UUW1624006

INTERNET OF THINGS

Program Studi Informatika
Fakultas Sains dan Matematika
Universitas Diponegoro

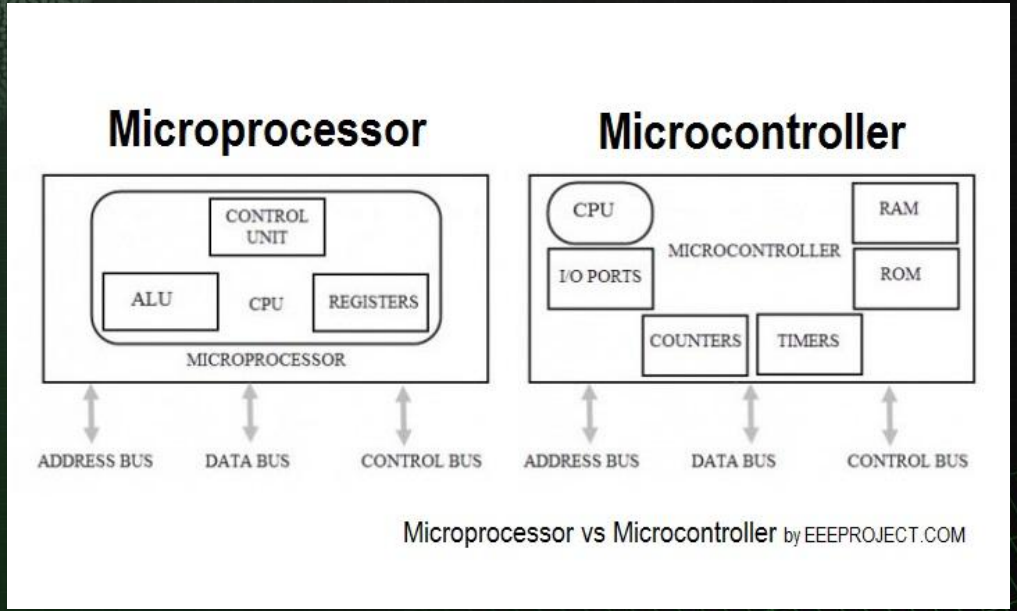
02

Pemrograman IoT

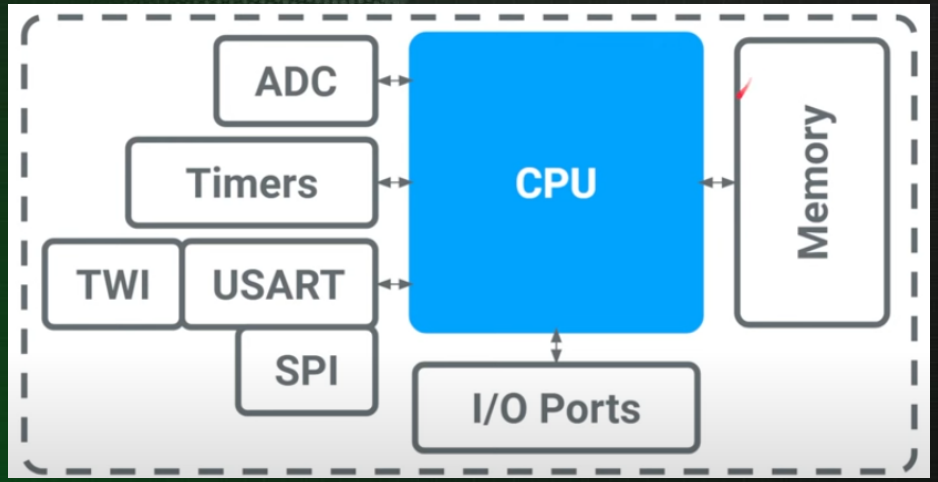
Mikrokontroler

- Mikrokontroler (MCU) adalah komputer kecil dalam satu sirkuit terpadu yang dirancang untuk mengontrol tugas-tugas spesifik dalam sistem elektronik.
- Mikrokontroler menggabungkan fungsi dari unit pemrosesan pusat (CPU), memori, dan antarmuka masukan/keluaran, semuanya dalam satu chip

AVR Microcontroller Family

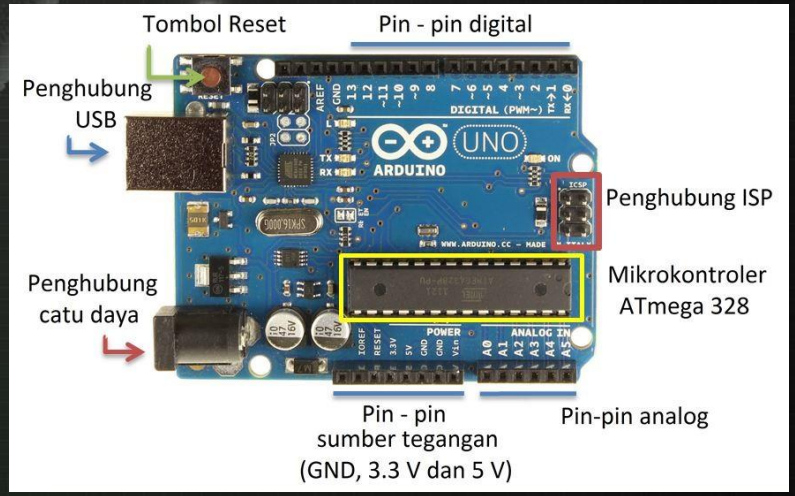


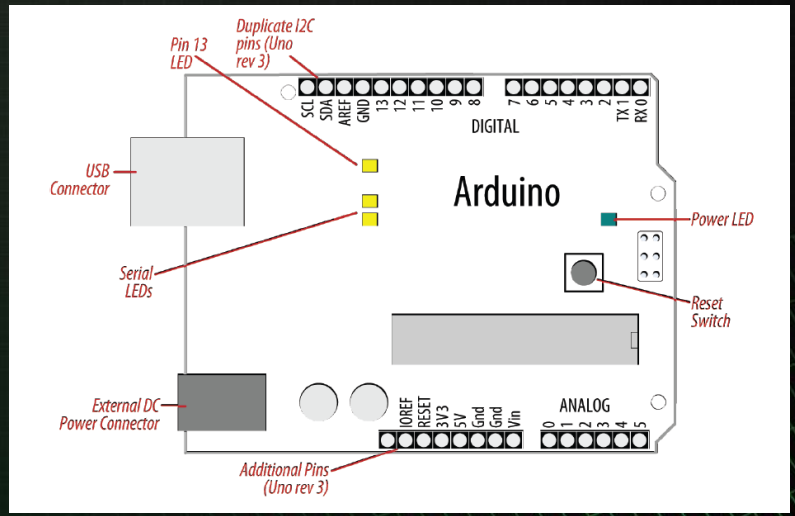
Mikrokontroler



Arduino

- **Arduino** bersifat open-source dengan input/output sederhana. Arduino merupakan kombinasi antara hardware, software, dan integrated development environment (IDE).
- **IDE** untuk Arduino merupakan sebuah software yang berperan untuk menulis program, mengkompilasi menjadi kode biner, dan mengupload kedalam memori mikrokontroller
- Beberapa jenis Arduino: Arduino Uno, Arduino Dueminalove, Arduino Leonardo, Arduino Mega, Arduino Nano





Arduino

Arduino bersifat open-source dengan input/output sederhana. Arduino merupakan kombinasi antara hardware, software, dan integrated development environment (IDE).

IDE untuk Arduino merupakan sebuah software yang berperan untuk menulis program, mengkompilasi menjadi kode biner, dan menguploud kedalam memori mikrokontroller

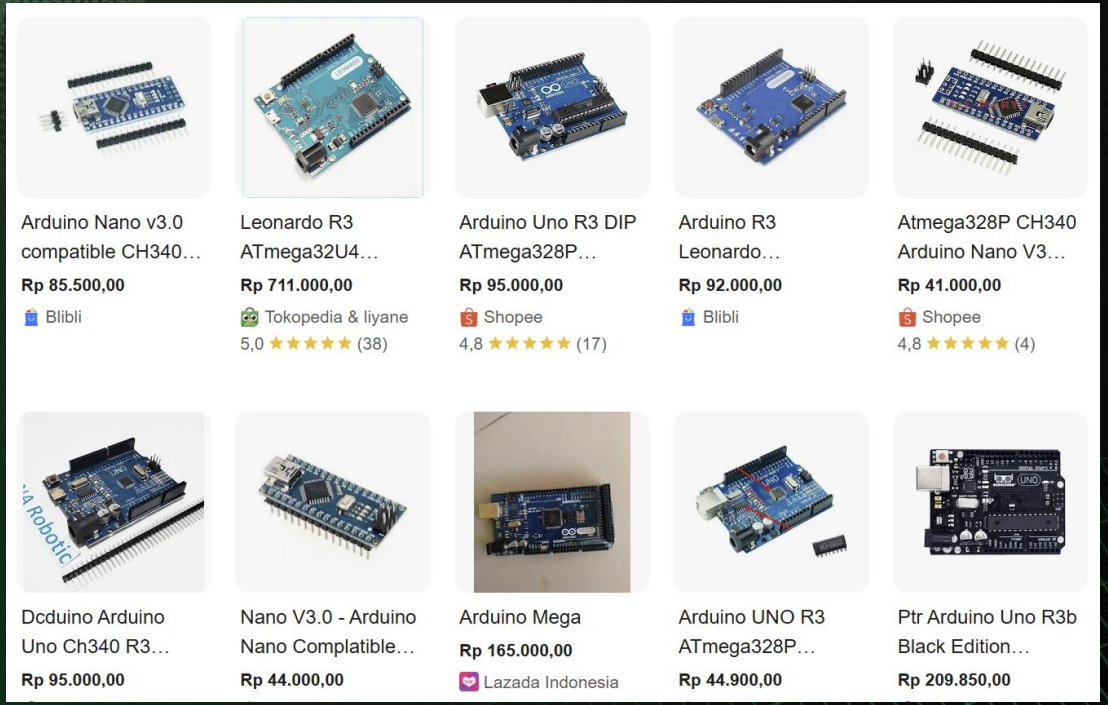
Beberapa jenis Arduino: Arduino Uno, Arduino Dueminalove, Arduino Leonardo, Arduino Mega, Arduino Nano

Arduino

Arduino bersifat open-source dengan input/output sederhana. Arduino merupakan kombinasi antara hardware, software, dan integrated development environment (IDE).

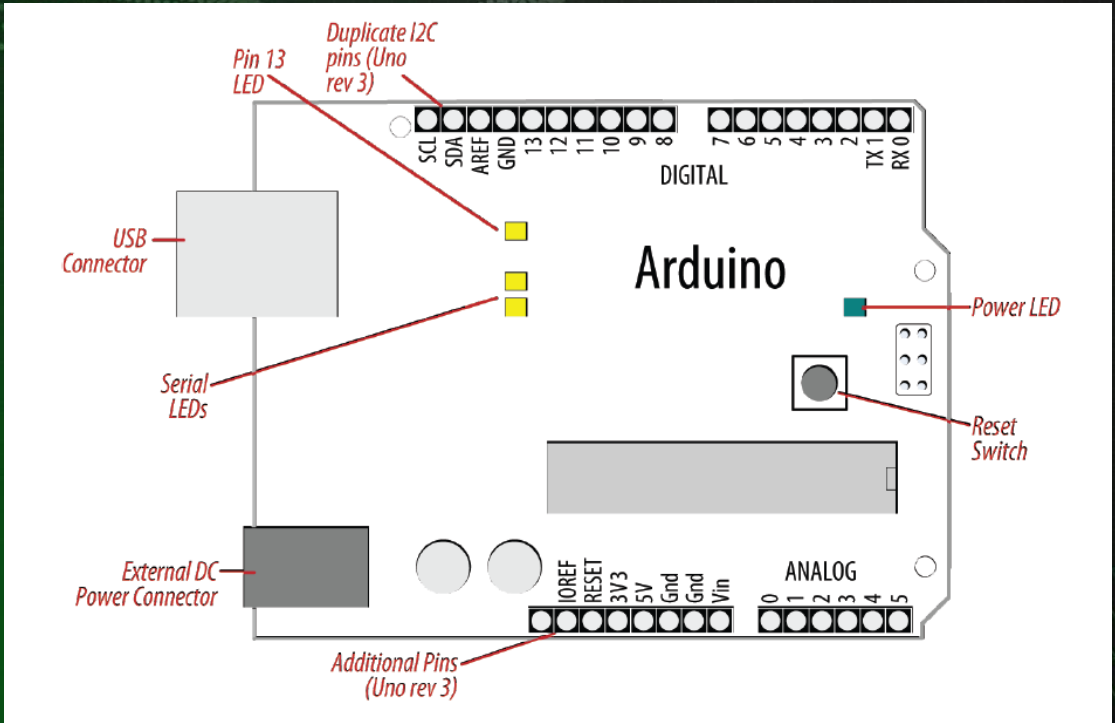
IDE untuk Arduino merupakan sebuah software yang berperan untuk menulis program, mengkompilasi menjadi kode biner, dan mengupload kedalam memori mikrokontroller

Beberapa jenis Arduino: Arduino Uno, Arduino Dueminalove, Arduino Leonardo, Arduino Mega, Arduino Nano

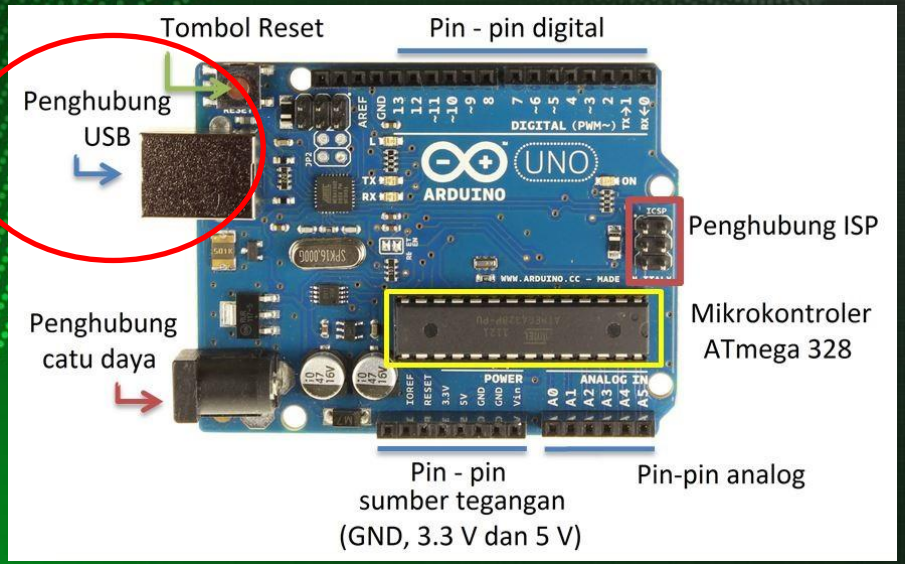


https://youtu.be/XWrAjvwHOr8?si=UHS6cx42V1ngPFGO

Board Arduino

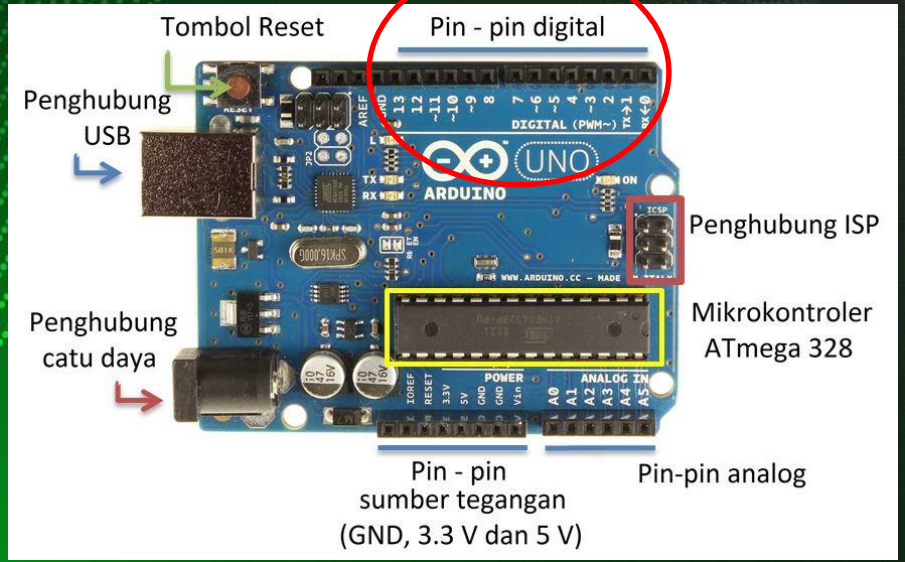


Board Arduino



USB Connector: memuat program dari komputer ke dalam papan komunikasi serial antara papan dan komputer
Memberi daya listrik kepada papan

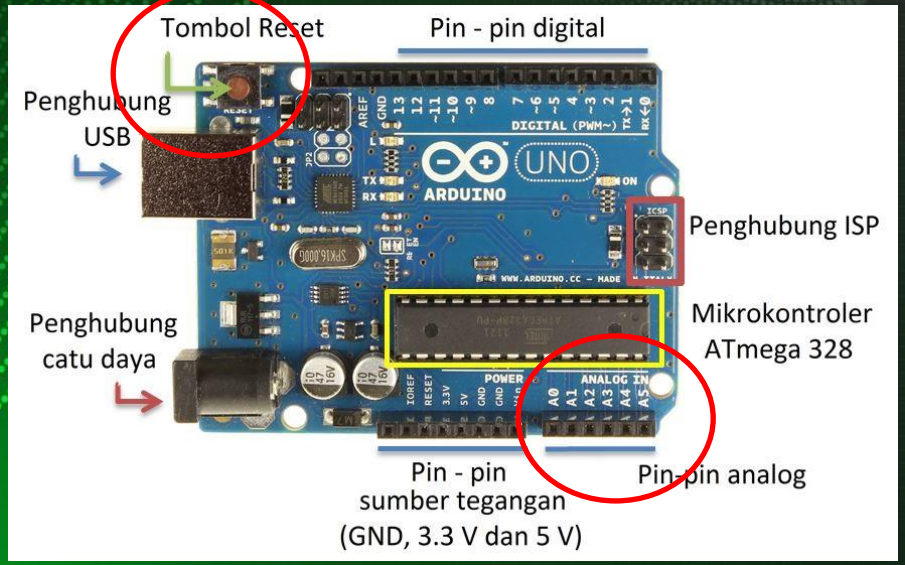
Board Arduino



Digital I/O Pins

- 14 pin input/output digital (0-13)
- Berfungsi sebagai input atau output, dapat diatur oleh program
- Khusus untuk 6 buah pin 3,5,6,9,10 dan 11 dapat juga berfungsi sbg pin analog output dengan tegangan bisa diatur
- Nilai sebuah pin output analog dapat diprogram antara 0-255 yang mewakili 0-5V

Board Arduino

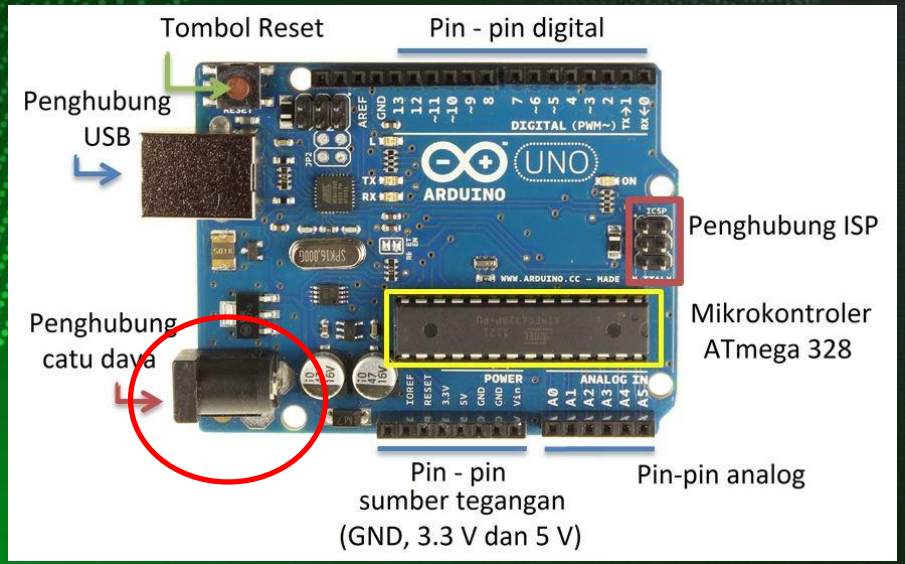


Analog Input Pins

- 6 pin input analog (0-5)
- Pin ini berguna untuk membaca tegangan yang dihasilkan oleh sensor analog.
- Program dapat membaca nilai sebuah pin input 0-5V

Reset button: mereset sehingga papan mulai dari awal

Board Arduino

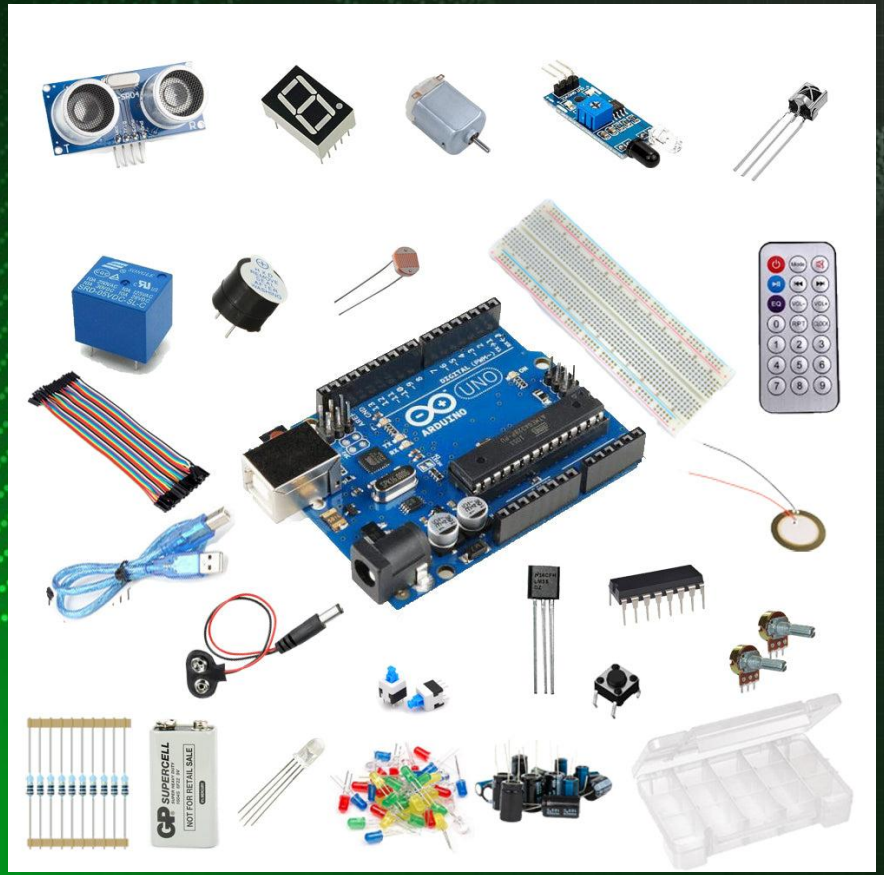


Microcontroller
- Microcontroller Atmega
- Komponen utama dari arduino, didalamnya ada CPU, ROM, RAM

Power Suppley
- Pin power dan ground

Power jack
- Jika hendak di supply dari dengan sumber daya eksternal

Arduino Starter kit



- Arduino Uno Board (papan biru di tengah).
- Breadboard (papan putih berlubang untuk merangkai rangkaian tanpa solder).
- Kabel jumper (male-male, warna-warni).
- Kabel USB (untuk menghubungkan Arduino ke komputer).
- Resistor (berderet dengan kode warna).
- LED berbagai warna.
- Push button / tactile switch (hitam kecil).
- Potensiometer (komponen putar, ada 2 di kanan bawah).
- Kapasitor (elektrolit, hitam dengan kaki panjang-pendek).
- Transistor / sensor suhu (TO-92 package) (hitam kecil berkaki 3).
- IC (Integrated Circuit) (chip hitam panjang).
- Relay (biru besar).
- Buzzer piezo (hitam bulat dengan lubang).
- Motor DC kecil (abu-abu).
- Sensor ultrasonik HC-SR04 (biru dengan dua "mata").
- 7-Segment Display (angka digital merah).
- Sensor cahaya LDR (Light Dependent Resistor) (coklat bulat).
- Sensor infra merah (IR receiver) (hitam kecil berkaki 3).
- Remote control IR (untuk dikombinasikan dengan sensor IR).
- Piezo disk (lingkaran tipis dengan kabel merah-hitam).
- Modul sensor suara / sensor getar.
- Battery clip 9V (hitam dengan kabel merah-hitam).
- Baterai 9V
- Tombol saklar mini (putih kecil).

Arduino Starter kit

arduino Uno r3 starter kit paket…

Rp 379.000,00

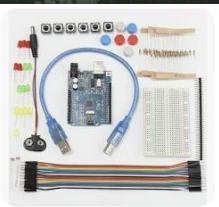
Shopee

4,6 ★★★★½ (138)

Arduino Uno Starter Learning Kit…

Rp 480.000,00

Shopee



Basic Starter Kit Arduino - UNO R3…

Rp 155.000,00

Lazada Indonesia

arduino Uno r3 dip starter kit paket…

Rp 429.500,00

Shopee

4,6 ★★★★½ (22)

Grove Beginner Kit For Arduino All In…

Rp 1.155.000,00

Tokopedia

Beginner Arduino Learning Kit

Rp 480.000,00

Shopee

Paket Arduino Uno R3 Starter Kit Pak…

Rp 200.000,00 Bekas

Shopee

Starter Kit Arduino uno ch340_

Rp 130.000,00

TikTok Shop - TikT…

Arduino Starter Kit Level 1

Rp 310.000,00

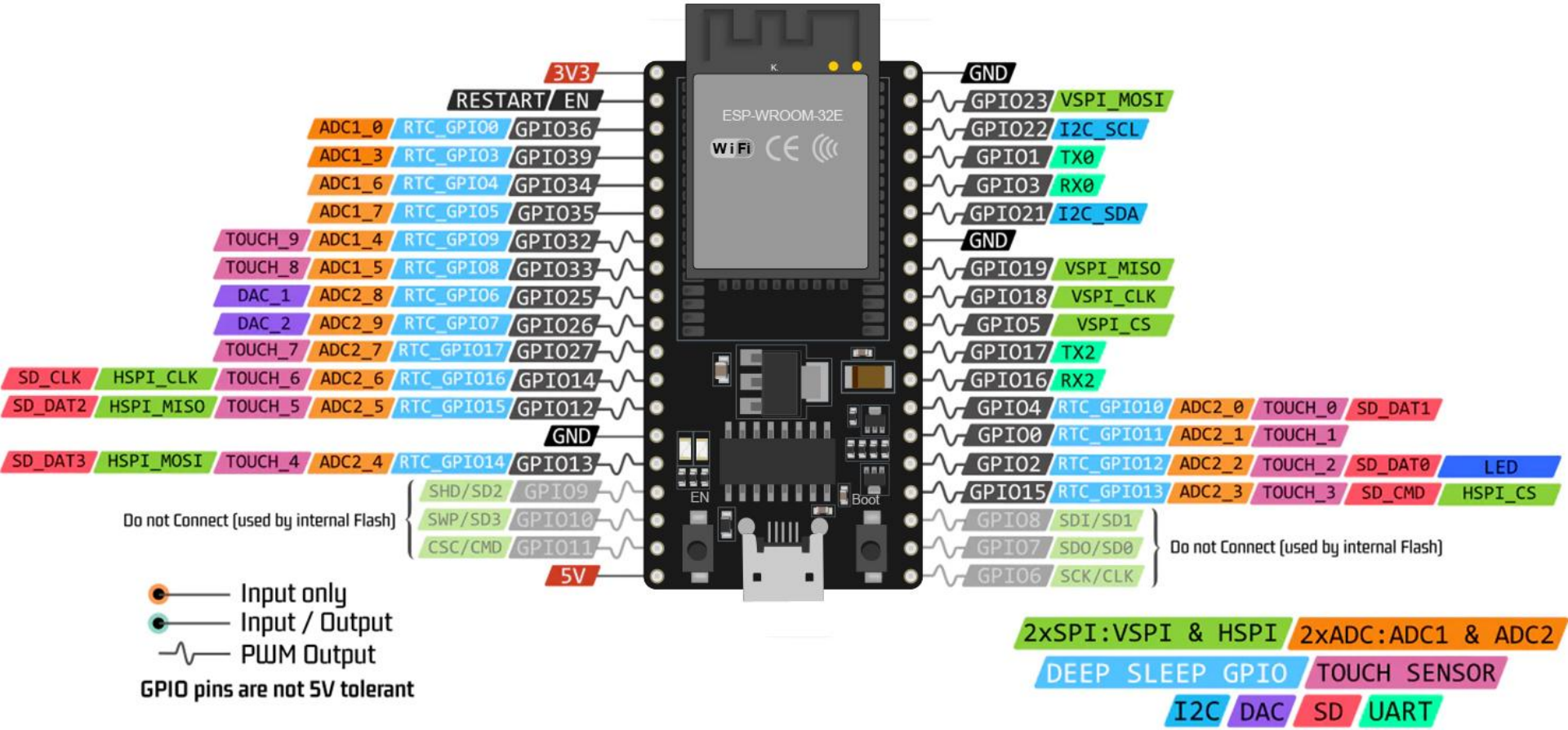
Shopee

Arduino Uno R3 Dip Learning Starter K…

Rp 425.000,00

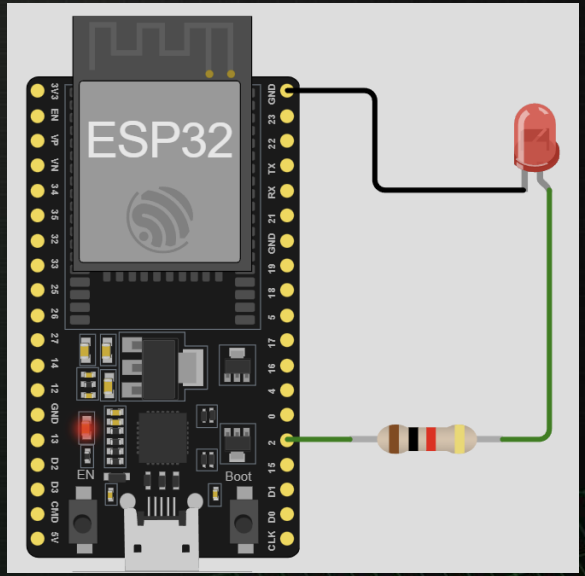
Lazada Indonesia

ESP32 WROOM 32E Pinout



ESP32

- ESP32 merupakan mikrokontroler modern 32-bit dengan prosesor lebih cepat, memori lebih besar, serta sudah terintegrasi Wi-Fi dan Bluetooth, sehingga jauh lebih powerful (dibandingkan dengan Arduino) dan fleksibel untuk proyek kompleks, khususnya yang membutuhkan koneksi internet dan aplikasi Internet of Things (IoT).
- **Arduino Uno**: cocok untuk belajar dasar mikrokontroler, rangkaian sederhana, kontrol LED, motor, sensor dasar.
- **ESP32**: cocok untuk proyek IoT, perangkat pintar, aplikasi yang butuh Wi-Fi/Bluetooth atau pemrosesan lebih cepat



https://youtu.be/Vqs1Qo0So1Q?si=NVQO_kPgmN8hlXb8

ESP32

Esp32 Module Wifi+Bluetooth…

Rp 91.000,00

Tokopedia

ESP32 DOIT WIFI IOT ESP-32S…

Rp 52.850,00

Shopee

4,7 ★★★★½ (55)

Wroom-32U Chip Esp32 Esp-32U W…

Rp 39.168,00

Lazada Indonesia

5,0 ★★★★★ (5)

Esp32 Papan Pengembangan…

Rp 230.300,00

Tokopedia

5,0 ★★★★★ (3)

PROMO

ESP32 ESP-32 DOIT WIFI…

Rp 71.900,00

~~Rp 92.900~~

Shopee

4,8 ★★★★★ (59)

ESP32 ESP-32 IOT Wireless Bluetoot…

Rp 72.000,00

Blibli

ESP32 Devkit V1 USB Type C Inter…

Rp 70.000,00

Shopee

5,0 ★★★★★ (5)

Esp32 Wifi

Rp 110.000,00

Tokopedia

4,4 ★★★★½ (9)

ESP32 WiFi + Bluetooth IOT Du…

Rp 53.800,00

Shopee

4,4 ★★★★½ (8)

Esp32 Esp-32 Wifi+Bluetooth Du…

Rp 15.000,00

Lazada Indonesia

ESP32 Starter kit

Paket Starter kit IoT ESP32 Nodemcu…

Rp 188.200,00

Shopee

3,7 ★★★½☆ (6)

Nk Professional Esp32 Automation…

Rp 1.251.000,00

Tokopedia

esp32 kit nodemcu paket iot internet o…

Rp 345.000,00

Shopee

5,0 ★★★★★ (7)

MC TRAINER KIT ESP32 MODULE…

Rp 599.900,00

Shopee

ESP32 WIFI IOT BASIC LEARNIN…

Rp 372.500,00

Shopee

5,0 ★★★★★ (1)

Papan inti ESP32-DevKitC ESP32-…

Rp 68.500,00

Tokopedia & liyane

4,6 ★★★★½ (201)

Esp32 starter kit Wifi module DIY ESP-…

Rp 340.000,00

Shopee

5,0 ★★★★★ (1)

M5Stack Core2 - ESP32 IoT…

Rp 1.753.830,00

Blibli

ESP 32 DEV KIT - ESP32 DEVKIT I…

Rp 59.500,00

Shopee

5,0 ★★★★★ (6)

Esp 32 Wroom 32 Dev Kit V1 Kit V1…

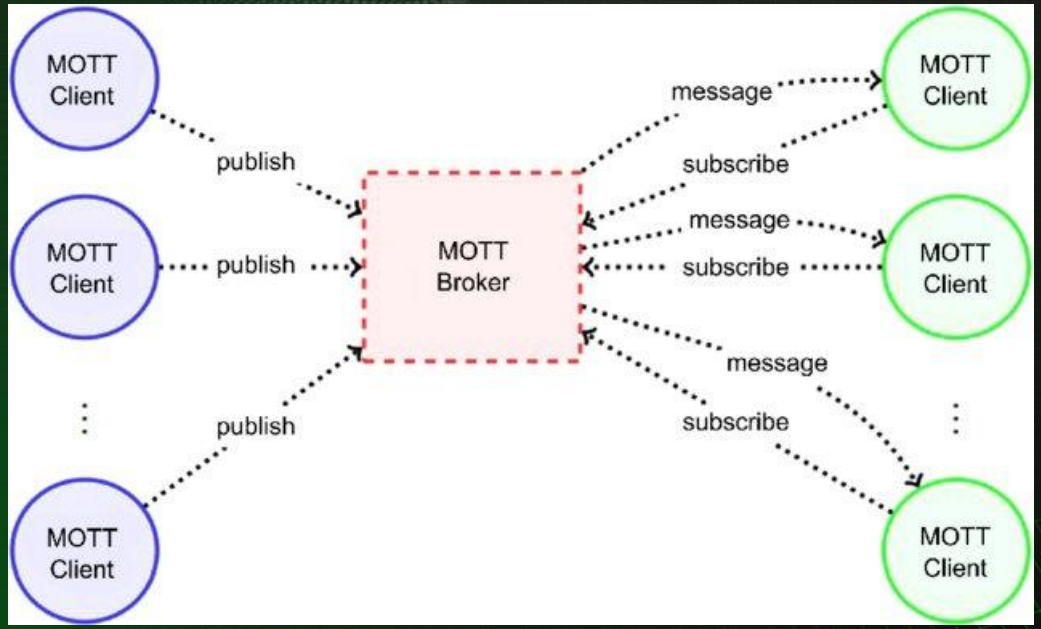
Rp 126.000,00

Tokopedia

Message Queuing Telemetry Transport (MQTT)

- MQTT dirancang untuk komunikasi ringan, efisien, dan andal di jaringan dengan bandwitdth rendah dan perangkat yang memiliki sumber daya terbatas, misalnya ESP32
- MQTT Adalah protocol komunikasi berbasis publish-subscribe (pub-sub)
- MQTT memungkinkan perangkat saling bertukar pesan melalui broker sebagai perantara
  - Perangkat yang mengirim disebut Publisher
  - Perangkat yang menerima disebut Subscriber
  - Publisher dan Subscriber terhubung ke Broker MQTT
- Pesan dikirimkan berdasarkan topik tertentu
- Keunggulan MQTT: ringan dan hemat bandwidth, asyncron dan real-time, skalabilitas tinggi, mendukung QoS

Message Queuing Telemetry Transport (MQTT)



Simulator Untuk Mikrokontroler

- Simulator untuk Mikrokontroler merupakan aplikasi yang memodelkan perilaku perangkat keras mikrokontroler beserta perangkat-perangkat yang berinteraksi dengannya tanpa menggunakan perangkat keras secara fisik
- Dengan menggunakan simulator, pengguna dapat menghemat biaya dan waktu, meminimalkan resiko (lebih aman), dibandingkan menggunakan perangkat keras fisik.
- Namun, ketersediaan perangkat yang tersedia dalam simulator terbatas.
- Contoh aplikasi simulator: Wokwi, Tinkercad, Proteus, SumulIDE, Friting, CircuitMaker, dsb

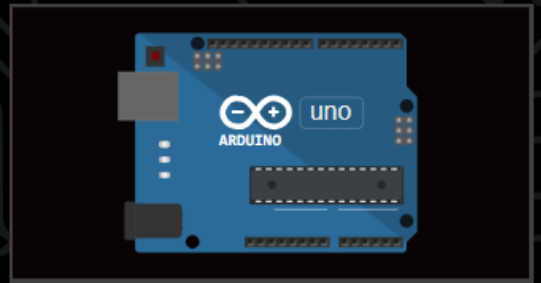
wokwi.com

WOKWI

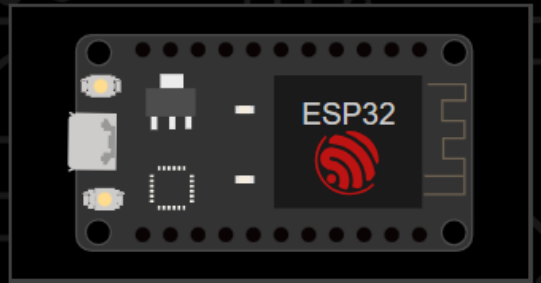
Simulate IoT Projects in Your Browser

Discord Community LinkedIn Group

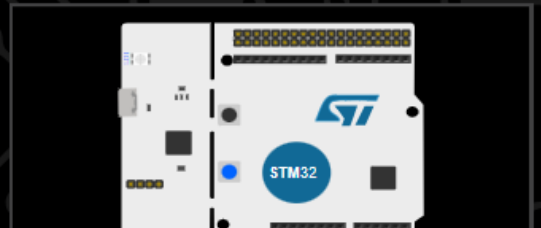
Simulate with Wokwi Online

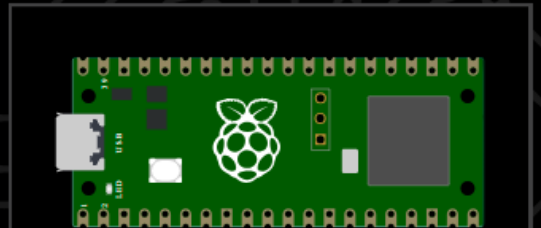


Arduino (Uno, Mega, Nano)



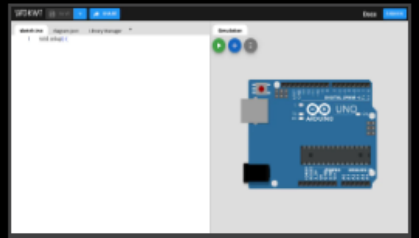
ESP32



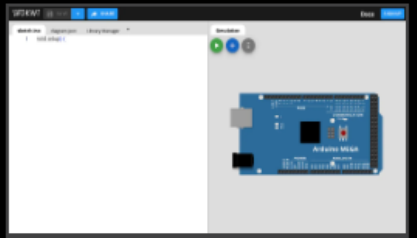


wokwi.com/arduino

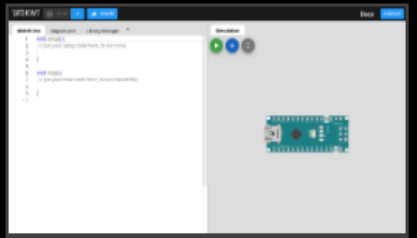
Start from Scratch



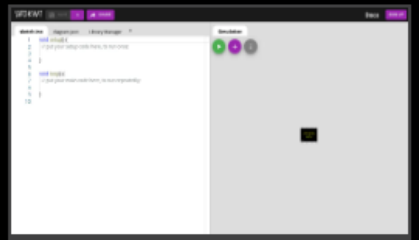
Arduino Uno



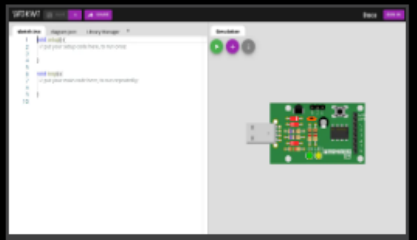
Arduino Mega



Arduino Nano



ATtiny85



Franzininho

Latest projects

Total of 1791536 Arduino projects

Wokwi Compiler

WOKWI SAVE SHARE

sketch.ino ● diagram.json ● libraries.txt ●

Library Manager

```
1  #include <WiFi.h>
2  #include <PubSubClient.h>
3
4  const char* ssid = "Wokwi-GUEST";
5  const char* password = "";
6  const char* mqtt_server = "broker.hivemq.com";
7
8  WiFiClient espClient;
9  PubSubClient client(espClient);
10
11 const char* topic = "Lights";
12 const int ledPin = 2;
13
14 void setup_wifi() {
15   WiFi.begin(ssid, password);
16   while (WiFi.status() != WL_CONNECTED) {
17     delay(500);
18   }
19 }
20
21 void callback(char* topic, byte* payload, unsigned
22   if ((char)payload[0] == '1') {
23     digitalWrite(ledPin, HIGH);
24   } else {
25     digitalWrite(ledPin, LOW);
26   }
27 }
```

Simulation

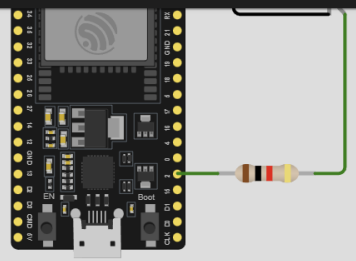
Start the simulation

Compiling project... 3.40s CANCEL

Upgrade to a paid plan to skip the build queue and compile faster.

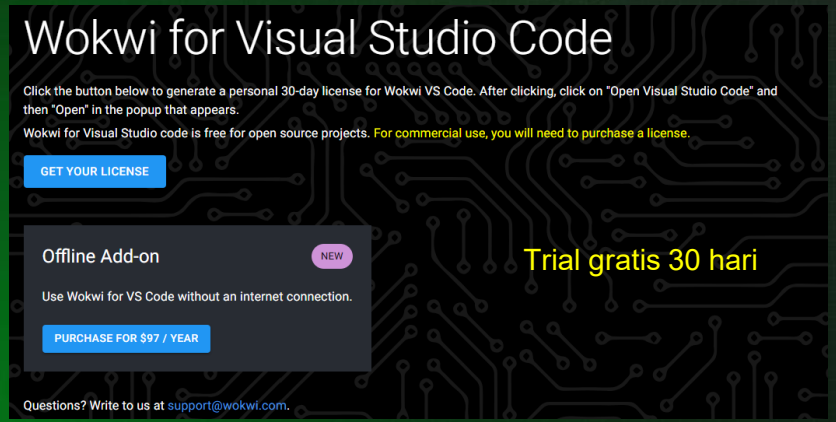
WOKWI PLANS

Build server load



Wokwi for VS Code

Tutorial install Wokwi di VS Code
https://youtu.be/gZ_NdmcSpcc?si=DEwKn6VCv0tU32Ac



Trial gratis 30 hari

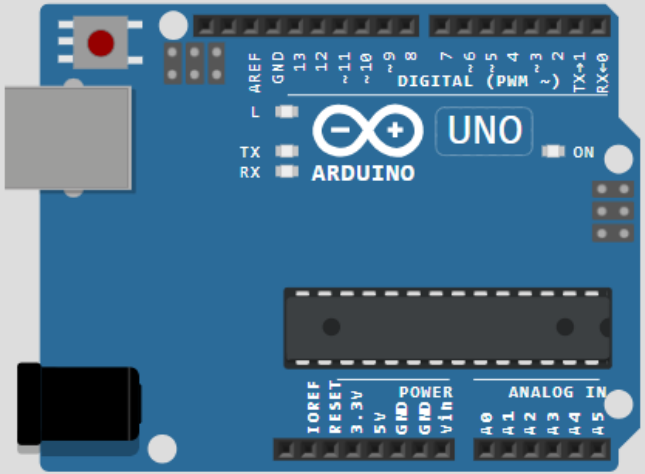
wokwi.com/projects/new/arduino-uno

WOKWI SAVE SHARE

sketch.ino | diagram.json ● | Library Manager

```
1  void setup() {
2    // put your setup code here, to run once:
3
4  }
5
6  void loop() {
7    // put your main code here, to run repeatedly:
8
9  }
10
```

Simulation



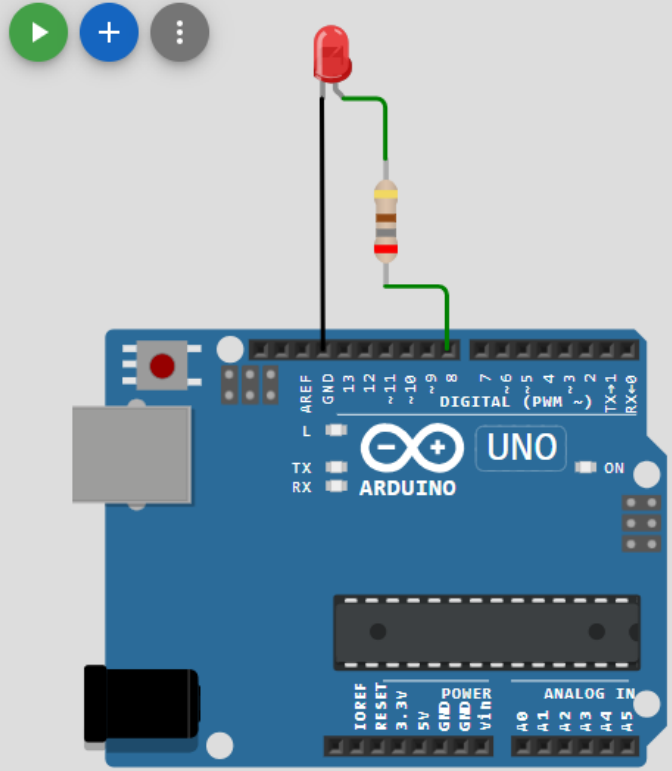
wokwi.com/projects/395914381723551745

WOKWI SAVE SHARE simple_led

sketch.ino ● diagram.json ● Library Manager

```
#define LED 8

void setup() {
  // put your setup code here, to run once:
  pinMode(LED, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(LED, HIGH);
  delay(500);
  digitalWrite(LED, LOW);
  delay(500);
}

```

Simulation



https://wokwi.com/projects/395914381723551745

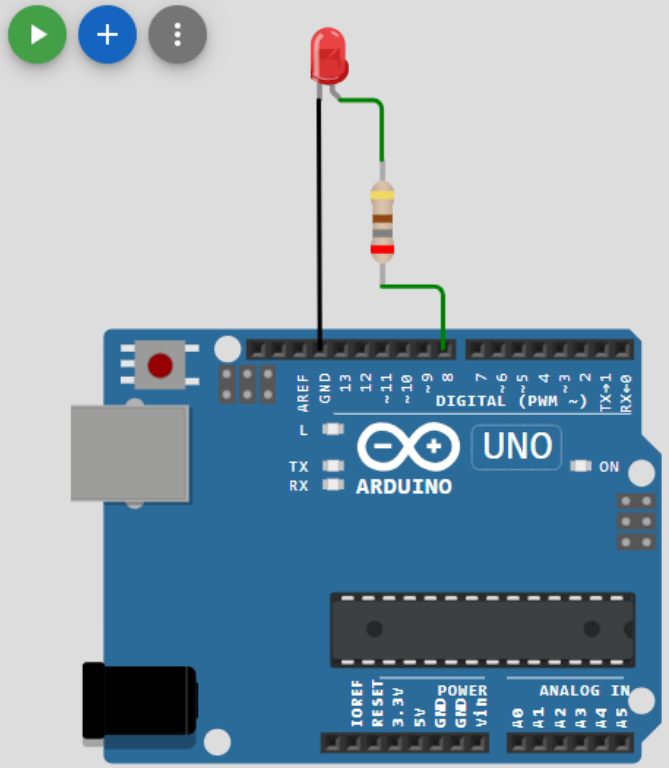
wokwi.com/projects/395914381723551745

WOKWI SAVE SHARE simple_led Docs

sketch.ino ● diagram.json ● Library Manager

```
{
  "version": 1,
  "author": "Indra Waspada",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-arduino-uno", "id": "uno", "top": 10.2, "left": -39, "attrs": {} },
    { "type": "wokwi-led", "id": "led1", "top": -138, "left": 61.4, "attrs": { "color": "red" } },
    {
      "type": "wokwi-resistor",
      "id": "r1",
      "top": -44.6,
      "left": 75.95,
      "rotate": 270,
      "attrs": { "value": "280" }
    }
  ],
  "connections": [
    [ "led1:C", "uno:GND.1", "black", [ "h0.4", "v115.2" ] ],
    [ "led1:A", "r1:2", "green", [ "h19.2", "v27.6" ] ],
    [ "r1:1", "uno:8", "green", [ "h0" ] ]
  ],
  "dependencies": {}
}
```

Simulation



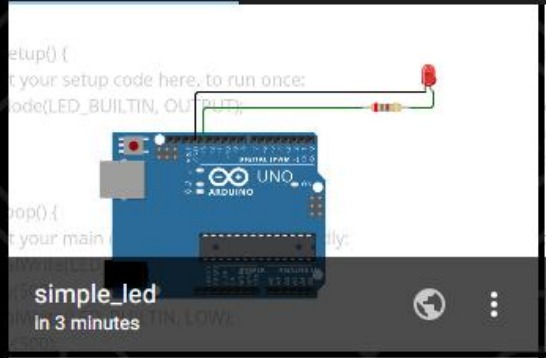
https://wokwi.com/projects/395914381723551745

wokwi.com/dashboard/projects

WOKWI

NEW PROJECT

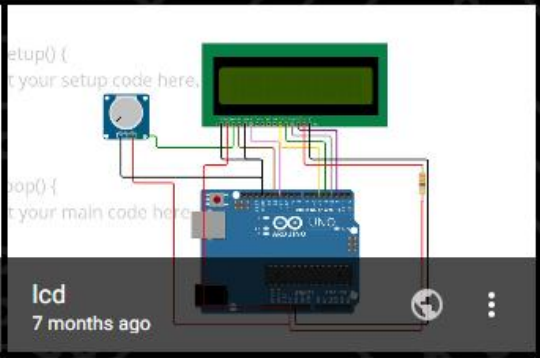
YOUR PROJECTS | YOUR LIKES ♥



simple_led
in 3 minutes

ESP32-2
7 months ago

ESP32-scan
7 months ago



lcd
7 months ago

Welcome to Wokwi!

Wokwi is an online Electronics simulator. You can use it to simulate Arduino, ESP32, STM32, and many other popular boards, parts and sensors.

Here are some quick examples of things you can make with Wokwi:

- Arduino Uno "Hello World"
- Blink an LED on ESP32
- Monitor the weather on ATtiny85
- Control 32 Servos with Arduino Mega
- Animate an LED Matrix with FastLED
- 7 Segment Counter with MicroPython on ESP32

Why Wokwi?

Start right now. No waiting for components, or downloading large software. Your browser has everything you need to start coding your next IoT project in seconds.

Mistakes are okay. You can't destroy the virtual hardware. Trust us, we tried. So don't worry about frying your precious components. And unlike real hardware, you can always undo.

Easy to get help and feedback. Sharing a link to your Wokwi project is all you need.

Gain confidence in your code. Separate hardware and software issues.

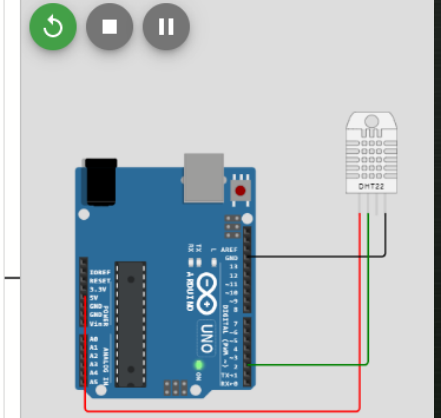
Unlimited hardware. No need to scavenge parts from old projects. Use as many parts as you need, without worrying about project price and stock.

Pengukur Suhu Ruangan

sketch.ino ● | diagram.json | libraries.txt | Library Manager

```
#include <DHT.h>

#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

float temp;


void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  // put your main code here, to run repeatedly:
  temp = dht.readTemperature();
  Serial.print(F("Temperature: "));
  Serial.println(temp);
  delay(1000);
}

```

Simulation



```
Temperature: 37.40
Temperature: 37.40
Temperature: 37.40
Temperature: 37.40
```

Project Libraries

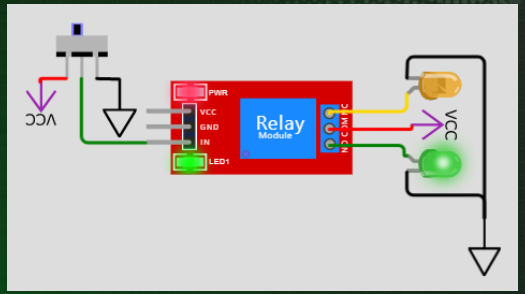
Installed Libraries

DHT sensor library

https://docs.wokwi.com/parts/wokwi-dht22

Digital Humidity and Temperature sensor.
DHT22

Relay



Relay modul
Electrically operated switch

https://docs.wokwi.com/parts/wokwi-relay-module

Name	Description
VCC	Supply voltage
GND	Ground
IN	Control signal (e.g. from micro-controller)
NC	Normally closed
COM	Common pin
NO	Normally open

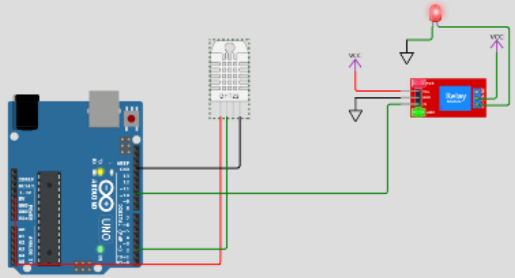
Pengatur Suhu Ruangan

sketch.ino ● | diagram.json ● | libraries.txt | Library Manager

```ino
#include <DHT.h>

#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

float temp;
const int relayPin = 10;

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

void loop() {
  temp = dht.readTemperature();
  Serial.print(F("Temperature: "));
  Serial.println(temp);
  if (temp<37){
    digitalWrite( relayPin, HIGH);
  } else if (temp>39){
    digitalWrite(relayPin, LOW);
  }
  delay(1000);
}
```

Simulation

Editing DHT22
Temperature: 29.3°C
Humidity: 50.5%



```
Temperature: 39.40
Temperature: 31.30
Temperature: 31.30
Temperature: 29.30
Temperature: 29.30
Temperature: 29.30
Temperature: 29.30
```

Sensor = suhu (DHT22)
Actuator/ output = lampu (led)

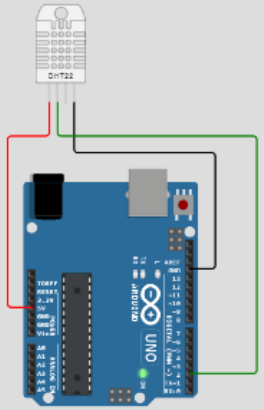
Lampu ON = Menghangatkan ruangan

Pengukur kelembaban ruangan

sketch.ino ● diagram.json ● libraries.txt ● Library Manager

```
#include <DHT.h>

#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

float hum;


void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  // put your main code here, to run repeatedly:
  hum = dht.readHumidity();
  Serial.print(F("Humidity: "));
  Serial.print(hum);
  Serial.println("%");
  delay(1000);
}

```

Simulation



```
Humidity: 65.00%
Humidity: 65.00%
Humidity: 65.00%
Humidity: 65.00%
```

Project Libraries

Installed Libraries

DHT sensor library

https://docs.wokwi.com/parts/wokwi-dht22

Digital Humidity and Temperature sensor.
DHT22

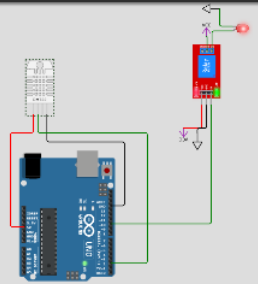
Pengatur kelembaban ruangan

sketch.ino ● diagram.json ● libraries.txt Library Manager

```ino
#include <DHT.h>

#define DHTPIN 2
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

float hum;
const int relayPin = 10;

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

void loop() {
  hum = dht.readHumidity();
  Serial.print(F("Humidity: "));
  Serial.print(hum);
  Serial.println("%");
  if (hum<40){
    digitalWrite(relayPin, LOW); // exhaust fan OFF
  }else if (hum>70){
    digitalWrite( relayPin, HIGH); // exhaust fan ON
  }
  delay(1000);
}
```

Simulation

Editing DHT22
Temperature: 24.0°C
Humidity: 71.5%



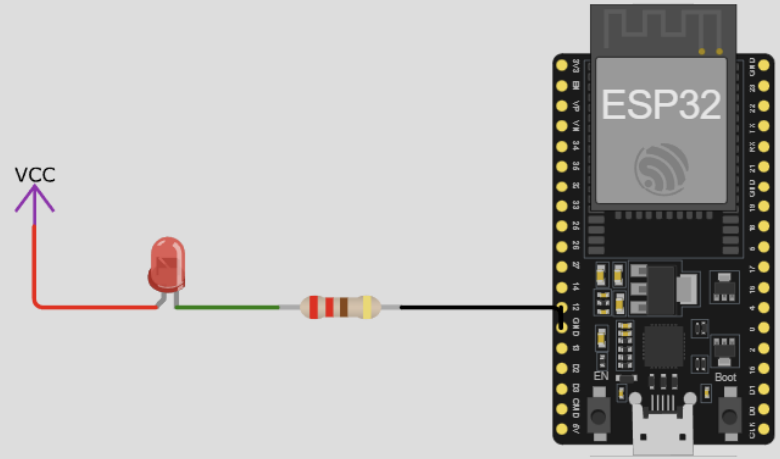
```
Humidity: 60.00%
Humidity: 70.00%
Humidity: 70.00%
Humidity: 71.50%
Humidity: 71.50%
Humidity: 71.50%
Humidity: 71.50%
```

WOKWi SAVE SHARE

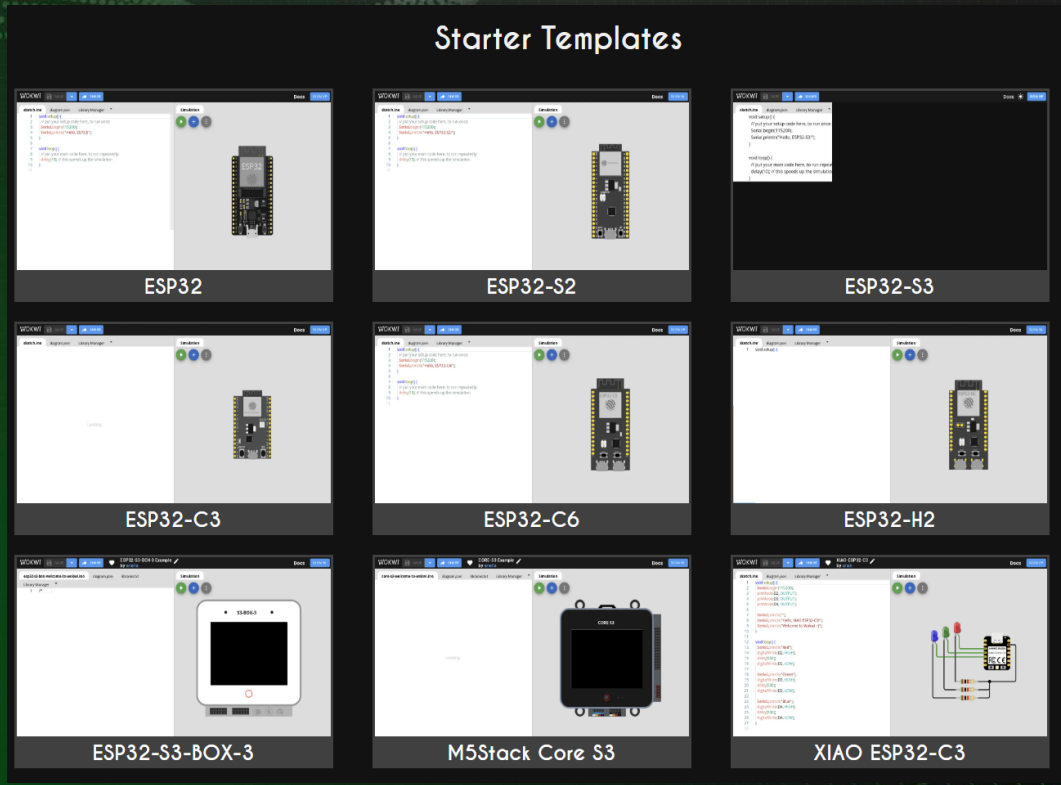
sketch.ino ● diagram.json ● Library Manager

```
1  #define LED1 12
2
3  void setup() {
4    pinMode(LED1, OUTPUT);
5  }
6
7  void loop() {
8    digitalWrite(LED1, HIGH);
9    delay(1000);
10   digitalWrite(LED1, LOW);
11   delay(1000); // this speeds up the simulation
12 }
13
```

Simulation



ESP32 di Wokwi



File Edit Selection View Go Run ···

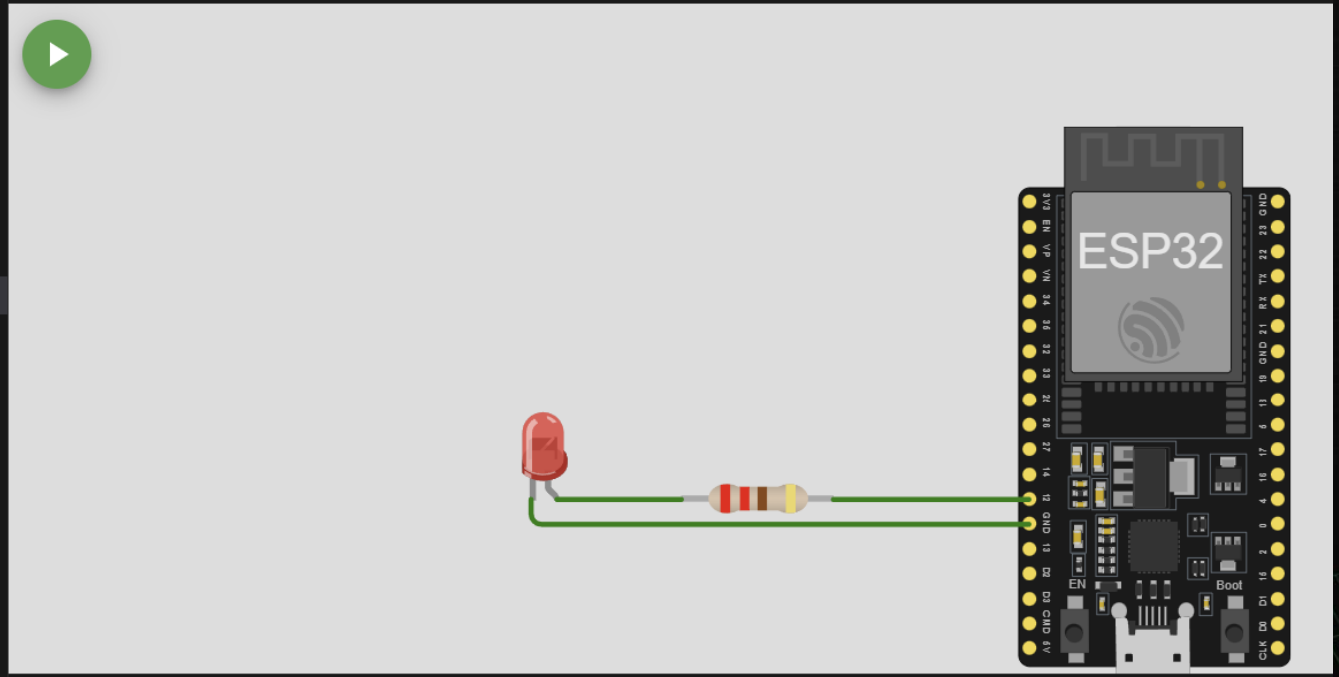
Untitled (Workspace)

EXPLORER: UNTITLED (W...

- Test
  - .pio
  - .vscode
  - include
  - lib
  - src
    - main.cpp
  - test
  - .gitignore
  - diagram.json
  - platformio.ini
  - wokwi.toml

main.cpp | platformio.ini | diagram.json | Wokwi Simulator | wokwi.toml | diagram.j

WOKWi Simulator

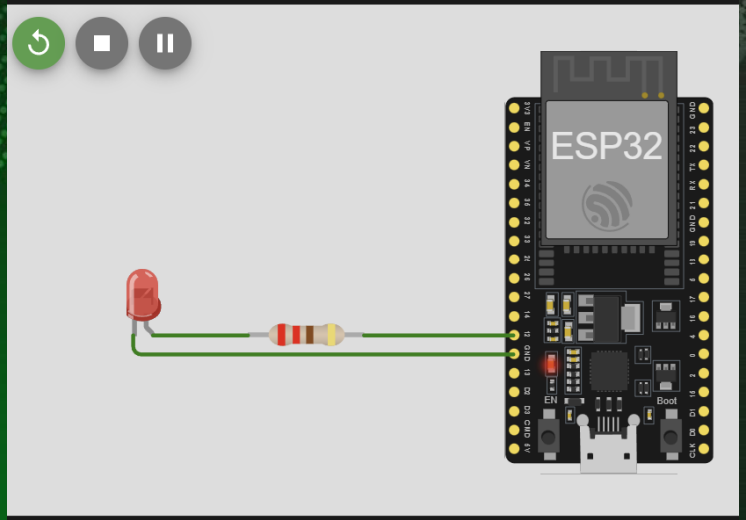


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
LED HIGH
LED LOW
LED HIGH
LED LOW
LED HIGH
```

Contoh Modifikasi

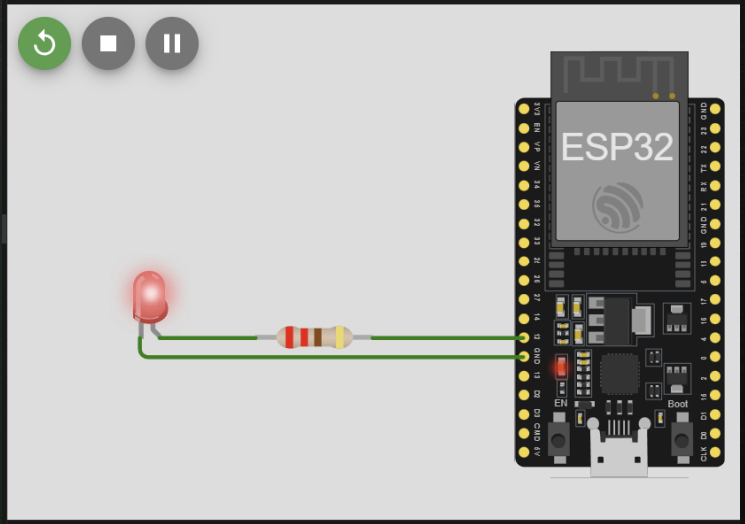
WOKWi Simulator



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Permainan Tebak Angka 0-20
Tebak angka yang benar! Anda punya 4 kesempatan.
Masukkan tebakan Anda: 10 : Terlalu besar.
Sisa kesempatan: 3
Masukkan tebakan Anda: 5 : Terlalu besar.
Sisa kesempatan: 2
Masukkan tebakan Anda: 2 : Terlalu besar.
Sisa kesempatan: 1
Masukkan tebakan Anda: 1 : Terlalu besar.
Kesempatan habis! Angka yang benar: 0
```

WOKWi Simulator



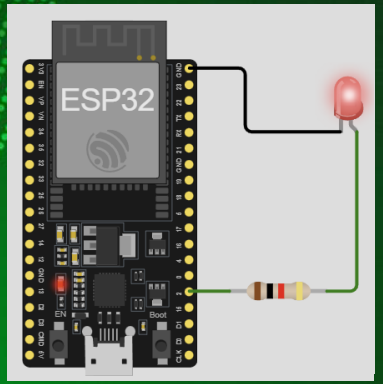
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
load:0x40080400,len:2972
entry 0x400805dc
Permainan Tebak Angka 0-20
Tebak angka yang benar! Anda punya 4 kesempatan.
Masukkan tebakan Anda: 10 : Terlalu kecil.
Sisa kesempatan: 3
Masukkan tebakan Anda: 15 : Terlalu besar.
Sisa kesempatan: 2
Masukkan tebakan Anda: 13: Selamat! Tebakan Anda benar.
Score Anda: 50
```

Program
Kontrol LED via MQTT

MQTT CLIENT

Subscriber: Actuator

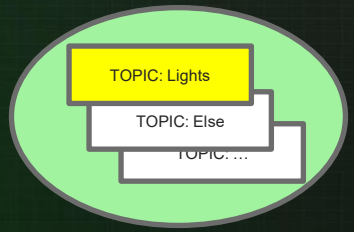


Subscribe to topic: Lights

Publish: On

HIVEMQ CLOUD

MQTT BROKER



Publish to topic: Lights

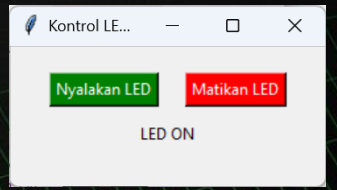
Publish: On

Tkinter

MQTT CLIENT

Publisher:

UI Button “On”



Kode Program

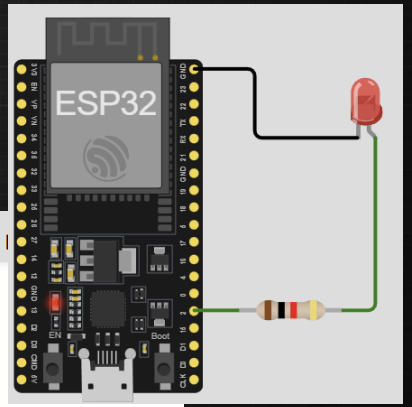
sketch.ino ● diagram.json ● libraries.txt ● Library Manager

```
#include <WiFi.h>
#include <PubSubClient.h>

const char* ssid = "Wokwi-GUEST";
const char* password = "";
const char* mqtt_server = "broker.hivemq.com";

WiFiClient espClient;
PubSubClient client(espClient);

const char* topic = "IoT-IF-Undip";
const int ledPin = 2;

void setup_wifi() {
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
}

void callback(char* topic, byte* payload, unsigned int length) {
  if ((char)payload[0] == '1') {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

HIVEMQ CLOUD

sketch.ino ● diagram.json ● libraries.txt ● Library

```
}

void reconnect() {
  while (!client.connected()) {
    if (client.connect("esp32-client")) {
      client.subscribe(topic);
    } else {
      delay(1000);
    }
  }
}

void setup() {
  pinMode(ledPin, OUTPUT);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
}
```



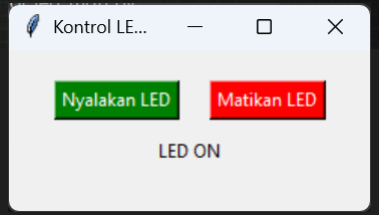
Kode Program

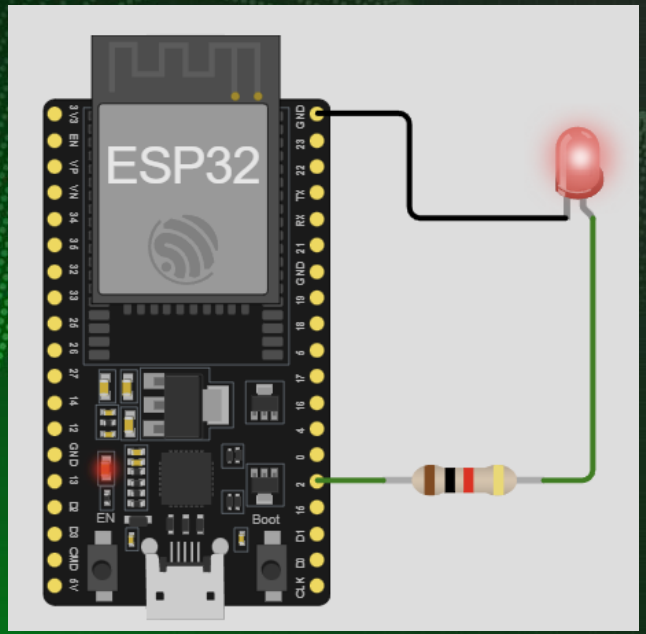
kontrol_led_mqtt.py > ...

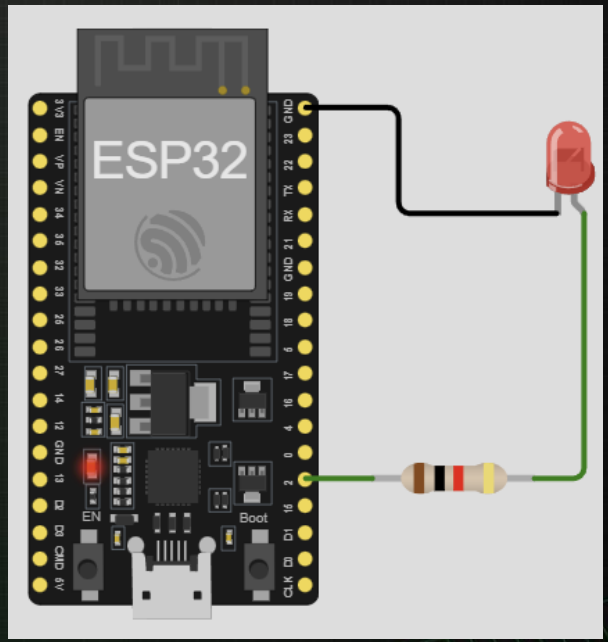
```python
import tkinter as tk
import paho.mqtt.client as mqtt

broker = "broker.hivemq.com"
topic = "IoT-IF-Undip"

def send_on():
    client.publish(topic, "1")
    status_label.config(text="LED ON")

def send_off():
    client.publish(topic, "0")
    status_label.config(text="LED OFF")

client = mqtt.Client()
client.connect(broker, 1883, 60)
```

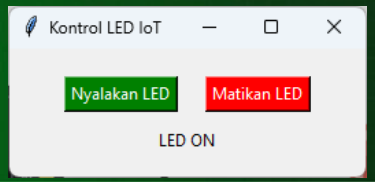
HIVEMQ CLOUD

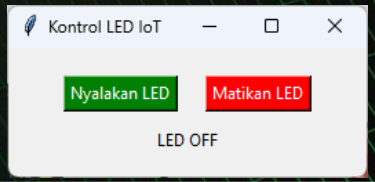
```python
# --- GUI ---
root = tk.Tk()
root.title("Kontrol LED IoT")

frame = tk.Frame(root, padx=20, pady=20)
frame.pack()

on_btn = tk.Button(frame, text="Nyalakan LED", command=send_on, bg="green", fg="white")
on_btn.grid(row=0, column=0, padx=10)

off_btn = tk.Button(frame, text="Matikan LED", command=send_off, bg="red", fg="white")
off_btn.grid(row=0, column=1, padx=10)

status_label = tk.Label(frame, text="Status: -")
status_label.grid(row=1, column=0, columnspan=2, pady=10)

root.mainloop()
```




ESP32
EN
Boot


ESP32
EN
Boot


Kontrol LED IoT
Nyalakan LED
Matikan LED
LED ON


Kontrol LED IoT
Nyalakan LED
Matikan LED
LED OFF

Modifikasi dan Pengembangan?

Proyek 01

Kesimpulan

- Mikrokontroler (MCU) adalah komputer kecil dalam satu sirkuit terpadu yang dirancang untuk mengontrol tugas-tugas spesifik dalam sistem elektronik.
- ESP32 sudah terintegrasi Wi-Fi dan Bluetooth, sehingga sangat sesuai dan fleksibel untuk proyek kompleks, khususnya yang membutuhkan koneksi internet dan aplikasi Internet of Things (IoT).
- Dengan menggunakan simulator, pengguna dapat menghemat biaya dan waktu, meminimalkan resiko (lebih aman), dibandingkan menggunakan perangkat keras fisik.
- MQTT Adalah protocol komunikasi berbasis publish-subscribe (pub-sub) yang memungkinkan perangkat saling bertukar pesan melalui broker sebagai perantara

Terimakasih