Photoresistor (LDR)

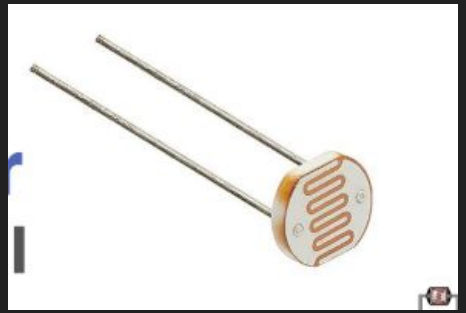
Lesson 8

What is a Photoresistor (LDR)?

- A photoresistor or LDR (Light Dependent Resistor) is a light sensor.
- It changes its resistance depending on the amount of light that falls on it.

More light → lower resistance.

Less light → higher resistance.



How We Use It with Arduino

5V --- [LDR] --- A0 --- [10kΩ resistor] --- GND

The voltage at A0 depends on the LDR’s resistance.

When it’s bright → resistance low → voltage on A0 rises.

When it’s dark → resistance high → voltage on A0 drops.

Typical Uses

- Automatic street lights
- Night lamps
- Smart curtains / blinds
- Line-following robots
- Solar panel trackers

Labs > ldr.ino

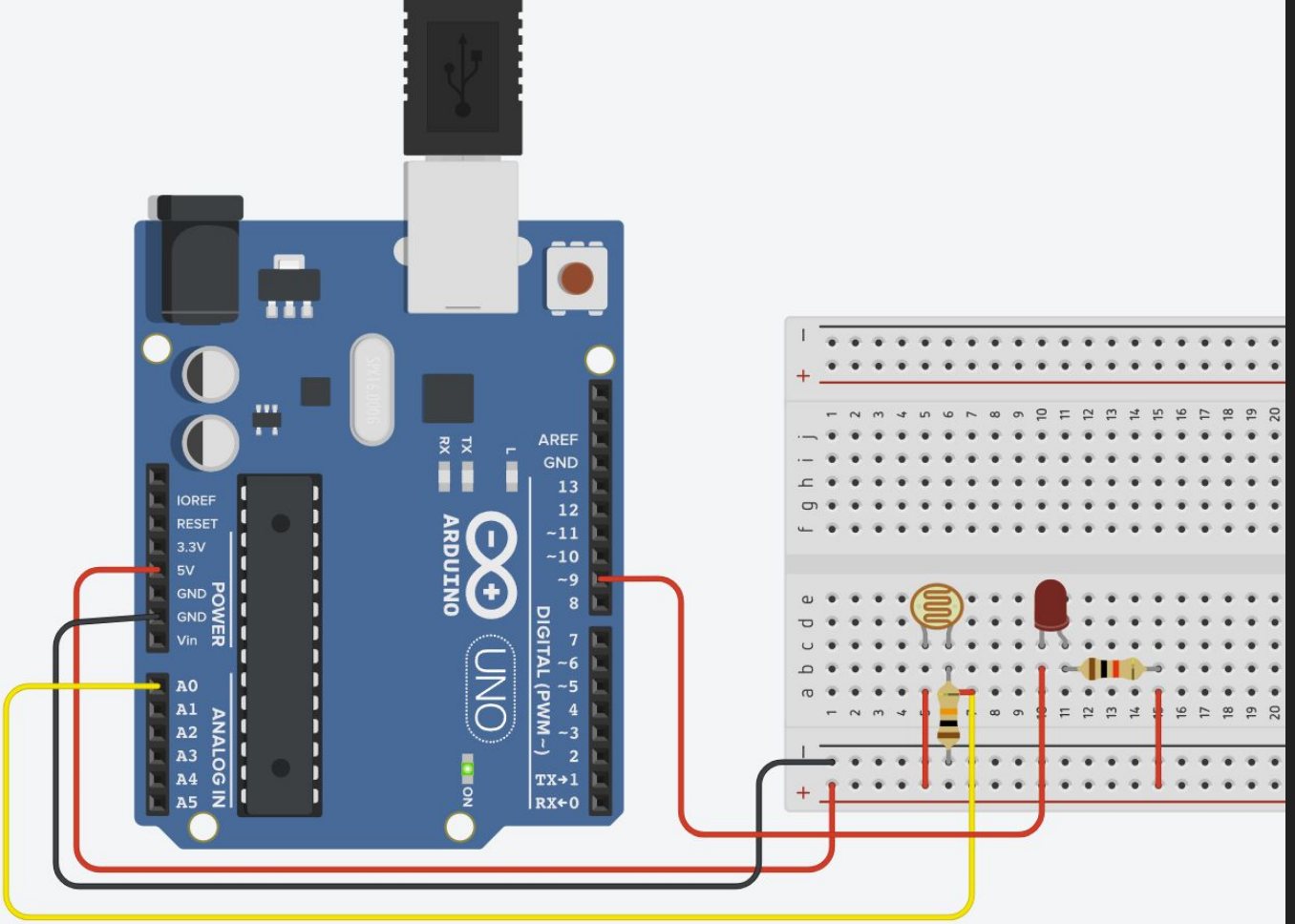
```
const int ldrPin = A0;
const int ledPin = 9;

void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int lightValue = analogRead(ldrPin);
  Serial.println(lightValue);

  if (lightValue < 400) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);

  delay(200);
}
```

Wiring Instructions

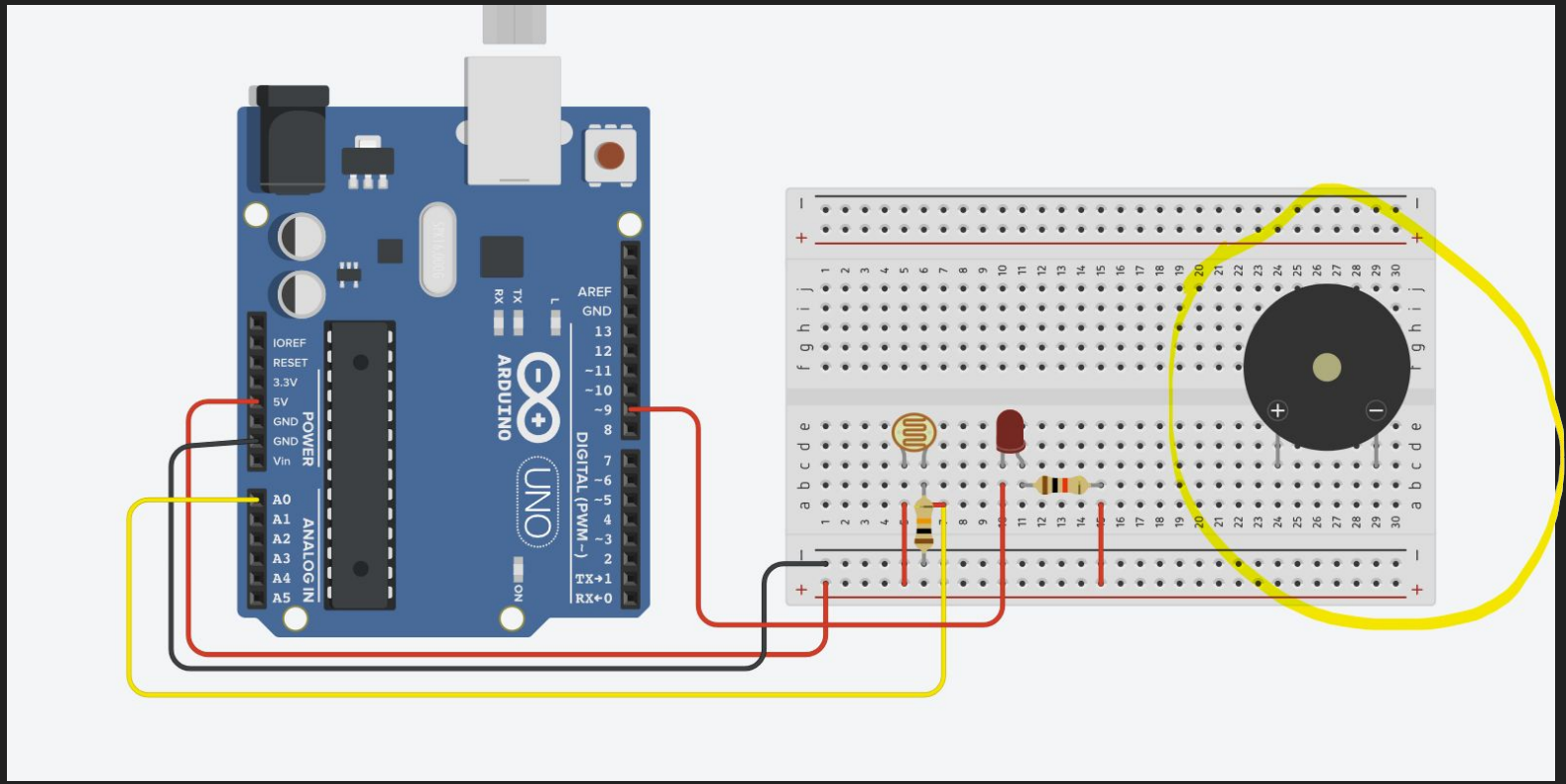
5V --- [LDR] --- A0 --- [10kΩ resistor] --- GND

The LDR and resistor form a voltage divider, and the ratio between them determines the voltage Arduino reads.

With a 1 kΩ resistor, most of the voltage drop will happen on the LDR — but since 1 kΩ is much smaller than the LDR’s resistance in darkness, the divider becomes unbalanced.


IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
TX
RX
L
ON
ARDUINO
UNO
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)

Lets Add Buzzer



How It Works

The LDR senses the light level (0–1023).

When it’s dark (lightValue < 400):

The LED lights up,

The buzzer makes a sound.

When it’s bright, both turn off.

Labs > ldr.ino

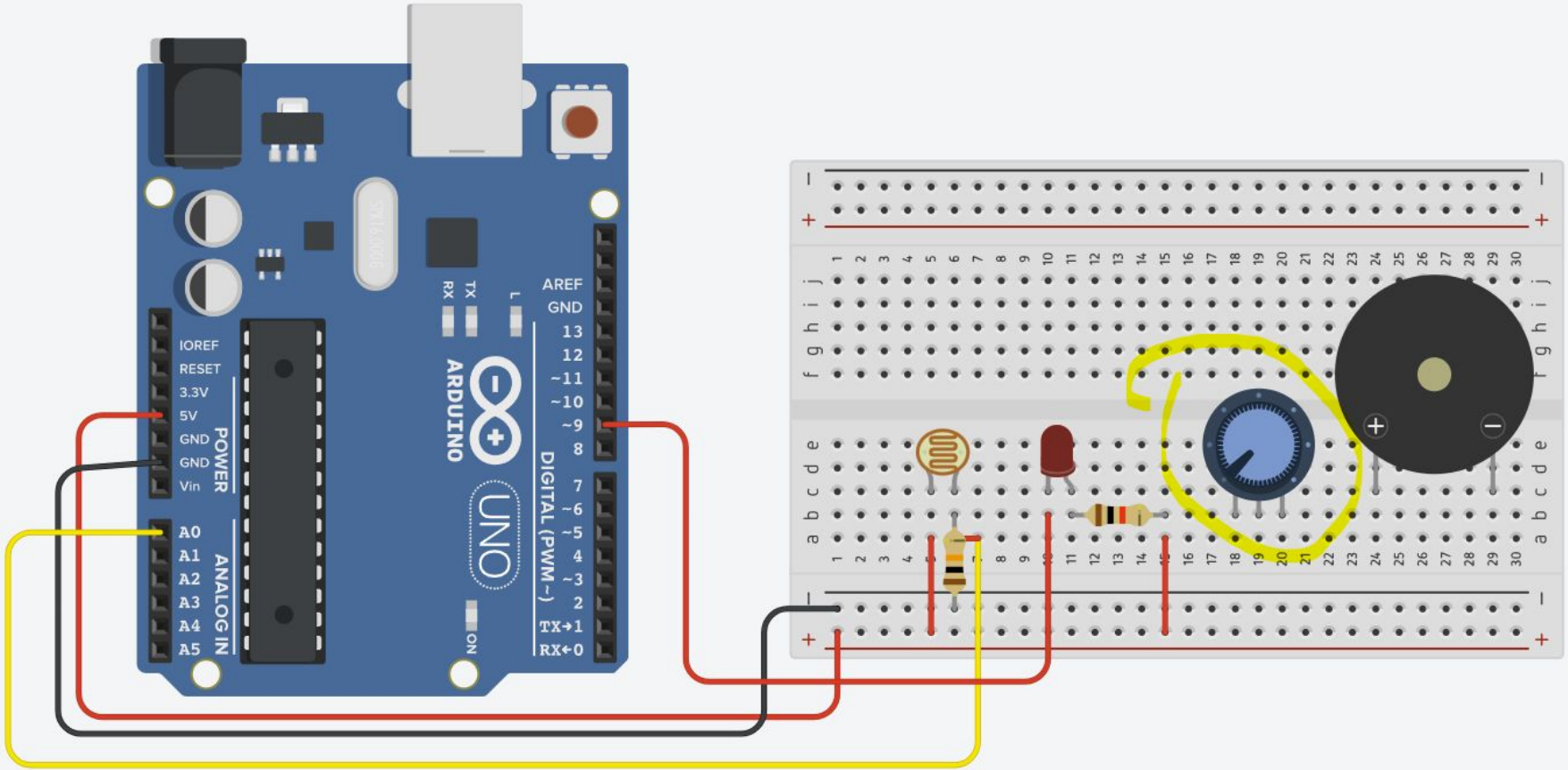
```
// LDR + LED + Buzzer Alarm System

const int ldrPin = A0;
const int buzzerPin = 8;
const int ledPin = 9;

void setup() {
  pinMode(buzzerPin, OUTPUT);
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int lightValue = analogRead(ldrPin);
  Serial.println(lightValue);


  if (lightValue < 400) {

    digitalWrite(ledPin, HIGH);
    digitalWrite(buzzerPin, HIGH);
  } else {

    digitalWrite(ledPin, LOW);
    digitalWrite(buzzerPin, LOW);
  }

  delay(200);
}

```

Lets add: Potentiometer (Adjustable Sensitivity)

What it does

Turn the knob to set the trip point between bright and dark in real time.

No need to re-upload code to change sensitivity.


IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
ARDUINO
UNO
RX
TX
L
ON

Wiring

Potentiometer (3 pins):

One side → 5V

Other side → GND

Middle (wiper) → A1

What Is Hysteresis?

Гистерезис — это когда система не реагирует мгновенно на малейшие изменения, а ждёт, пока разница станет заметной.

Это помогает избежать мигания и частых переключений, когда входной сигнал колеблется около порога срабатывания.

It’s a way to make your system more stable and avoid flickering or rapid switching when the input value (like light level) is close to the trigger point.

Imagine your LDR’s light reading moves around your threshold value (let’s say 400):

Light: 399 → ON

Light: 401 → OFF

Light: 398 → ON

Light: 402 → OFF

If the light keeps changing slightly (like your hand’s shadow moving),

the LED and buzzer will rapidly blink or beep many times per second

That’s because Arduino reacts every time the value crosses 400.

Example: With Hysteresis

Now let’s add a gap, like ±15 around the threshold (called hysteresis).

Turn ON when light < 385

Turn OFF when light > 415

So when light hovers near 400, nothing changes — the system stays calm.

It only switches when the difference is big enough.

```
const int ldrPin     = A0;
const int potPin     = A1;
const int ledPin     = 9;
const int buzzerPin  = 8;


const int hysteresis = 15;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  // Read sensors
  int lightValue = analogRead(ldrPin);
  int potValue   = analogRead(potPin);


  int threshold  = potValue;
```

Дополнительное задание

1. New Module: Mute/ Silence Button

What is does:

A pushbutton toggles buzzer mute without affecting the led alarm.