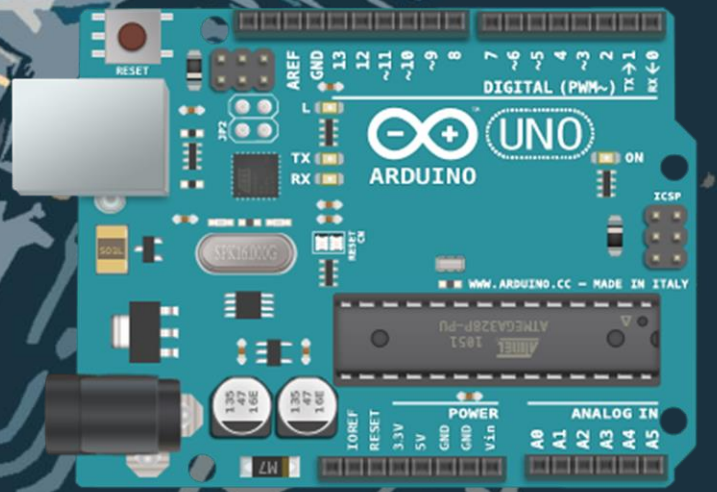
MINICURSO DE ARDUINO

ARDUINO

Dia 1

Filipe Augusto
Danillo Rodrigues

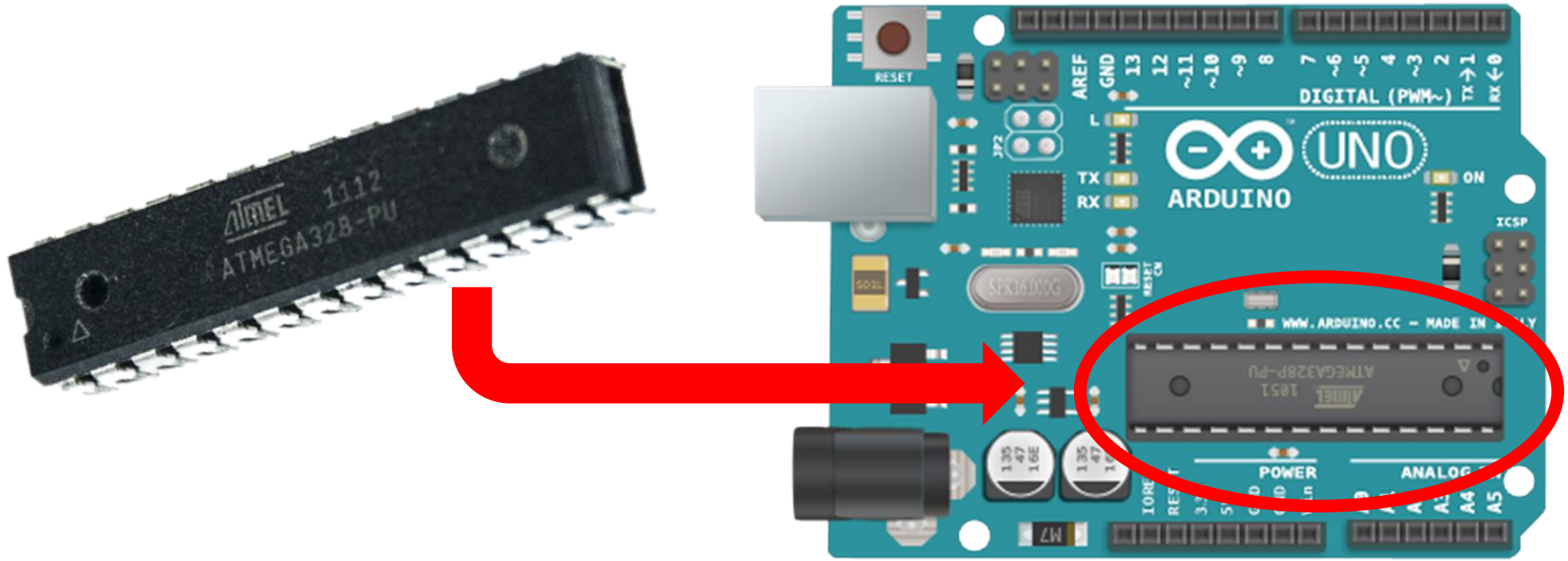


O que é Arduino?

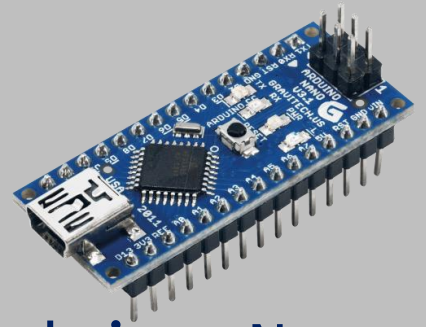
Microcontroladores

O que é Arduino?

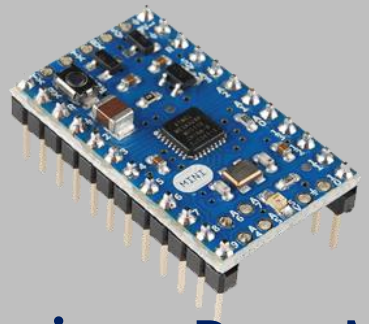
Microcontroladores



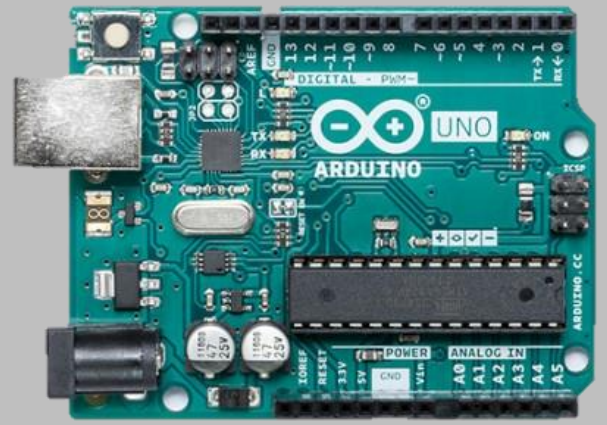
Tipos de Arduino



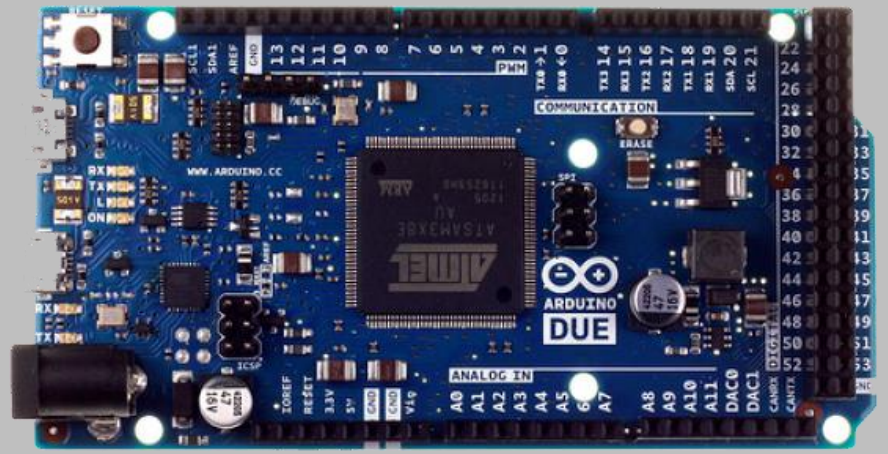
Arduino Nano



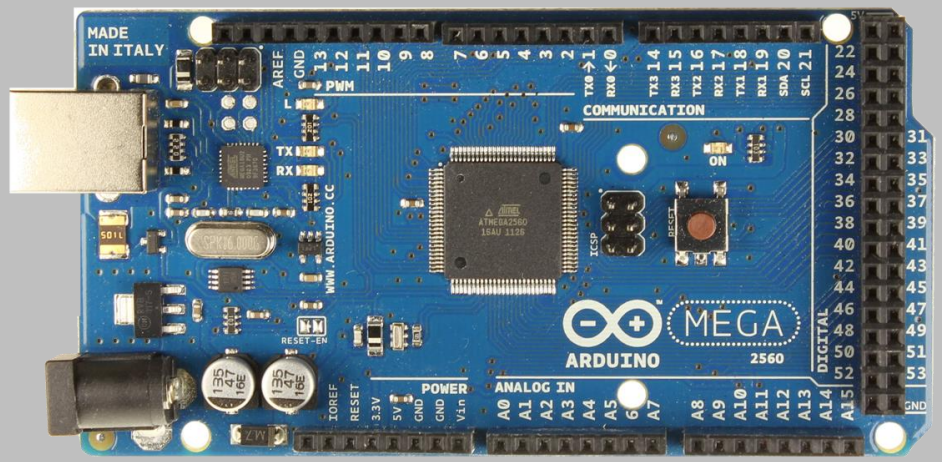
Arduino Pro Mini



Arduino UNO

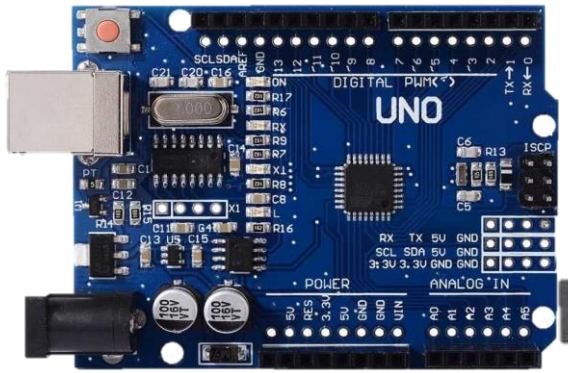


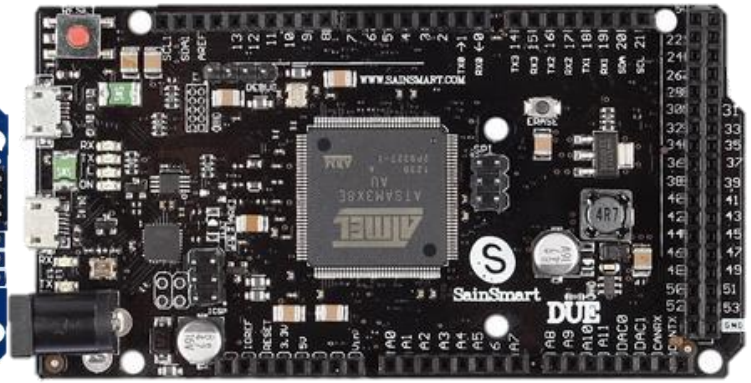
Arduino DUE

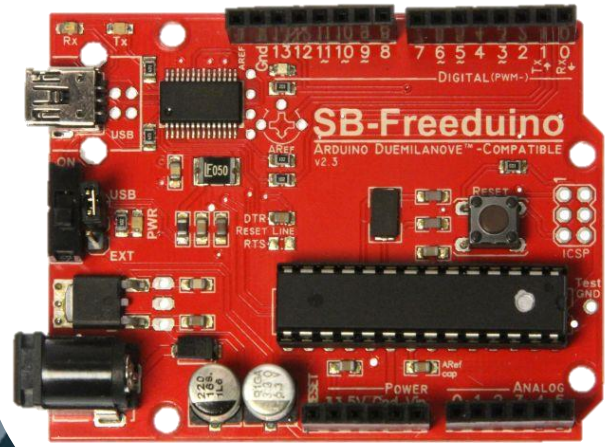


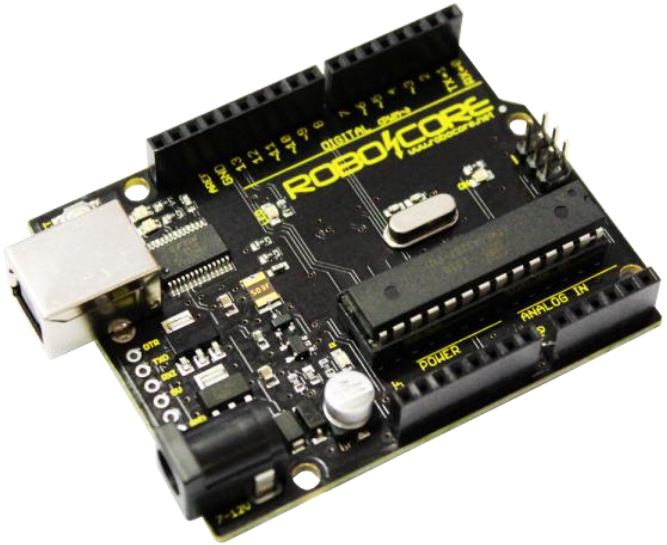
Arduino MEGA

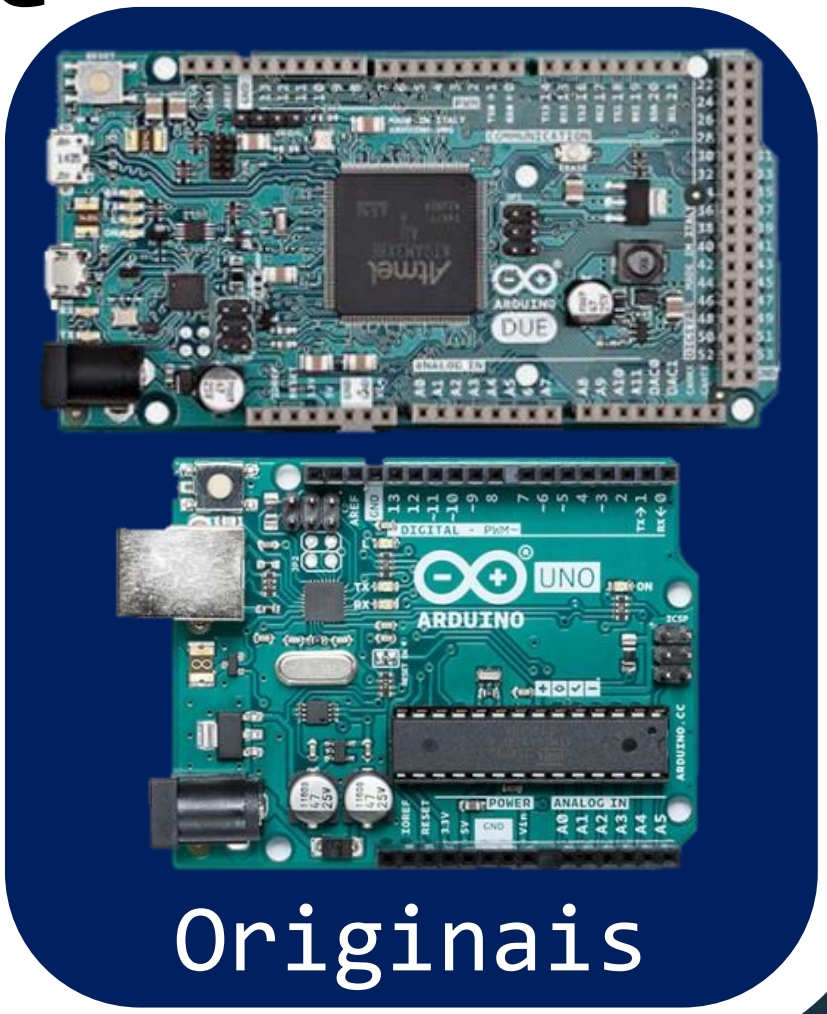
Hardware Open-Source









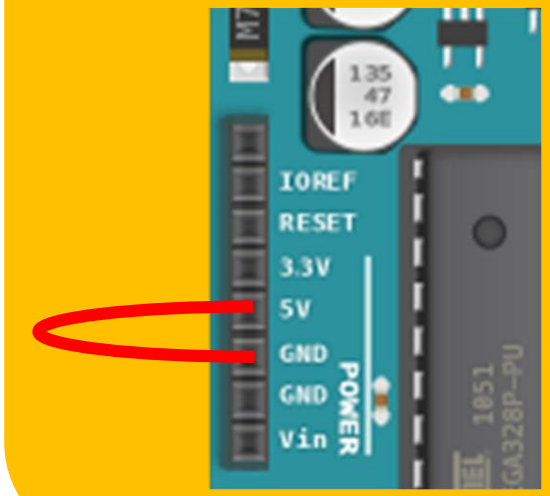


Originais

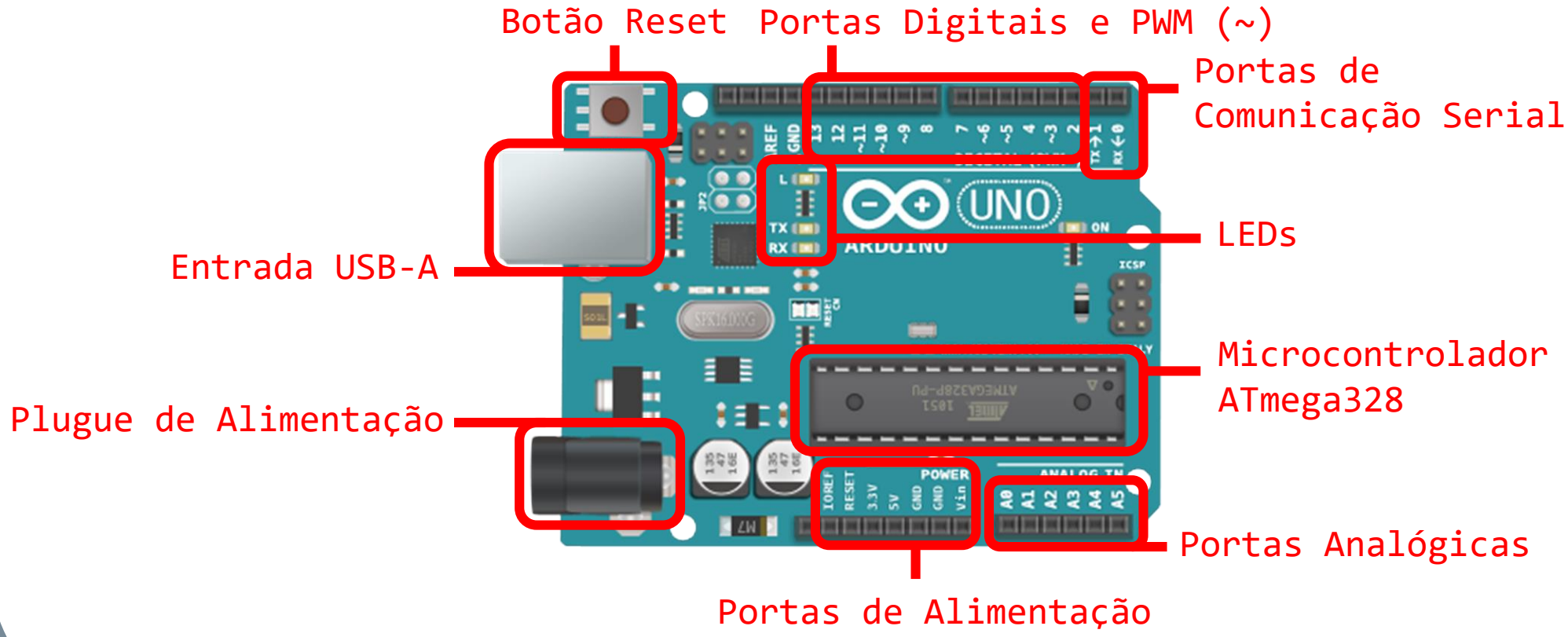
Cuidados com o Arduino

Eletricidade Estática

Curto Circuito

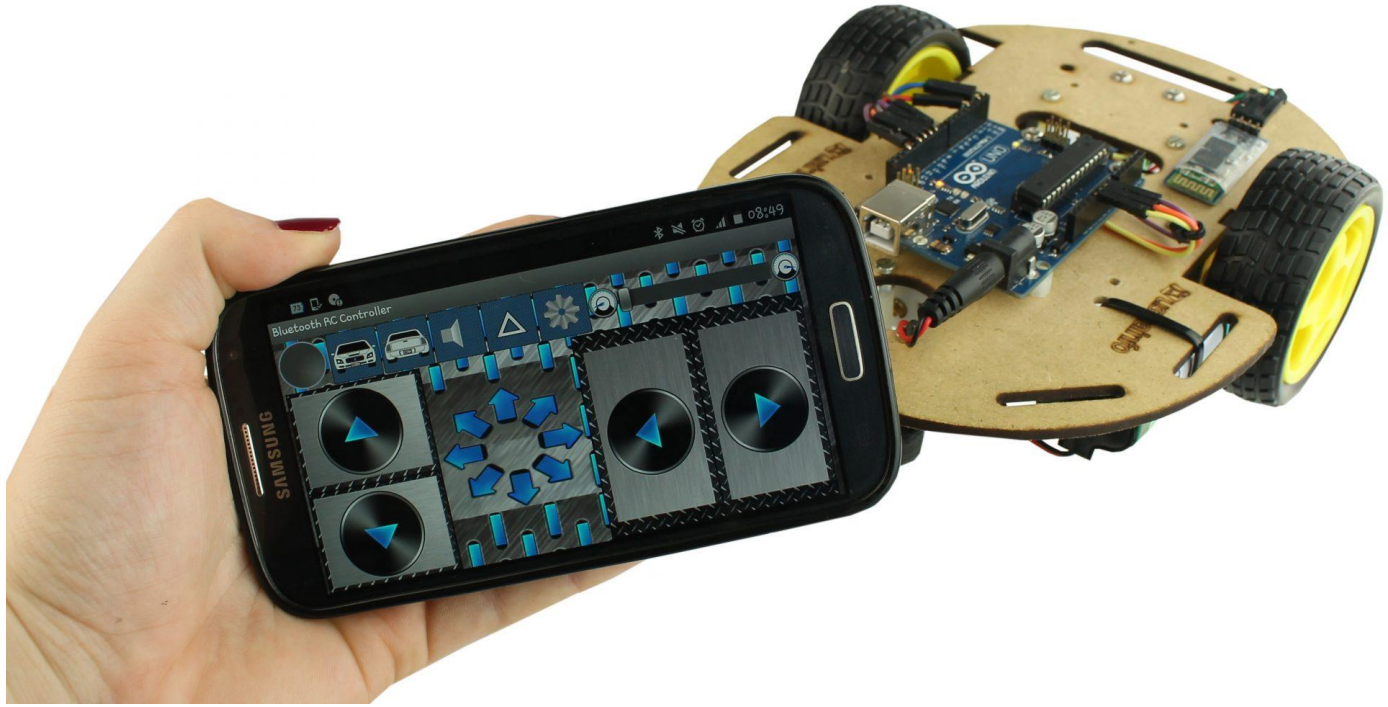


O que é cada parte da placa?



E suas Aplicações?

Robótica

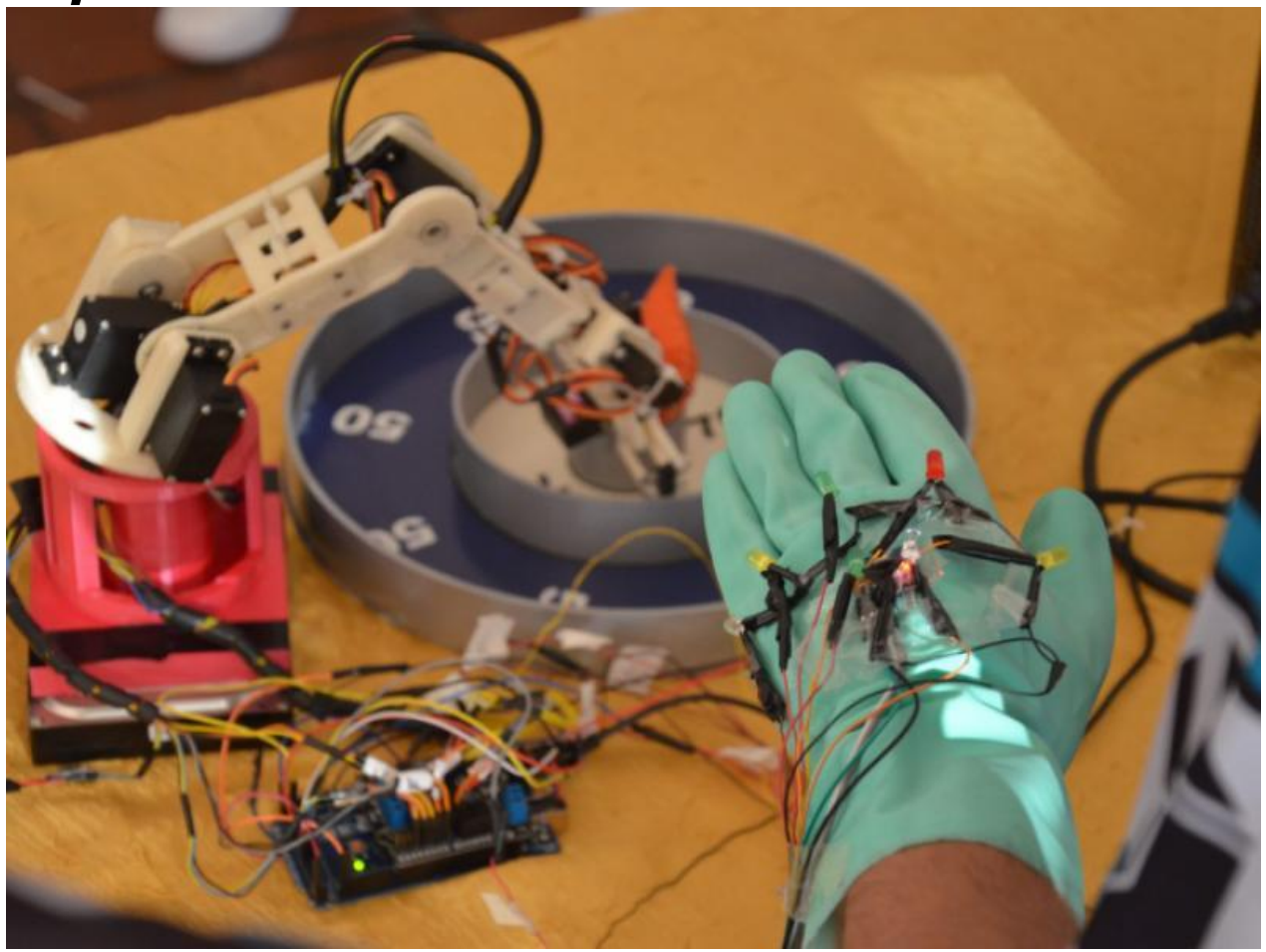


E suas Aplicações?

Robótica

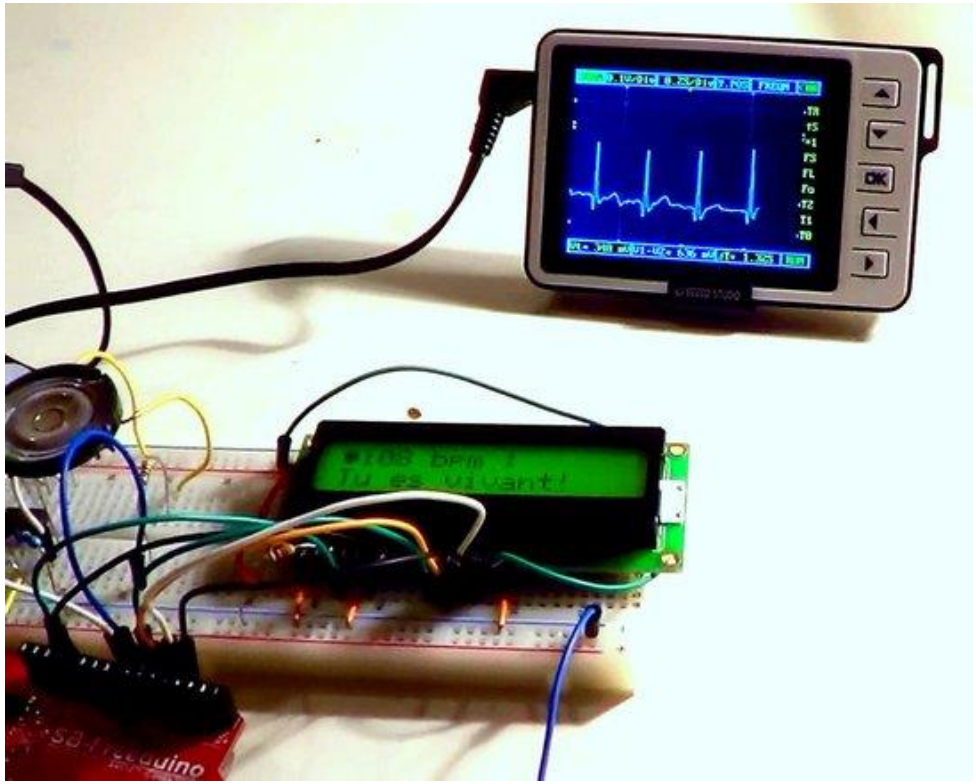
HOOK

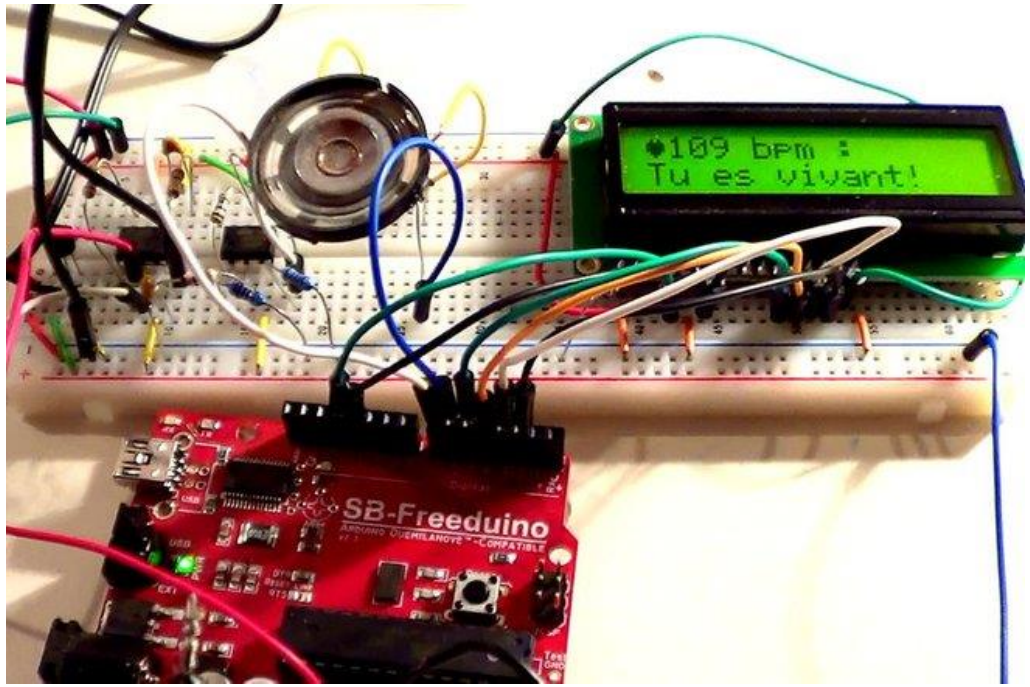
```
void PET_Engenharia_Biomédica ()
{
  DESAFIO CAPITÃO GANCHO;
  Edição = 1;
  start ();
}
```



E suas Aplicações?

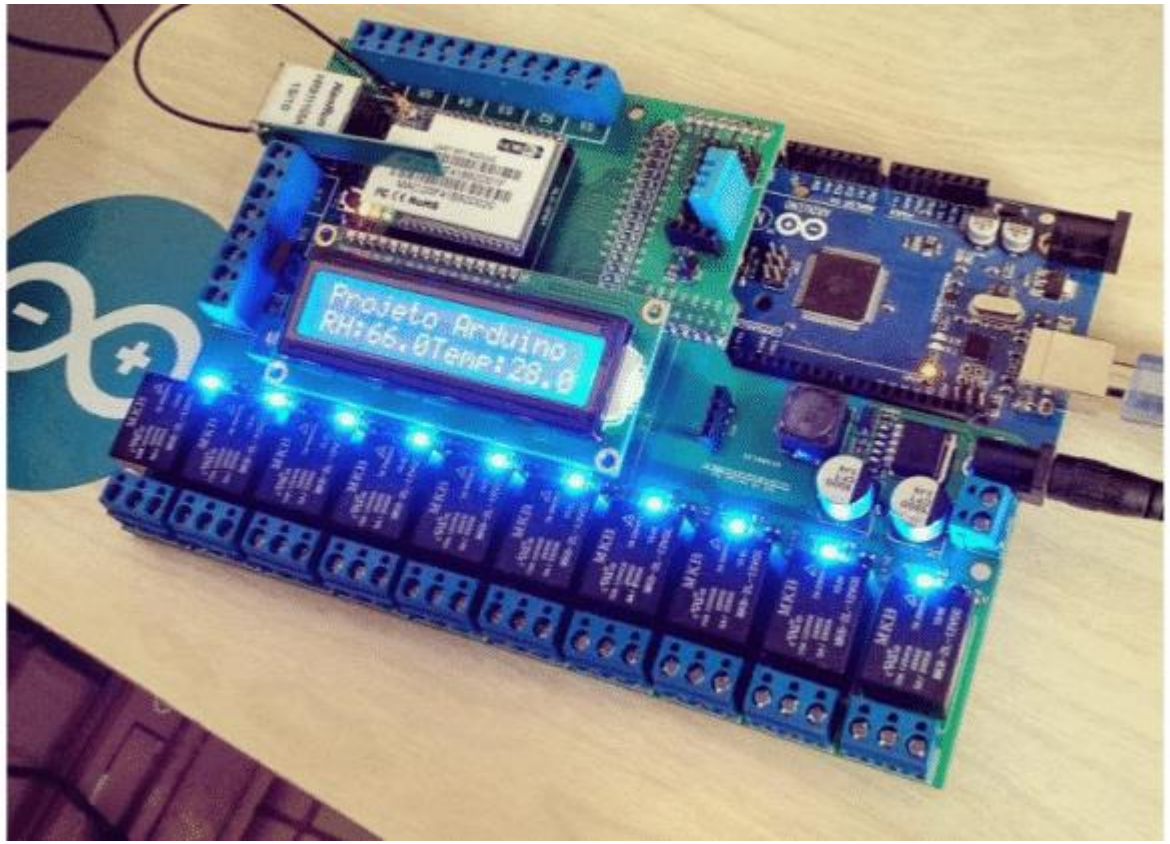
Monitoramento de Sinais Fisiológicos





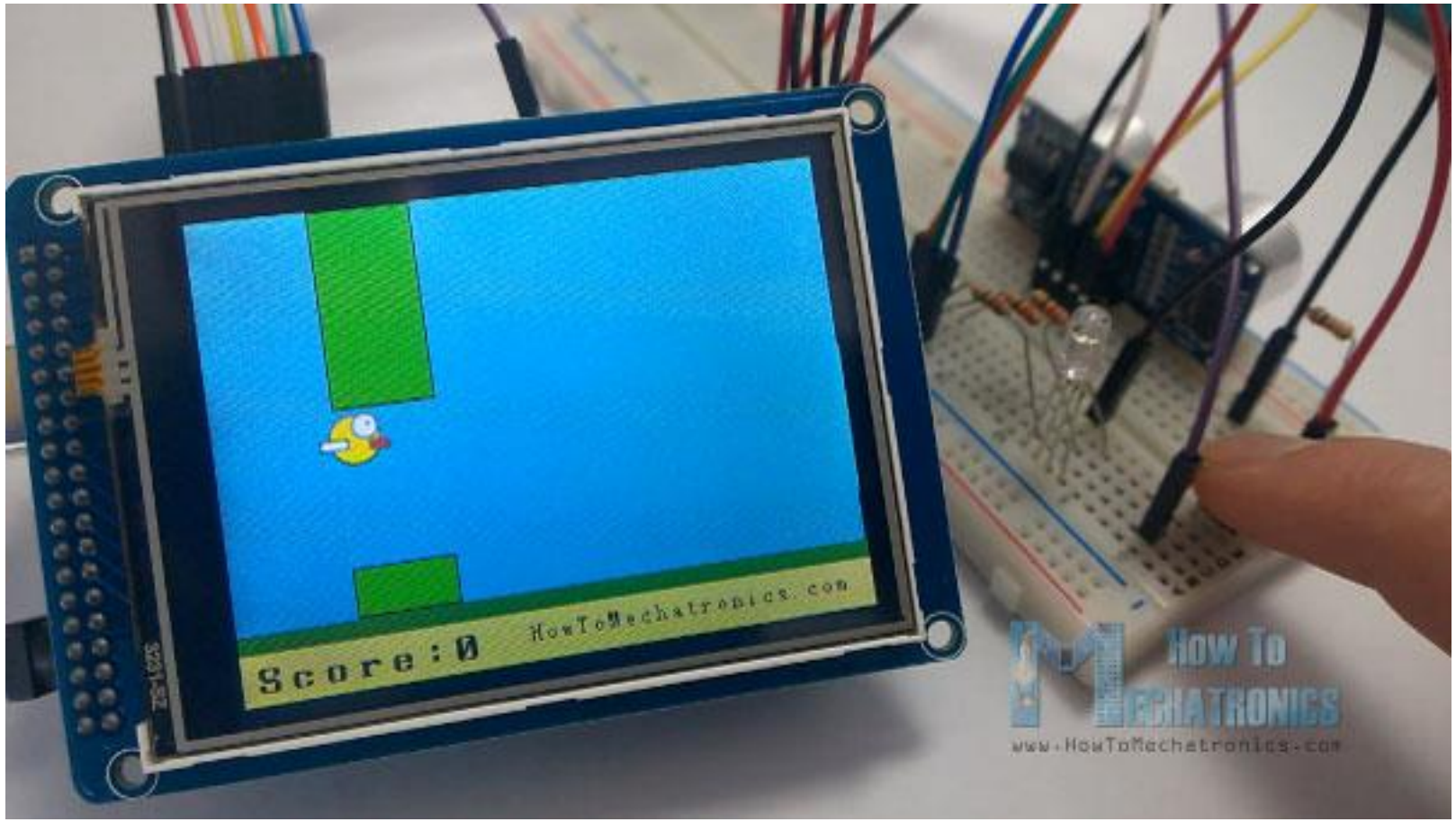
E suas Aplicações?

Projetos de Automação



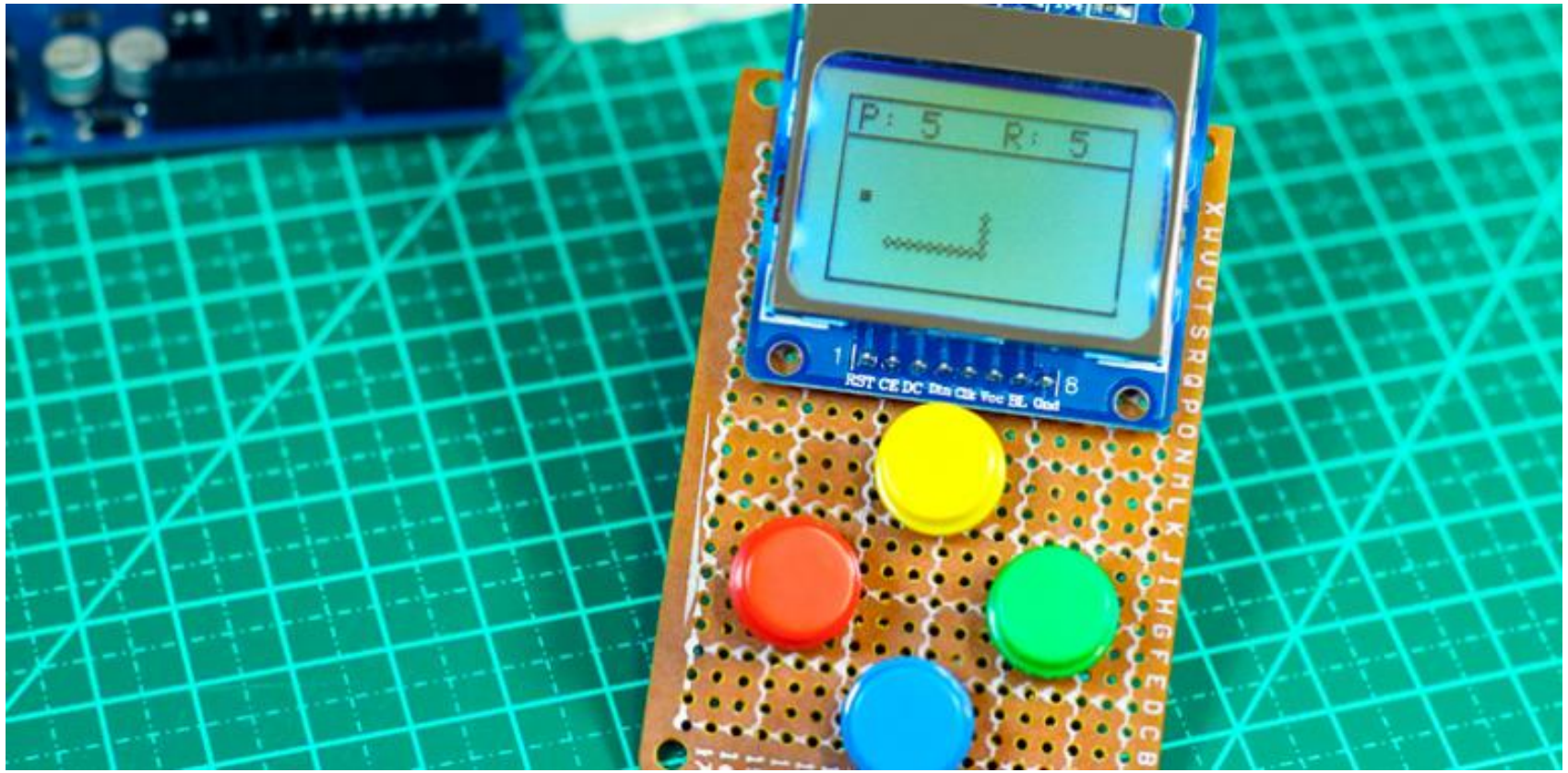
E suas Aplicações?

Jogos



E suas Aplicações?

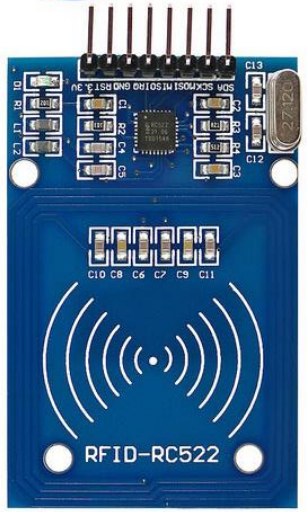
Jogos



E suas Aplicações?

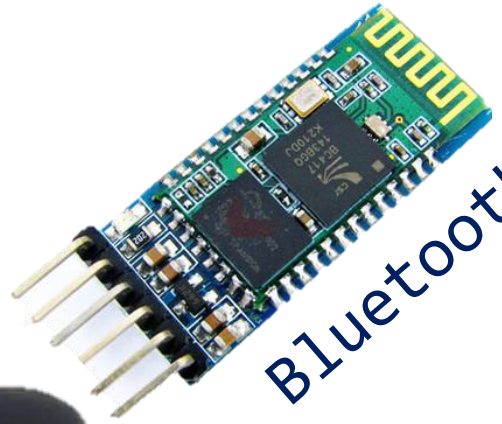
Coisas aleatórias ¯_(ツ)_/¯

Módulos!



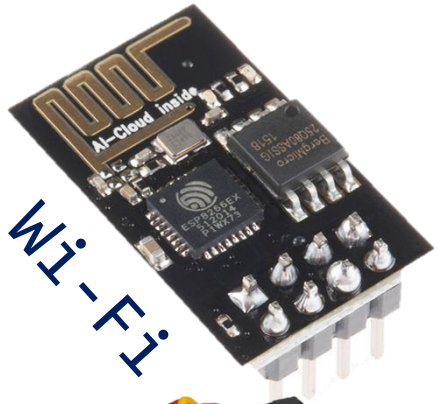
RFID

Ultrassom



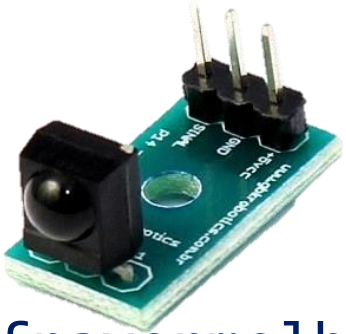
Bluetooth

Batimentos Cardíacos



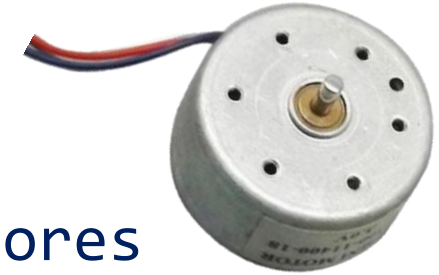
Wi-Fi

Joystick



Infravermelho

Teclado



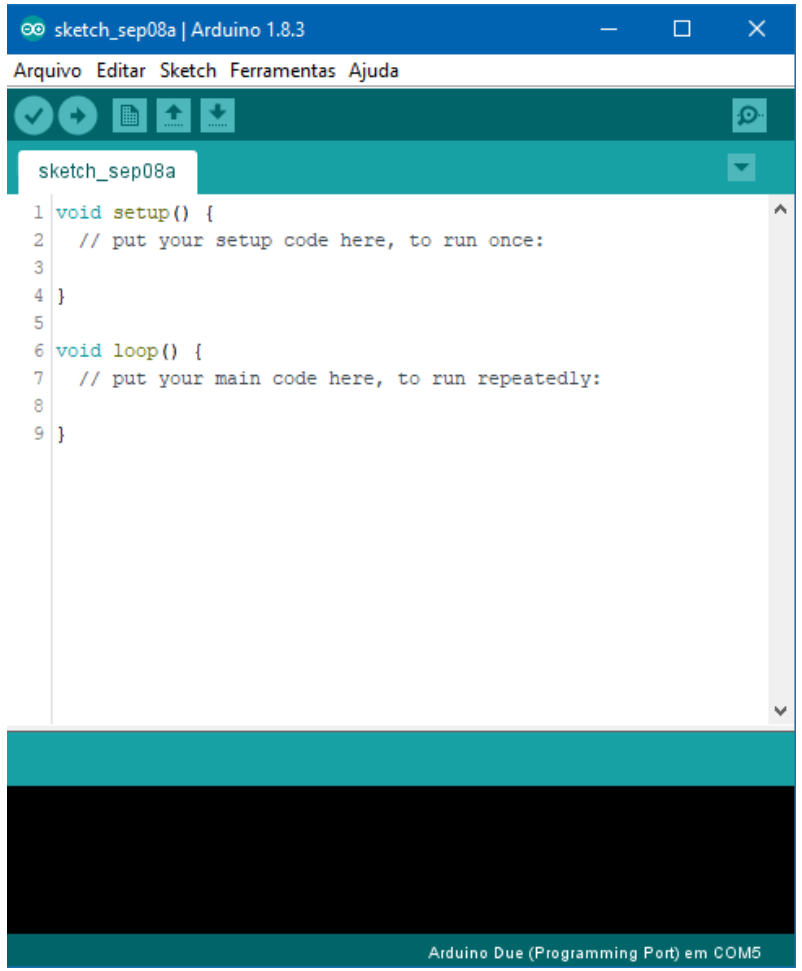
Motores

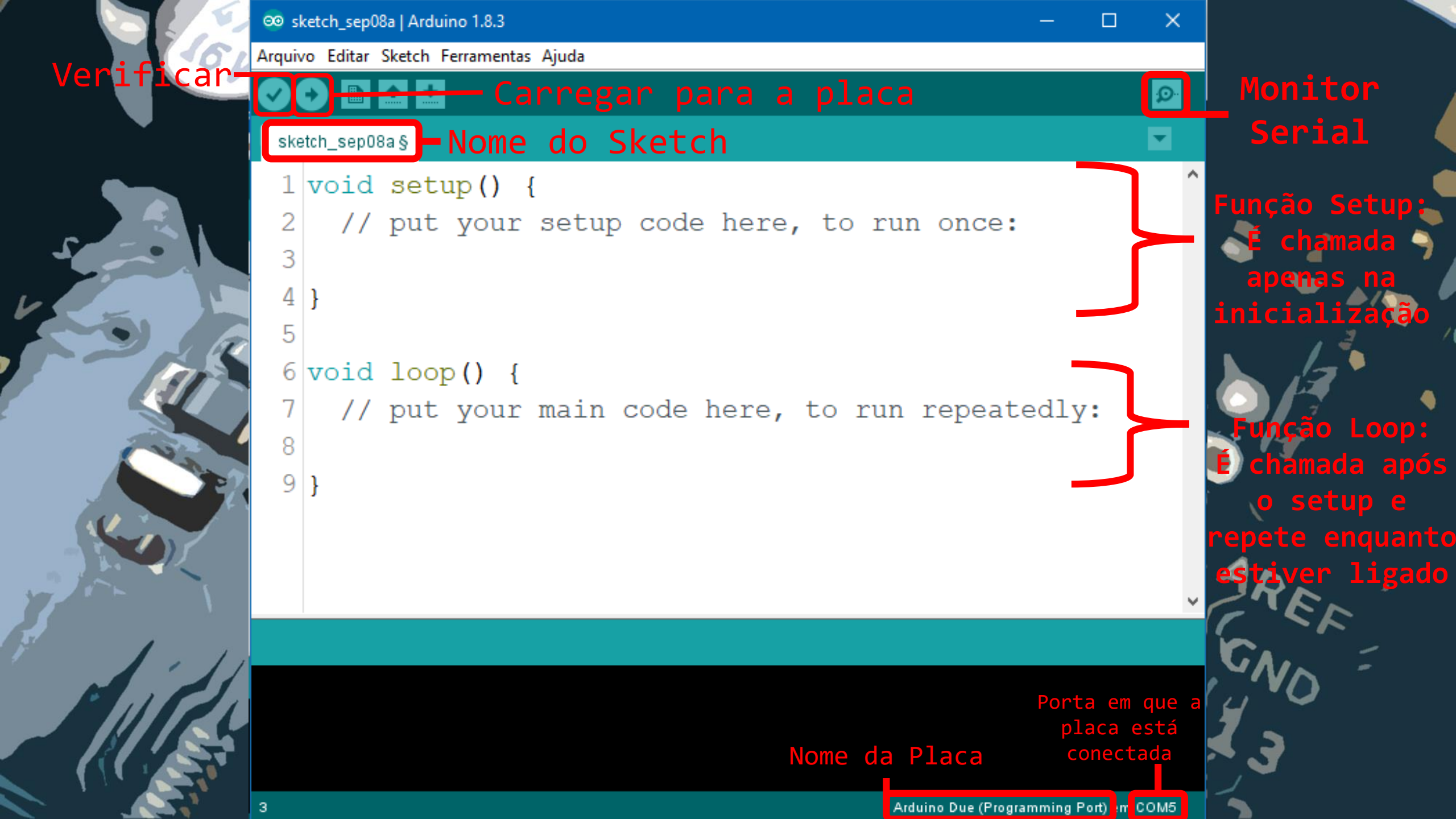
Como utilizar um Arduino?

IDE Arduino

Linguagem: C++ (modificada)

Cada programa é um "sketch"



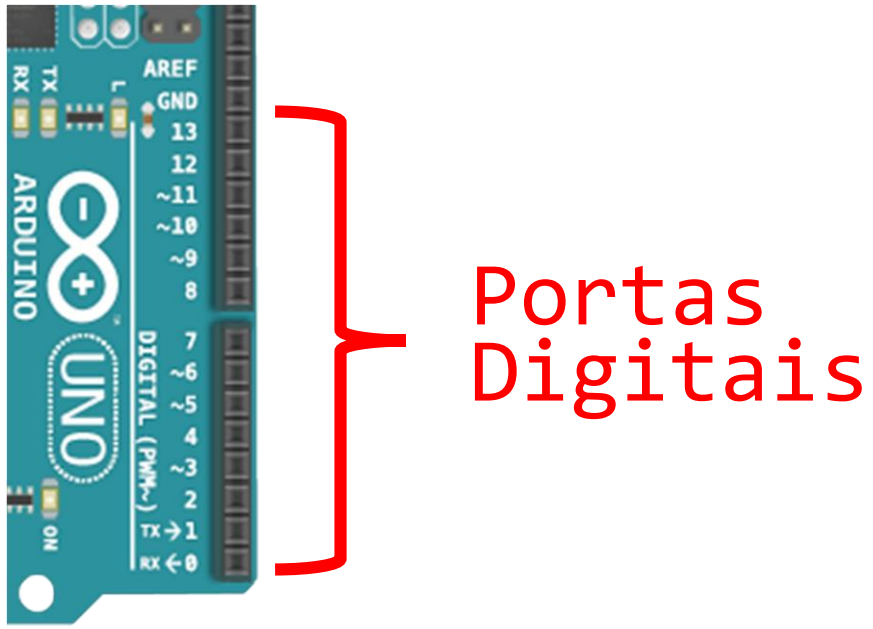

sketch_sep08a | Arduino 1.8.3
Arquivo Editar Sketch Ferramentas Ajuda
Verificar
Carregar para a placa
Monitor Serial
sketch_sep08a §
Nome do Sketch
1 void setup() {
2 // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7 // put your main code here, to run repeatedly:
8
9 }
Função Setup: É chamada apenas na inicialização
Função Loop: É chamada após o setup e repete enquanto estiver ligado
Nome da Placa
Porta em que a placa está conectada
3
Arduino Due (Programming Port)
COM5

Vamos à mão na massa!

1º Programa: Blink

Objetivo:

Fazer um led piscar com intervalos de tempo definidos



Conceito: Saída de Portas Digitais

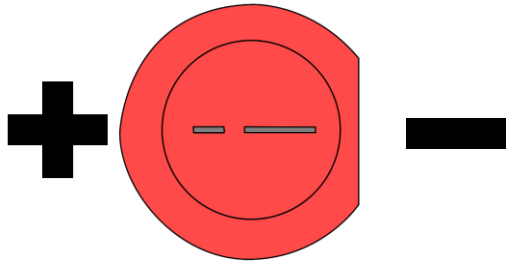
Portas digitais fornecem apenas dois valores de tensão:

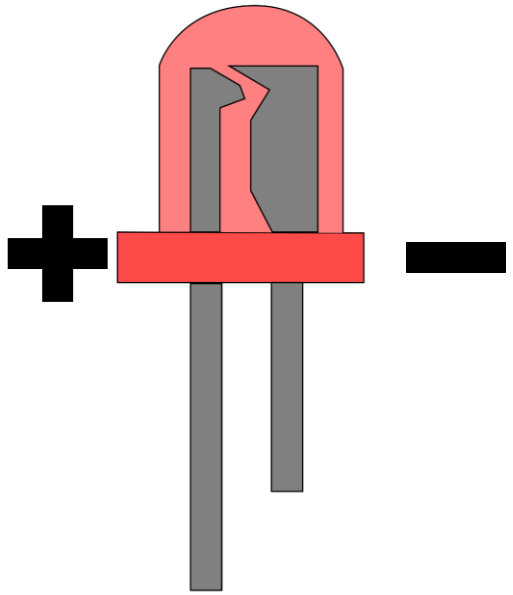
Alto: 5V e Baixo: 0V (GND)

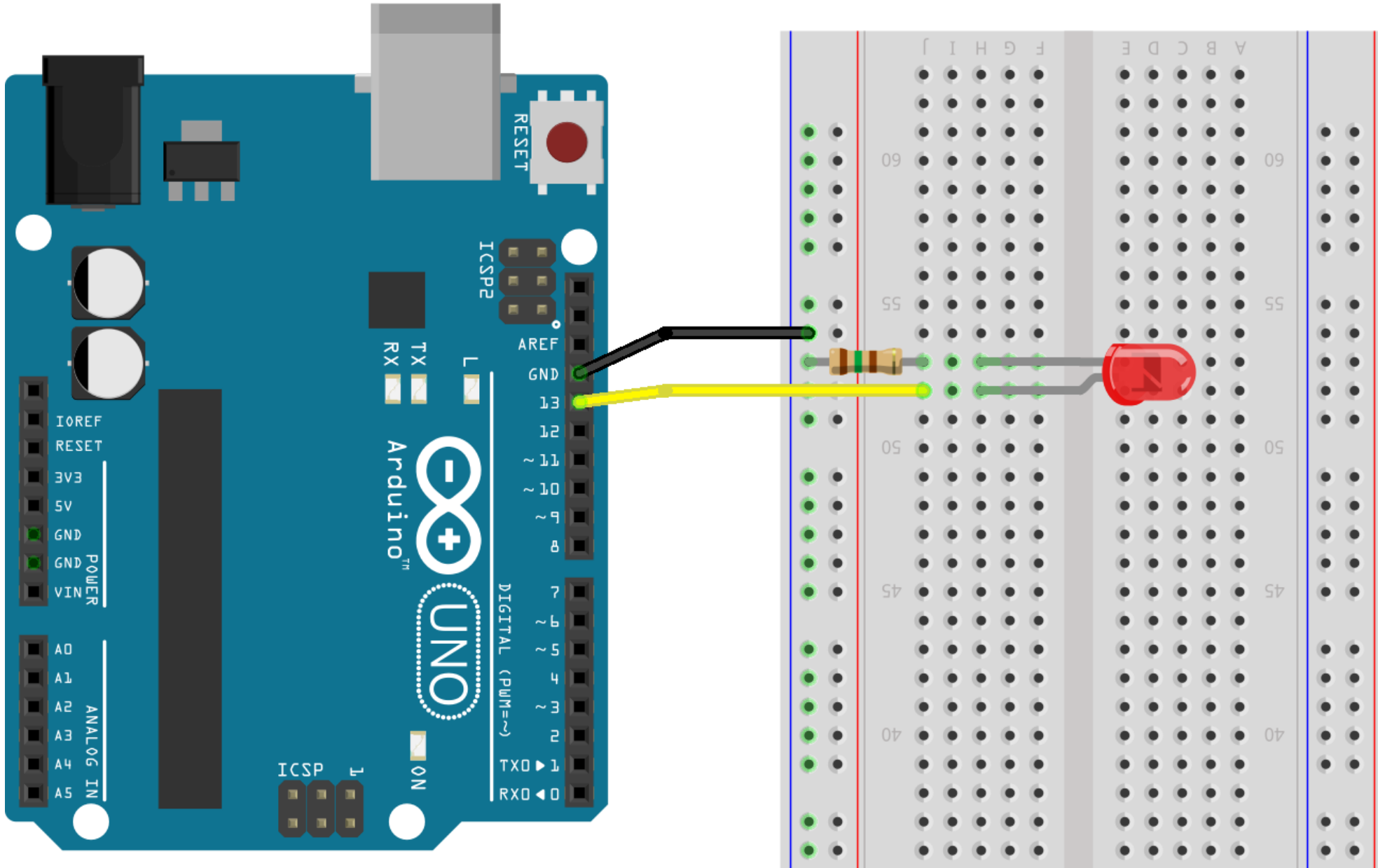
1º Programa: Blink

Montagem:

Obs.:







1º Programa: Blink

Funções:

pinMode(pino, modo)

- Nº do pino
- modo:
  - INPUT: "Entra" energia
  - OUTPUT: "Sai" energia

digitalWrite(pino, nível)

- Nº do pino
- nível:
  - HIGH: 5V (nível alto)
  - LOW: 0V (nível baixo)

delay(tempo) → Tempo em milissegundos (1000 ms = 1 segundo)

2º Programa: LED controlado por botão

Objetivo:

Fazer um led piscar quando um botão for apertado

Conceito: Entrada de Portas Digitais

Portas digitais identificam apenas dois valores de tensão:

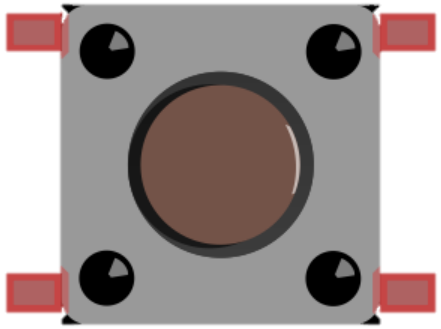
Alto: 5V e **Baixo:** 0V (GND)

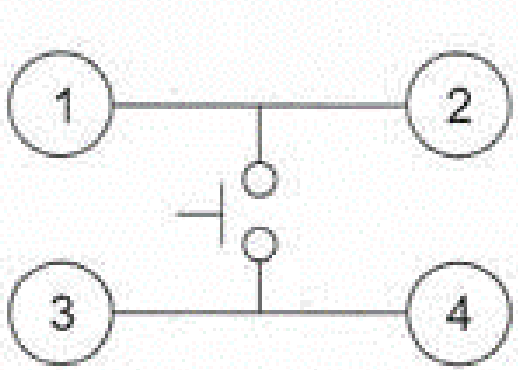
Valores intermediários retornam valores aleatórios!

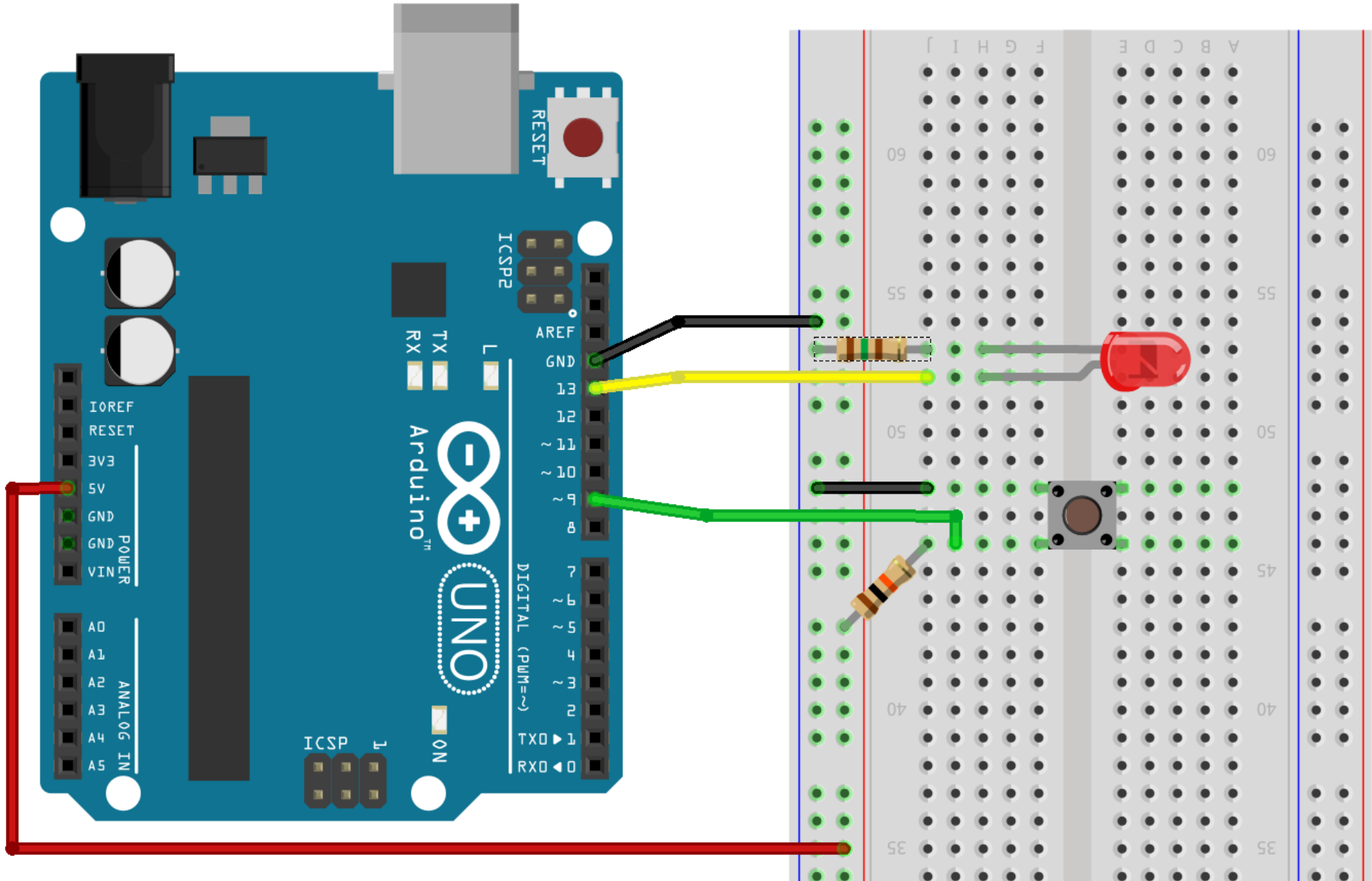
2º Programa: LED controlado por botão

Montagem:

Obs.:







2º Programa: LED controlado por botão

Função:

```
digitalRead(pino)
```

Nº do pino

Retorna:
1 para 5V e 0 para 0V

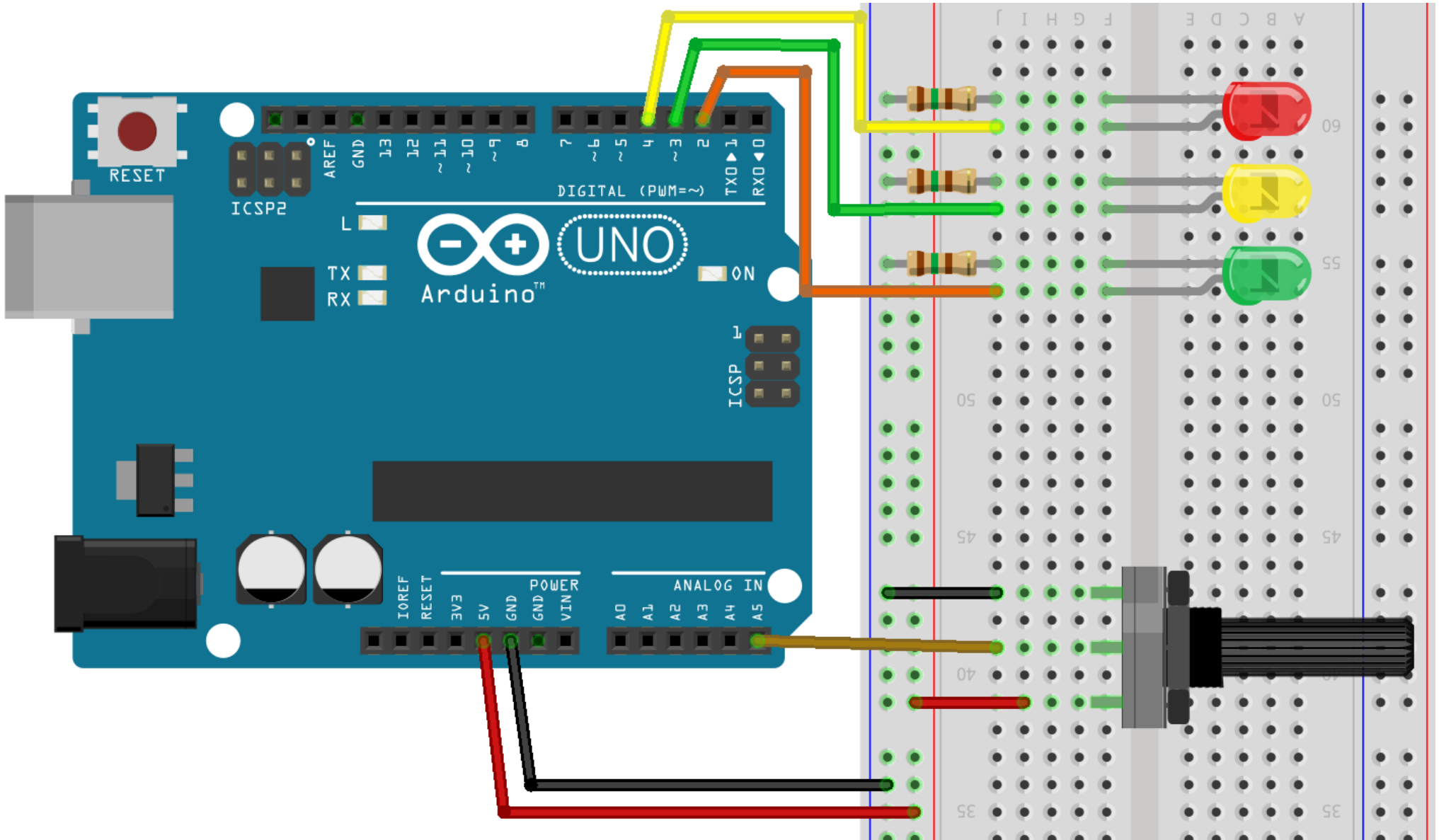
3º Programa: Semáforo com controle de tempo

Objetivo:
Elaborar um semáforo com 3 LEDs cujo tempo de operação varie de acordo com a tensão de saída do potenciômetro.

Conceito: Entrada de Portas Analógicas
Portas analógicas conseguem identificar valores de entrada tensão entre 0 e 5 Volts.

3º Programa: Semáforo com controle de tempo

Montagem:



3º Programa: Semáforo com controle de tempo

Função:

`analogRead(pino)`

Nº do pino

Retorna:
1023 para 5V e 0 para 0V

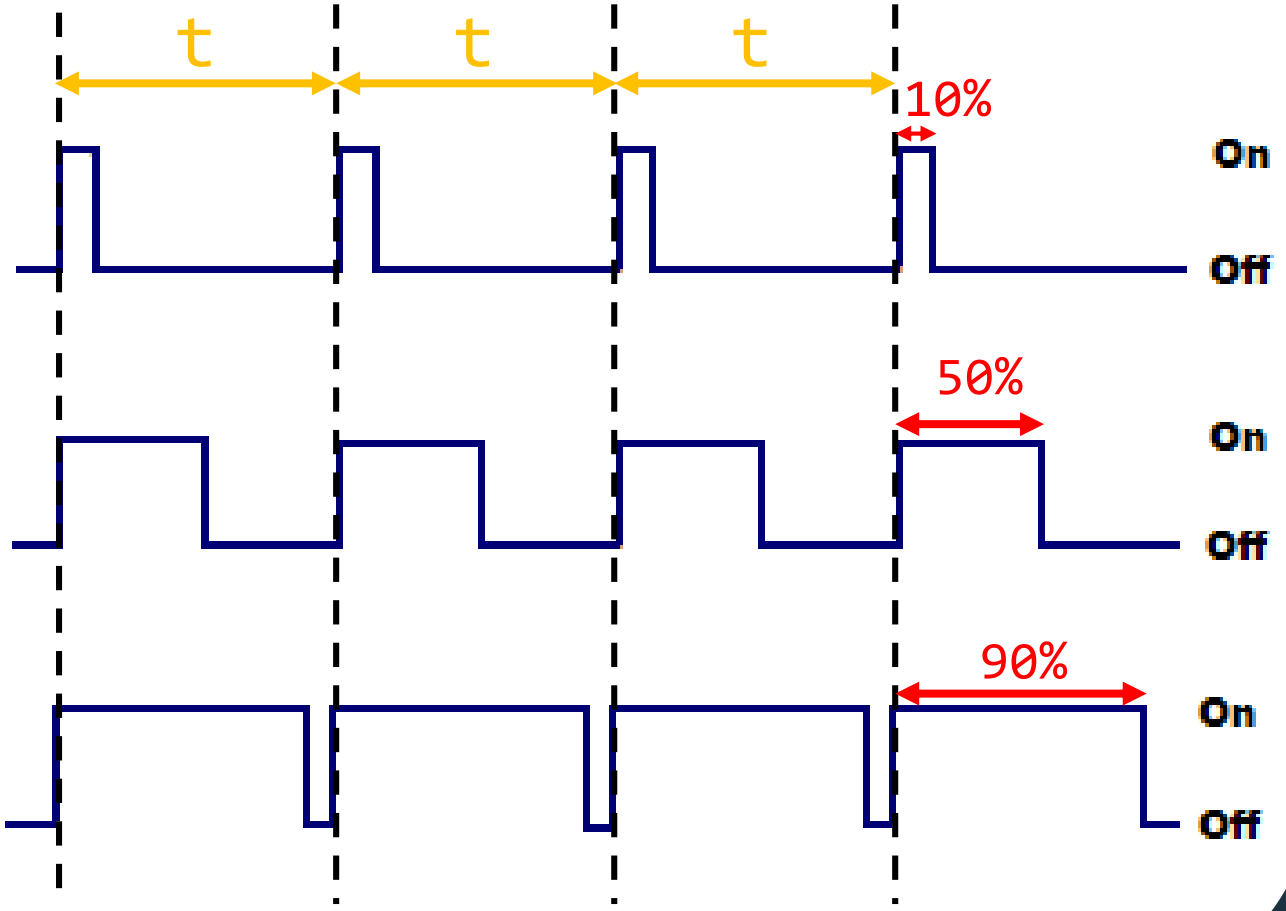
Arduino UNO: resolução de 10 bits

10 bits = armazenam até 1023 em valores decimais

$\frac{5\,V}{1023} \cong 0,005 = 5\,mV$ Portanto, cada bit equivale a $5\ mV$.

Introdução às saídas PWM

- Todas as saídas do Arduino são digitas
- Diferentes larguras de pulso, mesma frequência
- As saídas PWM são controladas pela função analogWrite



As saídas PWM são identificadas por um til (~)

```
analogWrite(pino, valor)
```

Nº do pino

Qualquer valor entre 255 e 0, onde:
255: 100% largura de pulso (5V)
0: 0% de largura de pulso (0V)

Exemplo: analogWrite(10, 127)
escreve aprox. 2,5 V no pino 10

4º Programa: Dimmer

Objetivo:

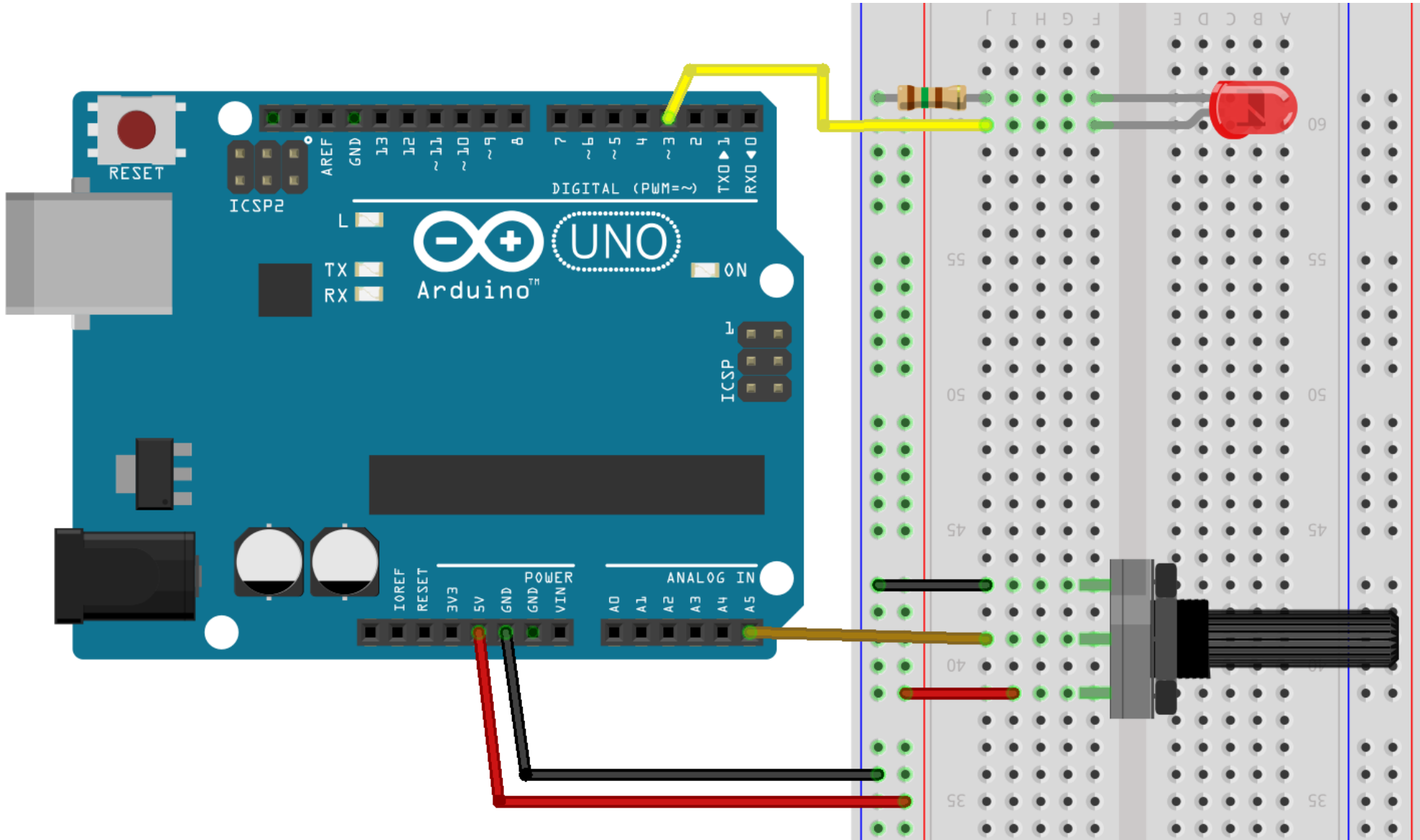
Controlar o brilho de um LED a partir da leitura analógica da saída de um potenciômetro

Conceito: Saídas PWM

Portas PWM fornecem valores de tensão de 0 a 5V de acordo com a mudanda a largura de pulso.

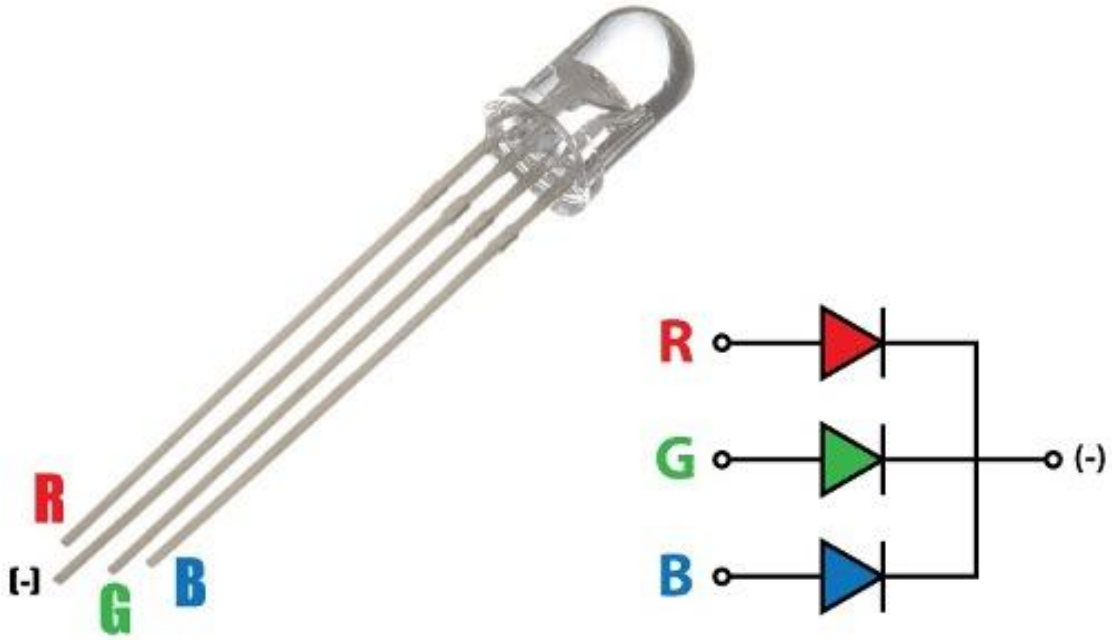
4º Programa: Dimmer

Montagem:



LED RGB

- O LED RGB é caracterizado por possuir o equivalente a 3 LEDs no mesmo encapsulamento, onde suas cores são vermelho, verde e azul;
- Cada cor é controlada por um terminal (perninha);
- O maior terminal é o Ground (-).



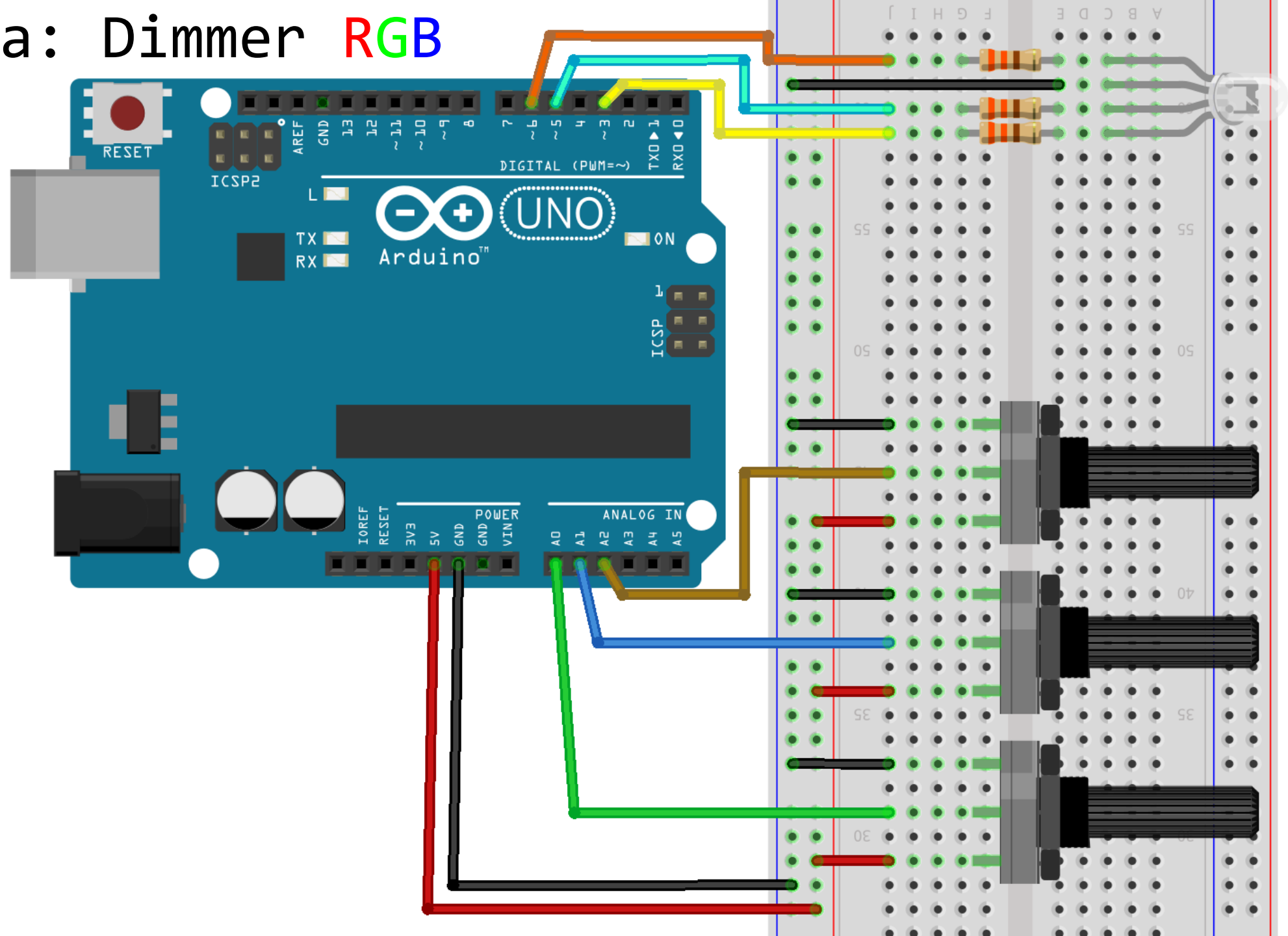
5º Programa: Dimmer RGB

Objetivo:

Controlar o brilho de cada cor de um LED RGB de forma independente através de potenciômetros

5º Programa: Dimmer RGB

Montagem:



THAT'S ALL FOR TODAY FOLKS