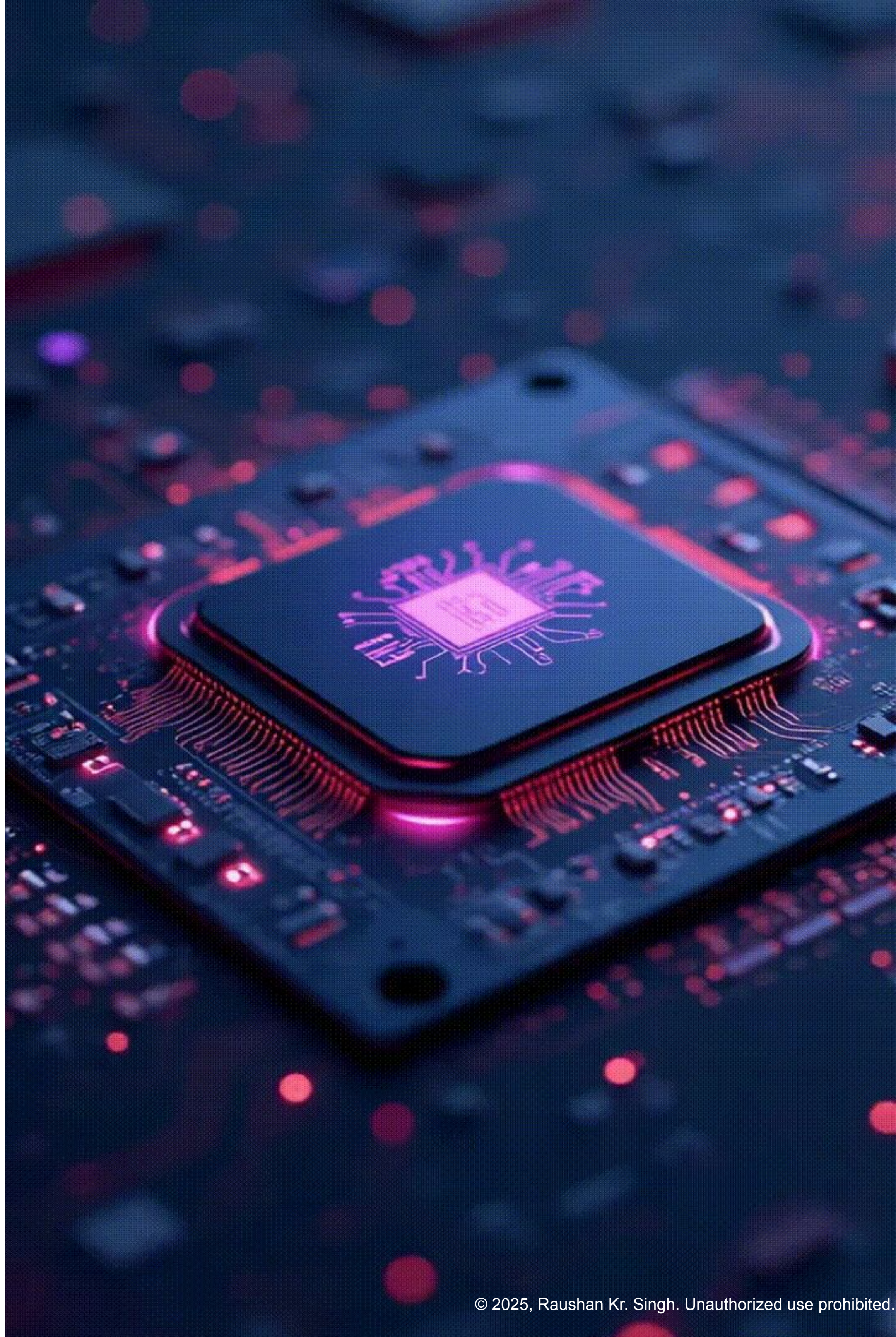
2.
TinyML
Hardware & Wokwi

Raushan Kr. Singh

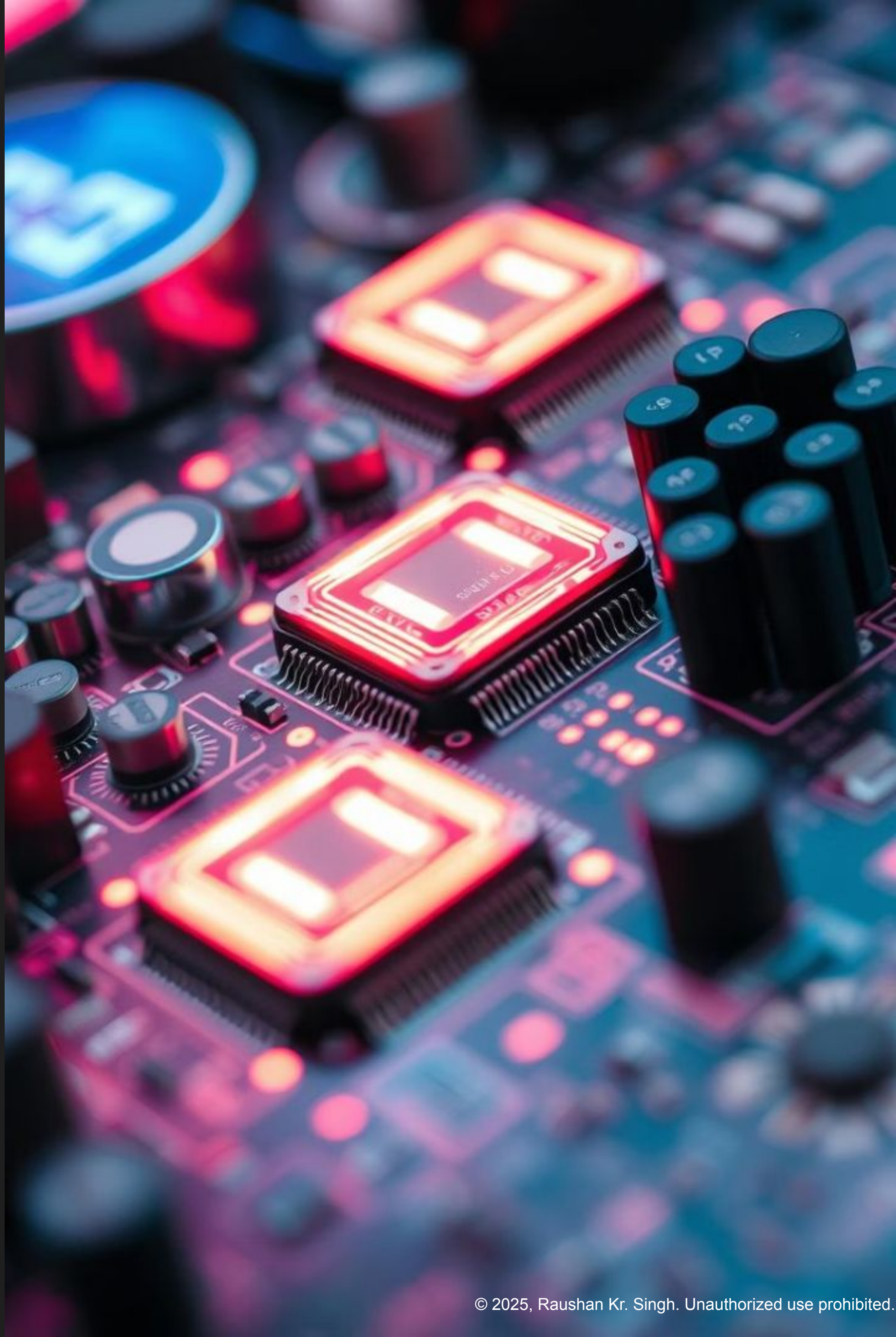
CEO, Fulectronix Technologies

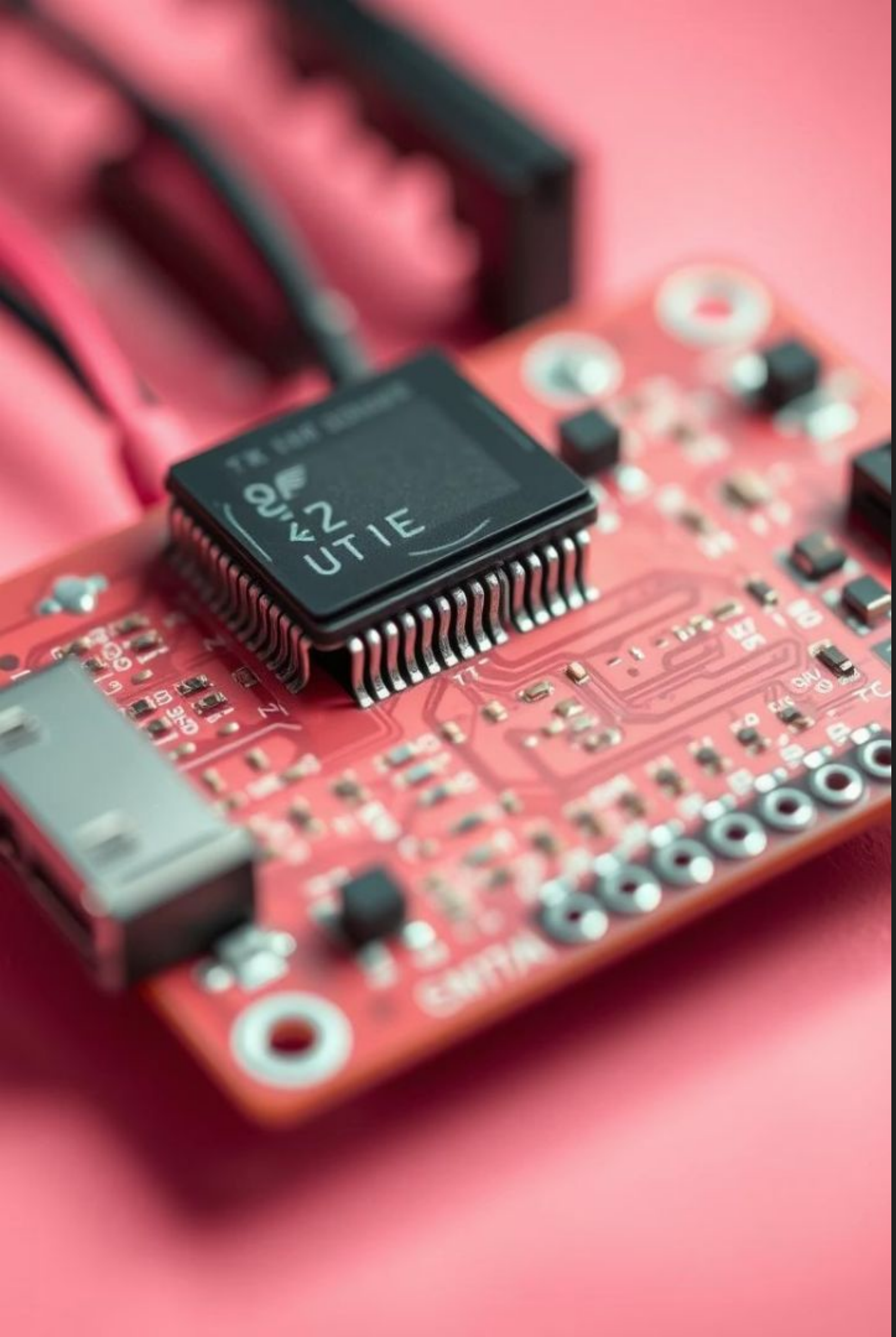
IIT Ropar



MCUs, CPUs, and NPUs

These processors play crucial roles in modern electronics, powering everything from simple devices to complex AI systems.





Microcontroller Units (MCUs)

Definition

A self-contained system on a chip integrating processor, memory, and peripherals.

Functions

Executes specific tasks within embedded systems with low power consumption and cost-effectiveness.

Examples & Market

- Controls appliances, automotive systems, IoT devices
- Market size: $20B in 2023; projected $30B by 2028 (CAGR 8.4%)

Central Processing Units (CPUs)

Definition

General-purpose processors designed to execute a wide variety of instructions.

Capabilities

Offers high performance but consumes significant power, suitable for computers and servers.

Market & Leaders

- Dominated by Intel (70%) and AMD (20%)
- Market size: $90B in 2023

Neural Processing Units (NPUs)

Purpose

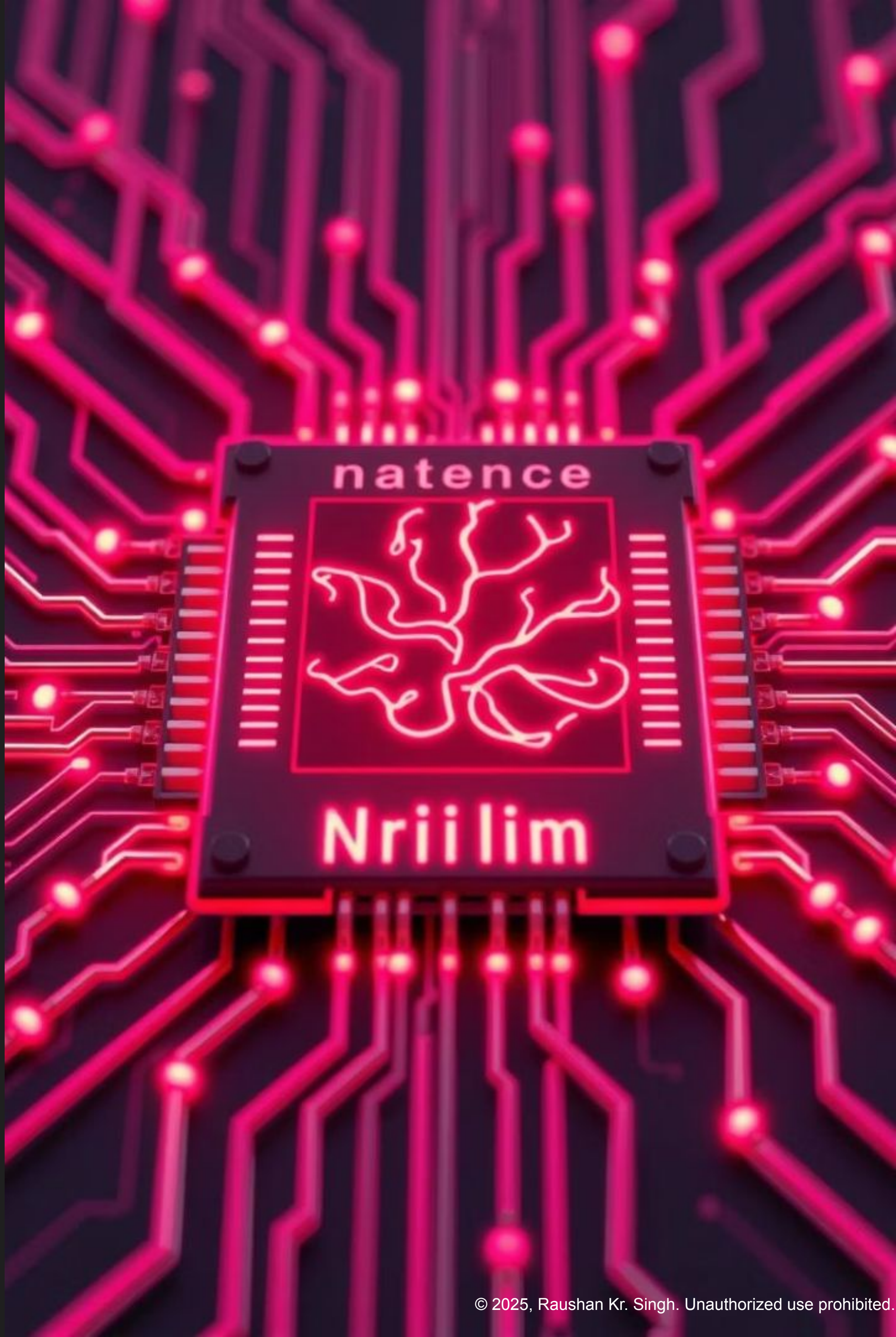
Specialized processors optimized for machine learning and neural network operations.

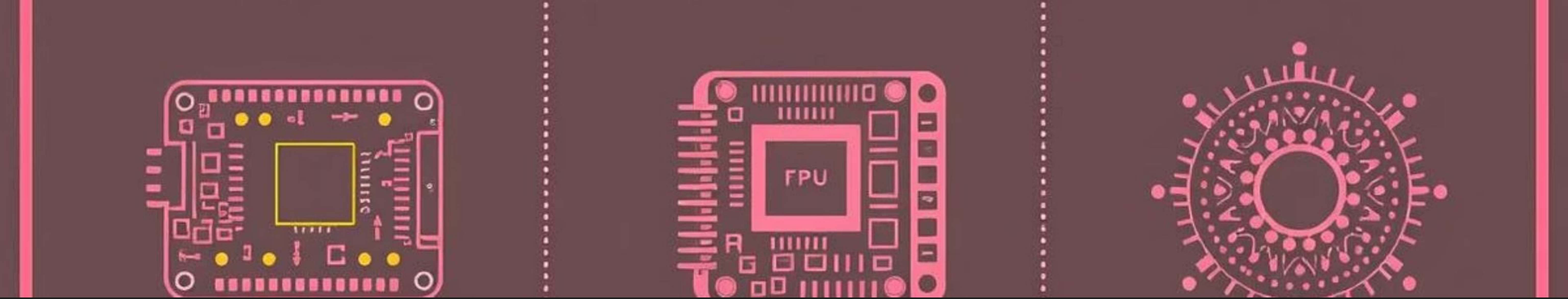
Advantages

Highly power-efficient for AI workloads, excelling at matrix multiplications.

Applications & Market

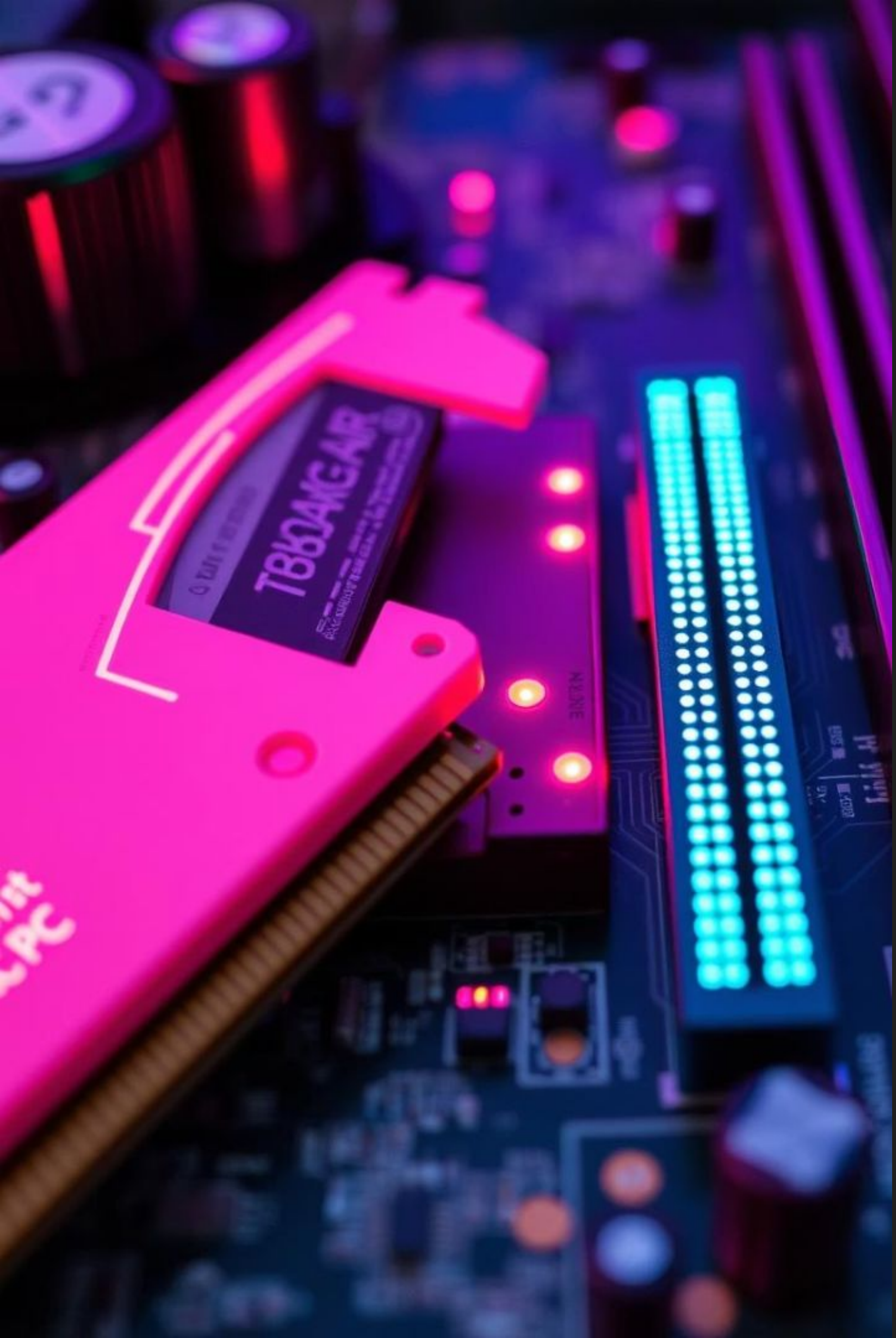
- Used in smartphones, AI accelerators, autonomous vehicles
- Market growth: $8B in 2023 to $60B by 2030 (CAGR 33%)





MCU vs. CPU vs. NPU: Key Differences

	MCU	CPU	NPU
Power Consumption	Lowest	Highest	Medium
Performance	Low	High	Specialized (AI)
Use Cases	Embedded Systems	General Computing	Machine Learning & AI
Examples	Arduino, ESP32	Intel i9, AMD Ryzen	Google TPU, Huawei Ascend



RAM, Flash Memory, and Clock Speed Explained

RAM

Volatile memory used for temporarily holding active data and instructions; faster access speeds.

Flash Memory

Non-volatile storage for long-term data and firmware retention; slower than RAM.

Clock Speed

Determines how fast a processor executes instructions, measured in GHz; higher speeds enable better performance but increase power consumption.

Current Market Trends in Processors

1. **MCUs**

   Increasing integration of wireless communication for smarter embedded systems.

2. **CPUs**

   Adoption of heterogeneous architectures combining CPUs with integrated GPUs for enhanced performance.

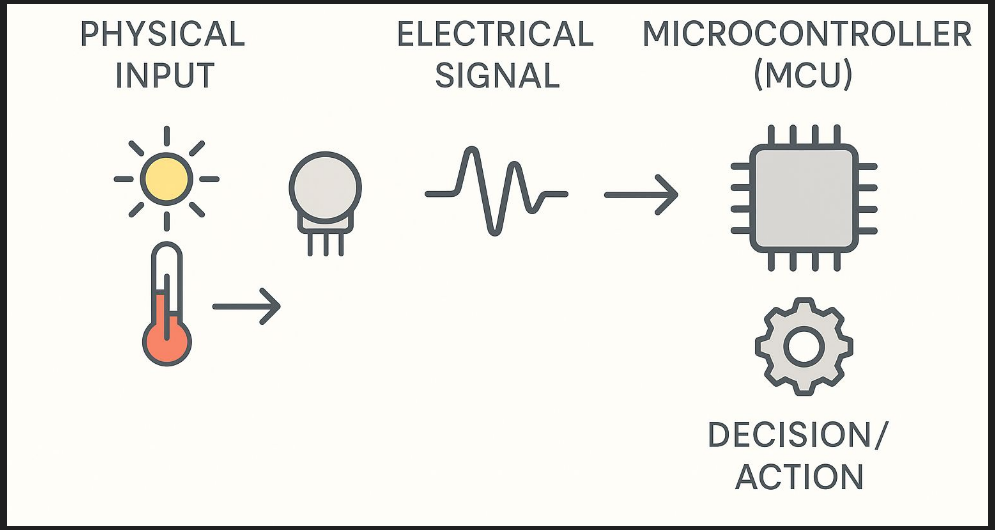
3. **NPUs**

   Rapid growth in edge computing and IoT devices deploying AI acceleration capabilities.

4. **Emerging AI-Enabled MCUs**

   Projected annual growth of 40% through 2027, enabling more intelligent low-power devices.

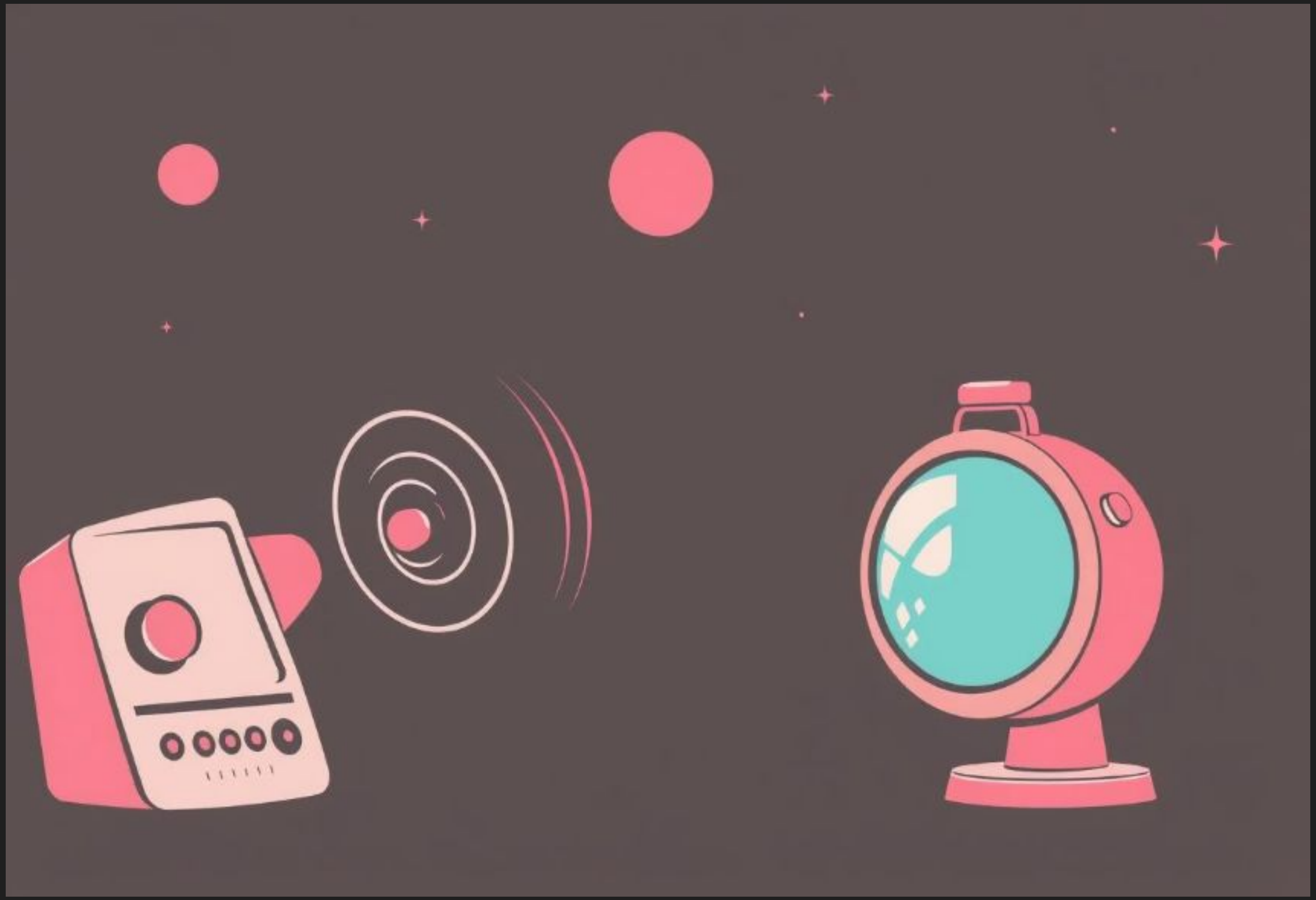
Sensors



A sensor is a device that detects changes in physical or environmental conditions and converts them into measurable signals.

Active vs. Passive Sensors

Active Sensors



Active sensors emit their own energy to detect environmental changes.

Passive Sensors

Passive sensors detect existing energy or stimuli without emitting energy themselves.

Active Sensors

- Ultrasonic Sensor
- LIDAR
- Radar Sensor
- Hall Effect Sensor

Passive Sensors

- LDR (Light Dependent Resistor)
- Strain Gauge
- Piezoelectric Sensor
- Accelerometer

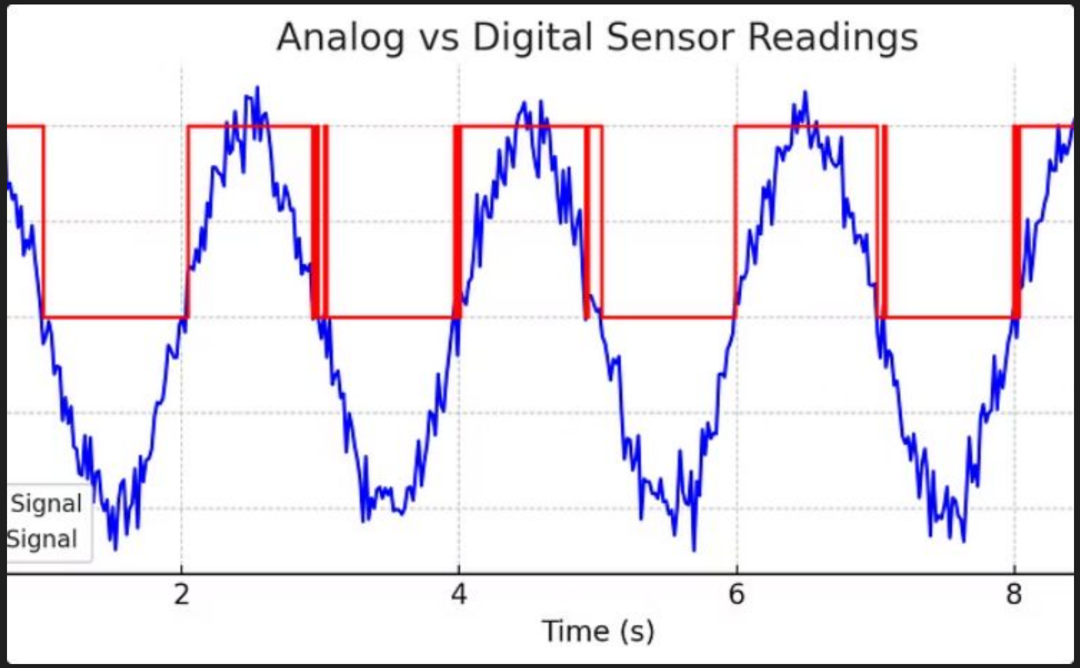
Analog vs. Digital Sensors

Analog Sensors

Continuous, smooth output signal with high-resolution measurements.

Digital Sensors

Discrete output: 0 or 1, offering strong noise immunity and reliability



Common Sensor Types

Temperature Sensors

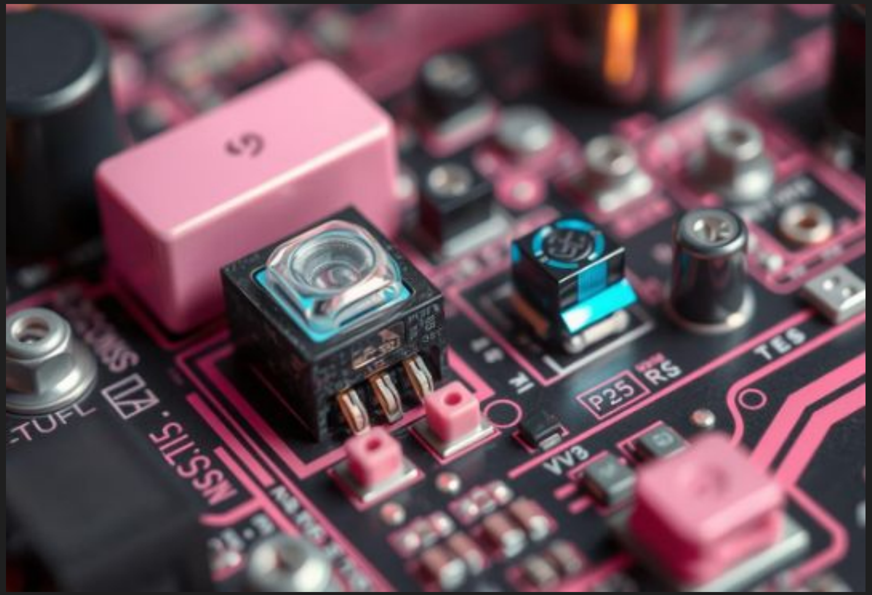
Measure heat or cold to regulate environments or processes. Found in thermostats, weather stations, and industry equipment.

- Thermistors
- Thermocouples
- RTDs (Resistance Temperature Detectors)

Pressure Sensors

Detect force applied per unit area and relay vital information in various settings, from automotive to healthcare.

- Strain Gauges
- Piezoelectric Sensors



Light Sensors

Gauge light intensity for adaptive applications in smartphones, cameras, and security systems.

- Photodiodes
- Photoresistors

Advanced Sensor Technologies

Image Sensors

Capture visual data for cameras, medical devices, and autonomous vehicles. Approximately 90% of these sensors utilize CMOS technology to deliver enhanced efficiency and superior image quality.

Inertial Sensors

Measure motion and orientation for navigation systems, robotics, and wearable technologies. For example, the iPhone 14 employs a dual-core accelerometer for precise movement tracking.

Gas Sensors

Detect various gases for applications in air quality monitoring, safety, and industrial environments using sophisticated sensor technologies.

Sensor Applications: Shaping Our World

Automotive

Sensors enable autonomous driving, collision avoidance, and parking assistance. Modern cars typically feature over 100 sensors to enhance safety and performance.

Fact: Cars contain 100+ sensors.

Healthcare

Wearable health trackers and remote monitoring devices rely heavily on sensors, supporting diagnostics and personalized care. The market is rapidly expanding, valued at $24 billion globally.

Fact: Global medical sensor market = $24B.

Smart Homes

Sensors automate lighting, climate controls, and security, improving comfort and energy efficiency in houses worldwide.

Fact: Smart homes incorporate 50+ sensors.

Industrial Automation

Sensors optimize process control, enable predictive maintenance, and drive robotics in factories, reducing downtime significantly.

Fact: Sensors can reduce downtime by 30%.

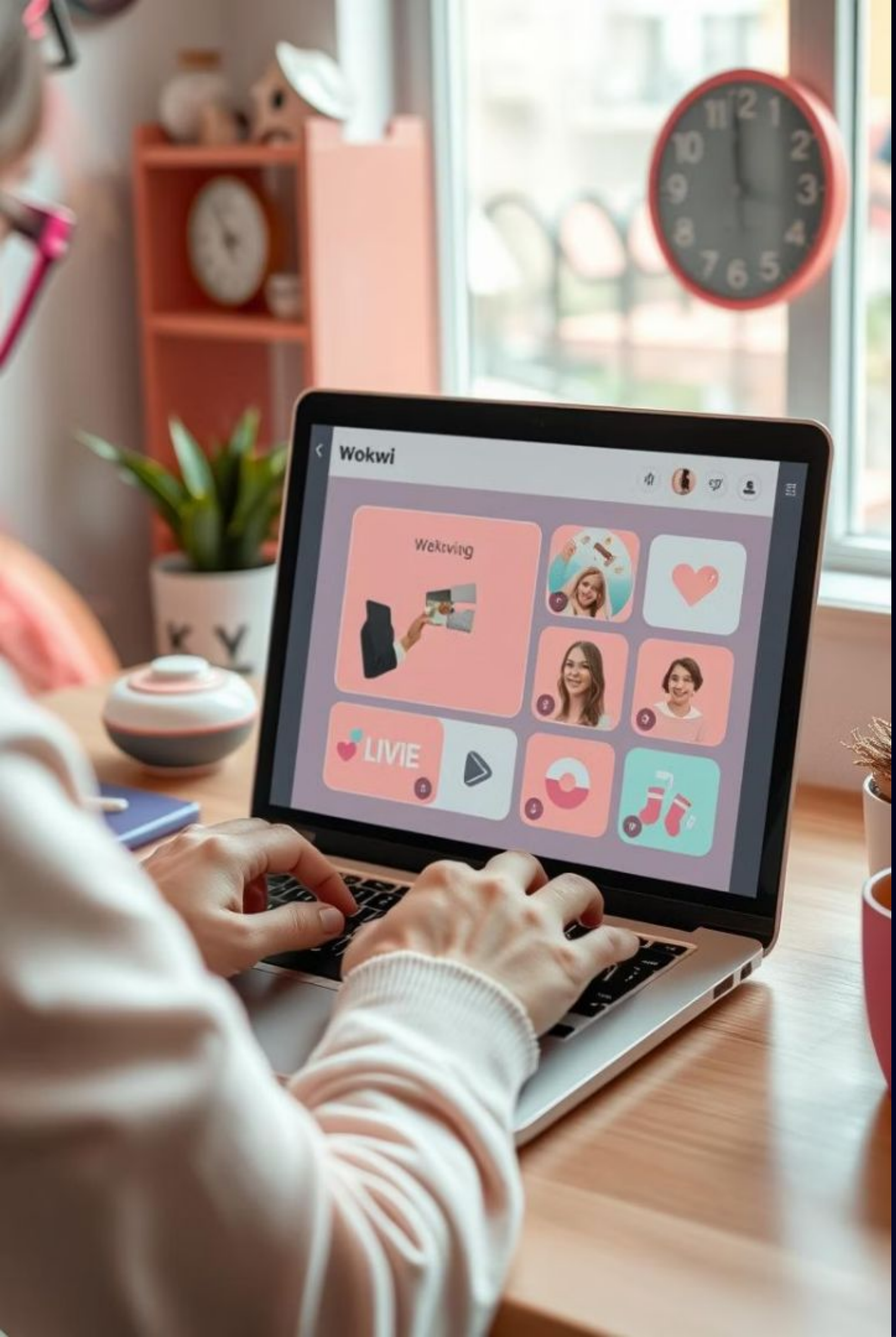
Wokwi Simulator

Wokwi Simulator is an online platform that allows users to simulate Arduino, ESP32, and other microcontroller projects in a virtual environment.

It supports

- Real-time code execution
- Virtual components
- Circuit design

making it ideal for prototyping and learning without physical hardware.



Features of Wokwi

1. Drag-and-Drop Interface
2. Real-Time Simulation
3. Integrated Code Editor
4. Serial Monitor

Benefits of Wokwi for TinyML Prototyping

Rapid Prototyping

Cost Efficient

Risk Minimization

Collaborative Development

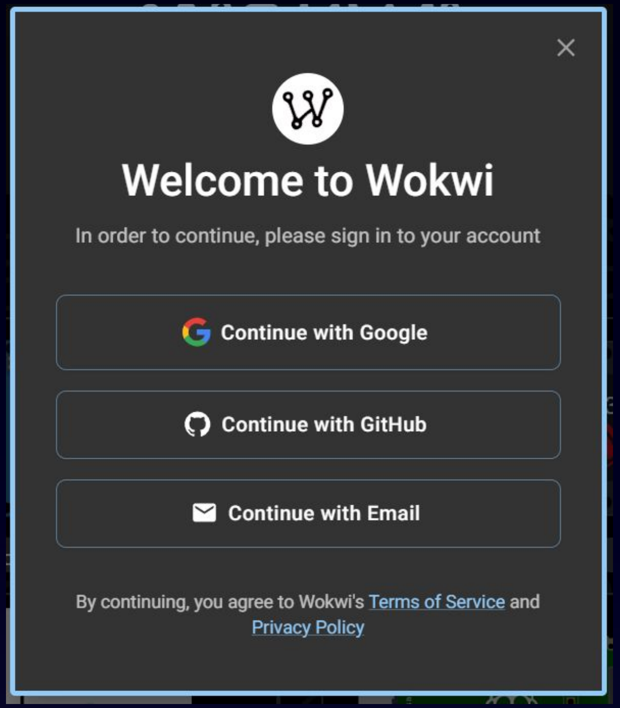
WOKWI

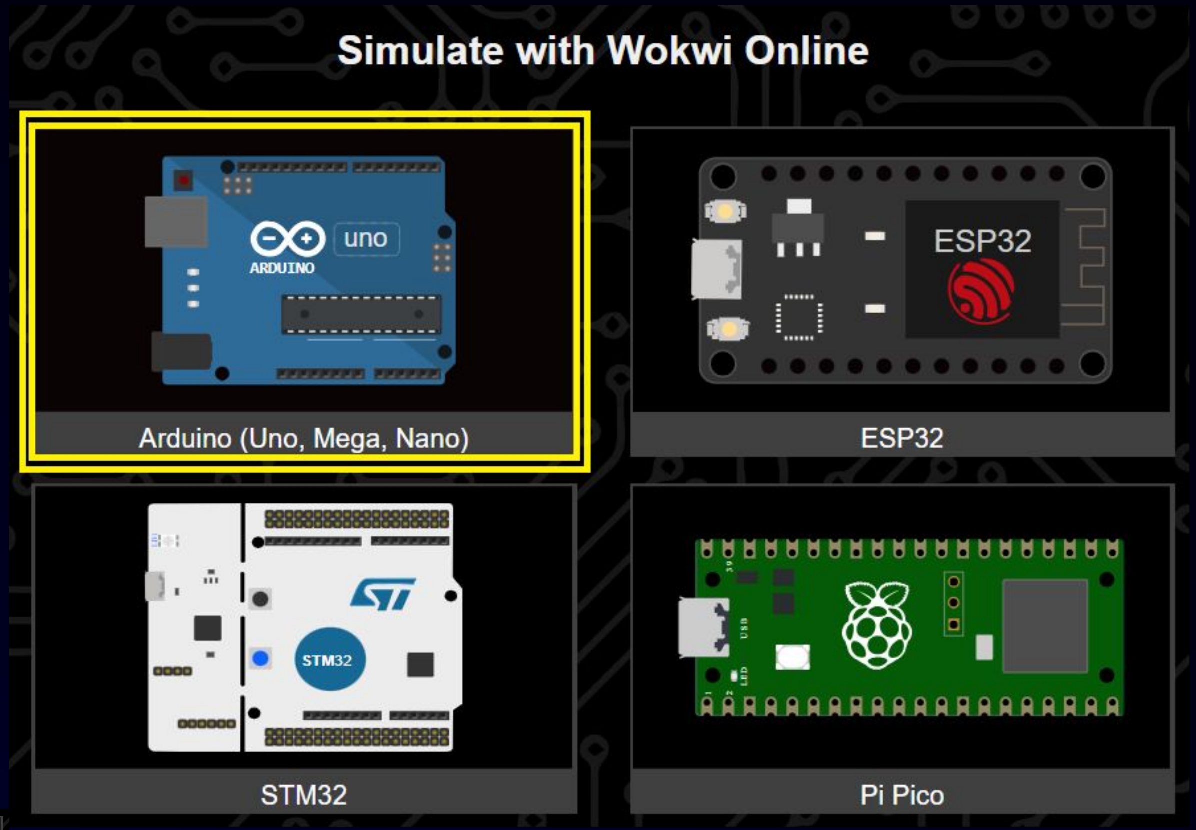
World's most advanced ESP32 simulator

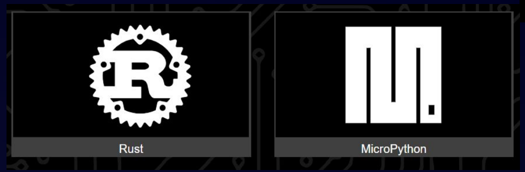
Simulation

Simulation : Step 1: Wokwi Account Creation

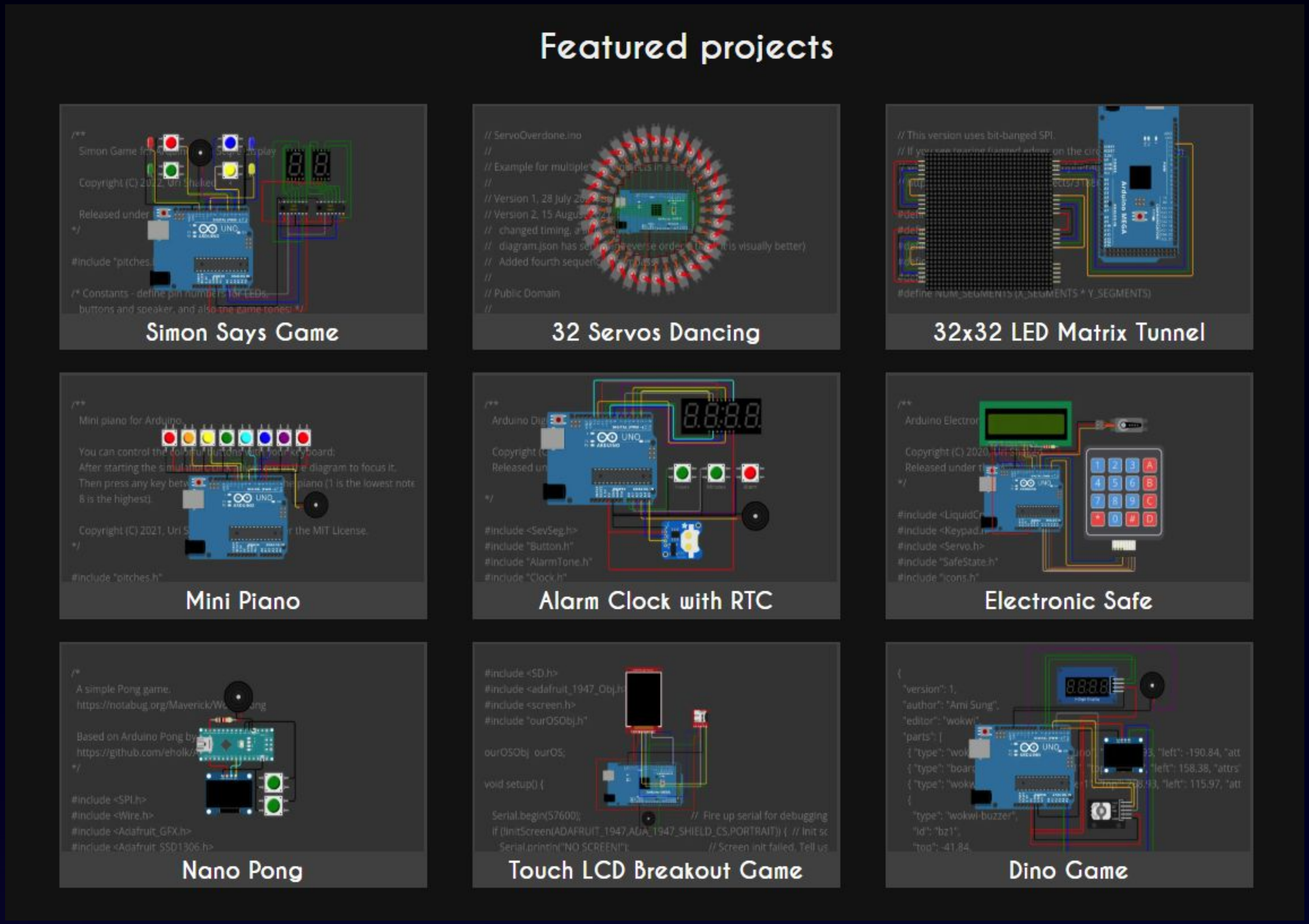
Wokwi Simulator Website : www.wokwi.com

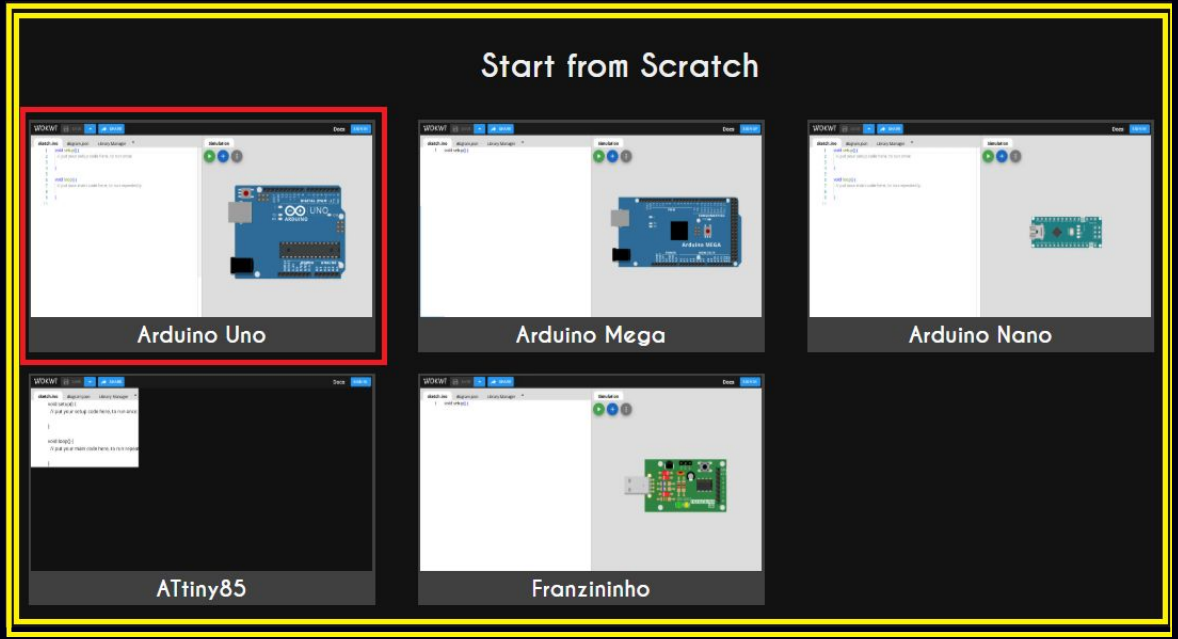


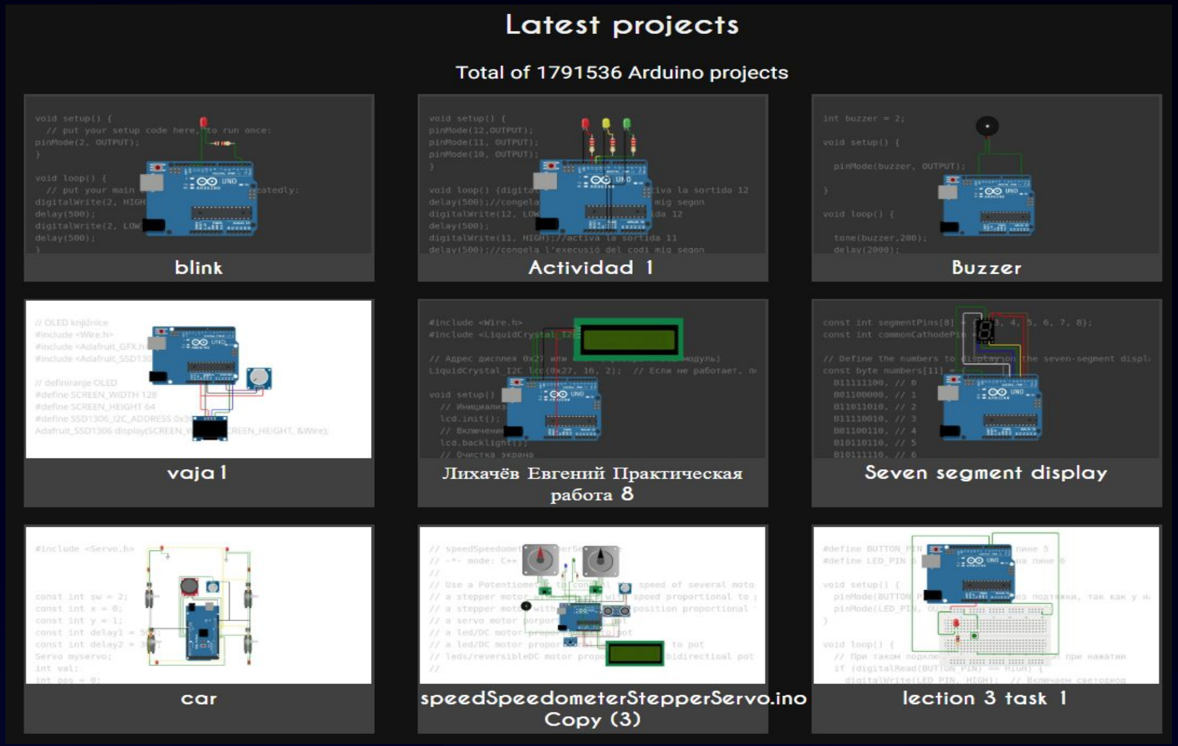




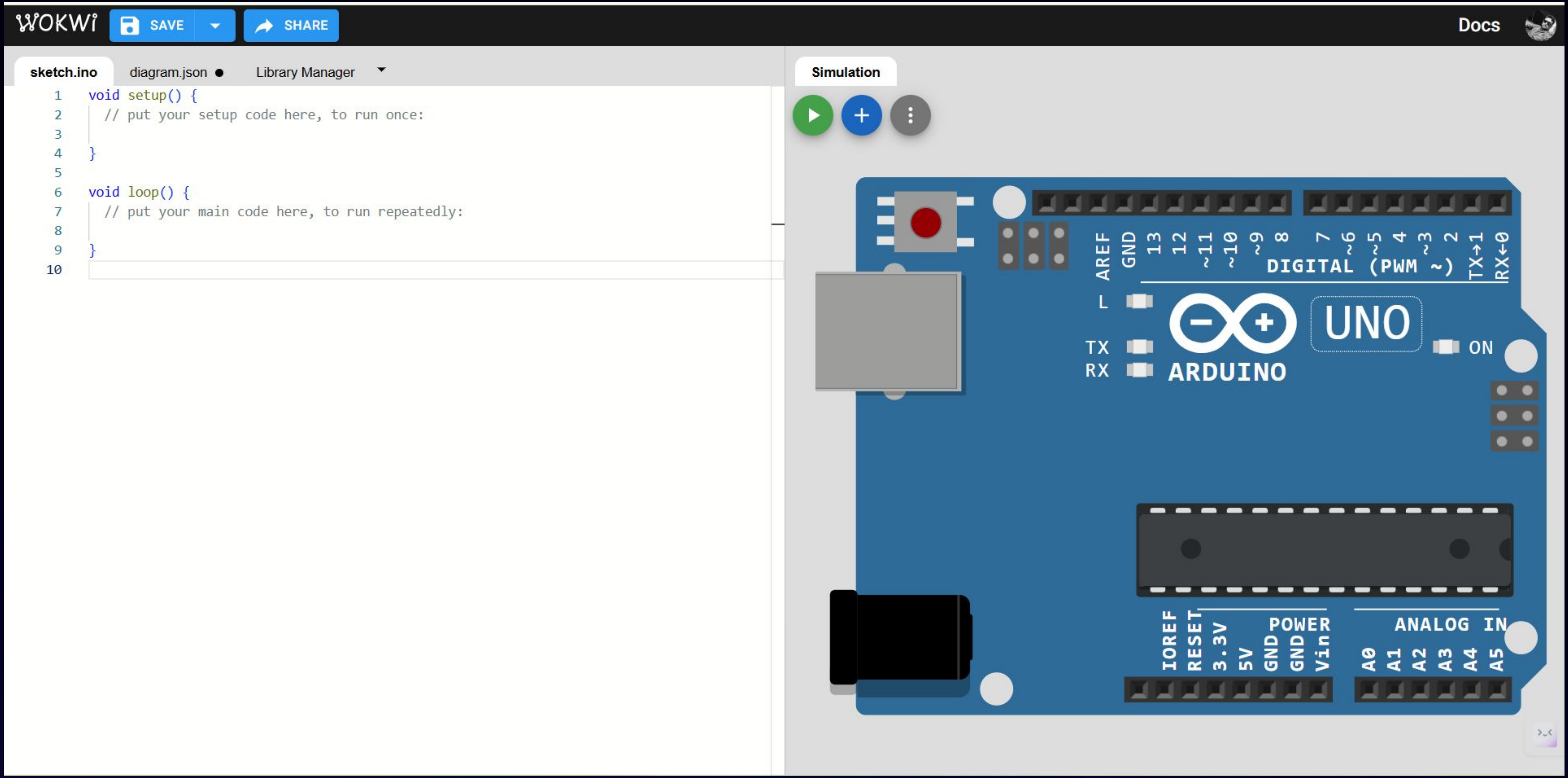
Step 2: Featured Projects / Start from Scratch / Latest P







Step 3: Arduino and Programming

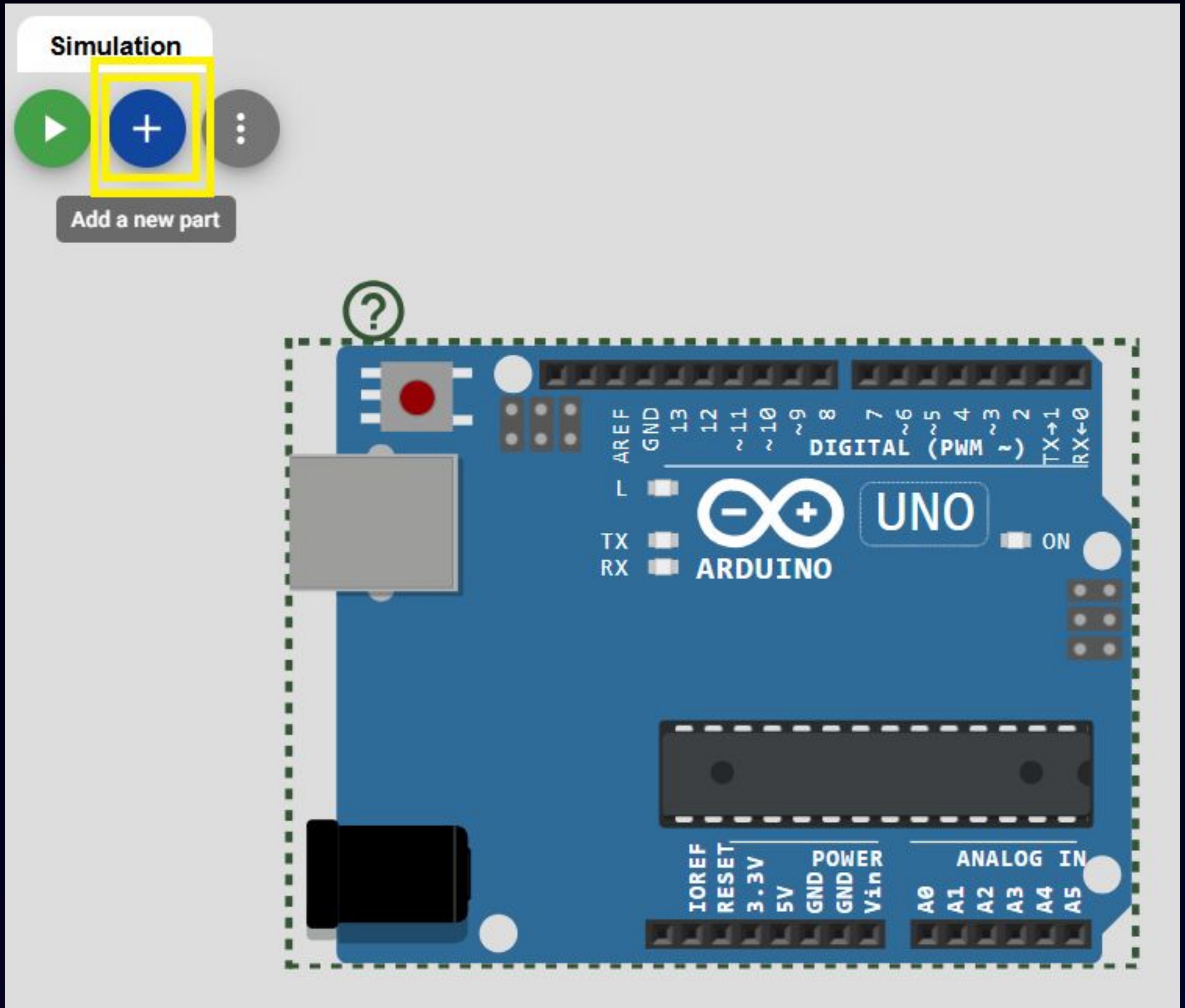


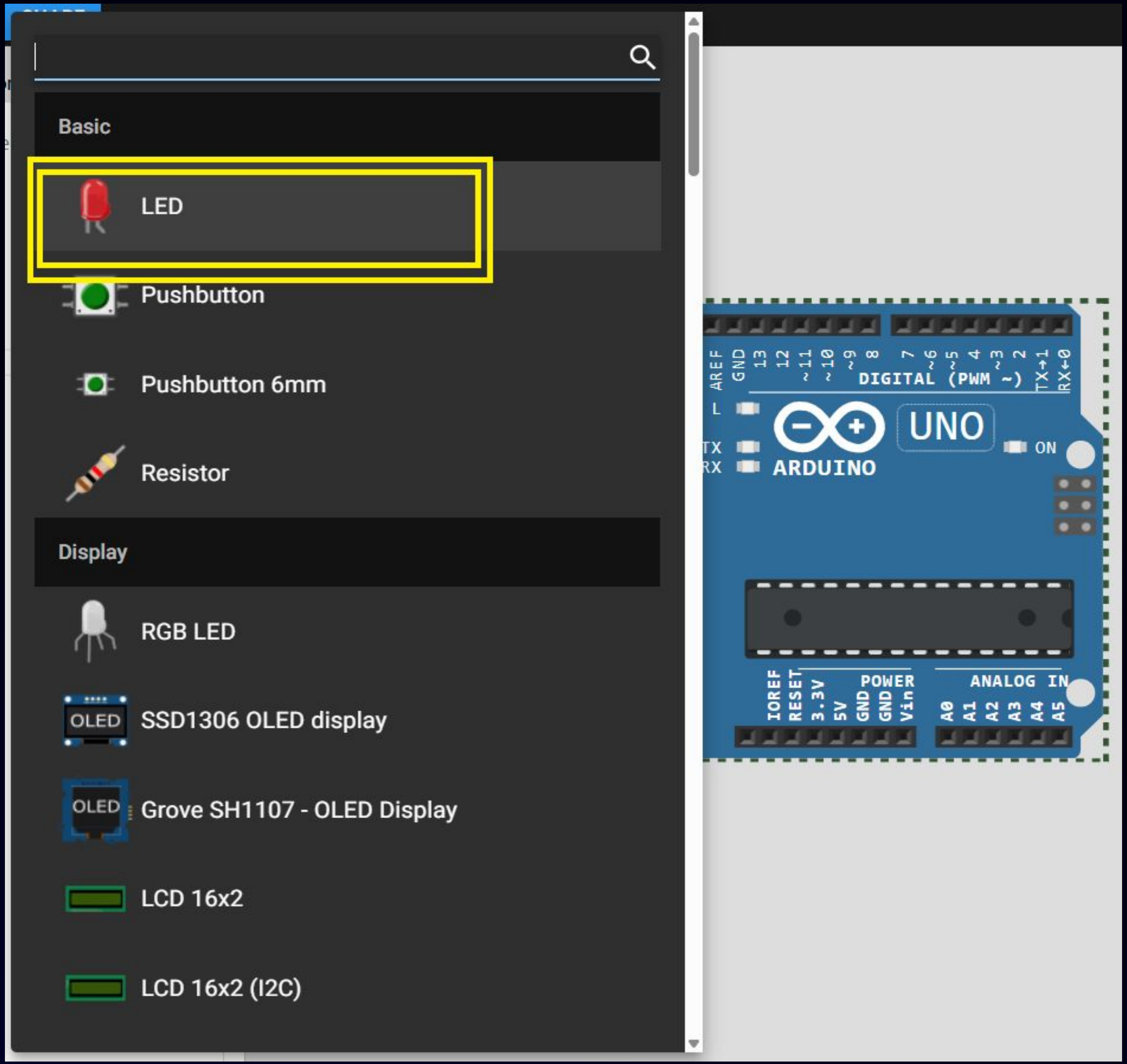
WOKWI

World's most advanced ESP32 simulator

LED Blink

Step 4: Components Selection : LED & Resistor

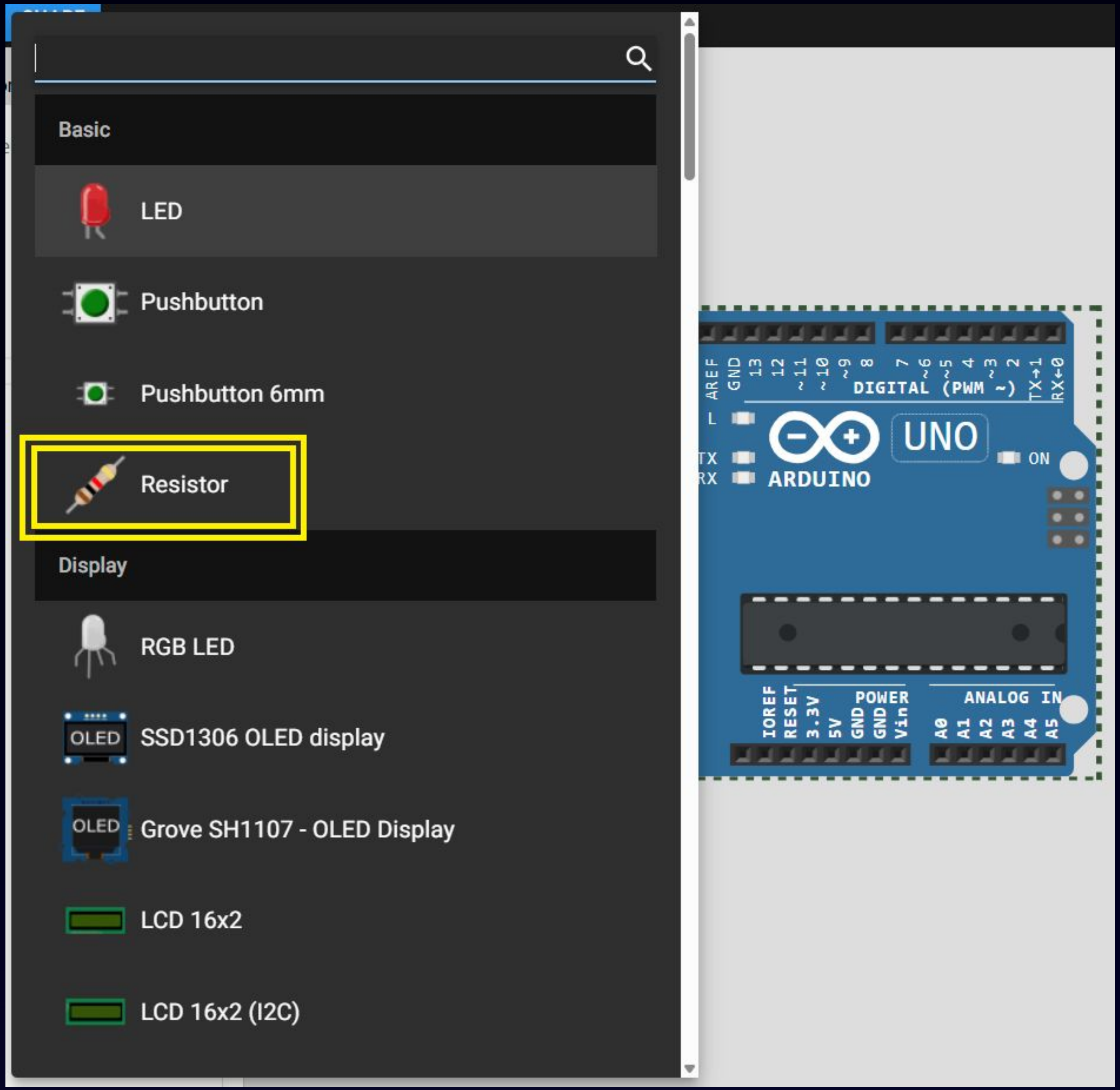


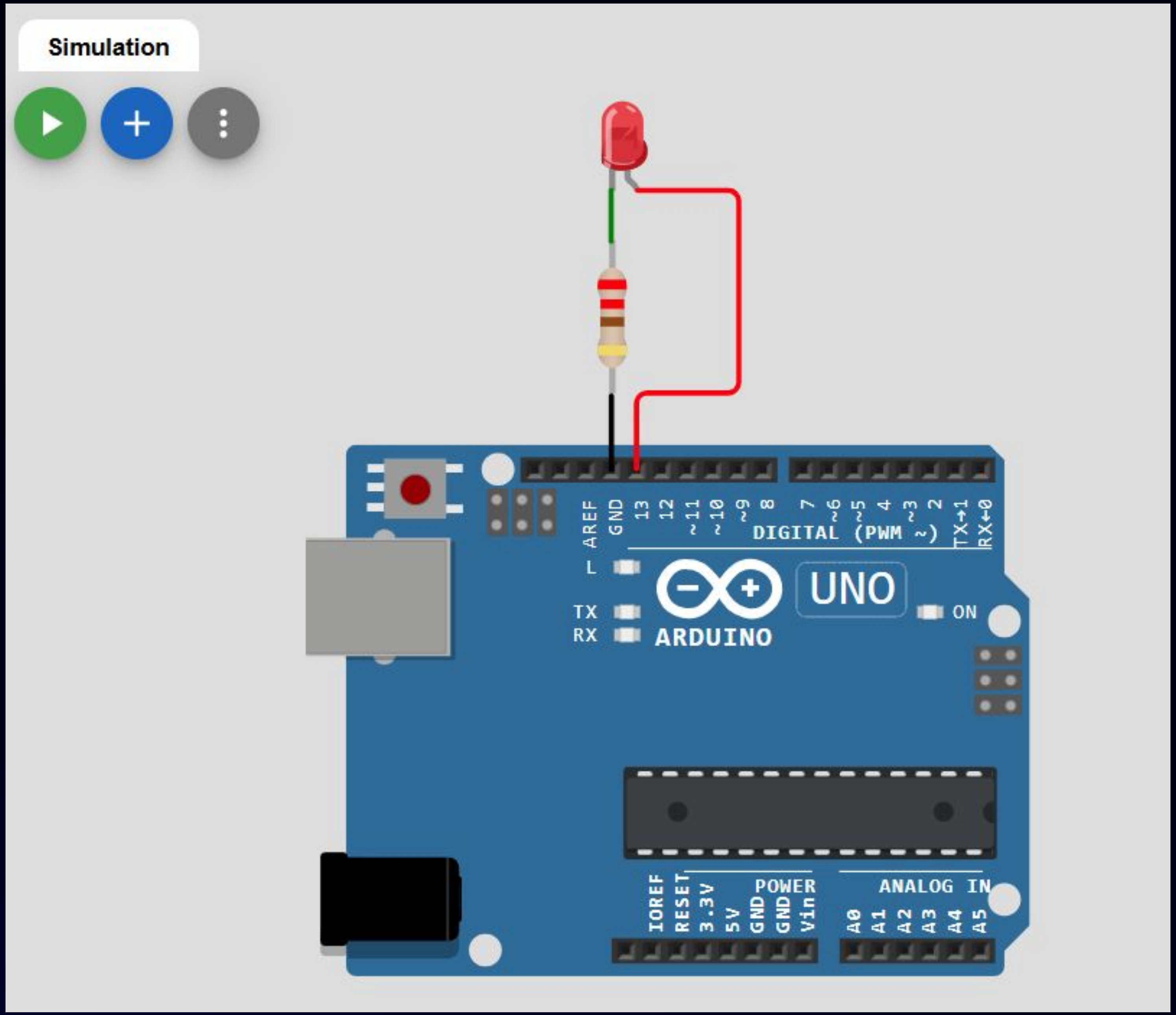


Choose + to add component

Select LED

Step 5: Components Selection and Circuit Design





Select Resistor - 220 Ohm

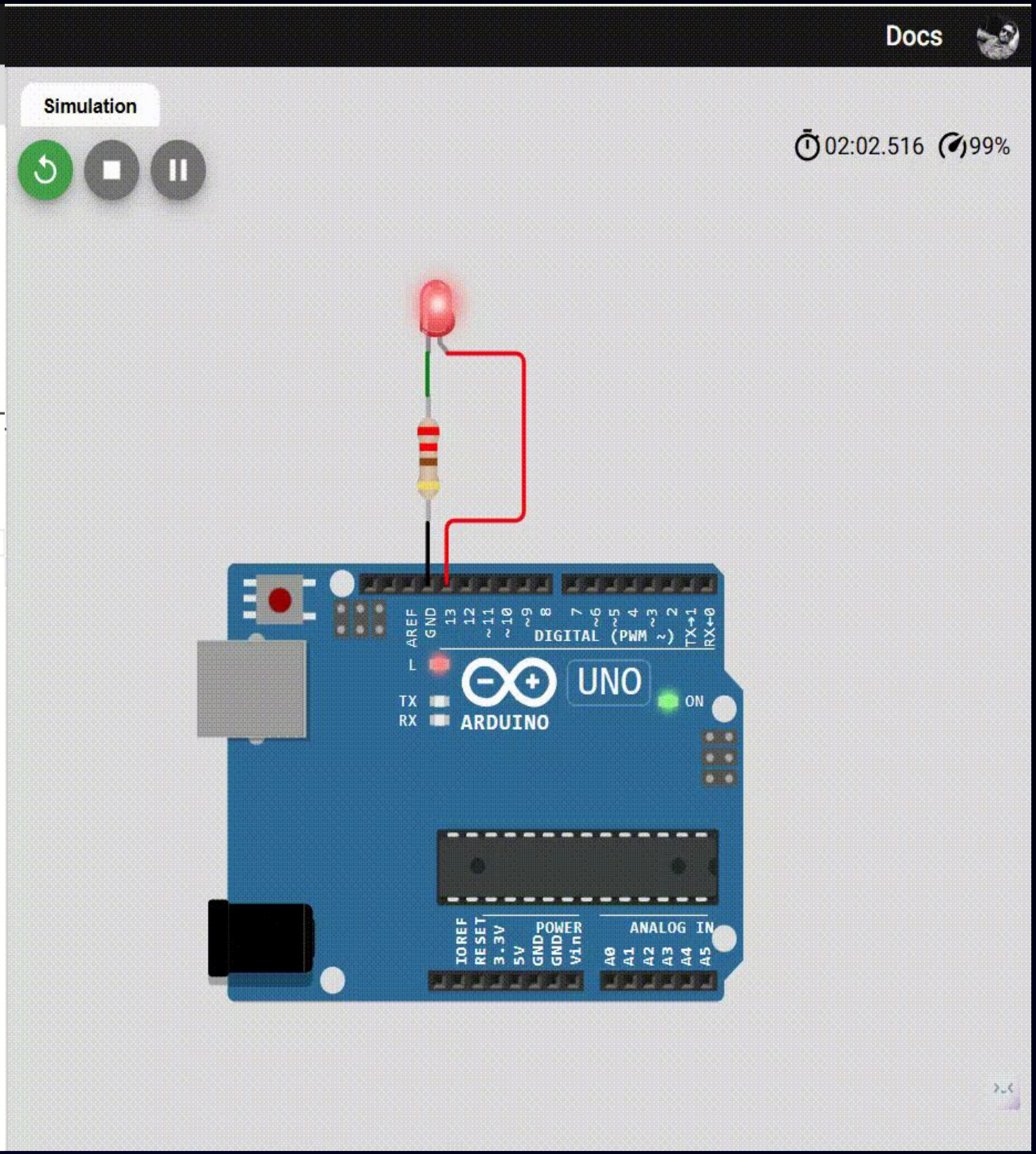
Connect Components

Step 6: Write Program and Hit Simulation

WOKWI SAVE SHARE

sketch.ino diagram.json Library Manager

```
const int LED = 13;

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

void loop()
{
  digitalWrite(LED, HIGH);
  delay(500);
  digitalWrite(LED, LOW);
 delay(500);
}
```

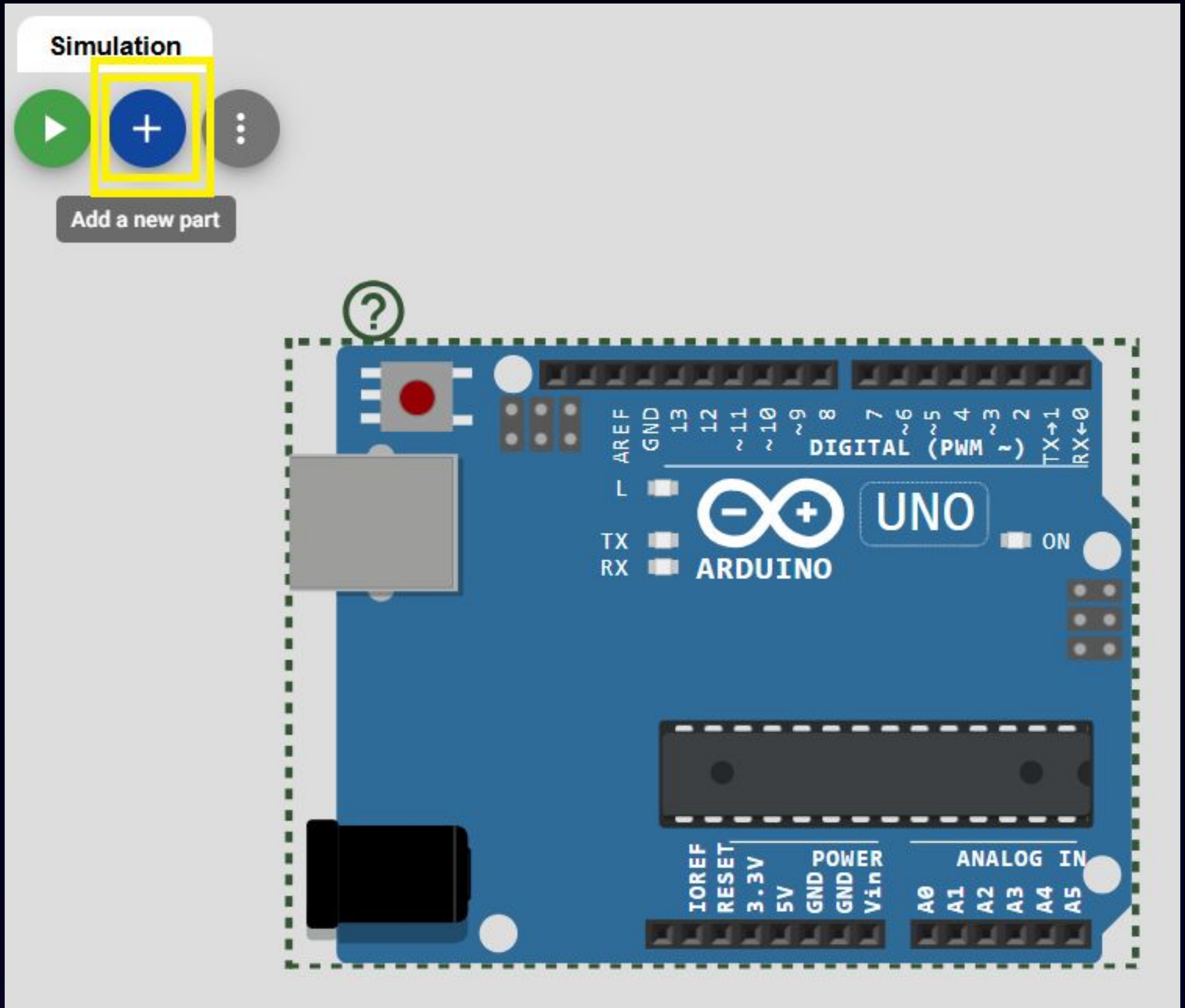


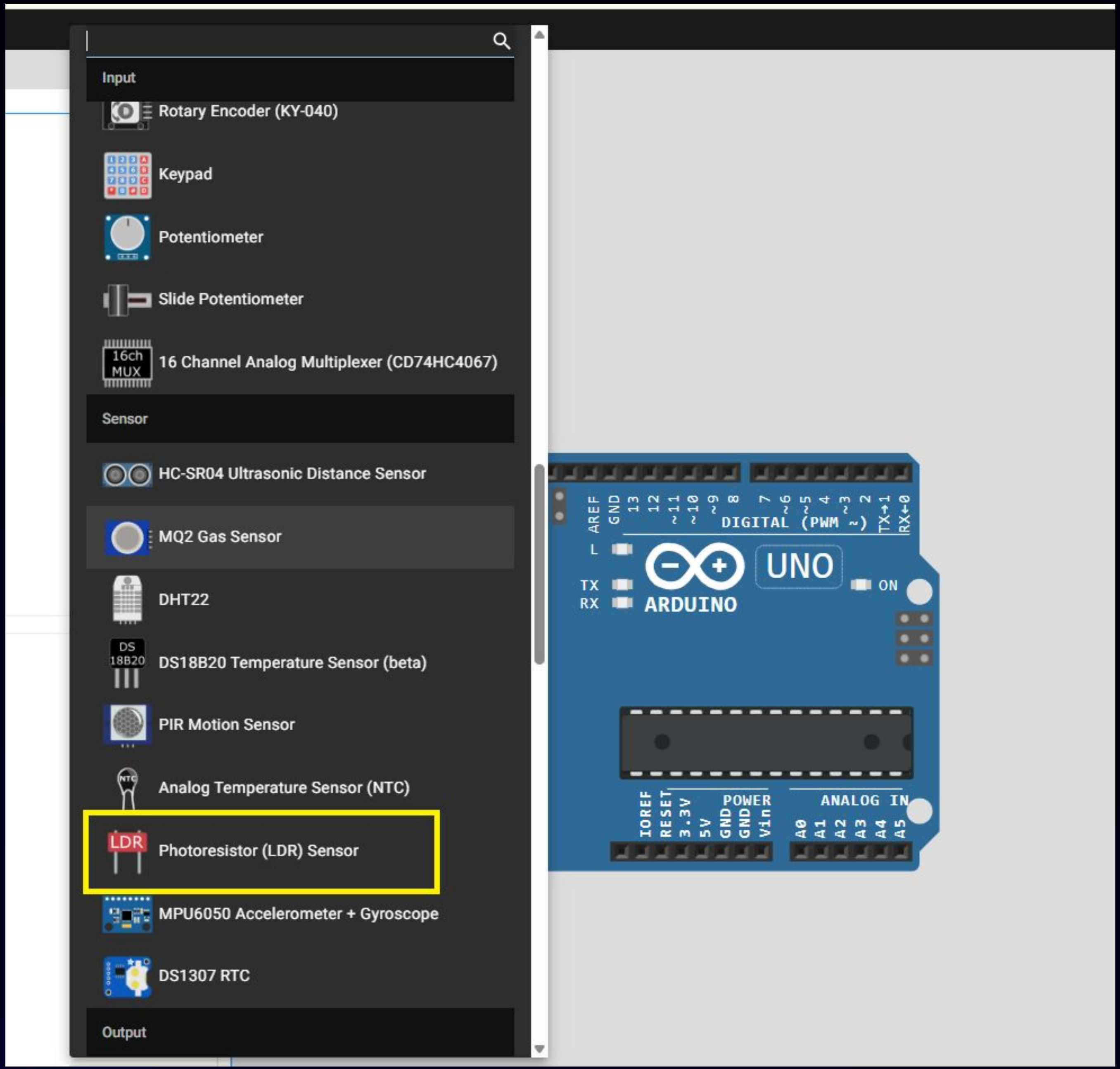
WOKWI

World's most advanced ESP32 simulator

LDR Sensor - Analog

Step 1: Components Selection : LED & Resistor

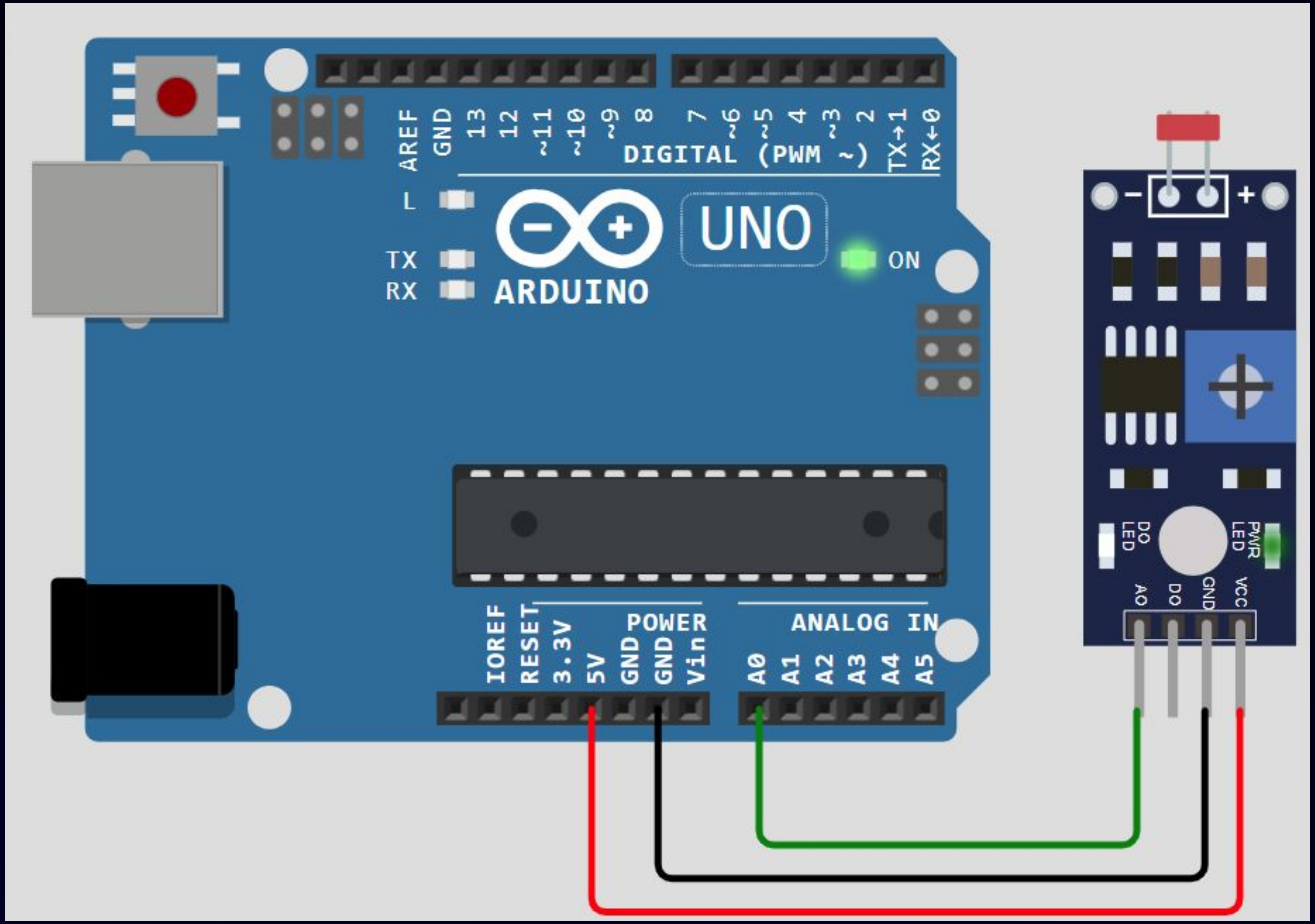




Choose + to add component

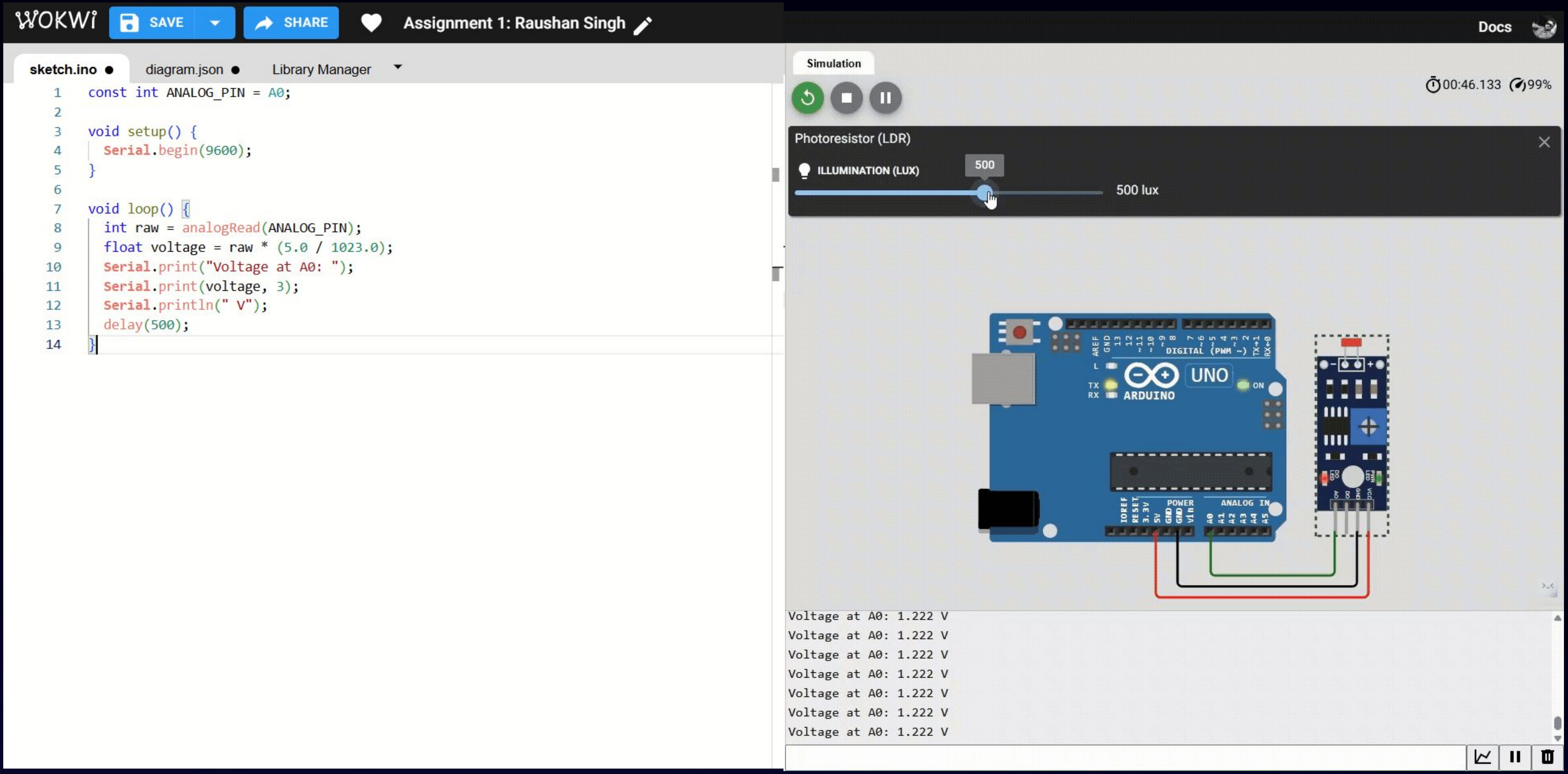
Select LDR Sensor

Step 2: Wire Connections



LDR Sensor Connections

Step 3: Write Program and Hit Simulation



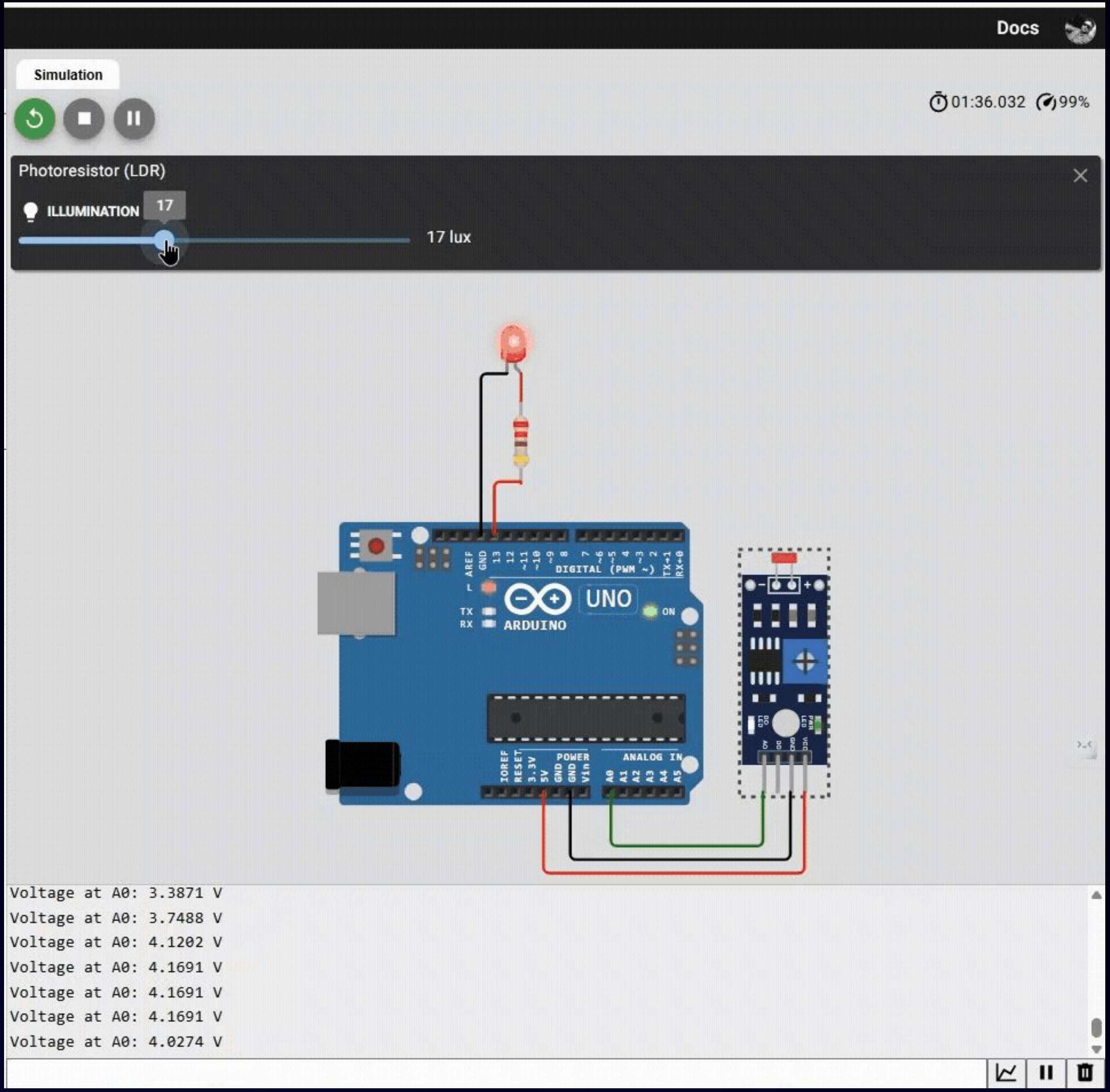
Practice 1

Write an Arduino sketch that reads the analog voltage from an LDR sensor on A0, compares it against a 3 V threshold, and drives the LED on pin 13 such that:

- **LED ON when the measured voltage is below 3 V**
- **LED OFF when the measured voltage is 3 V or above**

Solution - 1

```
const int LDR_PIN = A0;
const int LED_PIN = 13;

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

void loop()
{
  int raw = analogRead(LDR_PIN);
  float voltage = raw * (5.0 / 1023.0);
  Serial.println(voltage);

  if (voltage < 3.0) {
    digitalWrite(LED_PIN, LOW);
  } else {
    digitalWrite(LED_PIN, HIGH);
  }

  Serial.print("Voltage at A0: ");
  Serial.print(voltage, 4);
  Serial.println(" V");
  delay(500);
}
```

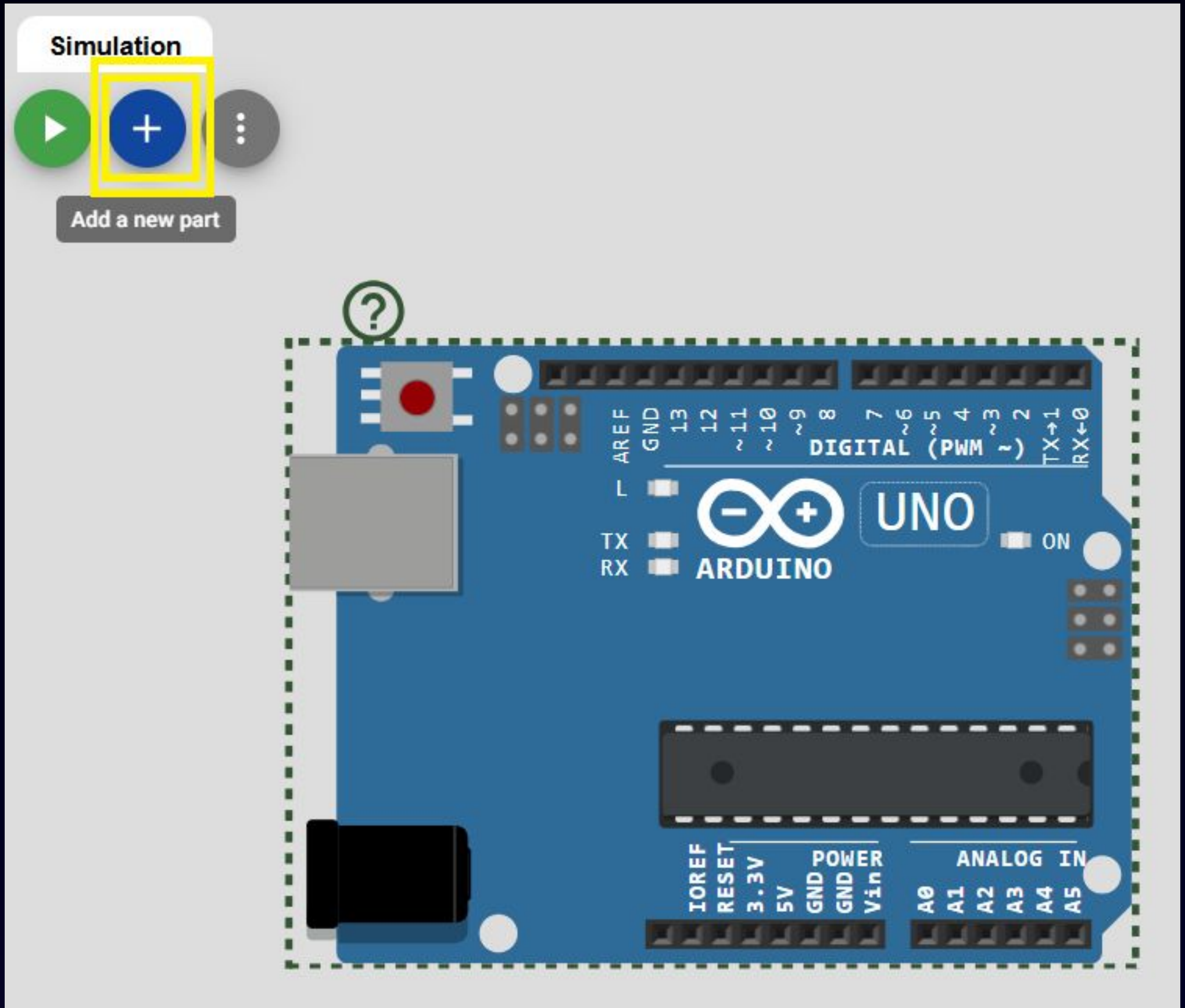


WOKWI

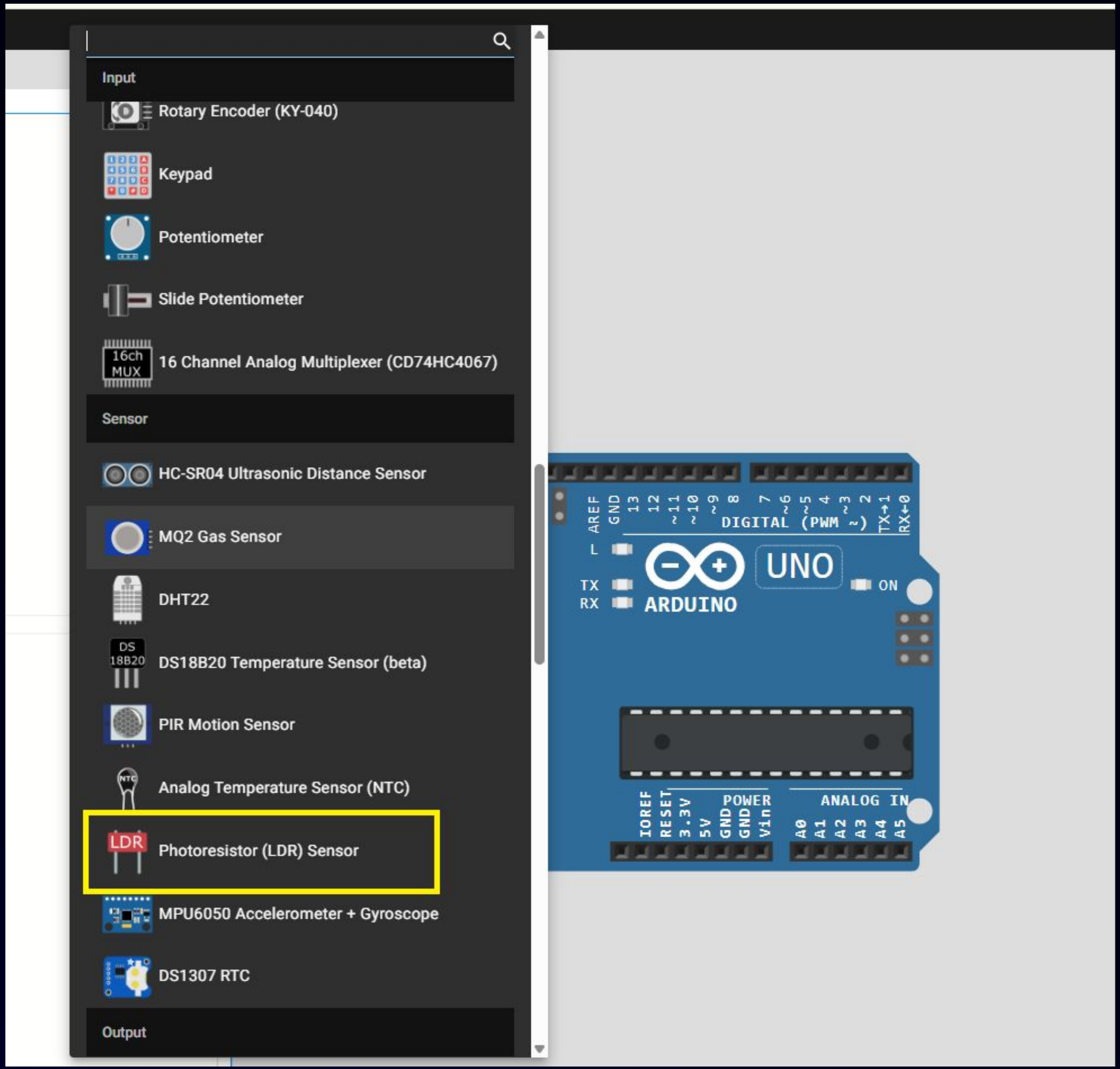
World's most advanced ESP32 simulator

LDR Sensor - Digital

Step 1: Components Selection : LED & Resistor

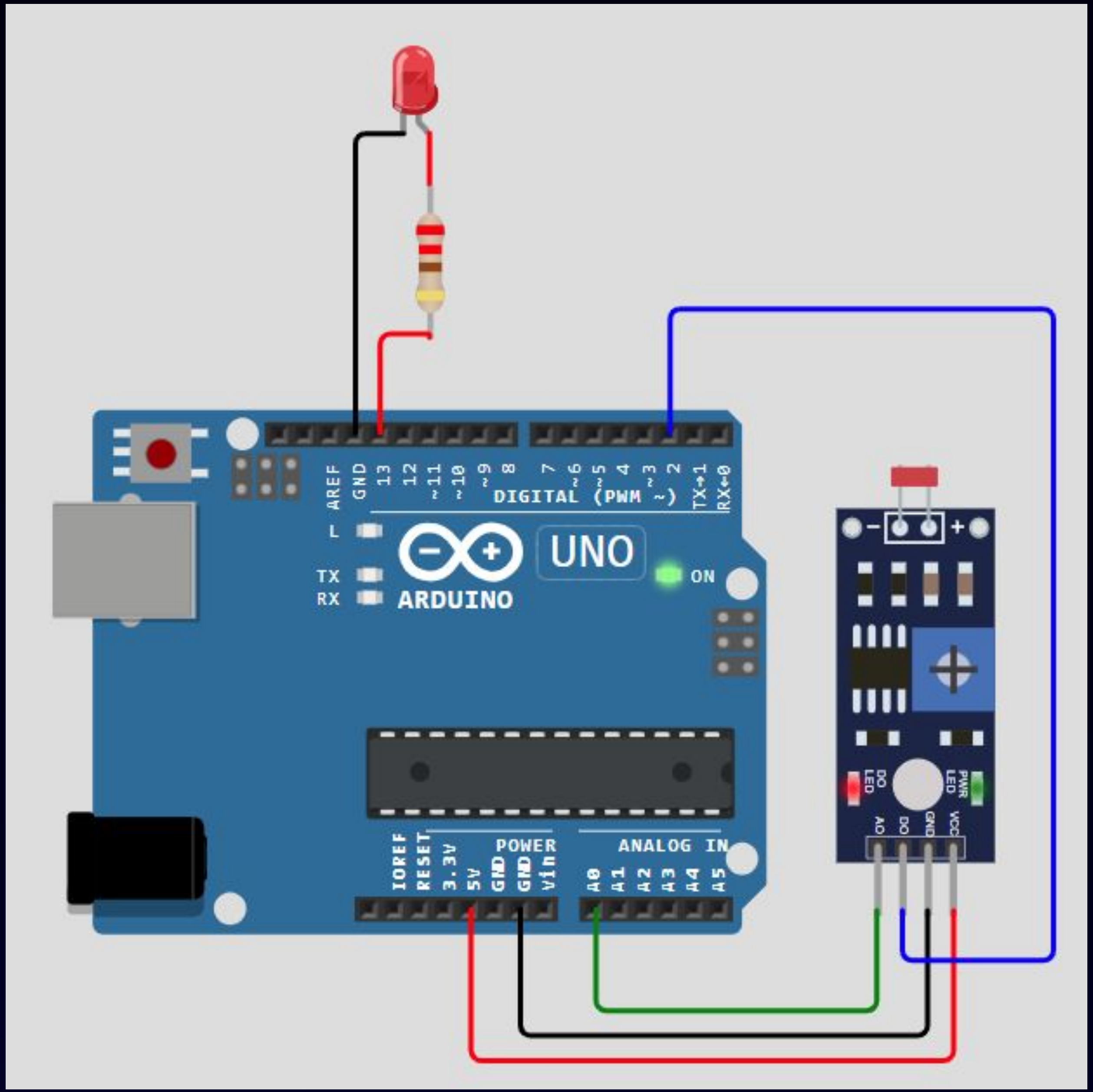


Choose + to add component



Select LDR Sensor

Step 2: Wire Connections



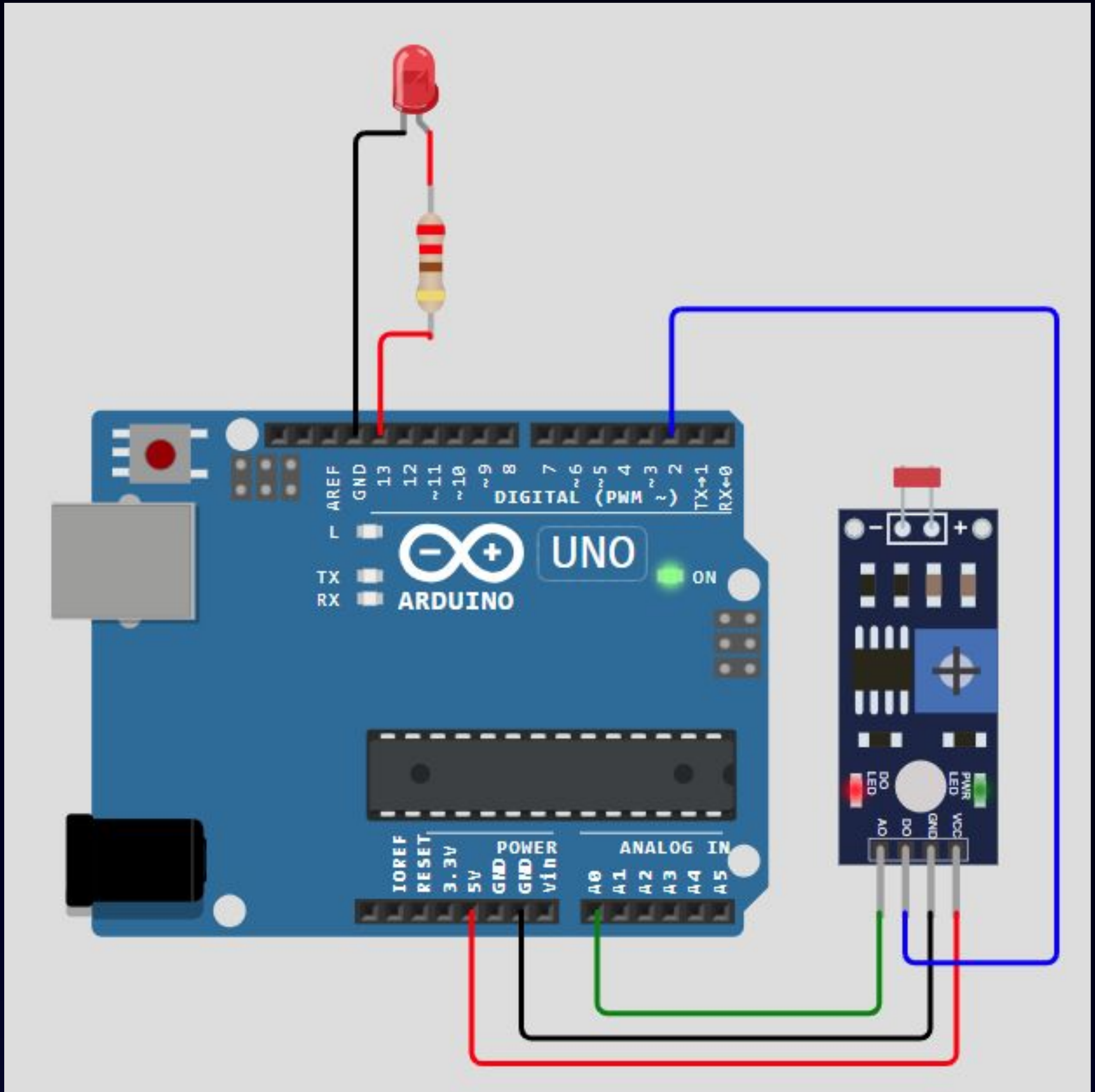
LDR Sensor Connections

Practice 2

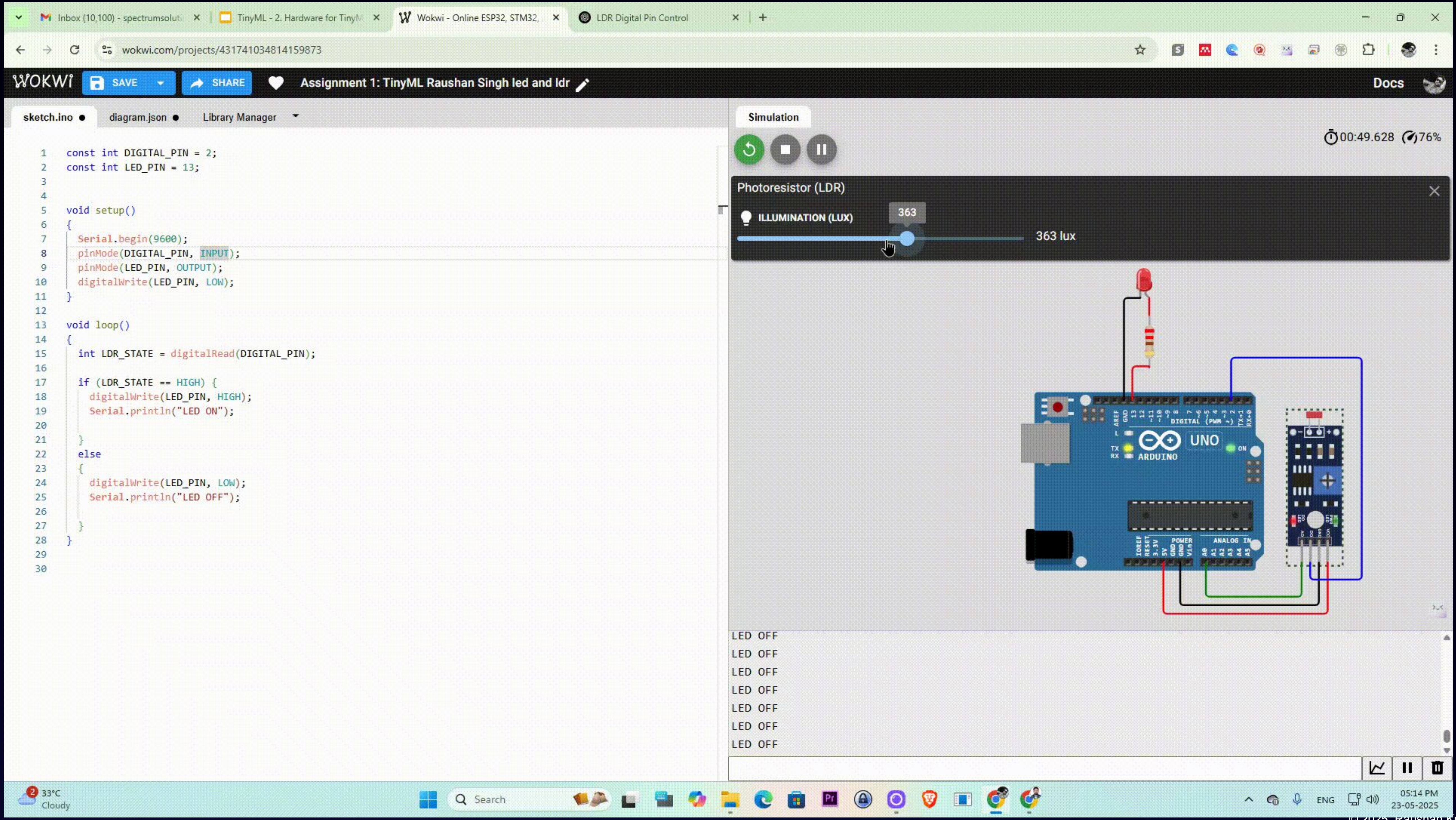
Write an Arduino sketch that reads the digital pin of LDR sensor and control LED when its getting high and low when its getting low.

Solution - 2

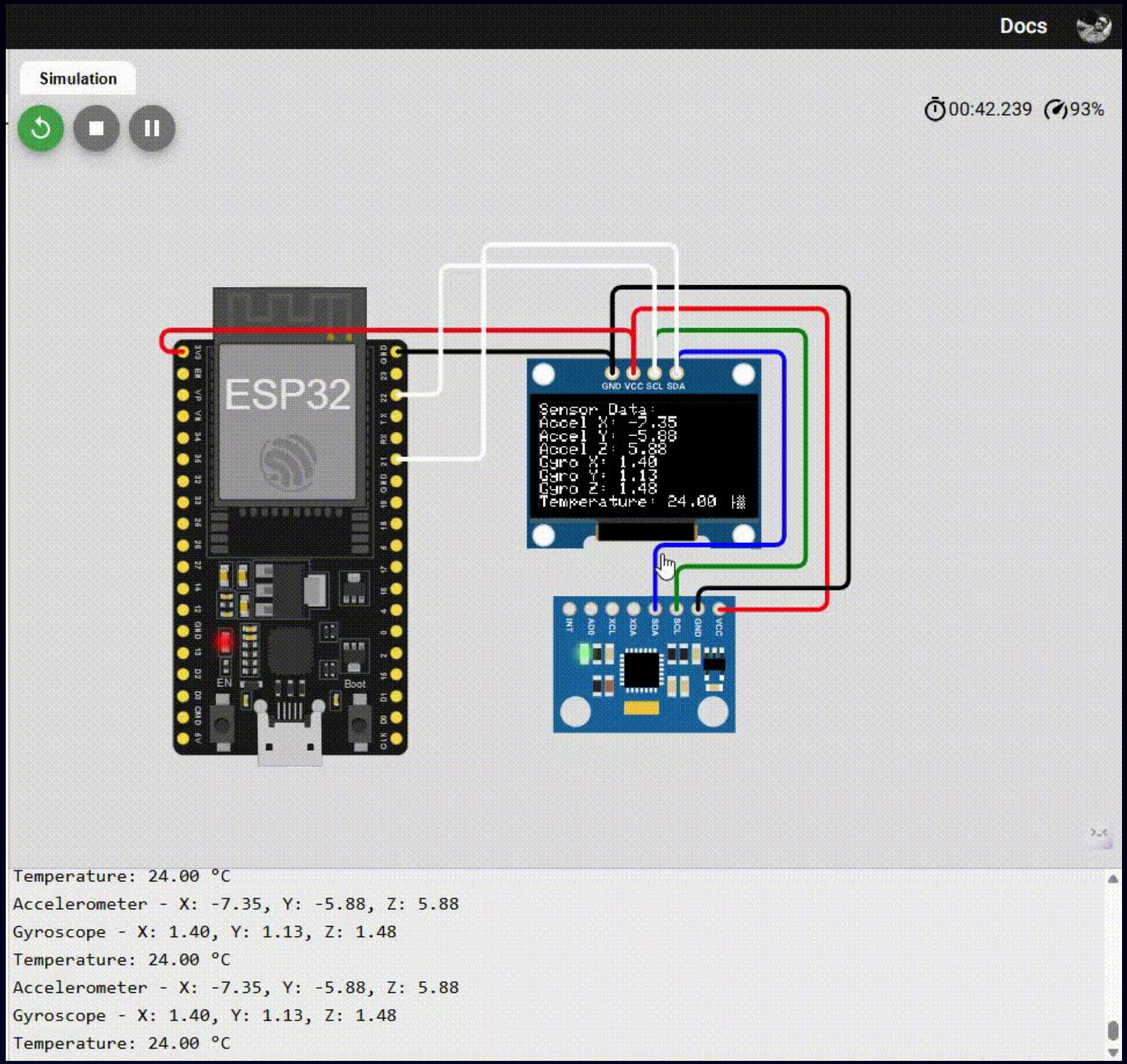
```
const int DIGITAL_PIN = 2;
const int LED_PIN = 13;

void setup() {
  Serial.begin(9600);
  pinMode(DIGITAL_PIN, INPUT);
  pinMode(LED_PIN, OUTPUT);
  digitalWrite(LED_PIN, LOW);
}

void loop() {
  int LDR_STATE = digitalRead(DIGITAL_PIN);


  if (LDR_STATE == HIGH) {
    digitalWrite(LED_PIN, HIGH);
    Serial.println("LED ON");

  }
else
{
    digitalWrite(LED_PIN, LOW);
    Serial.println("LED OFF");

  }
}
```



Step 3: Write Program and Hit Simulation



Assignment - 1



Key Takeaways

- ★ MCUs are central to TinyML for low-power, real-time edge AI tasks.
- ★ Know the trade-offs between MCUs, CPUs, and NPUs for optimal hardware selection.
- ★ Sensor choice and memory (RAM/Flash) impact performance significantly.
- ★ Wokwi is a powerful online simulator to prototype TinyML systems without hardware.

Thank You for Your Attention