



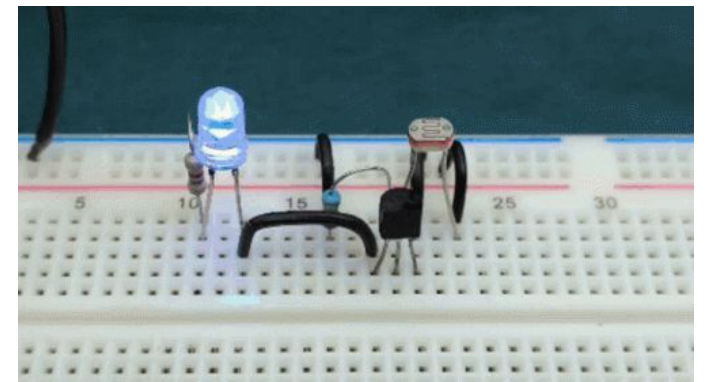
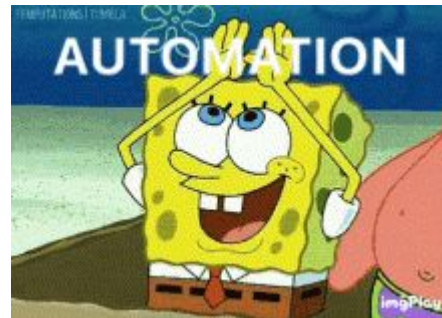
Android + Arduino

SEU APP CONTROLANDO O MUNDO REAL VIA BLUETOOTH

A solid blue horizontal bar at the bottom of the slide.

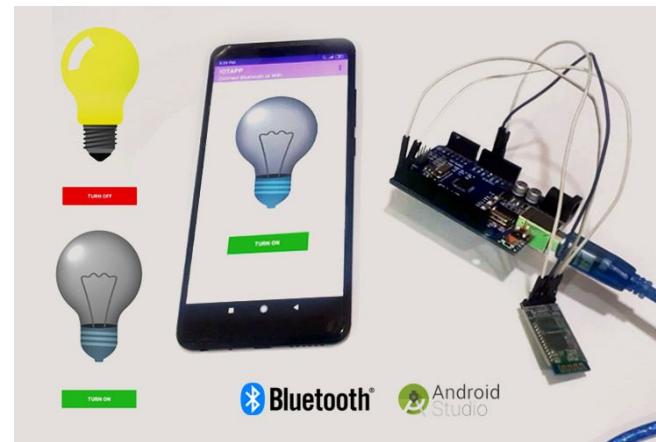
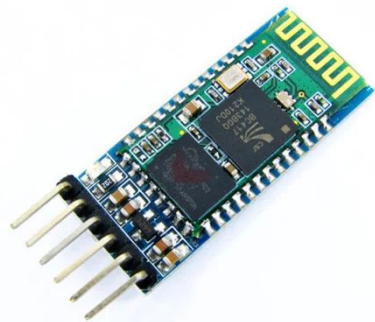
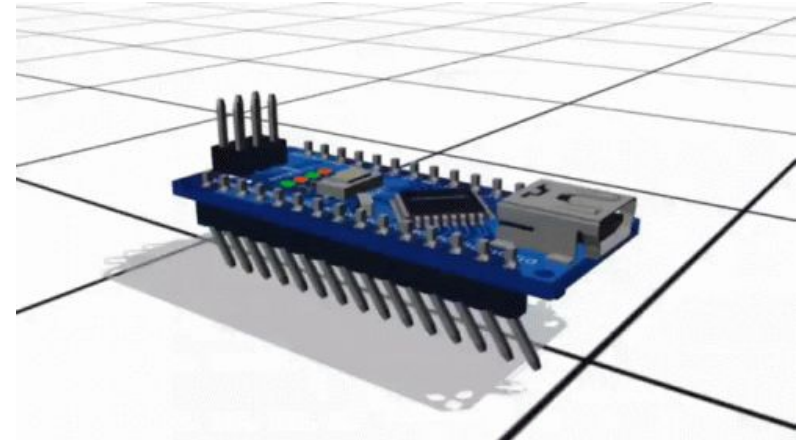
Por que esse tema?

- Trazer o Android para além da tela
- Integração com hardware acessível
- Abrir portas para IoT, automação residencial, wearables DIY
- Projeto simples e interessante



O que vamos ver hoje

1. Comunicação Bluetooth com Android
2. Como funciona o HC-05 com Arduino
3. Criando o app Android para controlar LED
4. Demonstração ao vivo
5. Ideias de expansão



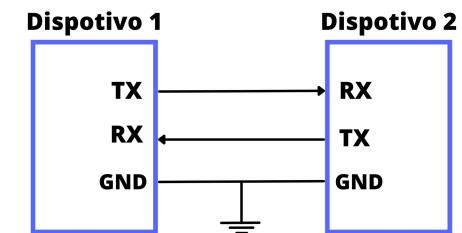
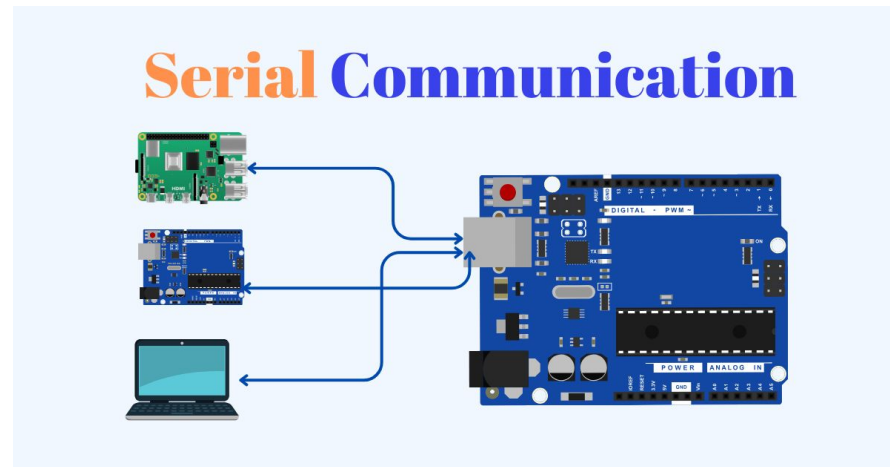
Comunicação Bluetooth: visão geral

Bluetooth Clássico vs BLE

App Android se conecta via socket RFCOMM

Comunicação serial (9600 bps no HC-05)

Simples, rápido e confiável





Materiais utilizados

Arduino Uno ou Nano

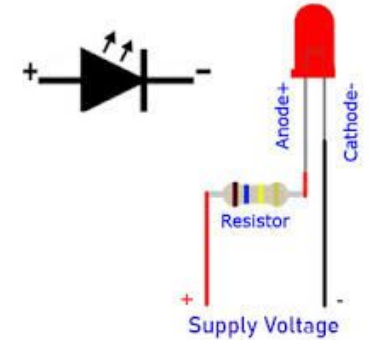
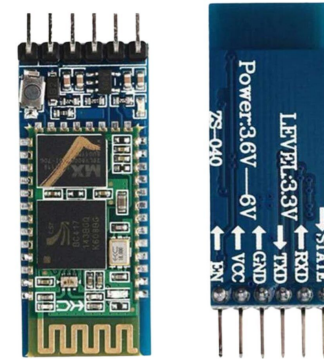
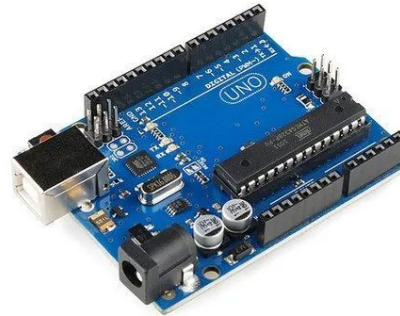
Módulo Bluetooth HC-05

LED + resistor 220Ω

Cabos jumper

Smartphone Android (API 23+)

IDE Arduino e Android Studio



Esquema de conexão

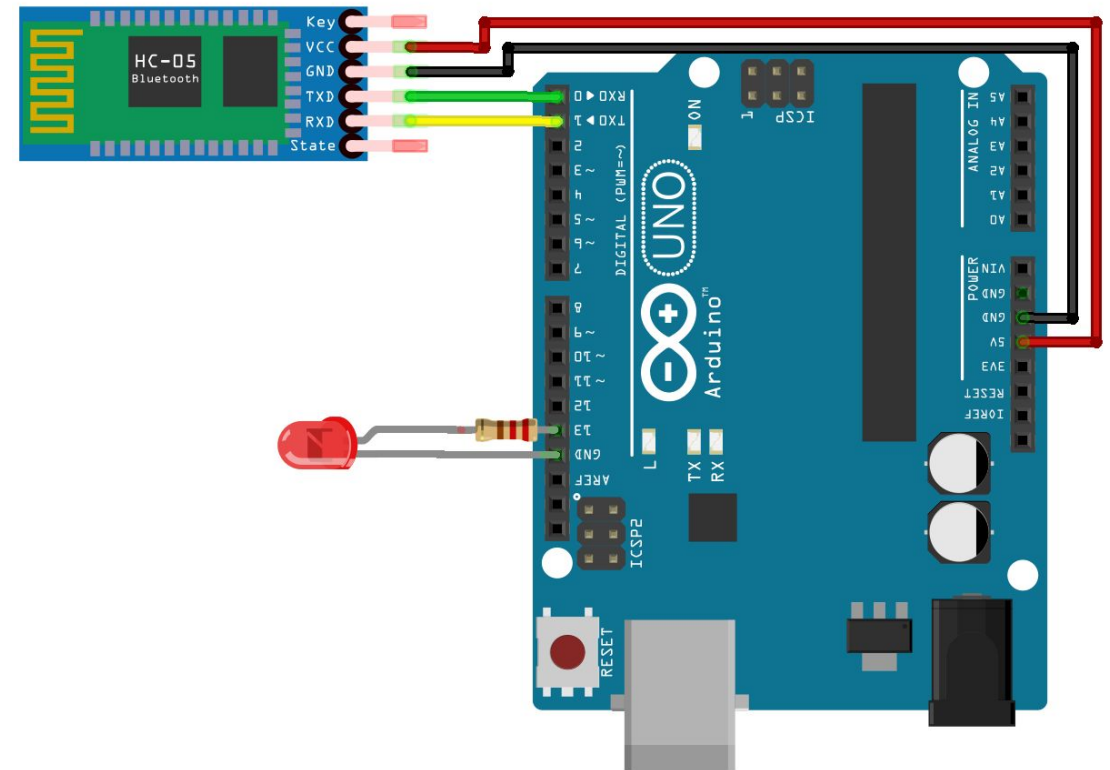
VCC HC-05 → 5V Arduino

GND HC-05 → GND Arduino

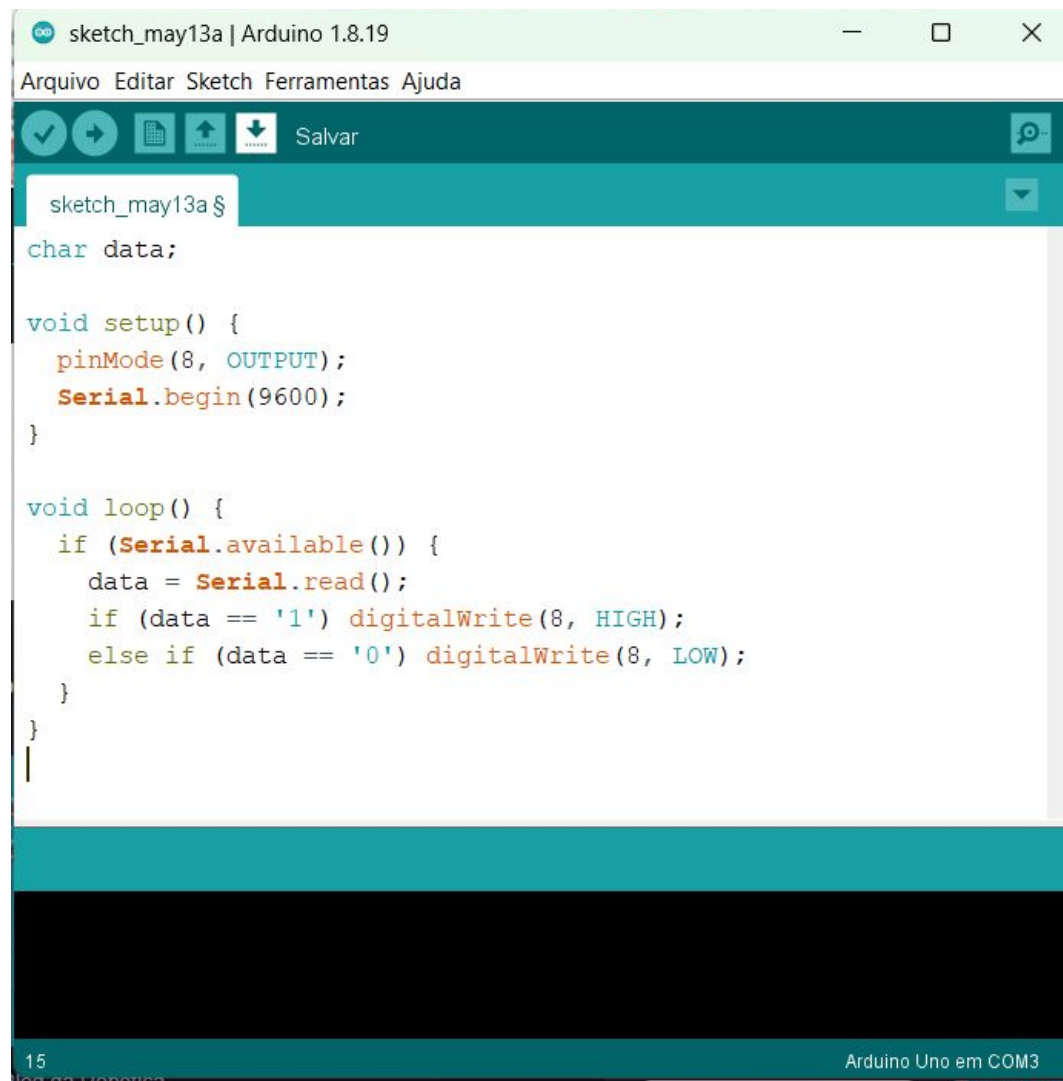
TX HC-05 → RX Arduino (com divisor de tensão)

RX HC-05 → TX Arduino

LED → pino digital D8 com resistor



Código Arduino



The image shows the Arduino IDE interface. The title bar reads 'sketch_may13a | Arduino 1.8.19'. The menu bar includes 'Arquivo', 'Editar', 'Sketch', 'Ferramentas', and 'Ajuda'. The toolbar contains icons for checking, undo, redo, opening a file, saving, and a 'Salvar' button. The text area shows the following code:

```
sketch_may13a $
char data;

void setup() {
  pinMode(8, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  if (Serial.available()) {
    data = Serial.read();
    if (data == '1') digitalWrite(8, HIGH);
    else if (data == '0') digitalWrite(8, LOW);
  }
}
```

The status bar at the bottom indicates '15' on the left and 'Arduino Uno em COM3' on the right.

App Android: estrutura

Permissão Bluetooth no AndroidManifest.xml

Obter BluetoothAdapter

Buscar dispositivos pareados

Conectar via BluetoothSocket

Enviar comandos (1 e 0)

Código Android (Kotlin)

```
val device = bluetoothAdapter.getRemoteDevice(macAddress)
val socket = device.createRfcommSocketToServiceRecord(UUID.fromString(MY_UUID))
socket.connect()
val outputStream = socket.outputStream
btnOn.setOnClickListener { outputStream.write("1".toByteArray()) }
btnOff.setOnClickListener { outputStream.write("0".toByteArray()) }
```

but here's the coder

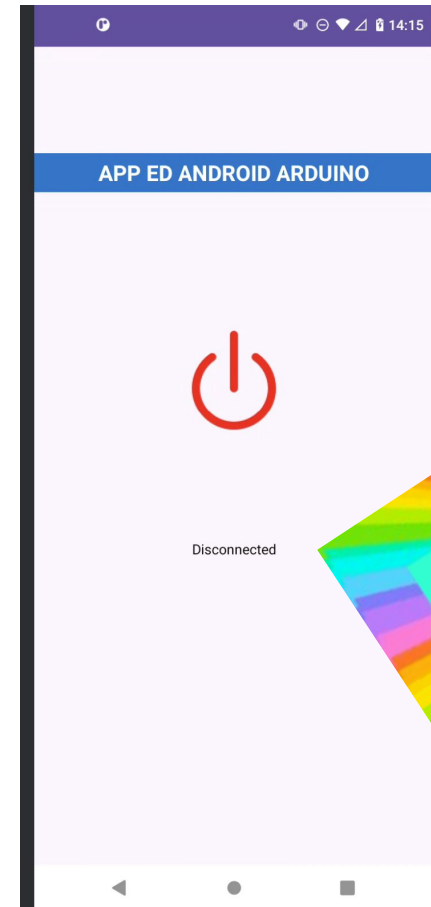


Layout Android

Botão “Acender LED”

Botão “Apagar LED”

Status de conexão



Demonstração prática

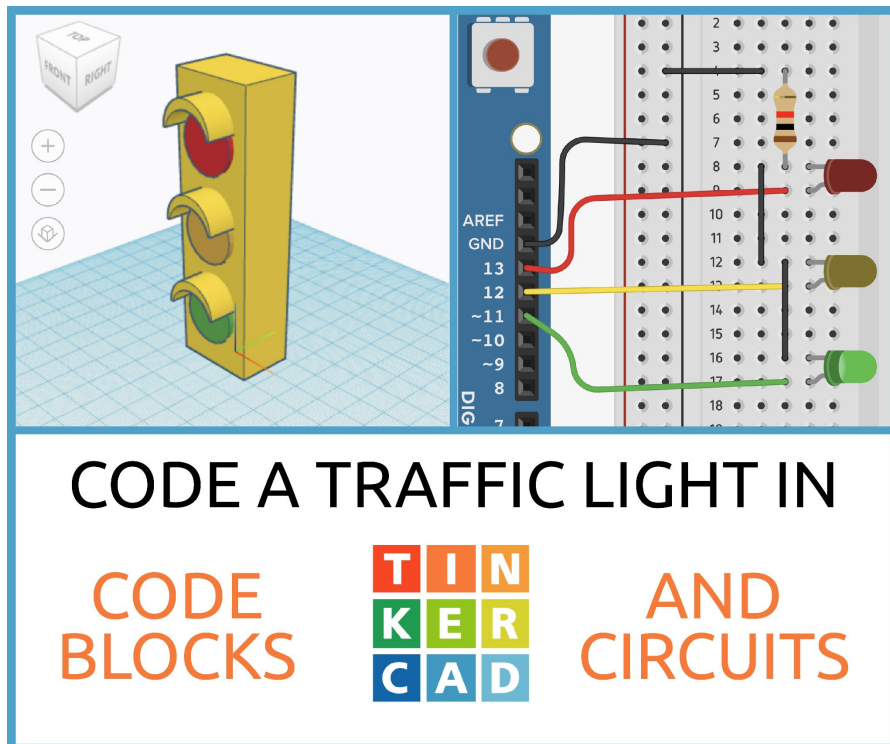
Conectar via Bluetooth

Acender e apagar LED em tempo real

Debug no Serial Monitor

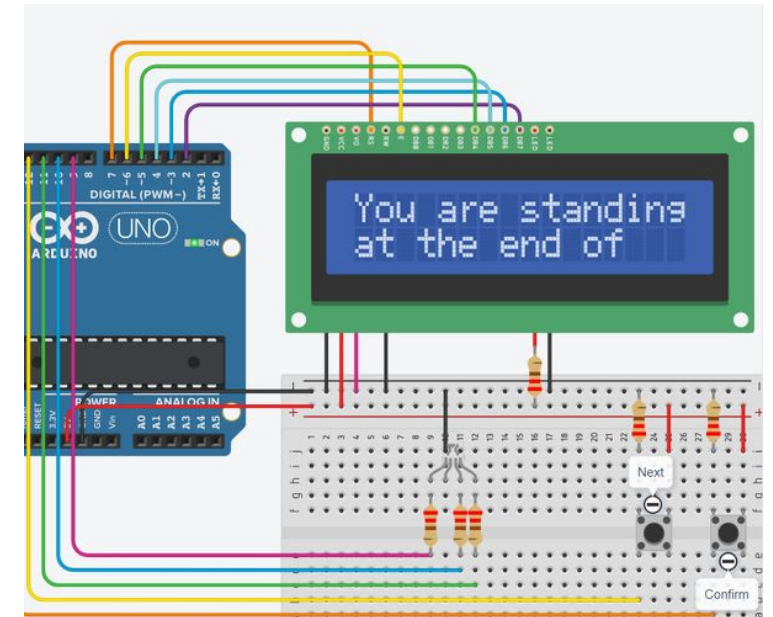
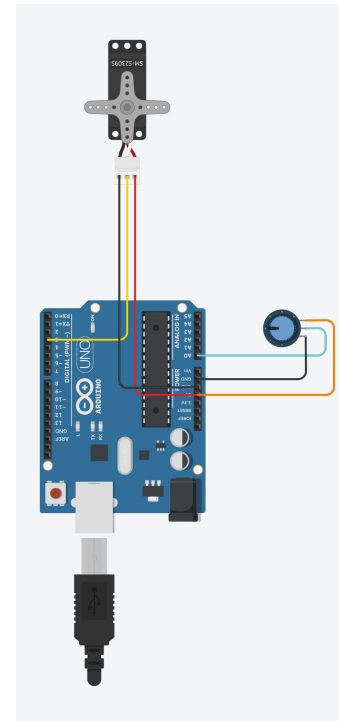


Também via emulador



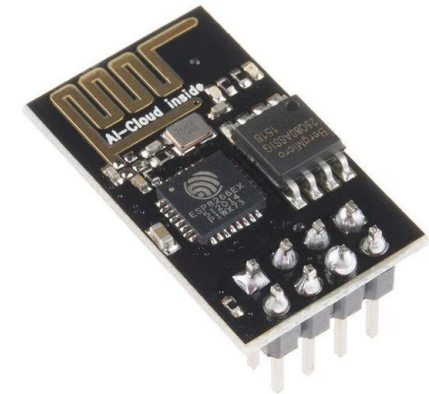
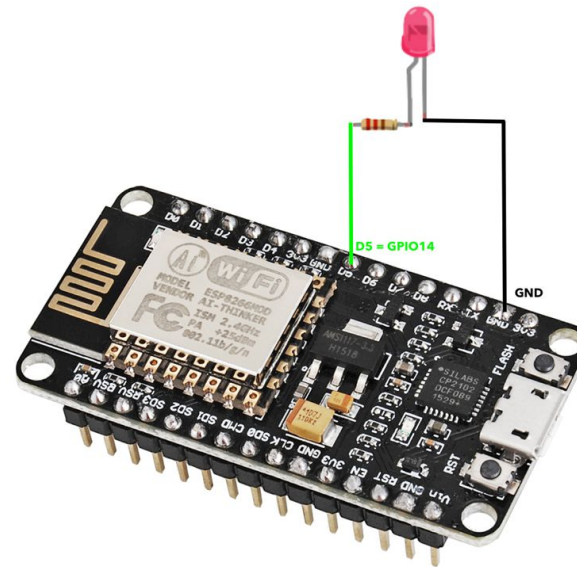
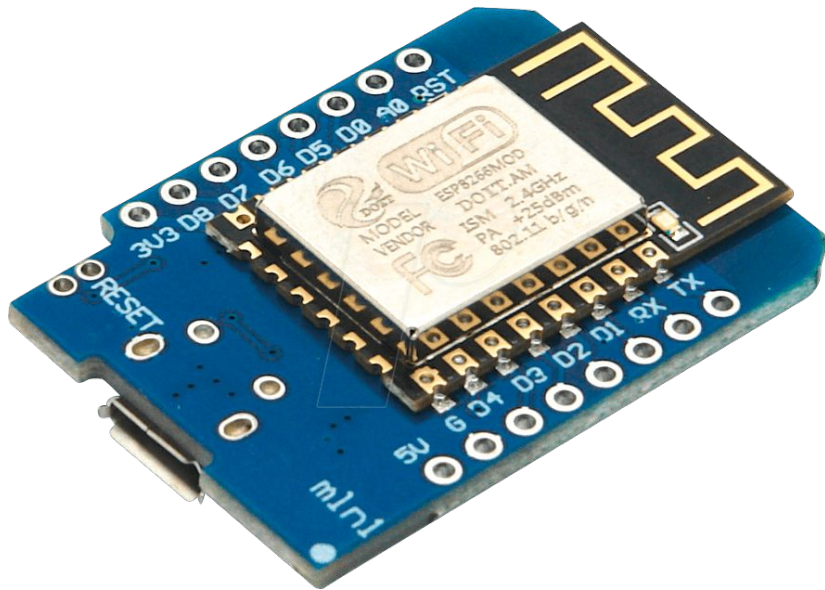
CODE A TRAFFIC LIGHT IN

CODE BLOCKS **TIN**
KER
CAD AND CIRCUITS



Alternativas

Wemos D1 | NodeMCU ESP8266 ESP32 | ESP01



Ideias para expandir

Comunicação bidirecional (leitura de sensores)

Controle via comandos de voz

Integração com Firebase

Controle múltiplo (motores, relés, sensores)

Conclusão e links

Android + Arduino é acessível e poderoso

Código e slides no GitHub

Contato para dúvidas

@edgarolv

linkedin.com/in/edgarolv



[https://github.com/EdgarOlv/Simple
ArduinoLedBluetooth](https://github.com/EdgarOlv/SimpleArduinoLedBluetooth)

Como diria minha ex

TERMINAMOS!!



End of discussion.



**Dapper
Dachshund**

