

Minicurso Arduino

🔍 Guia básico para iniciantes - Eletrônica Digital

Start!





Quem somos nós



[LinkedIn Marilene](#)

Marilene Andrade Garcia
Eng Comp 2017
Dev. Soft. Embarcado CNHI



[LinkedIn Gabriela](#)

Gabriela Barion Vidal
Eng Comp 2018
Dev. Machine Learning LSI-TEC

Links de Páginas de Estágio:

[Estágio de Verão Instituto Eldorado 2024](#)
[Estágio Case & New Holland Industrial](#)



Arduino UNO R3



```
Blink | Arduino 1.8.5

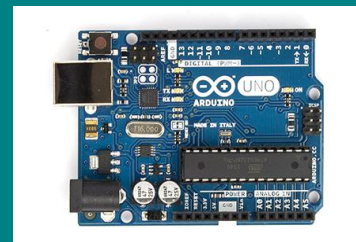
Blink §

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
}
```



Projeto Open Hardware com uma comunidade engajada que contribui com código, exemplos de uso e tutoriais.

Página Web Arduino

Massimo Banzi: Como o Arduino está tornando a imaginação Open-Source

Massimo Banzi
Co-founder, Chairman & CMO

David Cuartielles
Co-founder & Content Lead

Tom Igoe
Co-founder

David Mellis
Co-founder



Arduino e o Movimento Maker



open source
hardware

Faça Você mesmo - DIY ("Do it Yourself")

Projetos de Arduino

Faça seu Arduino



Adafruit METRO



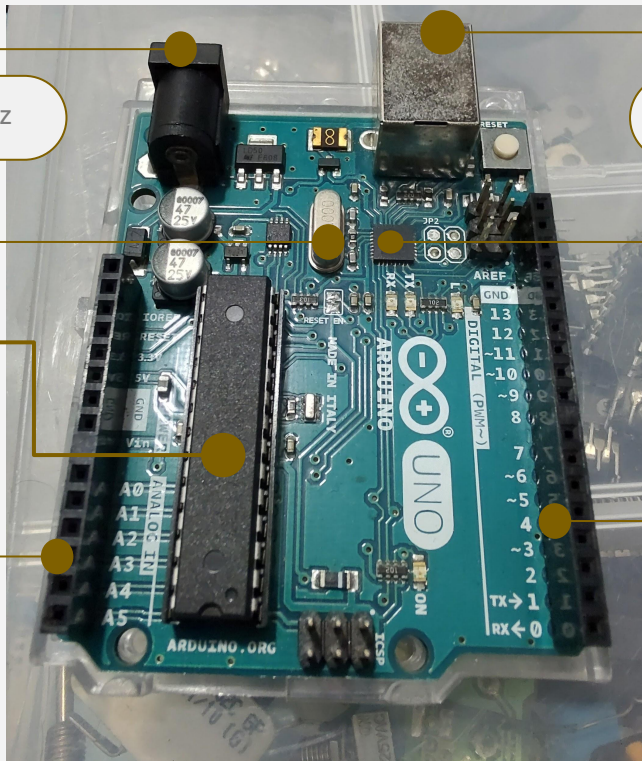
Limor Fried (ladyada)



adafruit



Conhecendo a Placa



Cristal Oscilador 16MHz

Microcontrolador Atmel
Atmega 16U2 (USB bridge)

Entrada de alimentação

Conector USB tipo B

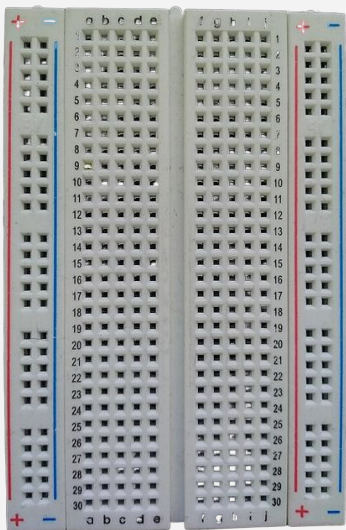
Microcontrolador Atmel
Atmega 328p

Portas Digitais

Portas Analógicas



Conhecendo os periféricos



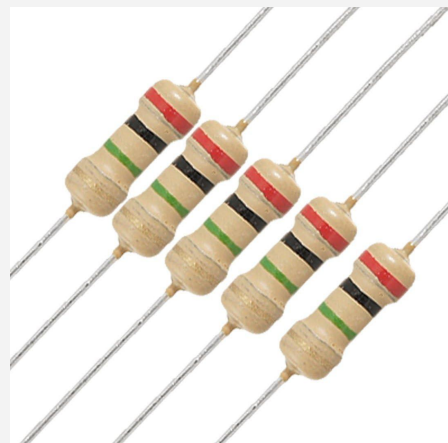
Protoboard



LED - Diodo emissor de luz



Botão



Resistor

No decorrer do curso vamos usar e entender cada um



Conhecendo a IDE

Ambiente de Desenvolvimento Integrado - IDE ("Integrated Development Environment")

Baixar a IDE

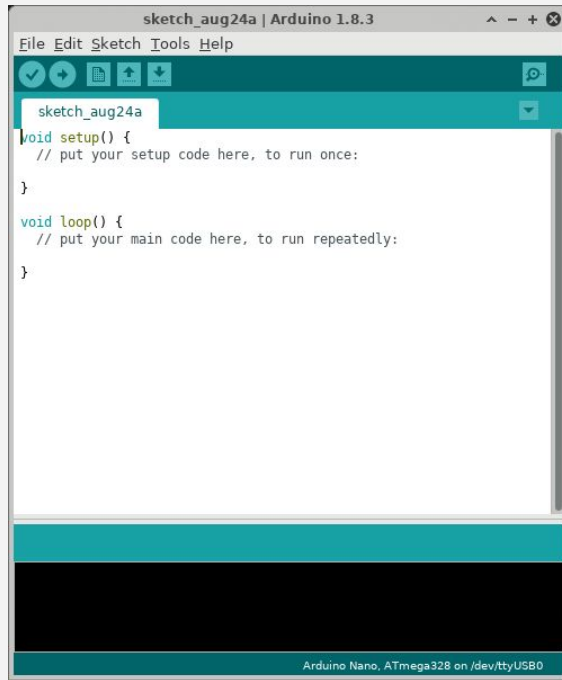
IDE online

Função **setup()**

Instruções executadas apenas uma vez, normalmente utilizada para configurações iniciais

Função **Loop()**

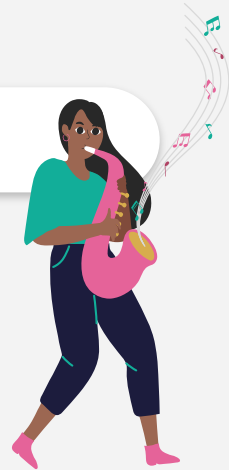
Instruções executadas continuamente (laço de repetição)



- Editor de texto
- Compilar
- Upload
- Novo
- Abrir
- Salvar
- Monitor serial



Prática 1 - Blink



O "Hello World" dos microcontroladores
Piscar um LED - Vamos usar o LED que tem na placa - Porta 13 Digital

Necessário configurar (Tools):

Board

Processor

Port

Etapas:

File -> Examples -> 01.Basics -> Blink

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                     // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                     // wait for a second
}
```




Entendendo o código

Funções:

pinMode(LED_BUILTIN, OUTPUT);

Configurar a porta 13 (LED_BUILTIN = 13) como de saída (OUTPUT/INPUT)

Como o LED da placa está ligado na porta 13 ele vai receber o sinal

digitalWrite(LED_BUILTIN, HIGH);

Colocando a voltagem alta (HIGH = 5V) na porta 13

digitalWrite(LED_BUILTIN, LOW);

Colocando a voltagem baixa (LOW = 0V) na porta 13

delay(1000);

“Trava” a execução de outras instruções por um determinado período de tempo

Prática 1.1: Alterar o valor do delay e substituir *LED_BUILTIN* pelo valor 13



No curso de hoje, iremos abordar apenas os sinais digitais

Q Diferença Eletrônica Digital/Analógica

Digital

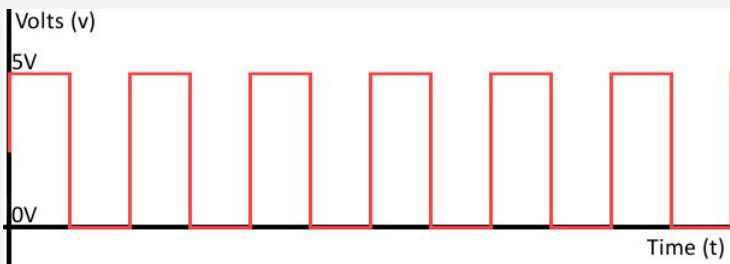
- Apenas dois valores possíveis
0 e 1 (Low e High)

Saída digital

Saída é 5V (1) ou 0V (0)

Entrada digital

Há tensão na porta (1) ou não há (0)



Analógica

- Sinais contínuos
Infinitos valores possíveis (ou quase)

Saída analógica

Valor entre 0V e 5V

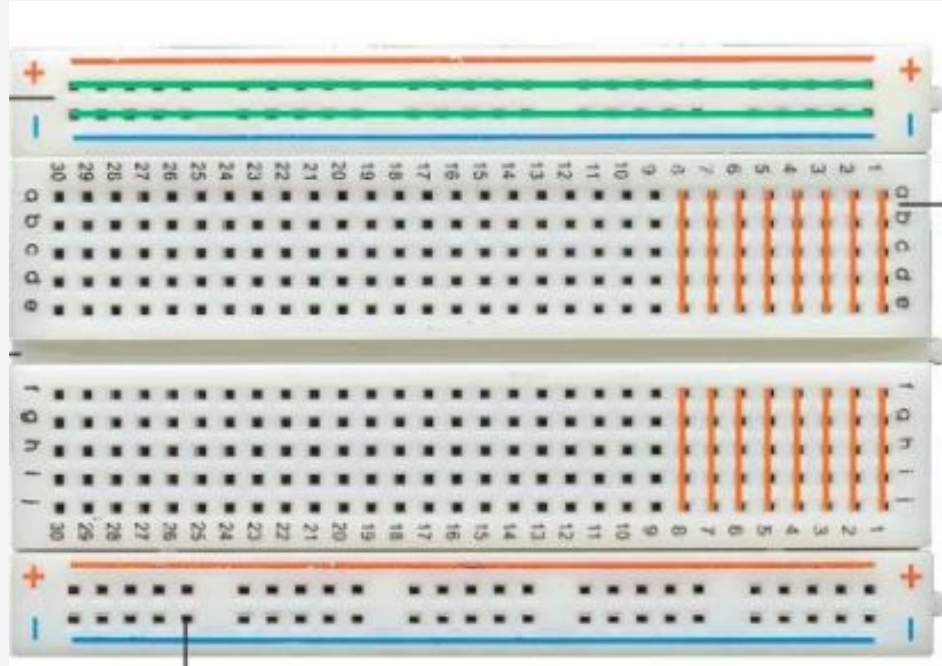
Entrada analógica

Valor da tensão na porta





Entendendo a Protoboard



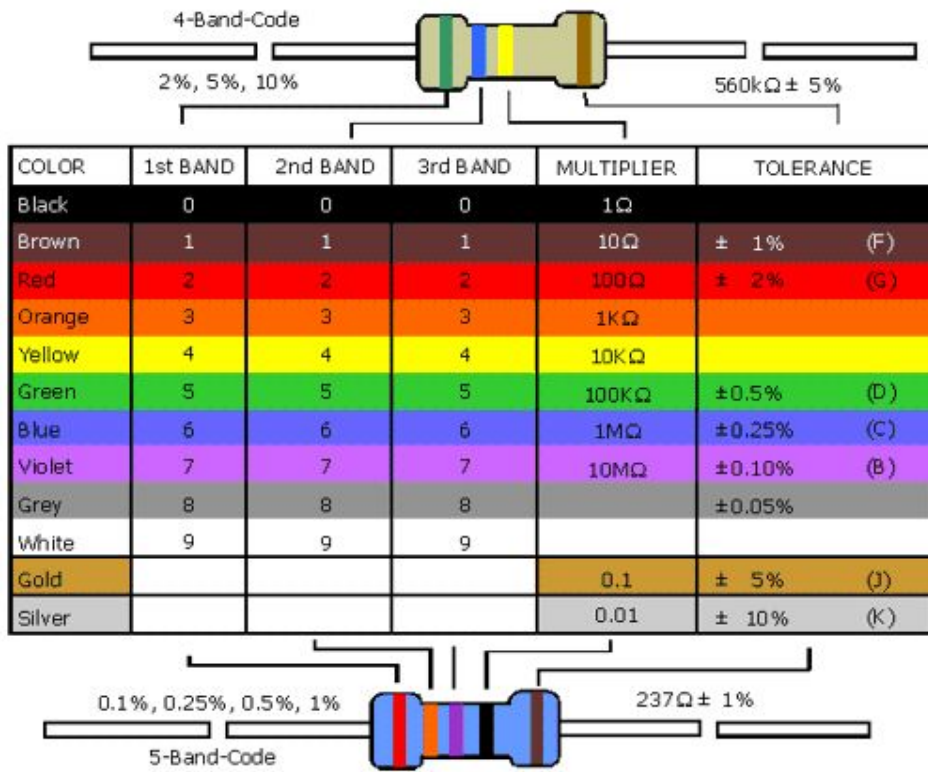
Placa para prototipagem de circuitos

Os furos do meio são interligados na vertical

Os furos nas extremidades são ligados na vertical



Entendendo o Resistor



As faixas coloridas nele indicam o valor da resistência (Ohm) e a precisão

- Limitar a corrente elétrica do circuito.
- Causam queda de tensão.

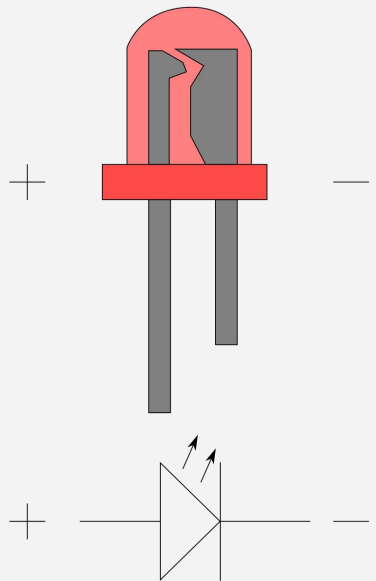
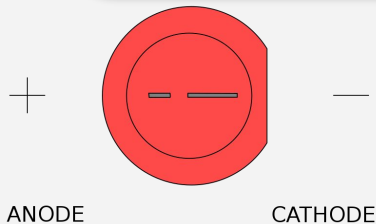
$$U = R.i$$



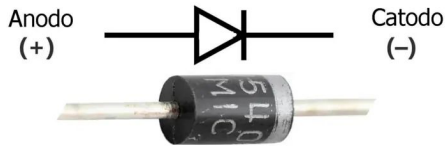
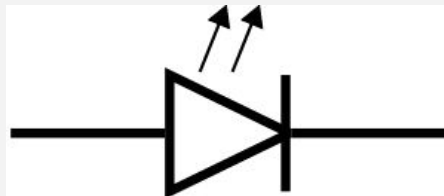
220 Ohms



Entendendo o LED



Então o LED tem
uma forma de ser
colocado no circuito

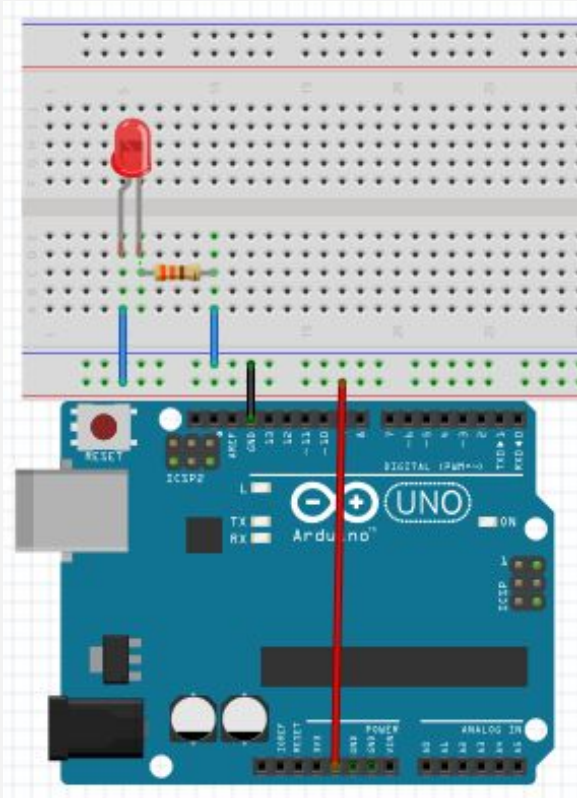


O que é um LED?
Diodo que emite luz
(“Light-Emitting Diode”)

O que é um Diodo?
Componente elétrico que
permite a passagem de corrente
apenas em um sentido

Todo LED é um diodo, mas nem
todo diodo é um LED

Q Como ligar o LED externo no Arduino?



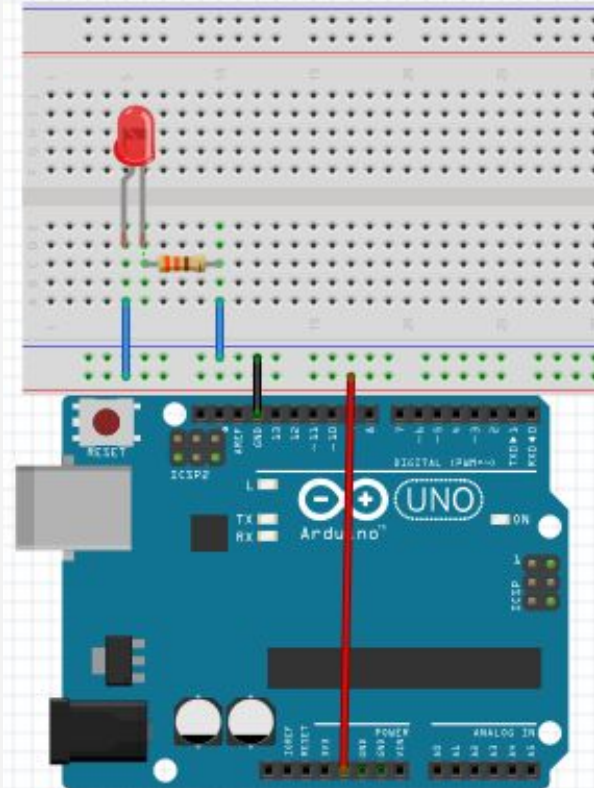
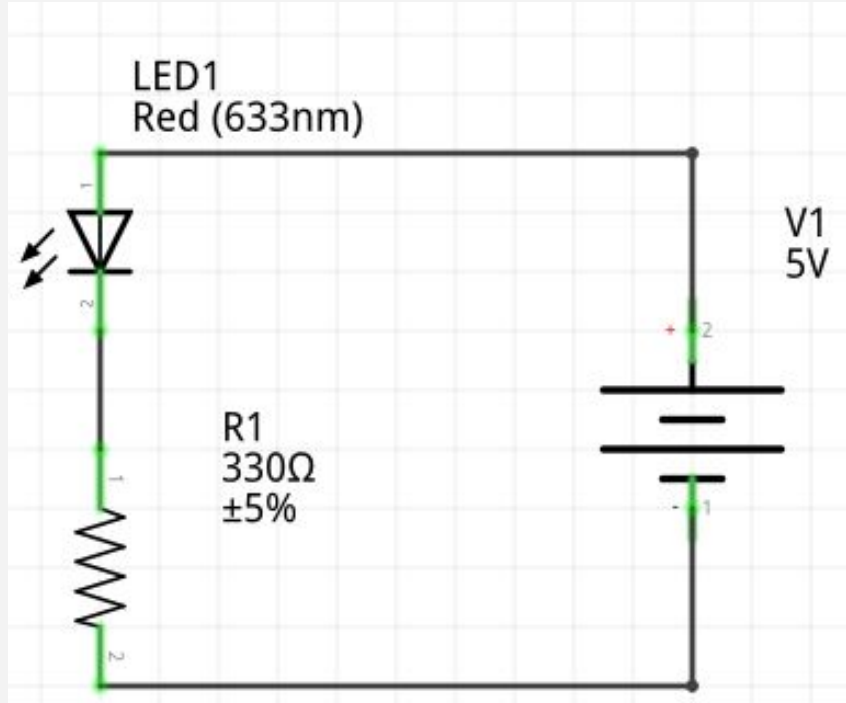
Calcular o resistor ideal para o LED

A conta é um pouquinho complicada, mas é mais ou menos assim:

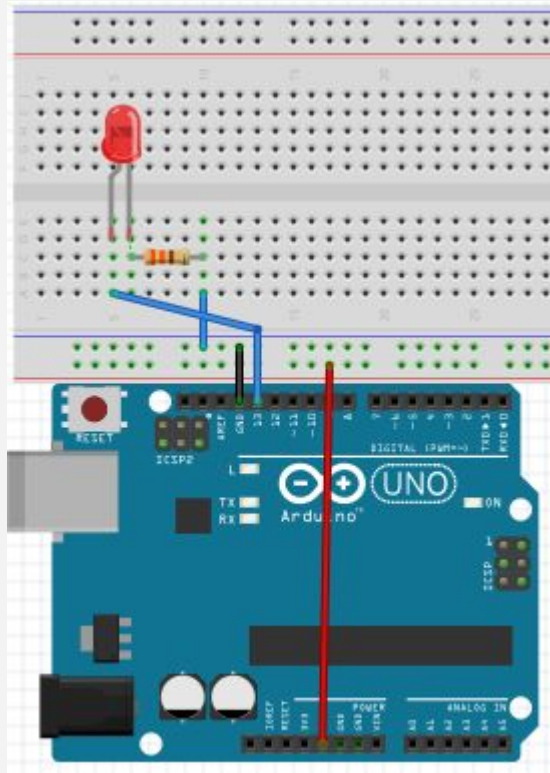
- Deve passar por volta de 20mA no LED
- Com esta corrente passando, ele dissipa uma tensão de aproximadamente 2V
- Alimentamos o circuito com 5V
- Então $(5V - 2V) / 20mA$ deve dar o valor da resistência

Mas não precisamos da potência toda do LED
Podemos colocar um resistor entre **200Ω e 400Ω**

Q Como ligar o LED externo no Arduino?



Q Como ligar o LED externo no Arduino?



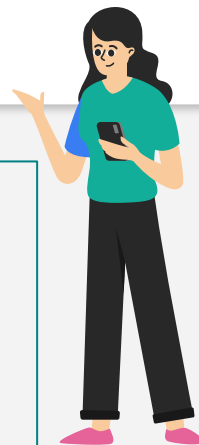
Ligando o LED em uma porta digital podemos controlar quando ele fica aceso ou apagado

Da mesma forma como fizemos na primeira prática

```
void setup() {  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);  
  delay(1000);  
  digitalWrite(13, LOW);  
  delay(1000);  
}
```




Prática 2 - Acendendo LEDs



1. Acender o LED Vermelho usando a Porta Digital 12 do Arduino
2. Fazer esse LED Vermelho piscar com um intervalo de 500 ms
3. Colocar outro LED de outra cor para piscar usando o mesmo intervalo de tempo, mas de maneira que enquanto um esteja acesso o outro esteja apagado.

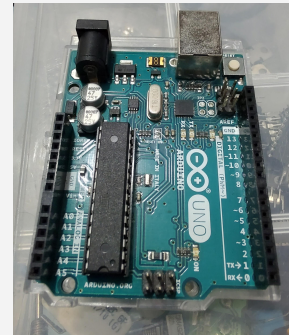
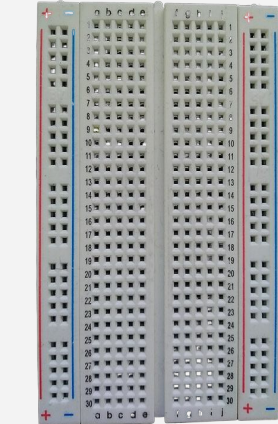
Desafio:

- Ligar 3 LEDs em 3 portas digitais diferentes. Acender o primeiro, depois o segundo, o terceiro e assim sucessivamente. Apenas um acesso por vez



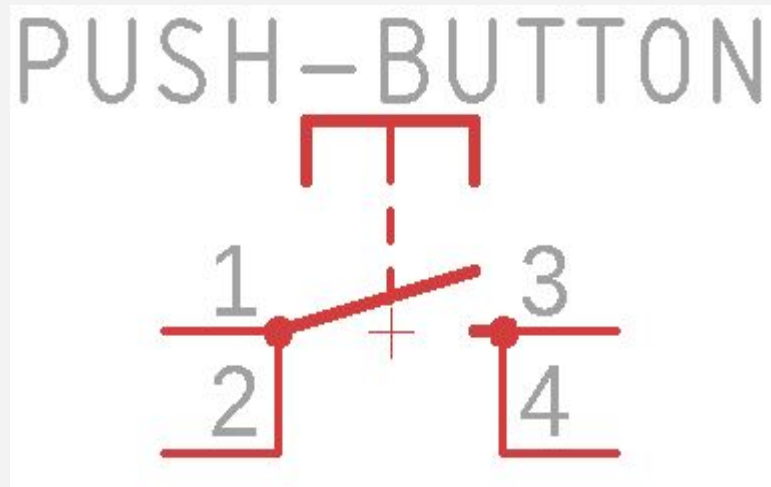
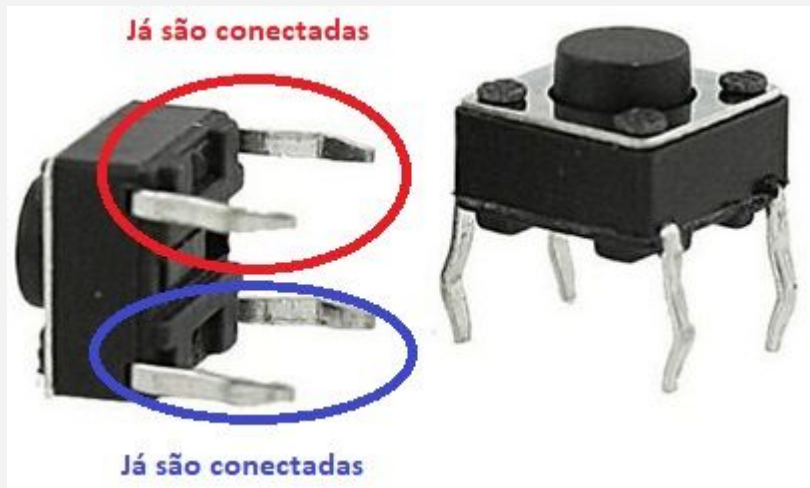
Recapitulando...

- Entendemos o que são o Arduino, a protoboard, o resistor e o LED
- Aprendemos como controlar uma saída digital de sinal (enviamos os sinais para acender os LEDs)
- Agora vamos aprender a ler uma entrada digital de sinal, utilizando um outro periférico, o botão ("push button")





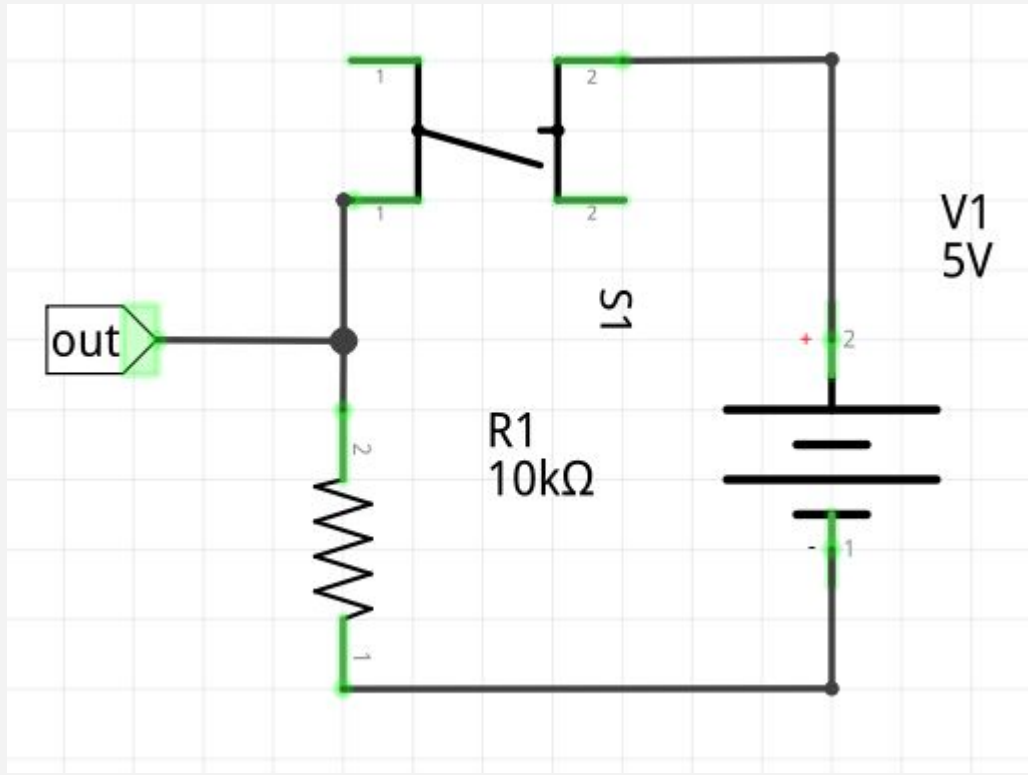
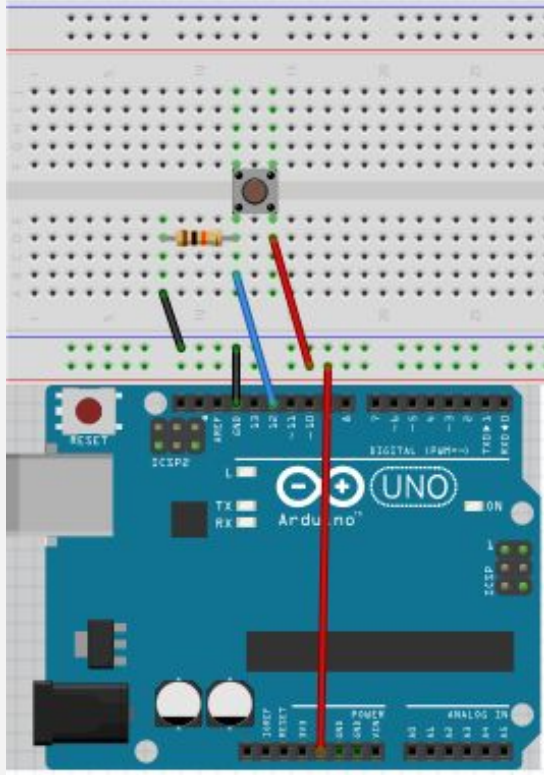
Entendendo o Push Button



- Enquanto o botão não está sendo apertado não há contato entre os pares de terminais (perninhas)
- Enquanto pressionamos o contato é fechado

Estamos usando push buttons de 4T (quatro terminais, quatro perninhas), mas vocês podem encontrar versões de dois terminais.

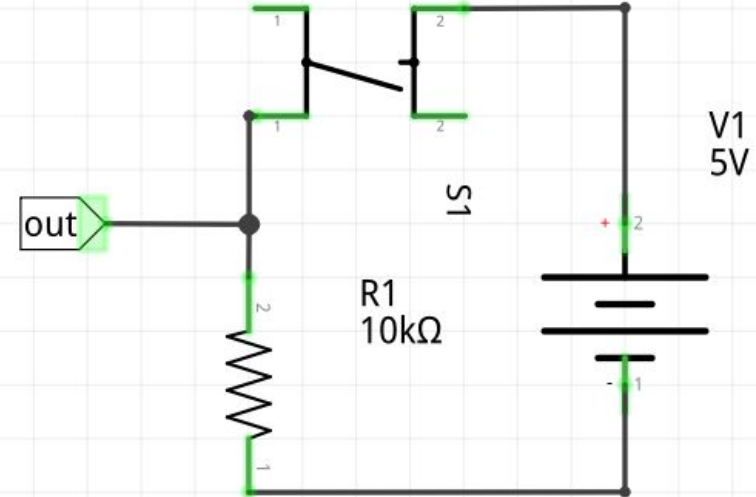
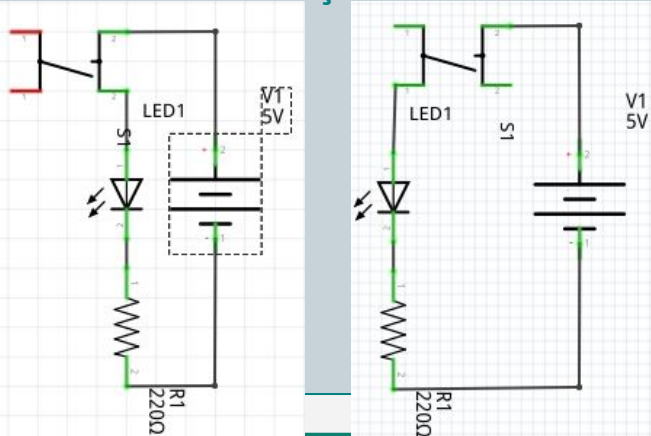
Q Como ligar o Push Button no Arduino?



Q Como ligar o Push Button no Arduino?

Qual o motivo do resistor de $10K\Omega$?
Sem ele, quando o botão fosse fechado, iria ocorrer um curto circuito. A tensão de 5V estaria diretamente ligado no GND ("ground") 0V.

Obs. Prestar atenção nos terminais da ligação



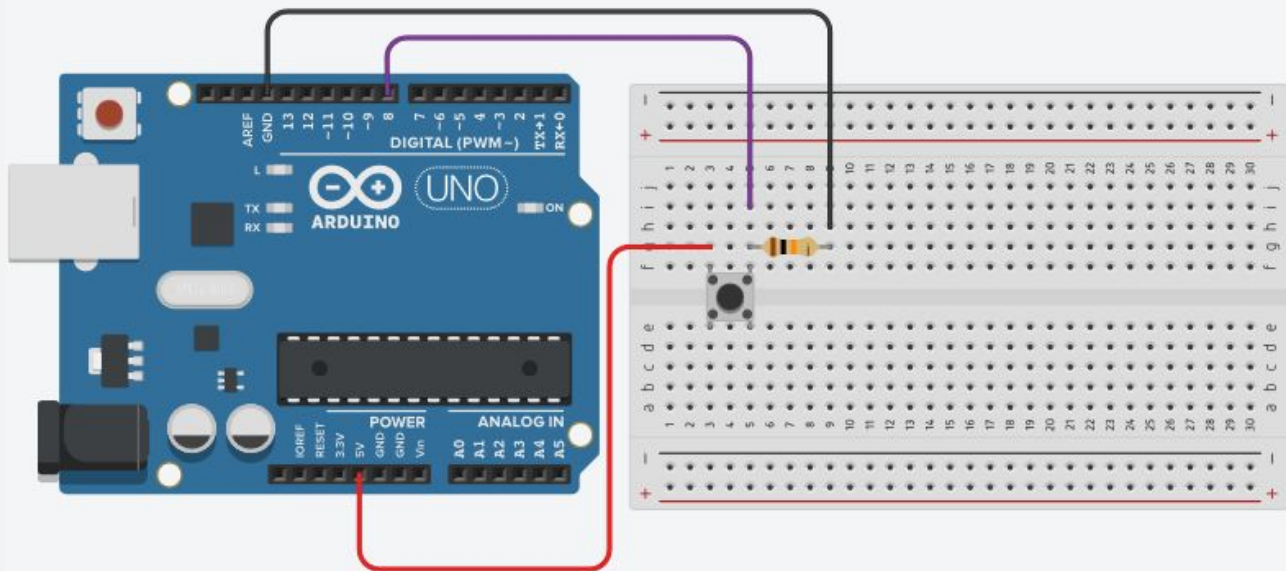


Lendo o Estado do Push Button

digitalRead(buttonPin);

Ler a voltagem na porta conectada ao botão (*buttonPin = 8*)

Possíveis valores (HIGH e LOW)





Exemplos de código

```
void setup() {  
    pinMode(12, INPUT) ;  
    pinMode(13, OUTPUT) ;  
}  
  
void loop() {  
    if (digitalRead(12)) {  
        digitalWrite(13, HIGH) ;  
    } else {  
        digitalWrite(13, LOW) ;  
    }  
}
```

```
// constants won't change. They're used here to set pin numbers:  
const int buttonPin = 2; // the number of the pushbutton pin  
const int ledPin = 13;   // the number of the LED pin  
  
// variables will change:  
int buttonState = 0; // variable for reading the pushbutton status  
  
void setup() {  
    // initialize the LED pin as an output:  
    pinMode(ledPin, OUTPUT);  
    // initialize the pushbutton pin as an input:  
    pinMode(buttonPin, INPUT);  
}  
  
void loop() {  
    // read the state of the pushbutton value:  
    buttonState = digitalRead(buttonPin);  
  
    // check if the pushbutton is pressed. If it is, the buttonState is HIGH:  
    if (buttonState == HIGH) {  
        // turn LED on:  
        digitalWrite(ledPin, HIGH);  
    } else {  
        // turn LED off:  
        digitalWrite(ledPin, LOW);  
    }  
}
```



Prática 3 - Lendo o Push Button



1. Se o botão estiver pressionado, o LED Amarelo acende, caso o botão não esteja pressionado ele apaga.
2. Ao apertar o botão, o LED Amarelo muda estado. Se estiver aceso vai ser apagado. Caso estiver apagado ira acender.

Desafio:

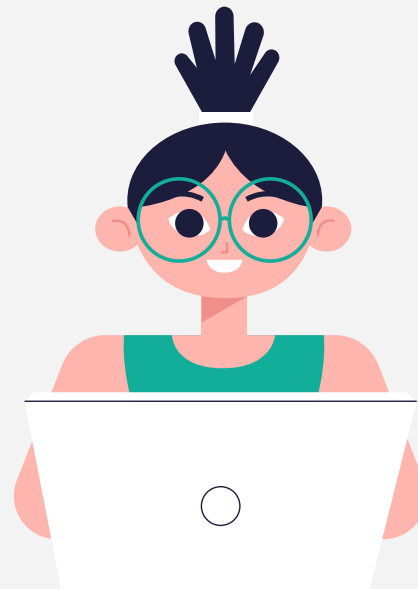
- Ligar três LEDs e um botão ao Arduino. O primeiro LED começa aceso. Ao pressionar o botão, apaga-se o LED aceso e acende o próximo (ciclicamente).



Desafio Final



Vocês Conhecem o Jogo Genius?



[Site Com Exemplo no Arduino](#)

Sugestão- Faça o seu, pesquise mais exemplos e ideias para se inspirar!



Extra - Eletrônica Analógica

1. Comunicação Serial
2. AnalogRead()
3. AnalogWrite() e PWM

Comunicação com o computador

- Envio de dados
- Recebimento de dados/comandos

```
void setup() {  
  Serial.begin(9600) ;  
}  
  
void loop() {  
  Serial.println("Hello World") ;  
  delay(500) ;  
}
```

Entrada analógica (AnalogIn)

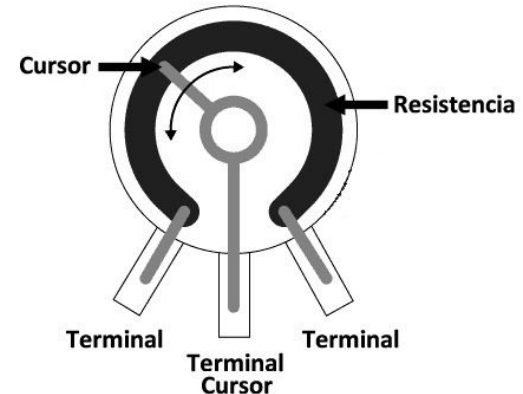
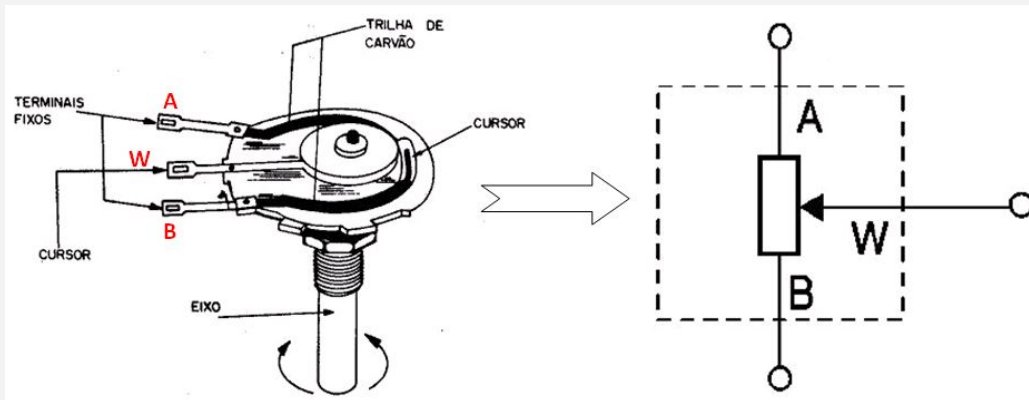
- Converte a tensão no pino para um valor entre 0 (0V) e 1023 (5V)
- Usamos para obter valores de sensores como:
 - Temperatura
 - Luminosidade
 - Nível de determinado gás
 - Leitura de um potenciômetro



Extra - Eletrônica Analógica

Potenciômetro

- Componente que permite variação na resistência
- O potenciômetro tem três terminais
- A resistência entre as duas extremidades é sempre fixa
- A resistência entre o pino do meio e qualquer uma das extremidades é variável
- Conseguimos variar essa resistência girando o pino central





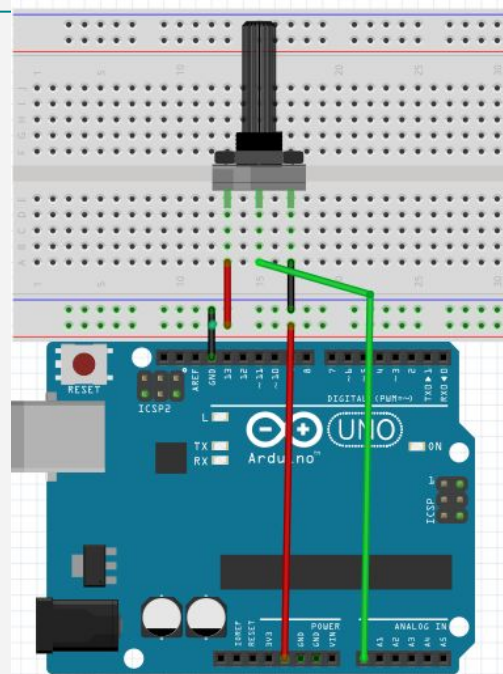
Extra - Eletrônica Analógica

Potenciômetro

- Conectamos o terminal central a uma das portas analógicas (A0 ~ A5)

```
void setup() {  
  Serial.begin(9600) ;  
}  
  
void loop() {  
  Serial.println(analogRead(0)) ;  
  delay(500) ;  
}
```

Prática Extra - Ler um potenciômetro com o terminal central na porta A0
`analogRead(porta)`

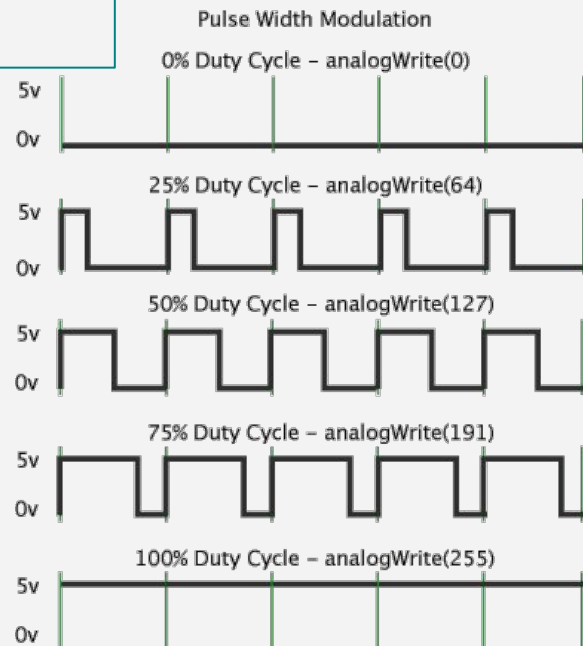
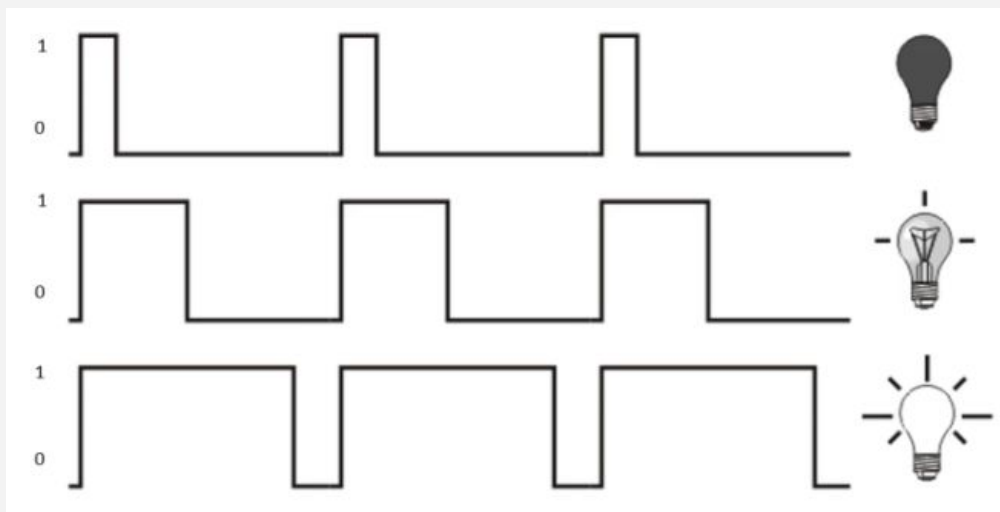




Extra - Eletrônica Analógica

Saída analógica - O Arduino não tem saída analógica! Ele usa PWM

- Portas Digitais com símbolo til (“~”) podem ser usadas como PWM (**Pulse Width Modulation**)
- Simula valores entre 0V (0) e 5V (255)





Extra - Eletrônica Analógica

Prática Extra - Alterar gradativamente o brilho de um LED ligado na Porta 11
`analogWrite(porta,valor)`

```
int val = 0 ;

void setup() {
  pinMode(11, OUTPUT) ;
}

void loop() {
  if (val >= 255) {
    val = 0 ;
  }
  analogWrite(11, val) ;
  val += 10 ;
  delay(200) ;
}
```



Muito Obrigada pela Presença!!



Agradecimentos Especiais:

- Bárbara pelo Convite
- Gabriela (Band) pela ajuda hoje
- ADA - Projetos em Engenharia de Computação - pelos Slides que usei de inspiração para montar essa apresentação
- Professora Sarita por providenciar o material
- Vocês pela presença

Dúvidas?



Thanks



Do you have any questions?

youremail@freepik.com
+91 620 421 838
yourcompany.com



CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

Please keep this slide for attribution.

