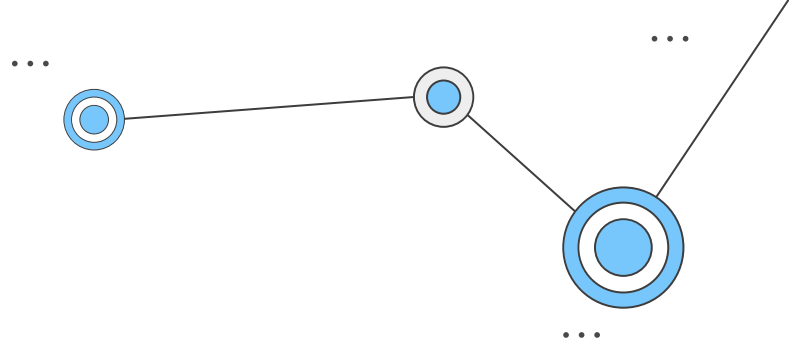




PUC Minas



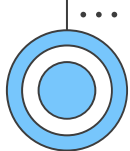
Laboratório de Iniciação à Programação

Prof. Dr. João Paulo Aramuni

Aula 12

Botão e Entrada Digital

LIP - Manhã



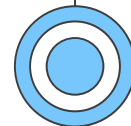
Novo componente: Botão táctil (Push Button)

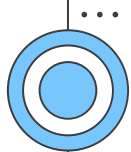
Um botão é um componente que simula um interruptor ou um botão físico.

Ele pode ser utilizado para controlar circuitos eletrônicos digitais.



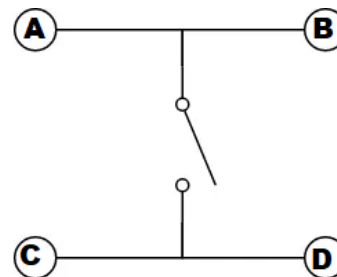
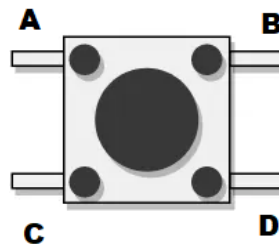
...



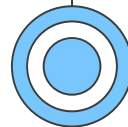


Novo componente: Botão táctil (Push Button)

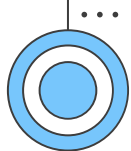
Observe que o Pino A é ligado “direto” com o Pino B, valendo a mesma estrutura entre o Pino C e o Pino D, ou seja, a ligação para o perfeito funcionamento do botão é em “X”, a ligação deve ser feita cruzada ligando o terminal A com o D ou o terminal C com o B, esse detalhe é muito importante, muitas pessoas se confundem e realizam a montagem do botão de forma errada no projeto.



...

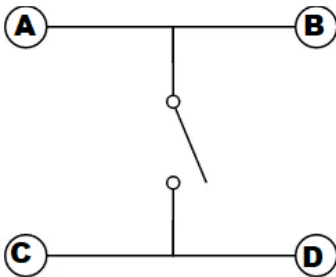
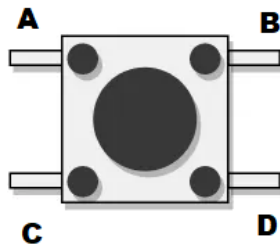


...

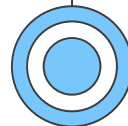


Novo componente: Botão táctil (Push Button)

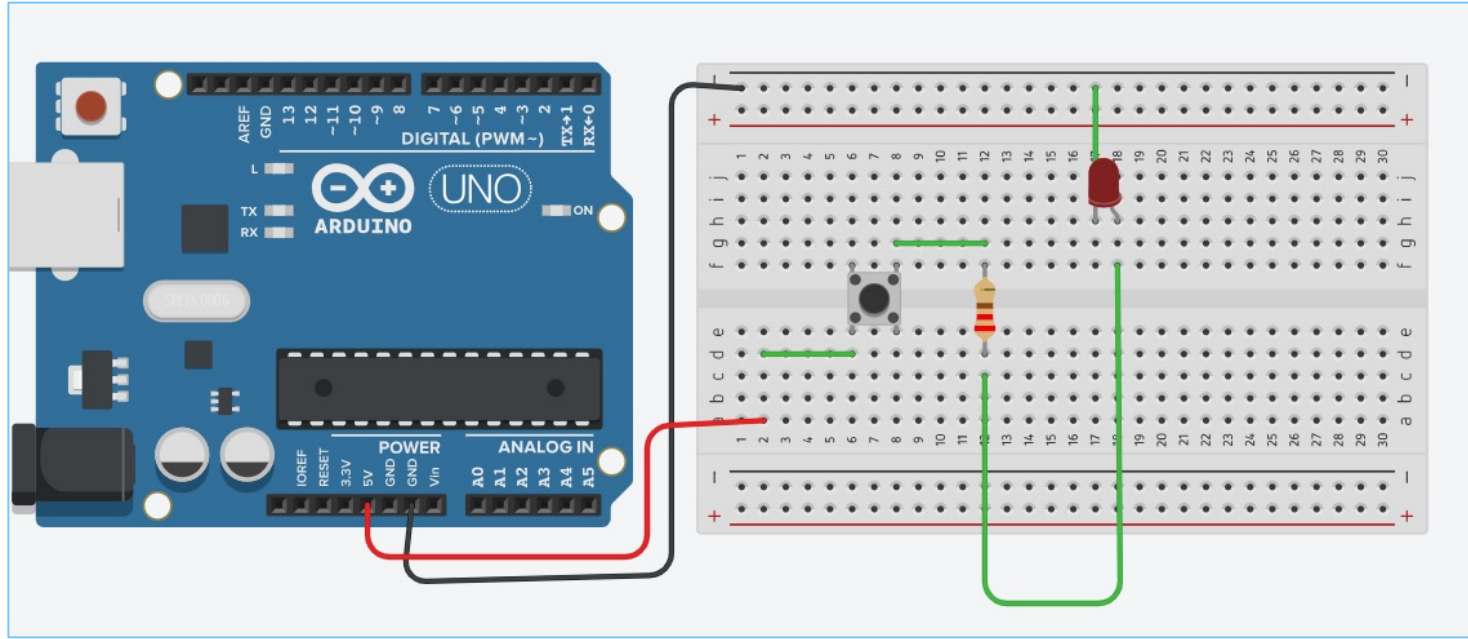
E pra que serve a conexão AB e CD? Simples, em alguns projetos você necessita fazer um "jumper" uma ponte entre uma área e outra da sua placa, essa conexão é uma opção para você.

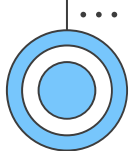


...



Montagem no Tinkercad





Entrada Digital

Os pinos de entrada digital em um Arduino são terminais físicos que podem ser usados para ler sinais digitais, ou seja, para detectar se um determinado nível de tensão está presente ou não.

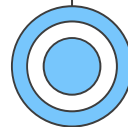
Cada pino de entrada digital pode estar em um de dois estados: HIGH (alto) ou LOW (baixo).



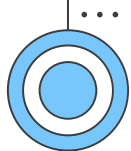
...



...



...



Entrada Digital

Um botão conectado a uma porta digital de entrada em um Arduino funciona basicamente como um interruptor simples.

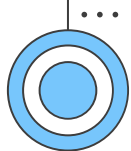
Quando o botão não está pressionado, a porta digital lê um estado LOW devido à presença de um resistor de pull-down que o conecta ao terra.



...



...



Entrada Digital

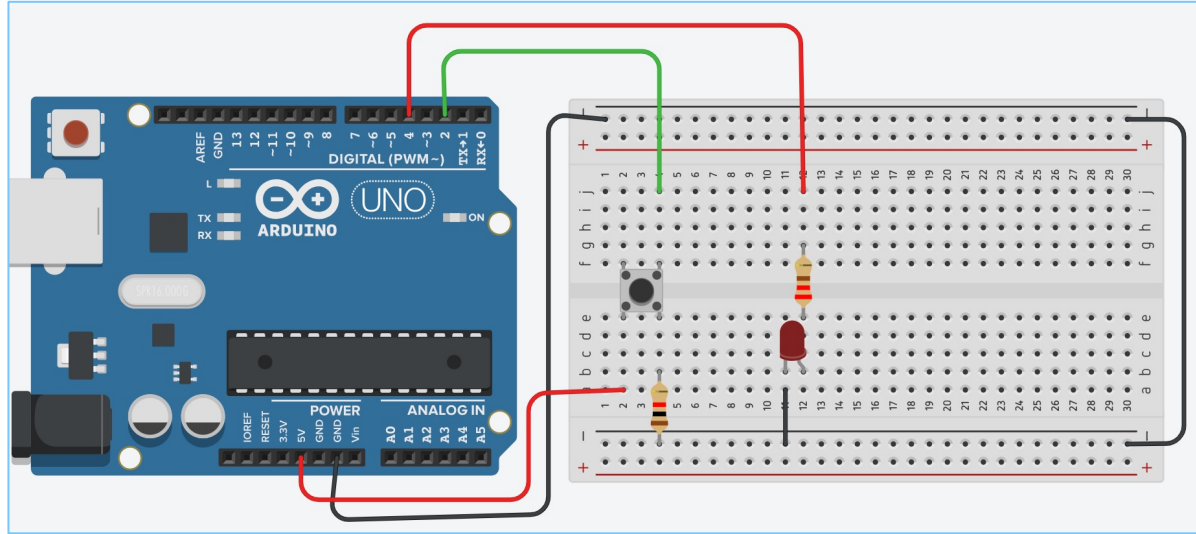
Ao pressionar o botão, a conexão entre a porta digital e o terra é fechada, fazendo com que a porta leia um estado HIGH.

O Arduino pode então detectar esse estado e executar ações específicas de acordo com o estado do botão, como acender um LED, iniciar um motor ou qualquer outra tarefa programada.



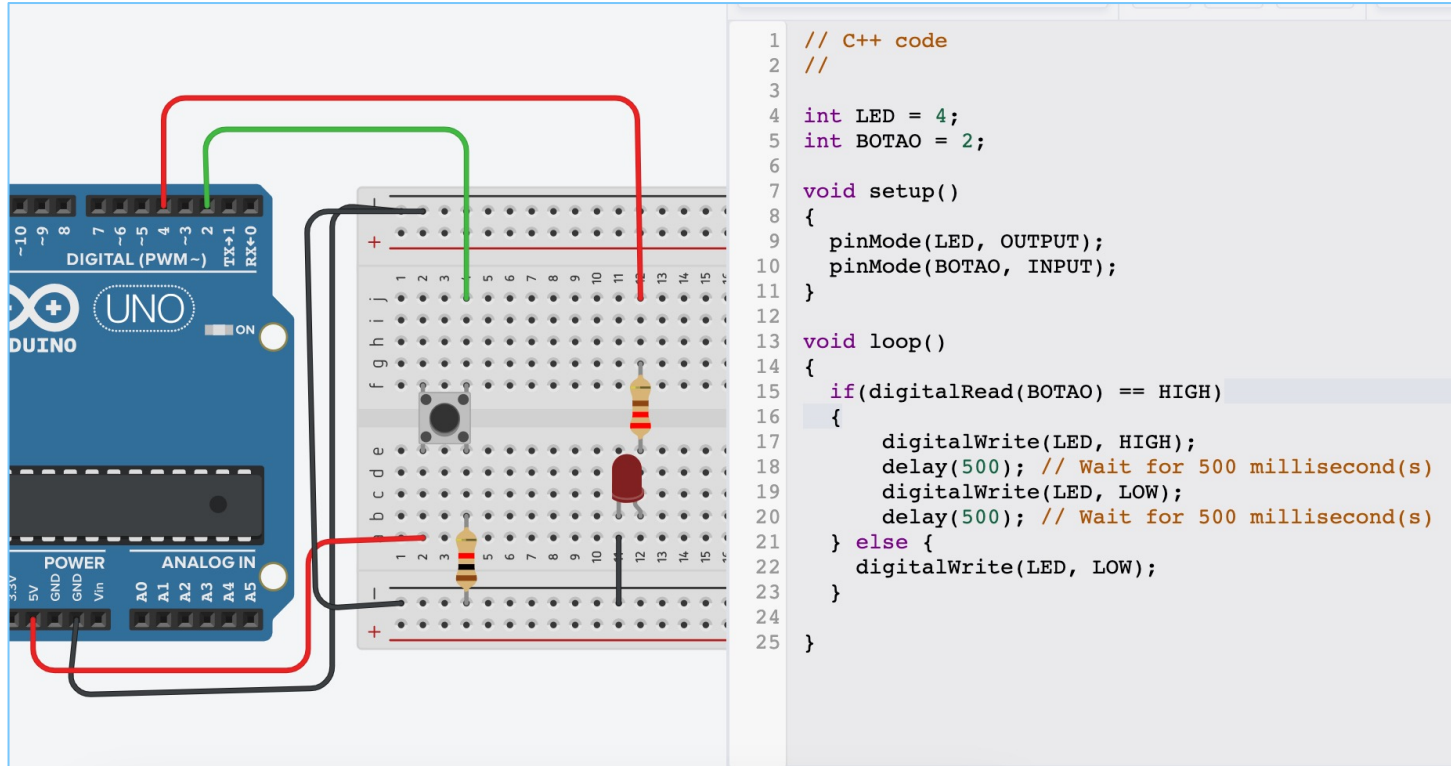
...

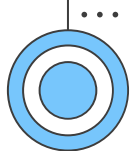
Exemplo Tinkercad – Entrada Digital



OBS: Conectar um dos terminais do botão ao terra é uma prática comum para garantir que o pino de entrada digital do Arduino permaneça em um estado conhecido (LOW) quando o botão não estiver pressionado. Isso evita que o pino fique "flutuante", o que poderia resultar em leituras imprecisas devido a interferências elétricas. A inclusão de um resistor tem a ver com a técnica chamada "pull-down resistor".. Esse resistor é colocado para garantir que, quando o botão não estiver pressionado, o pino do Arduino seja puxado para LOW de forma confiável.

Circuito para a montagem anterior





Exercício: Sistema de Controle de Luzes com Dependência

Objetivo: Criar um sistema de controle de luzes em uma maquete de casa onde a luz de um cômodo só pode ser ligada se as luzes dos outros cômodos estiverem apagadas.

Componentes:

3 botões (um para cada cômodo: sala, quarto e cozinha)

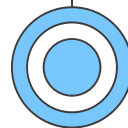
3 LEDs (um para cada cômodo)

6 resistores (um para cada botão e um para cada LED) Arduino

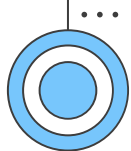
...



...



...



Exercício: Sistema de Controle de Luzes com Dependência

Instruções: Conecte um botão e um LED a cada cômodo da maquete (sala, quarto e cozinha).

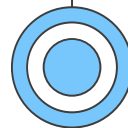
Conecte cada botão a uma entrada digital e cada LED a uma saída digital no Arduino, utilizando um resistor de pull-down para cada botão e um resistor limitador de corrente para cada LED.

Escreva um código que permita ligar as luzes dos cômodos apenas se as luzes dos outros cômodos estiverem apagadas.

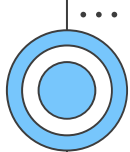
...



...



...



Resolução - pt1

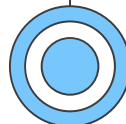
```
const int buttonSalaPin = 2; // Pino do botão da sala
const int buttonQuartoPin = 3; // Pino do botão do quarto
const int buttonCozinhaPin = 4; // Pino do botão da cozinha
const int ledSalaPin = 5; // Pino do LED da sala
const int ledQuartoPin = 6; // Pino do LED do quarto
const int ledCozinhaPin = 7; // Pino do LED da cozinha
```

```
bool salaLigada = false;
bool quartoLigado = false;
bool cozinhaLigada = false;
```

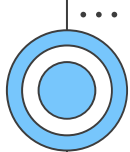
```
void setup() {
  pinMode(buttonSalaPin, INPUT);
  pinMode(buttonQuartoPin, INPUT);
  pinMode(buttonCozinhaPin, INPUT);
  pinMode(ledSalaPin, OUTPUT);
  pinMode(ledQuartoPin, OUTPUT);
  pinMode(ledCozinhaPin, OUTPUT);
}
```



...

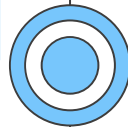


...



Resolução - pt2

```
void loop() {  
  // Controle da luz da sala  
  if (digitalRead(buttonSalaPin) == HIGH && !quartoLigado && !cozinhaLigada) {  
    salaLigada = !salaLigada;  
    digitalWrite(ledSalaPin, salaLigada);  
    delay(200); // Debounce  
  }  
  
  // Controle da luz do quarto  
  if (digitalRead(buttonQuartoPin) == HIGH && !salaLigada && !cozinhaLigada) {  
    quartoLigado = !quartoLigado;  
    digitalWrite(ledQuartoPin, quartoLigado);  
    delay(200); // Debounce  
  }  
  
  // Controle da luz da cozinha  
  if (digitalRead(buttonCozinhaPin) == HIGH && !salaLigada && !quartoLigado) {  
    cozinhaLigada = !cozinhaLigada;  
    digitalWrite(ledCozinhaPin, cozinhaLigada);  
    delay(200); // Debounce  
  }  
}
```



Obrigado!

Dúvidas?

joaopauloaramuni@gmail.com



[GitHub](#)



[LinkedIn](#)



[Lattes](#)

...