

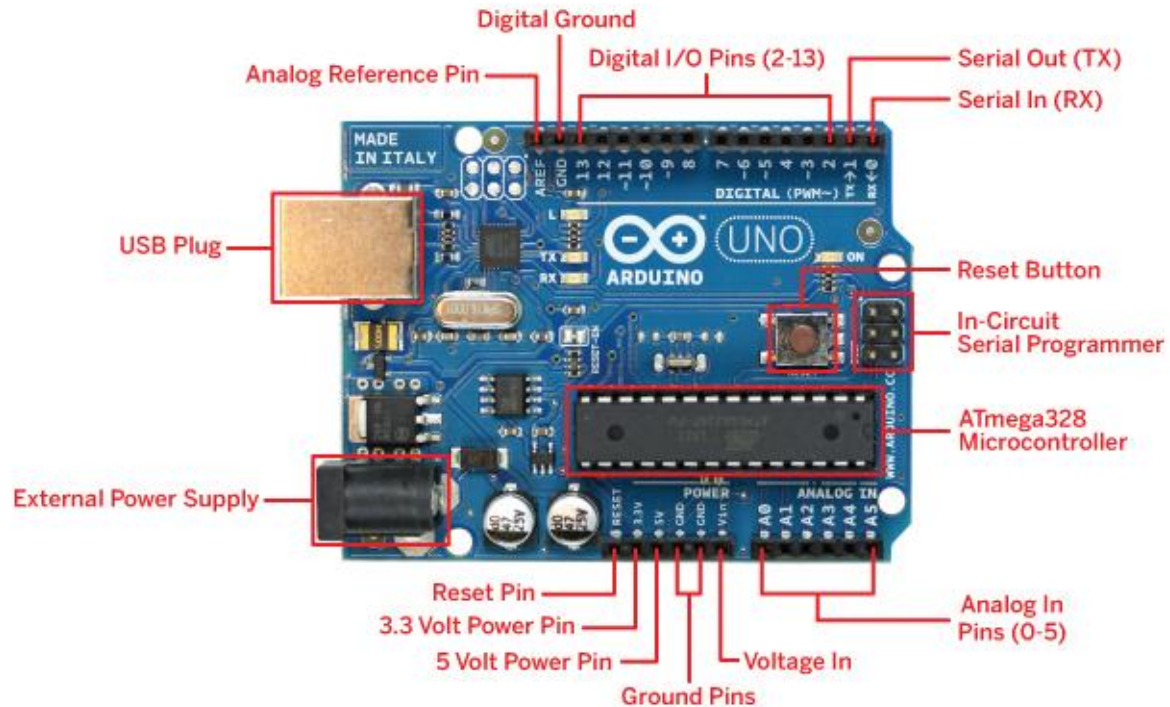
Hardware Livre
USP

Oficina de
Introdução ao
Arduino

Conteúdo da oficina

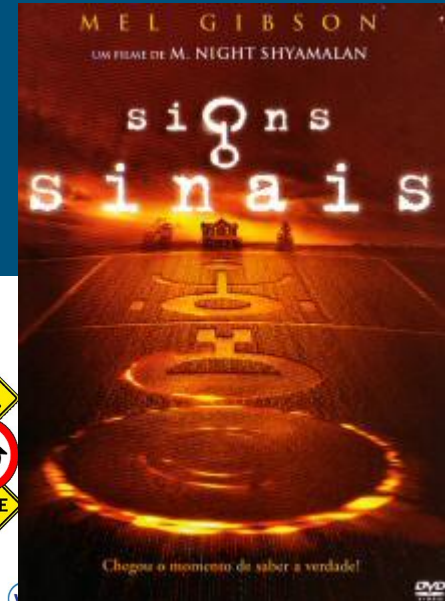
- Estrutura de uma placa Arduino
- Conceitos de sinais elétricos
 - Sinal digital, analógico, PWM
- Componentes eletrônicos básicos
 - LED, resistor, protoboard
- Ambiente de desenvolvimento
 - Os blocos básicos de um programa em Arduino
 - Receber e enviar sinais digitais
- Projeto Genius (Jogo da memória)

O Arduino UNO



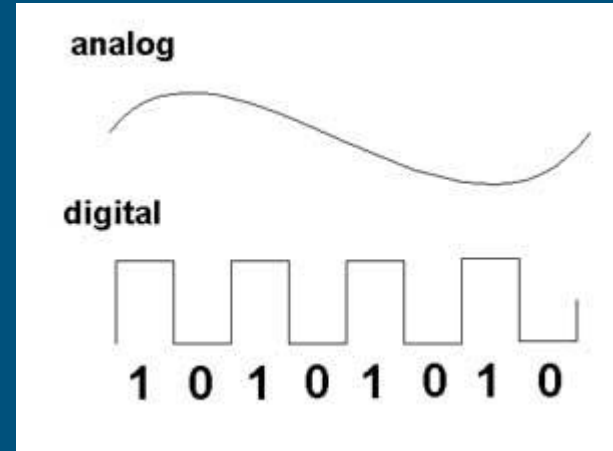
Entrada e Saída (I/O)

- Entrada e saída de dados / sinais em relação ao microcontrolador
- Entrada
 - Um sinal externo é enviado ao microcontrolador
- Saída
 - O microcontrolador gera um sinal e o envia ao meio externo
- Que sinais são esses?



Analógico Vs Digital

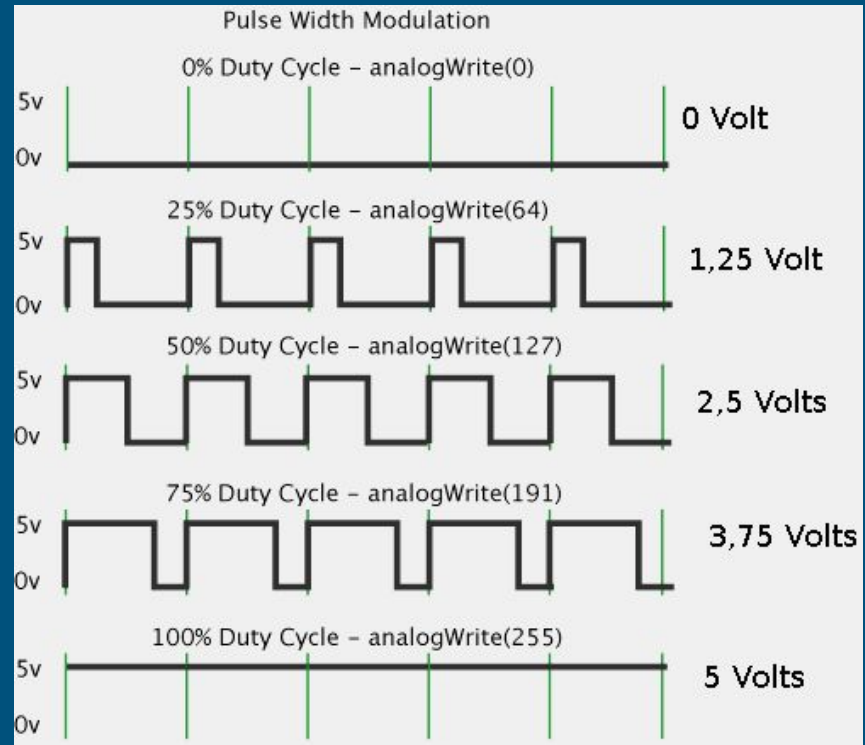
- Sinal analógico
 - Pode assumir qualquer valor entre o limite mínimo e máximo
- Sinal digital
 - Discretizado (geralmente apenas 0 ou 1)



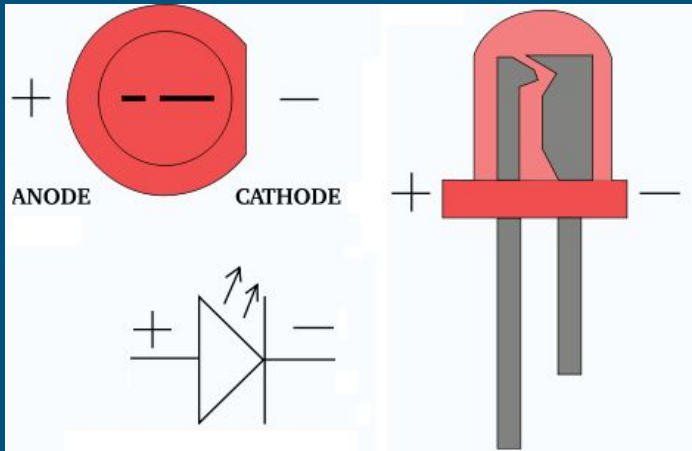
Pulse Width Modulation (PWM)

- Pulsos (ligado e desligado / 1 e 0) durante determinada proporção de tempo em um ciclo constante
- Simula um sinal analógico a partir de um sinal digital

<https://www.arduino.cc/en/Tutorial/PWM>



LED (Light Emitting Diode)



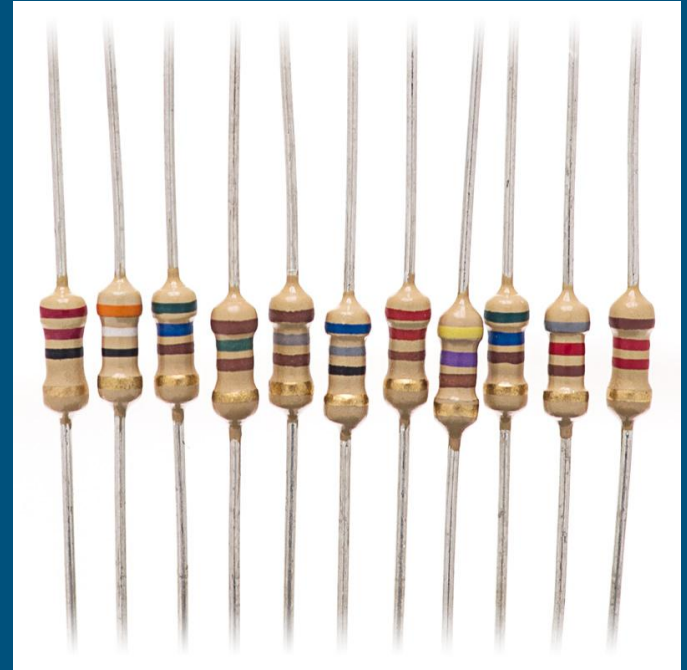
- Usualmente operam em um nível de tensão de 1,6 a 3,3 volts e sob uma corrente elétrica próxima de 20 mA
- Possuem polaridade
 - Perna maior é positiva
 - Perna menor é negativa

https://en.wikipedia.org/wiki/Light-emitting_diode

Resistores



- Dificultam a passagem de corrente elétrica
- Provocam queda do potencial elétrico de uma ponta para a outra

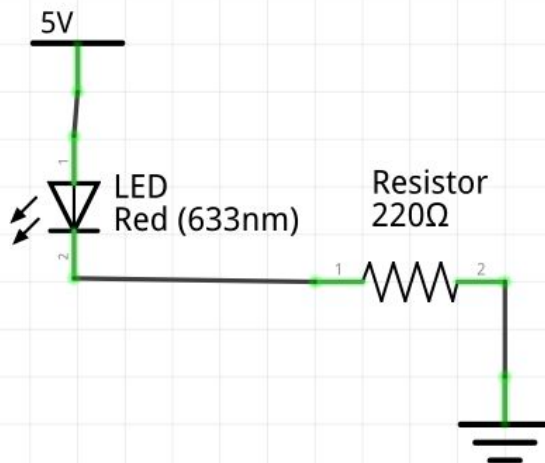


LED e Resistor

Normalmente junto ao LED é usado um resistor de 220 ou 330 ohms para evitar que o LED queime.

Fundamento teórico:

- Primeira lei de Ohm:
 $i = U / R$
- Lei de Kirchhoff das tensões (LKT)



Componentes

Core Parts

CORE

MINE

Visão Protoboard

Visão Esquemático

text

Visão PCB

text

Propriedades

text

text

text

text

text

text

text

text

text

text

text

text

text

text

text

text

text

text

text

text

text

text

text

text

text

text

text

text

text

text

text

text

text

Roteamento completo

Compartilhar

(x,y)=(4.857, -0.190) in 375 %

A IDE Arduino

- Instale a IDE do Arduino:
 - <https://www.arduino.cc/en/Main/Software>



The screenshot shows the Arduino IDE interface with a menu bar (File, Edit, Sketch, Tools, Help) and a toolbar. The main editor window displays a C++ sketch named 'memorygame'. The code includes a setup function and a loop function. The loop function contains a game logic section that checks for button presses and updates the game state. The status bar at the bottom indicates 'Done compiling.' and 'Binary sketch size: 7.912 bytes (of a 32.256 byte maximum)'.

```
File Edit Sketch Tools Help
memorygame
level = 1;
start = 1;
}

void loop()
{
  if(start) {
    for (int i = 0; i < length; i++) {
      playNote(notes[i], beats[i]*tempo);
      delay(tempo/2);
    }
    lcd.clear();
    lcd.print("Press Again to");
    lcd.setCursor(0, 1);
    lcd.print("Start");
    lcd.setCursor(5, 1);
    lcd.write((uint8_t)0);
    delay(100);
    lcd.setCursor(5, 1);
    lcd.write((uint8_t)1);
    delay(100);
  }

  if (getButtonPress()) {
    start = 0;
    lcd.clear();
    lcd.setCursor(0, 0);
    if (user == game && game != 0 && score < 10) {
      lcd.print("You Are Correct");
      score++;
      lcd.setCursor(0, 1);
      lcd.print("Score:");
      lcd.setCursor(6, 1);
    }
  }
}
```

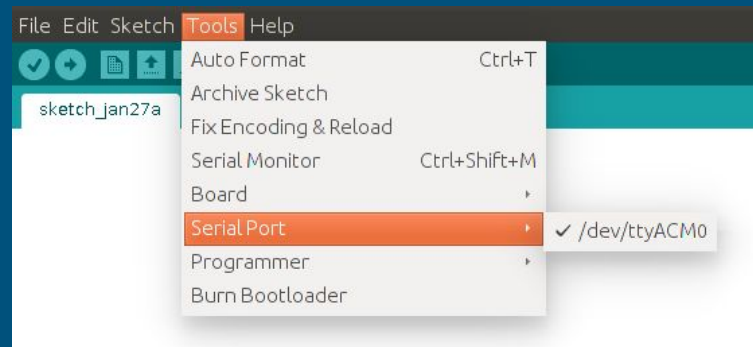
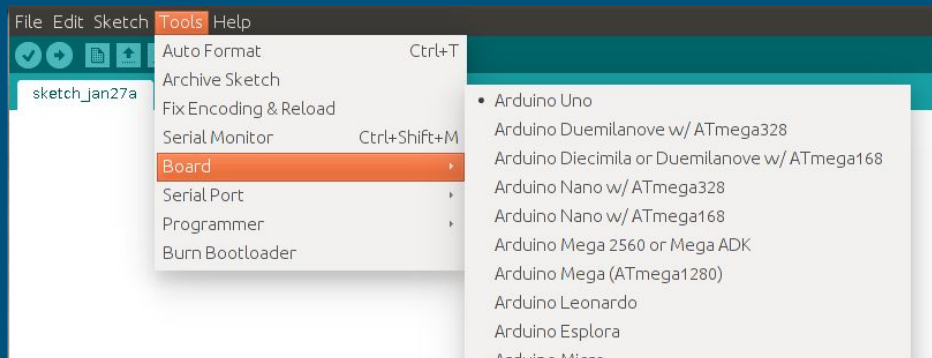
Done compiling.

Binary sketch size: 7.912 bytes (of a 32.256 byte maximum)

14

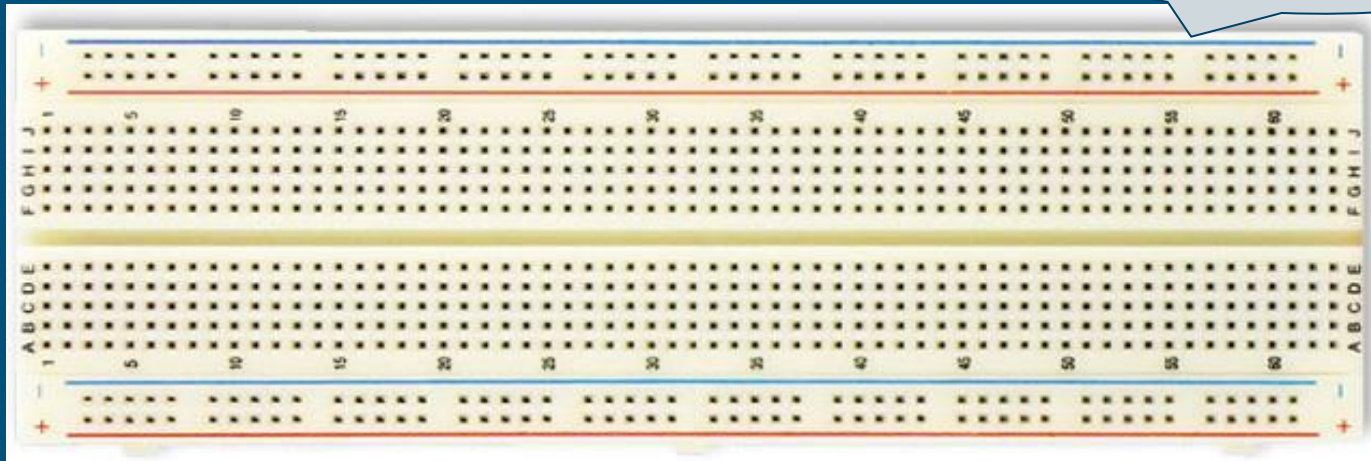
Checklist Inicial

- Conecte o Arduino em uma porta USB do computador
- Selecione a placa Arduino em Tools -> Board
- Selecione a porta USB em Tools -> Serial Port



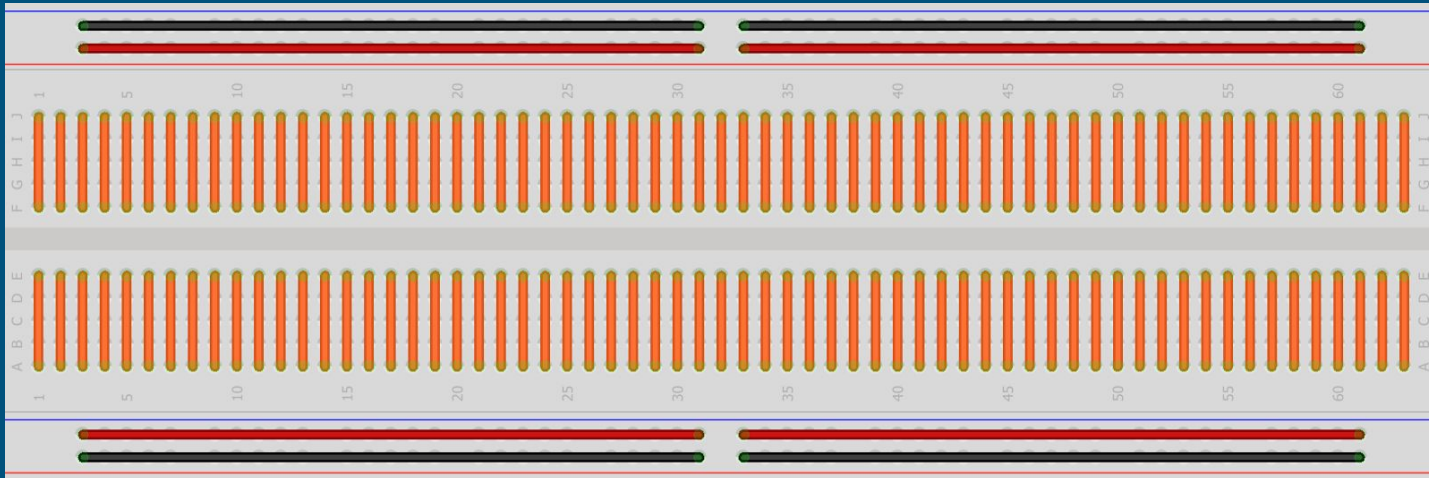
Protoboard / breadboard

Olá, lembra de mim?
Eu sou uma protoboard.



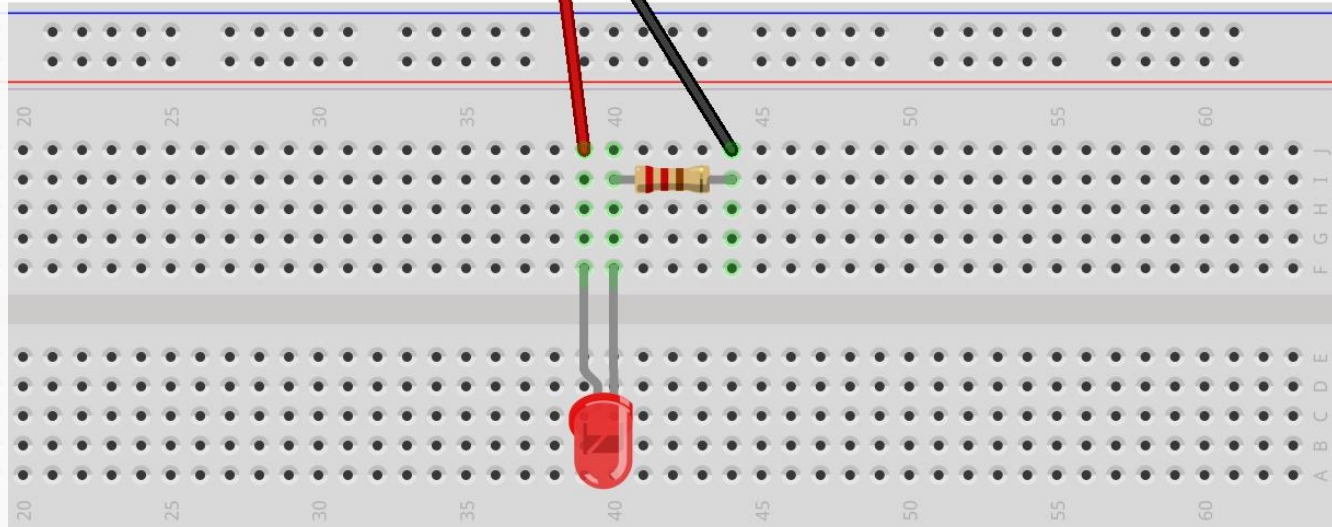
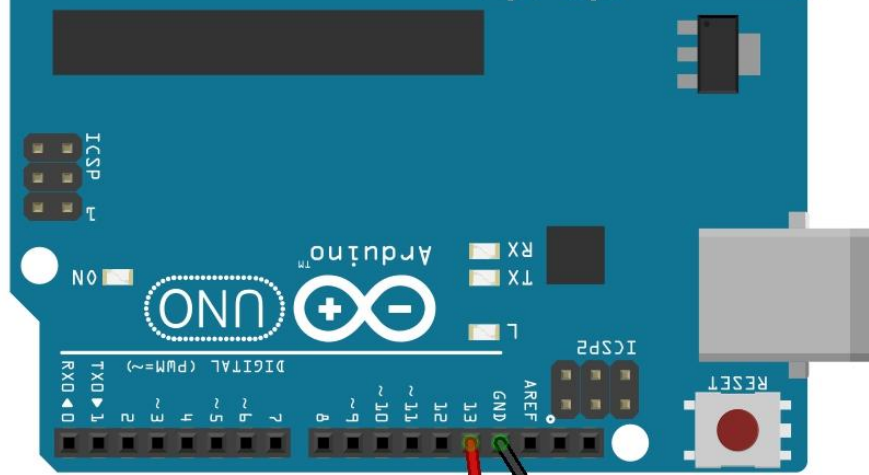
Protoboard / breadboard

- As ilhas no centro da placa estão conectadas “verticalmente”
- As ilhas nas laterais da placa estão conectadas “horizontalmente”





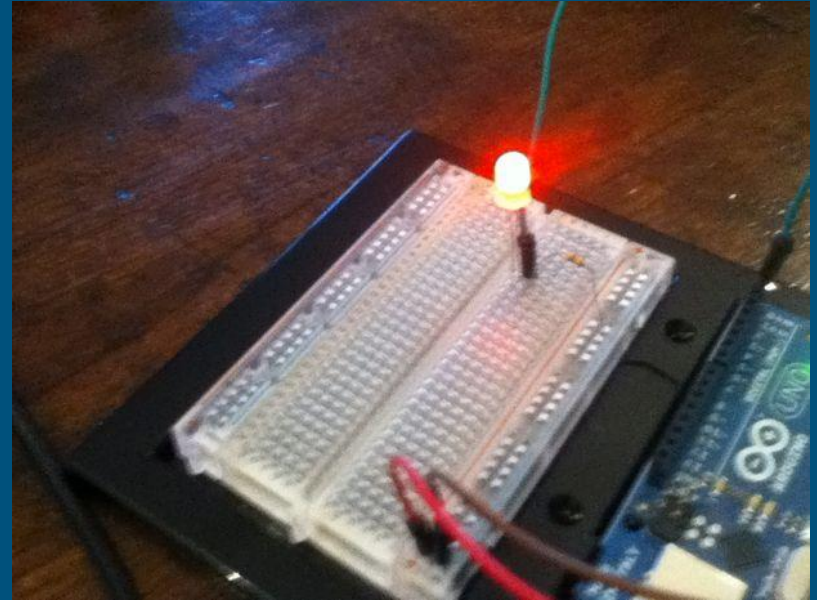
Projeto Blink

- Monte o projeto na protoboard
- Hardware necessário
 - Placa Arduino
 - Protoboard
 - 1 LED
 - 1 Resistor de 220 Ohms
 - Fios para conectar os componentes



Projeto Blink

- Abra o código do exemplo em
File -> Examples -> Basics -> Blink
- Compile o código 
- Envie o programa para o
Arduino 





The screenshot shows the Arduino IDE interface. The title bar reads "Blink | Arduino 2:1.0.5+dfsg2-4". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for checking, running, opening a file, uploading, and downloading. The code editor displays the following code:

```
Blink §  
  
// Blink  
  
int led = 13;  
  
void setup() {  
  pinMode(led, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(led, HIGH);  
  delay(1000);  
  digitalWrite(led, LOW);  
  delay(1000);  
}
```

Annotations with arrows point to specific parts of the code:

- An arrow points from the text "Apelidamos o pino 13 de led" to the variable `led` in `int led = 13;`.
- An arrow points from the text "Este conjunto de instruções é executado apenas uma vez sempre que o Arduino é inicializado" to the `setup()` function.
- An arrow points from the text "O pino do led vai ser usado para envio de sinais" to the `pinMode(led, OUTPUT);` line.
- A bracket on the right side of the `loop()` function is annotated with the text "Este conjunto de instruções é executado continuamente enquanto o Arduino estiver ligado".

Linguagem de programação

- Linguagem baseada em C/C++
- Estrutura simples
 - `setup()`
 - Executa apenas uma vez logo no início
 - `loop()`
 - Executa ciclicamente após o `setup()`

Atividades

- Modifique o projeto
 - Sugestões:
 - altere a frequência com que o LED pisca
 - faça dois LEDs piscarem juntos
 - faça dois LEDs piscarem alternadamente
- Utilize `analogWrite(led, <valor>)` no lugar de `digitalWrite(led, HIGH)`
 - <valor> é um número entre 0 e 255 (inclusive)
- Rode o exemplo em File -> Examples -> Basics -> Fade
 - Projeto completo disponível em <https://www.arduino.cc/en/Tutorial/Fade>

Referência da linguagem

<https://www.arduino.cc/en/Reference/HomePage>

Buy Download Products ▾ Learning ▾ Forum Support ▾ Blog

Structure

begin()
end()
loop()
setup()

Control Structures

if
if...else
if...else if... else
switch case
while
do... while
break
continue
return
goto

Other Syntax

semicolon)
(curly braces)
(inline namespace)

Variables

Constants

HIGH | LOW
INPUT | OUTPUT | INPUT_PULLUP
LED_BUILTIN
true | false
integer constants
floating point constants

Data Types

void
boolean
char
unsigned char
byte
int
unsigned int
word
long

Functions

Digital I/O

pinMode()
digitalWrite()
digitalRead()

Analog I/O

analogReference()
analogRead()
analogWrite() - PWM

Due & Zero only

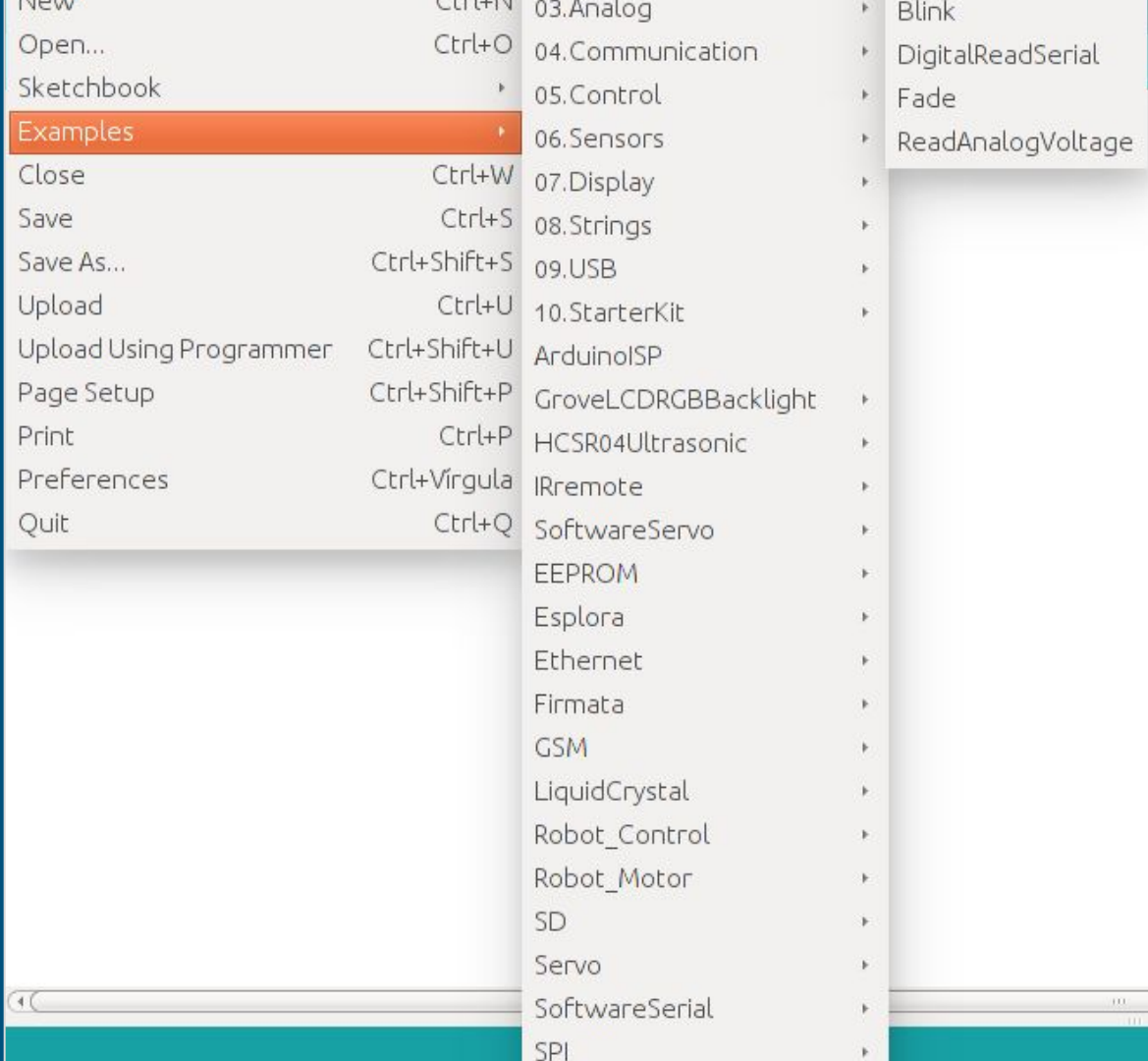
analogReadResolutions()
analogWriteResolutions()

Advanced I/O

tone()
noTone()
shiftOut()

Milhares de exemplos

A IDE Arduino vem com vários exemplos, dos mais simples (pisca um LED) aos mais elaborados envolvendo sensores, motores, etc..

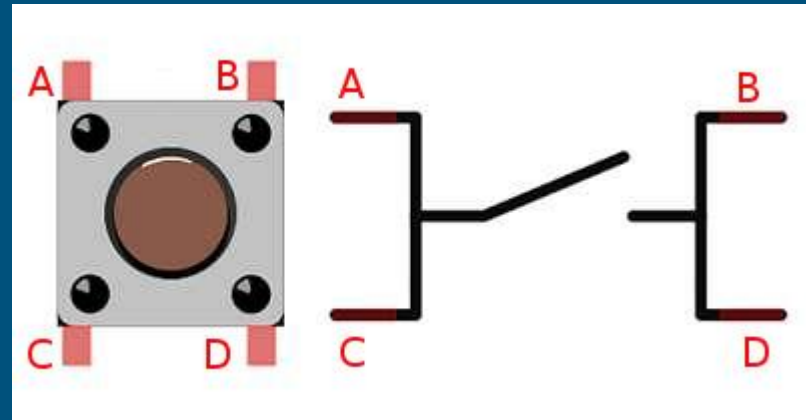


Recebendo e enviando sinais digitais

Objetivo: Ligar e desligar um LED de acordo com o estado de um botão.



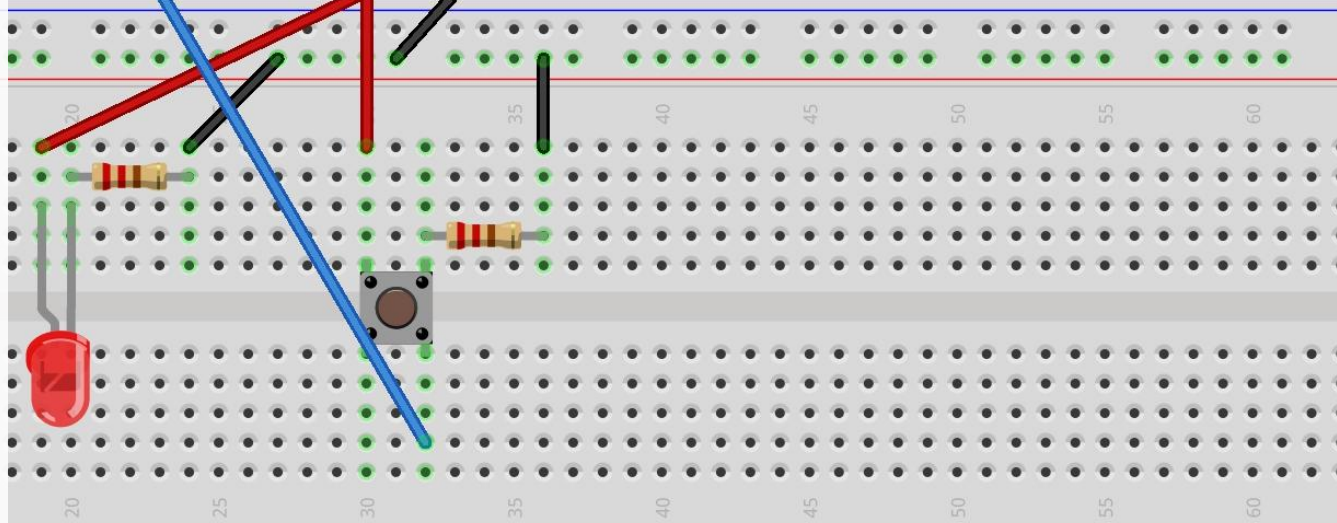
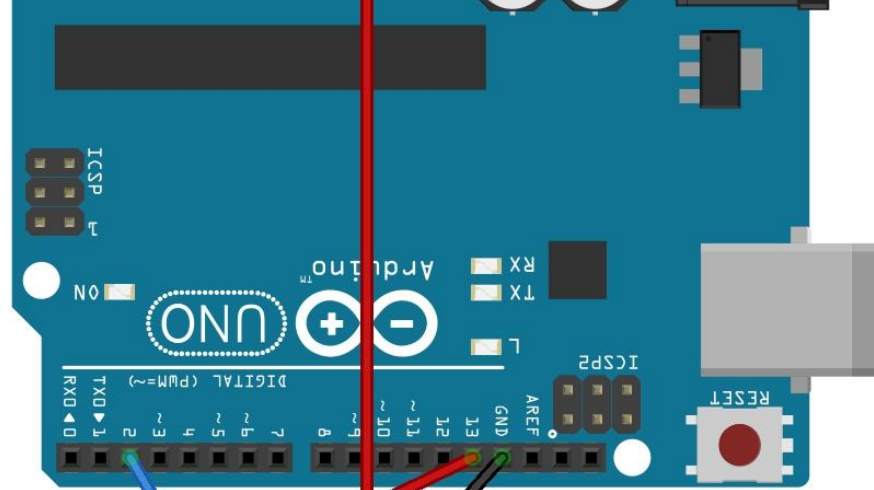
O botão / push button



Quando pressionado o botão conecta os contatos A e C aos contatos em B e D.

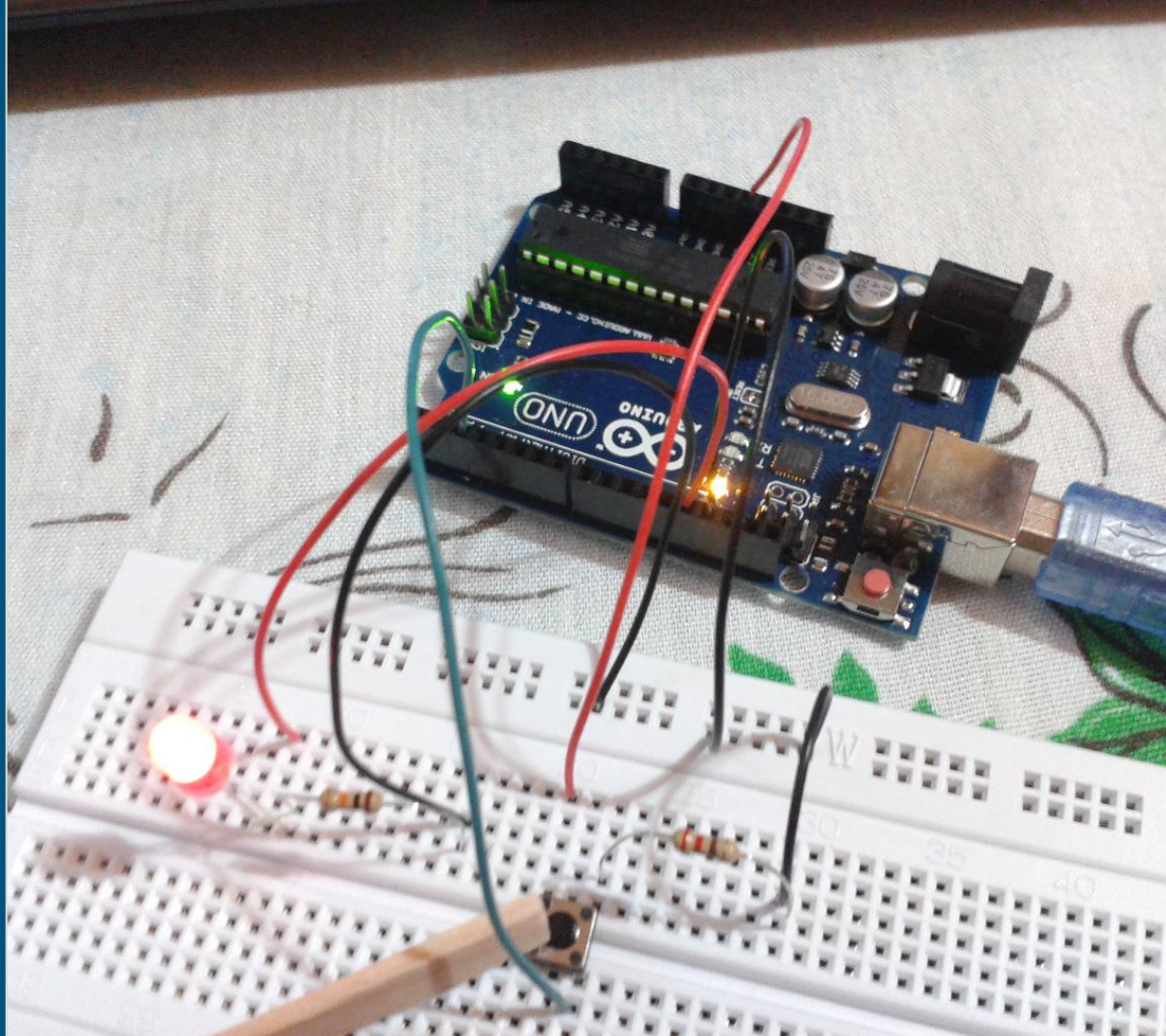
Projeto Botão

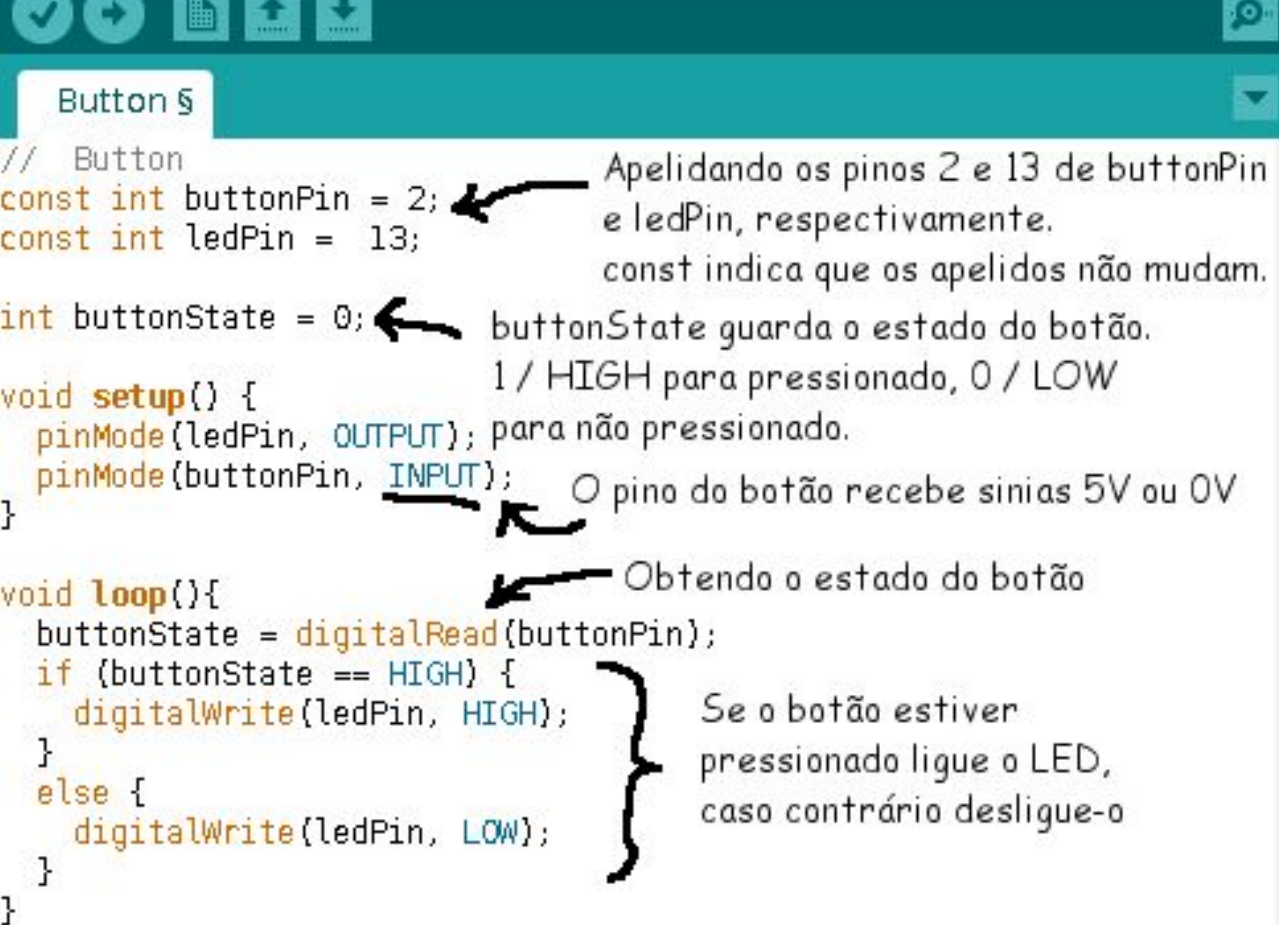
- Hardware necessário
 - Placa Arduino
 - Protoboard
 - 1 LED
 - 2 Resistores de 220 ohms
 - Fios para conectar os componentes
 - Um botão



Projeto Botão

- Código para o projeto em: File -> Examples -> Digital -> Button
- Compile o código (opcional)
- Envie o programa para o Arduino

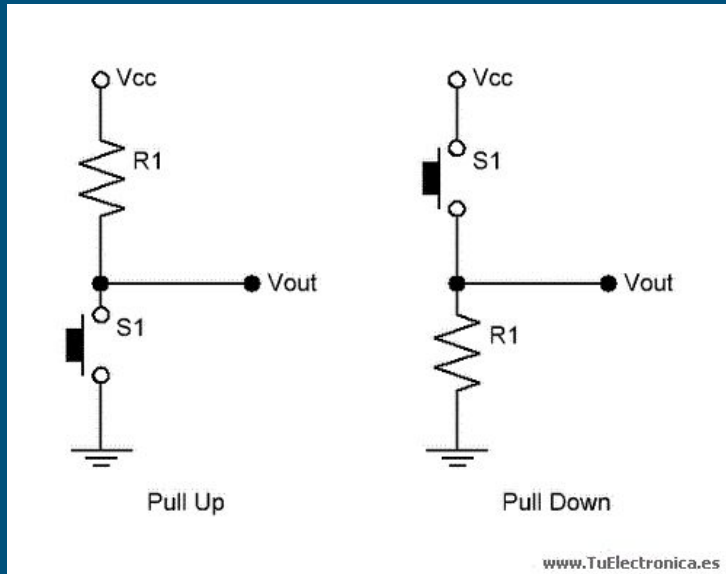




Melhores momentos do Projeto Botão

- Usamos a Arduino tando para receber quanto para transmitir sinais digitais
- Poderíamos ter ligado outra coisa que não um LED (um sensor por exemplo)

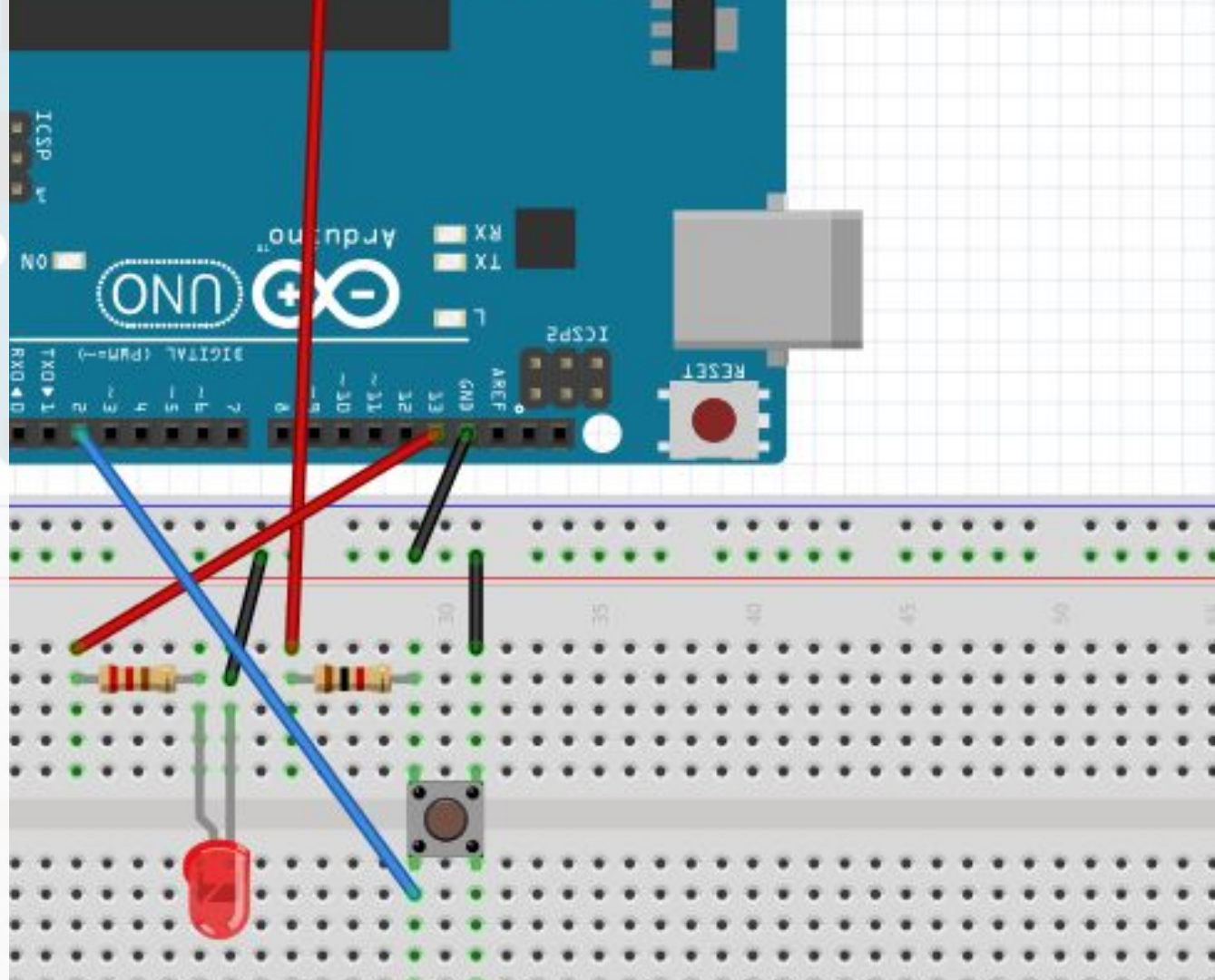
Pullup e pulldown



- Até agora usamos resistores pulldown
- As Arduinos têm resistores internos para uso como resistores pullup
- Acessados com `pinMode(<pino>, INPUT_PULLUP)` isso inverte o comportamento do pino de entrada
 - HIGH quando o sensor está desligado e LOW quando ligado

Projeto Botão (pullup)

- Hardware necessário
 - Placa Arduino
 - Protoboard
 - 1 LED
 - 2 Resistores de 220 ohms
 - Fios para conectar os componentes
 - Um botão



Atividades

- Mude o código para que o LED mude de estado (ligado/desligado) ao pressionar e soltar do botão
 - Botão liga e desliga
- Mude o projeto para que o LED ligue e permaneça ligado ao pressionar de um botão e desligue e permaneça desligado ao pressionar de outro botão.
 - Botão liga e (outro) botão desliga

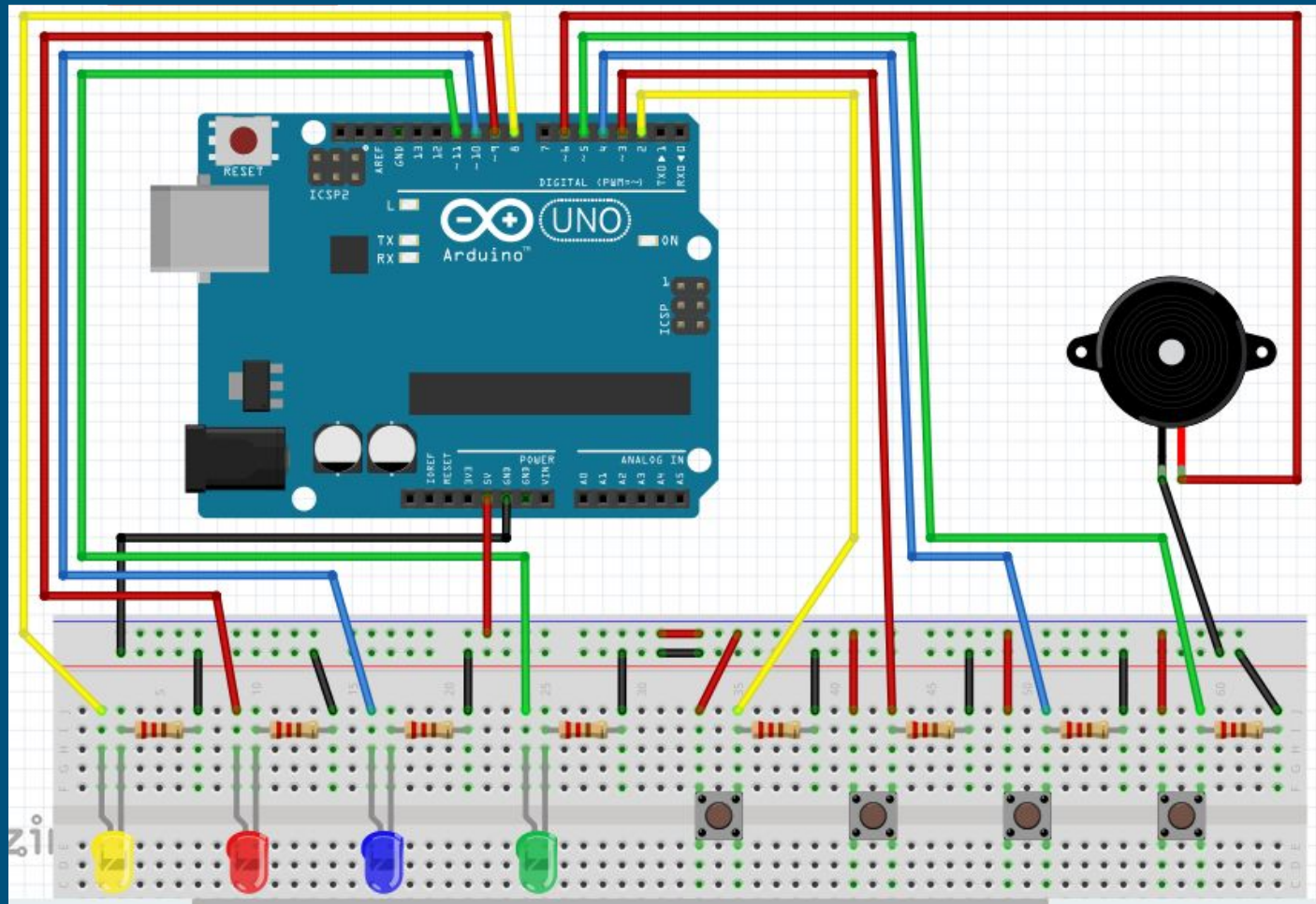
Atividades

- Estude o exemplo em <https://www.arduino.cc/en/Tutorial/InputPullupSerial>
- Faça a atividade anterior (botão liga e botão desliga) sem usar resistores
- Extra: estude o conteúdo em <https://www.arduino.cc/en/Tutorial/DigitalPins>

Projeto Genius (Jogo da memória)

- Hardware necessário
 - Placa Arduino
 - Protoboard
 - 4 LEDs
 - 4 Resistores de 220 ohms
 - 4 Resistores de 1000 ohms
 - 4 botões
 - 1 alto falante (speaker)
 - Fios para conectar os componentes



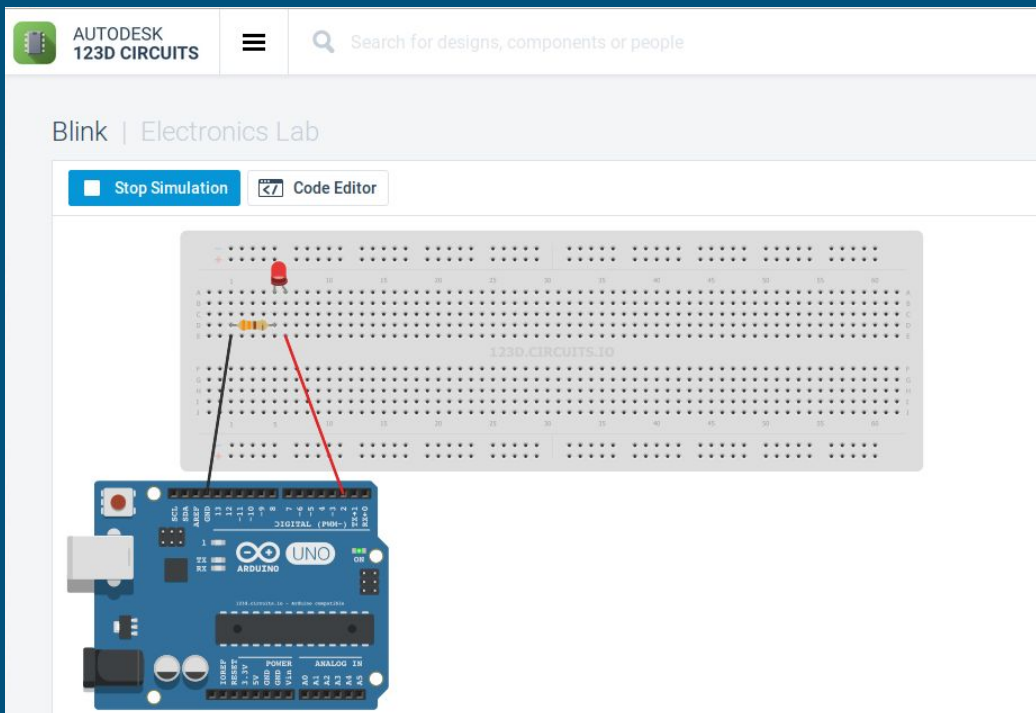


Código do projeto em
<https://goo.gl/0P3qGp>

The image shows an Arduino Uno R3 microcontroller board connected to a breadboard. The breadboard contains several electronic components: a yellow push-button, a green push-button, and four black push-buttons arranged in a row. Numerous jumper wires in various colors (red, blue, yellow, black, green) are used to connect the components to the Arduino's pins. The breadboard has a grid of holes with numbers 1 through 60 printed along the bottom edge. The text "Projeto Genius (Jogo da memória)" is overlaid in the center of the image.

Projeto Genius (Jogo da memória)

E se eu não tiver uma placa Arduino?



Se você não tiver uma placa Arduino visite

<https://123d.circuits.io/>

Grupo HardwareLivreUSP



Muito obrigado!

Curta nossa pagina no facebook:

www.facebook.com/Hardwarelivreusp

Participe do nosso grupo de e-mail:

hardwarelivreusp@googlegroups.com

Acesse nosso site:

<http://hardwarelivreusp.org>