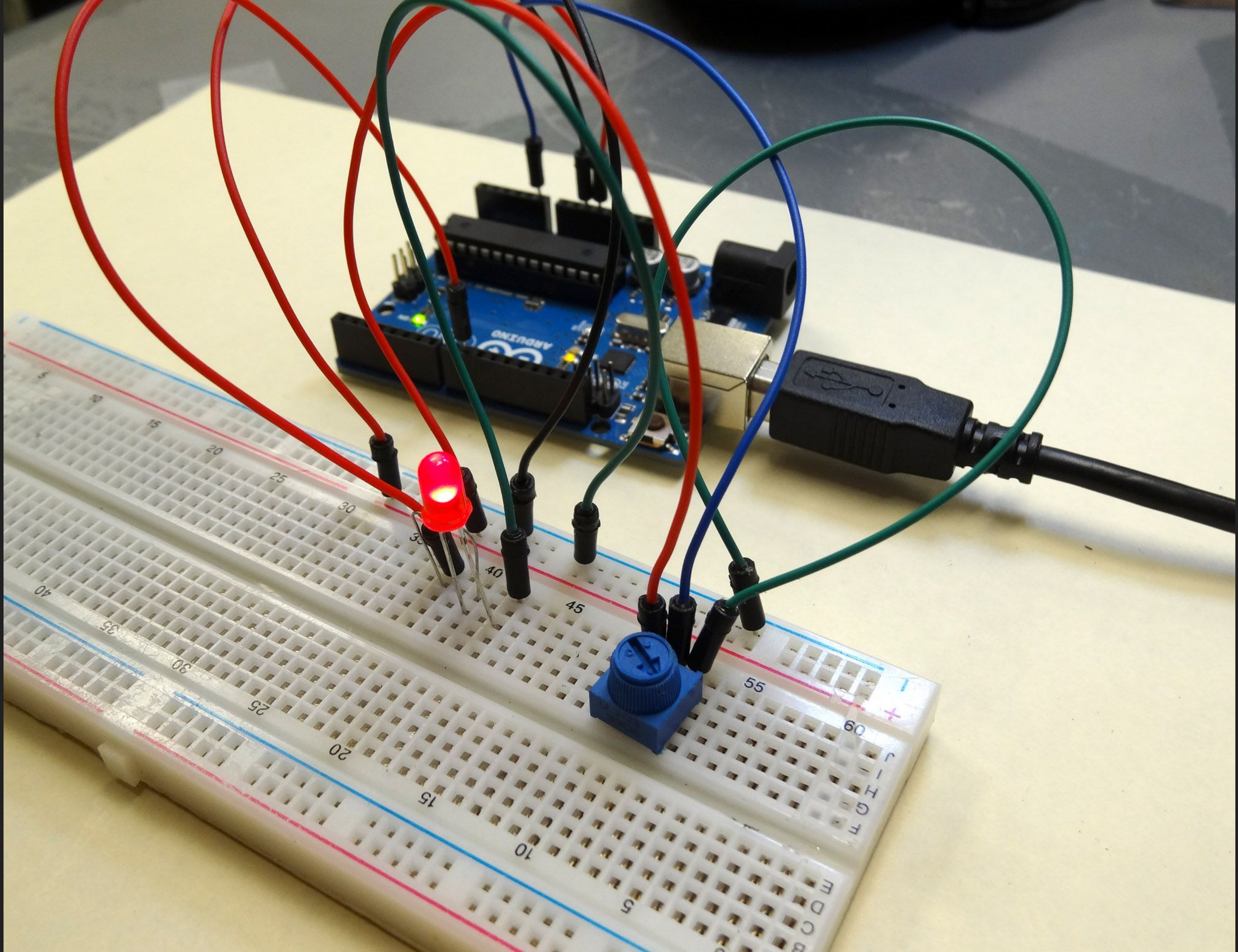
$$Vout=5V\times(R2/(R2R2))$$

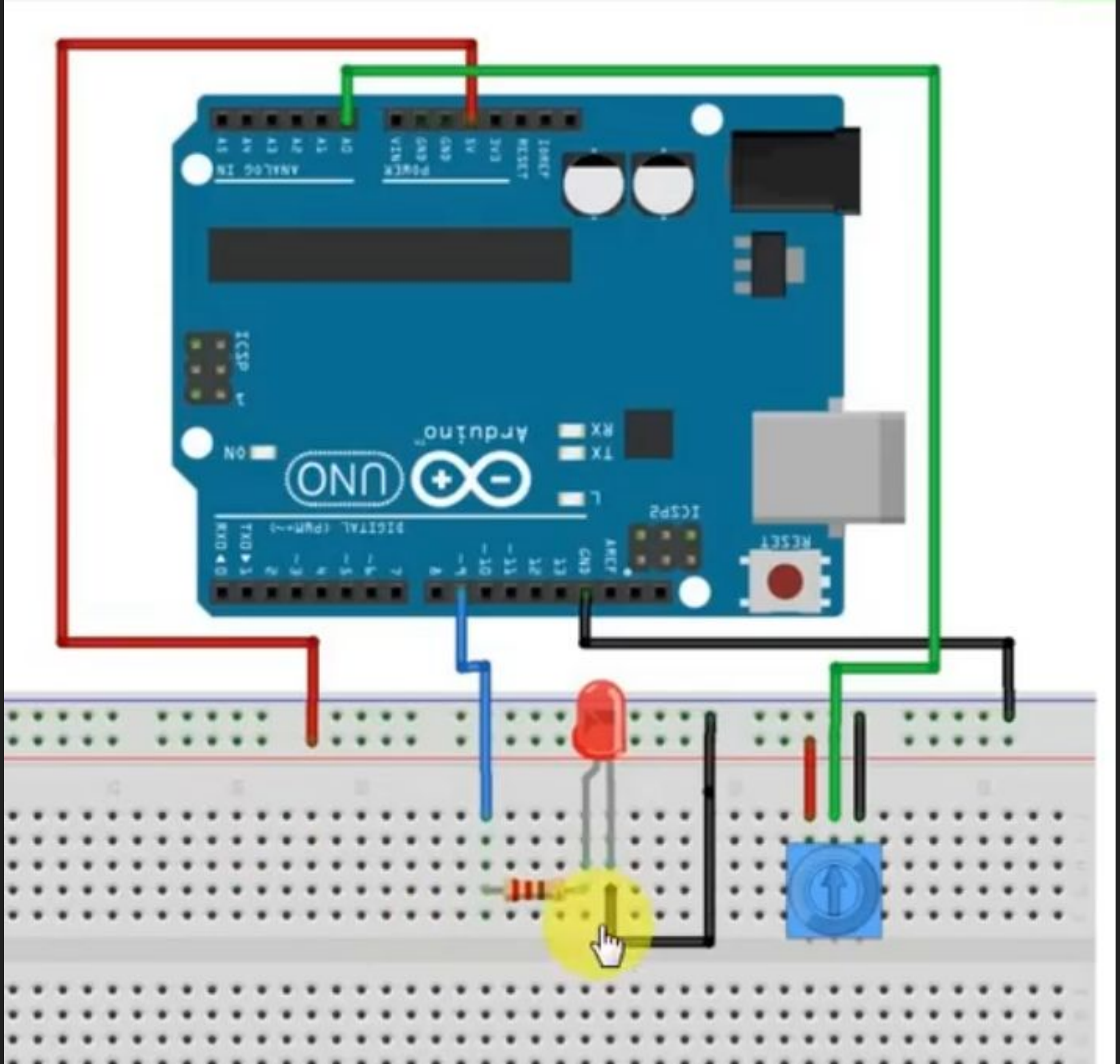
When you turn the knob:

- Rotate toward the 5 V side → R1 decreases, R2 increases → Vout rises.
- Rotate toward the GND side → R1 increases, R2 decreases → Vout falls.



Knob position	Approx. Vout	analogRead()
Fully CCW	0 V	~0
Center	2.5 V	~512
Fully CW	5 V	~1023


ARDUINO
5
10
15
20
25
30
35
40
45
55
60
40
35
30
25
20
15
10
5
J
I
H
G
F
E
D
C
B
-
+


ANALOG IN
A0
A1
A2
A3
A4
A5
POWER
IOREF
RESET
3V3
5V
GND
GND
VIN
ICSP
ON
UNO
Arduino
RX
TX
L
ICSP2
RESET
DIGITAL (PWM~)
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX▶1
RX◀0

Overview

An LED is either ON or OFF electrically.

To make it appear dimmer, Arduino uses PWM (Pulse-Width Modulation):

- It switches the LED on and off hundreds of times per second.
- The proportion of ON time (the duty cycle) sets the average voltage the LED “sees”.
- A higher duty cycle means brighter light.

`analogWrite(pin, value)` lets you set this duty cycle with a value between **0** and **255**

1023 from analogRead()

Arduino Reads analog 0 to 1023

We write Analog in the range of 0 to 255

The arudino has a 10-bit Analog-to-Digital Converter (ADC).

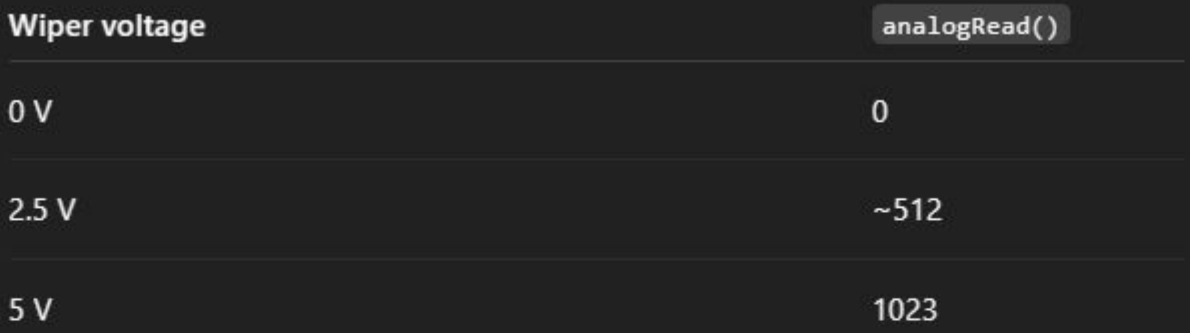
The potentiometer is wired between 5 V and GND; its center (wiper) outputs a voltage between 0 V and 5 V.

The ADC measures this and converts it to a number from 0 to 1023:

0 → about 0 V

1023 → about 5 V

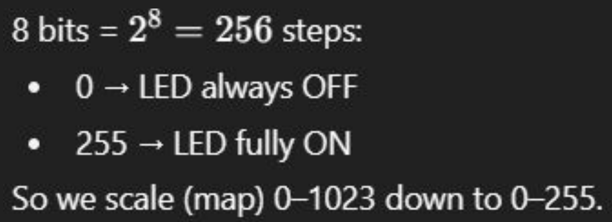
Example of Read



Wiper voltage	`analogRead()`
0 V	0
2.5 V	~512
5 V	1023

Why We Map to 255

analogWrite() on Arduino UNO uses 8-bit PWM (0–255).



8 bits = $2^8 = 256$ steps:

- 0 → LED always OFF
- 255 → LED fully ON

So we scale (map) 0–1023 down to 0–255.

```
int reading = analogRead(A0);           // 0..1023
int brightness = map(reading, 0, 1023, 0, 255);
analogWrite(9, brightness);             // PWM 0..255
```

Why We Use Pin A0 for the Potentiometer

Pins A0…A5 on Arduino UNO are analog input pins connected to the ADC.

They can measure voltages between 0 V and 5 V and return a precise 0–1023 reading.

Ordinary digital pins (D0–D13) can only read HIGH/LOW, not a continuous range of voltages.

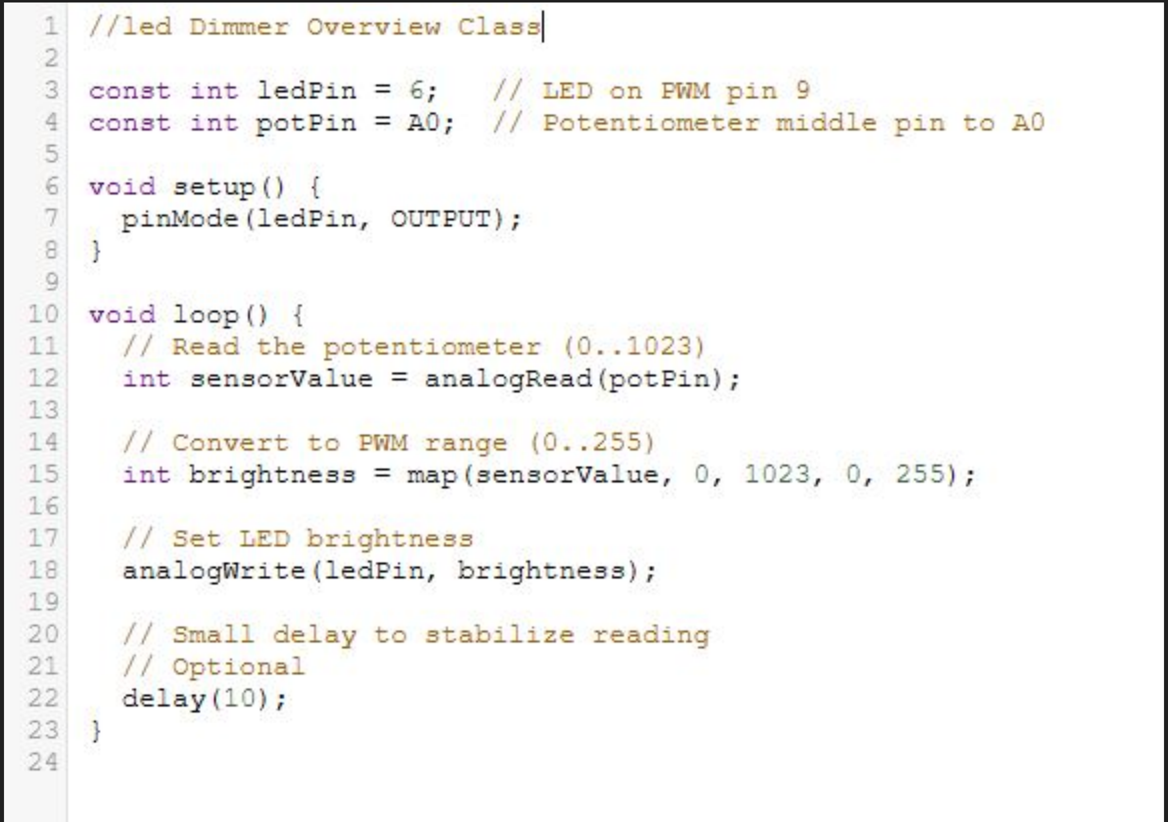
Overview

-Potentiometer divides 5 V into a variable output voltage.

-Analog input (A0) reads that voltage as a number 0–1023.

-Code maps 0–1023 → 0–255 to match the PWM range.

-analogWrite() produces PWM with that duty cycle, making the LED brighter or dimmer.



```
//led Dimmer Overview Class

const int ledPin = 6;   // 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);
}

```

Attempt 2

The Magic Lamp and the Knob

You know how you turn a volume knob on a radio to make the sound louder or softer?

A potentiometer is just a special knob for electricity.

- One side of the knob is full power (5 volts) – like a full glass of juice.
- The other side is zero power (0 volts) – like an empty glass.
- The middle pin is the magic straw that lets you take just the right amount of juice out.

When you twist the knob, you choose how much voltage (juice) goes to the Arduino.

How the Arduino Reads the Knob (Why 1023)

The Arduino has tiny “math eyes” called an ADC (analog-to-digital converter).

It looks at the juice coming from the knob and gives a number from 0 to 1023:

0 = no juice (knob all the way down)

1023 = full juice (knob all the way up)

Why 1023?

Because the Arduino’s “math eyes” have 10 little switches inside.

Ten switches can count from 0 to 1023. (That’s 2^{10} numbers!)

How the Arduino Makes the LED Shine (Why 255)

But our lamp isn’t just ON or OFF.

It can glow softly or brightly.

The Arduino can blink super-fast so our eyes see smooth light.

This super-fast blinking is called PWM – think of it as tiny electric flickers.

The Arduino can blink in 256 steps:

0 (always off) to 255 (always on).

So when we send a number from 0 to 255, we’re really telling it how long to stay on during each tiny blink.

How Everything Connects (Why we use pin A0)

The middle pin of the knob (the “magic straw”) goes to A0 on the Arduino.

“A” means Analog – it can feel all the gentle changes, not just ON or OFF.

Digital pins (D2, D3, etc.) can only feel simple ON or OFF like a light switch, so they wouldn’t work.

The Whole Adventure

1. You twist the knob (potentiometer).

2. Arduino feels how much “juice” is coming in on A0 and says:

“Hmm, that’s about 512 out of 1023.”

3. Arduino turns that into a number 0 to 255.

4. It blinks the LED super-fast for just the right fraction of time, so your eyes see a steady, gentle glow.

Class Task/ Homework

“Sound-and-Light Alarm” – Buzzer + LED Alert

Arm/Disarm logic: When the Arm button is pressed, toggle between armed and disarmed mode.

Use an LED blink pattern (e.g., one short flash) to confirm each state change.

Trigger alarm: Only when armed, pressing the Sensor button makes the buzzer beep and the LED flash.

Adjustable duration: The potentiometer controls how long the alarm keeps flashing and beeping (for example 1–10 seconds).

Goal of the Project

Собрать простую охранную сигнализацию, где:

: Кнопка «Вкл/Выкл» ставит систему на охрану или снимает с охраны.

: Потенциометр задаёт время срабатывания (как долго будут мигать свет и звучать зуммер).

: Когда система на охране, нажатие второй кнопки (датчик) запускает:

Красный светодиод с быстрым миганием

Зуммер с прерывистым сигналом

Требования к программе

Логика охраны

Нажатие кнопки «Вкл/Выкл» переключает систему в режим охраны или снята с охраны.

Для подтверждения можно сделать короткую вспышку светодиода.

Срабатывание тревоги

Только в режиме охраны нажатие кнопки «Датчик» включает мигание светодиода и прерывистый звук зуммера.

Регулировка времени срабатывания

Потенциометр определяет, сколько секунд работает сигнализация (например от 1 до 10 секунд).