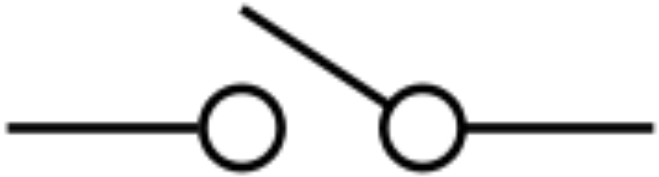
Entrada de dados com botões

Escola CDI

Professor: Paulo Marcos Trentin

Interruptor

Quando está aberto, nenhuma corrente passa por ele. Seu símbolo:



Ao pressioná-lo, toda corrente passa por ele, e esta pode ser medida facilmente.

Aberto seu estado lógico é 0, e fechado, 1

Circuito lógico

"É um circuito projetado para emitir uma saída de ligado ou desligado, representada pelos números binários 1 e 0, respectivamente"

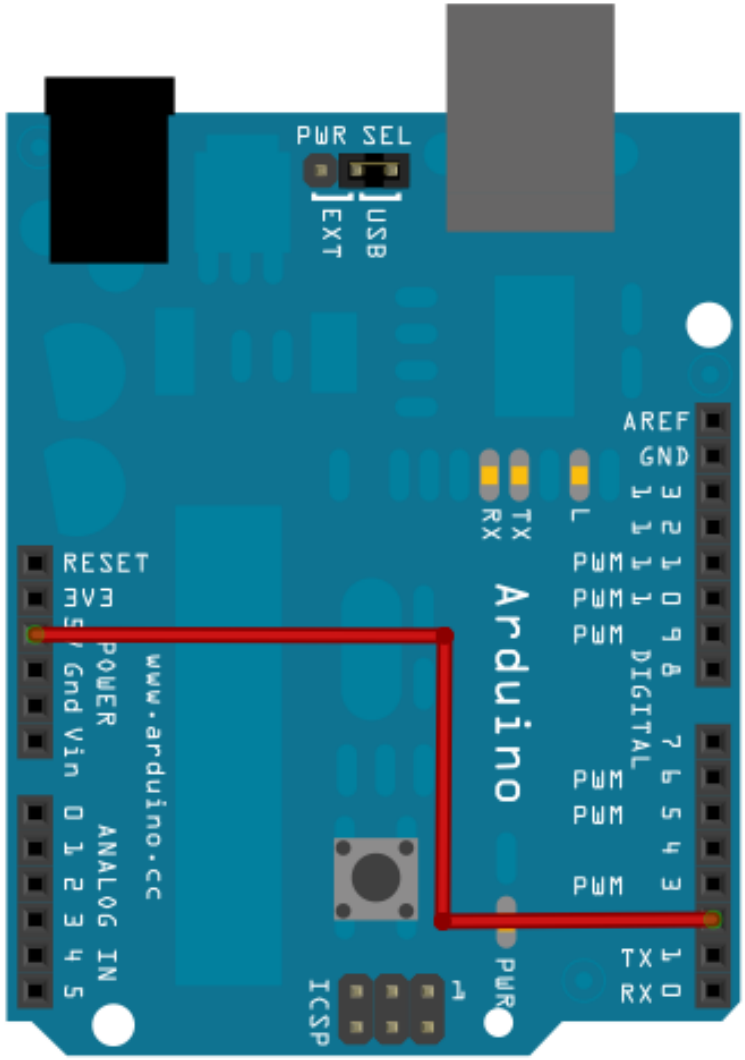
0 é representado por uma tensão próxima à 0 volts

1 é representado por uma tensão próxima do valor de alimentação do circuito (5 volts)

Fonte: Arduino Básico

Brincando com uma Entrada Digital

Faça a conexão no Arduino como visto ao lado, ou seja, conecte um fio do pino de 5 volts para o pino digital 2 do Arduino:



Brincando com uma Entrada Digital

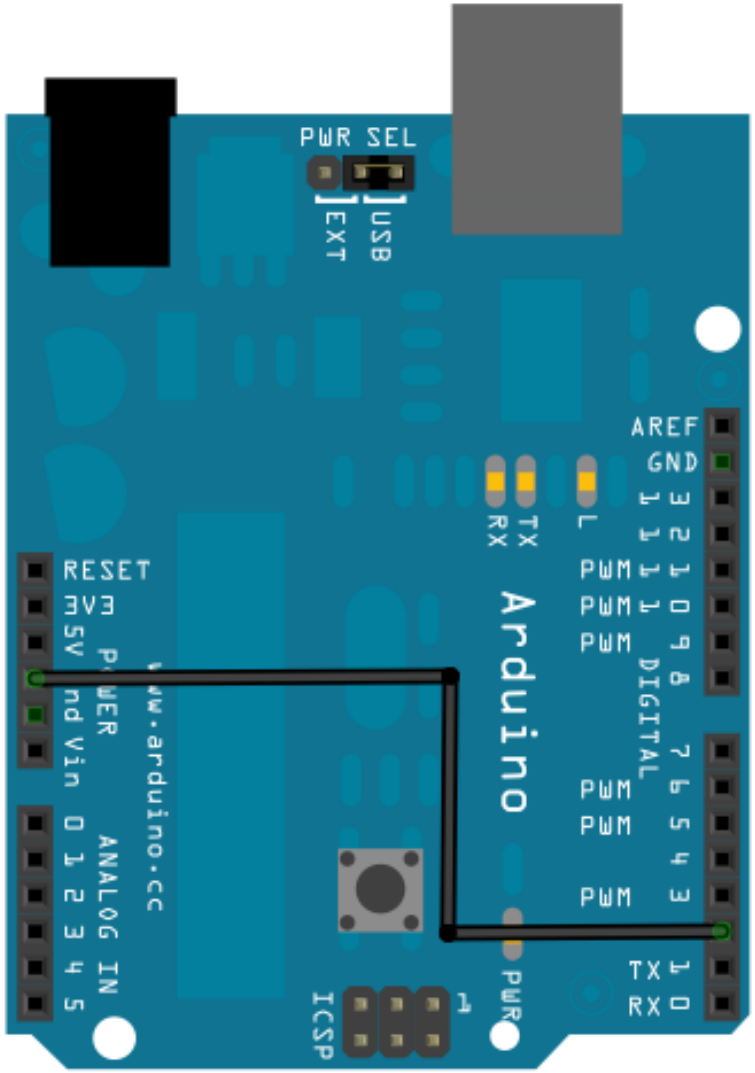
Desenvolva o seguinte código:

```
void setup(){
    Serial.begin(9600);
    pinMode(2, INPUT);
}
void loop(){
    int valor = digitalRead(2);
    Serial.print("Valor lido: ");
    Serial.println(valor);
    delay(100);
}
```

Repare a mudança do uso da função pinMode. Agora com o parâmetro INPUT

Brincando com uma Entrada Digital

Agora altere a conexão para o **GND**:



Brincando com uma Entrada Digital

Teste as saídas no monitor serial alternando o fio do pino digital 2 entre 5 volts e **GND** e respoda:

1. Qual o valor digital para a entrada de 5 e 0 volts?

2. O que acontece quando removemos o fio da entrada digital 2 e a deixamos desconectada?

Entrada digital

Tendo uma entrada digital onde 5 volts representa 1, e 0 volts representa 0 digital, você então deve GARANTIR que o sinal de entrada esteja próximo a esses valores.

Do contrário teremos um estado indefinido, também conhecido como *flutuante*, ou seja, não é 0 nem 1, o que irá comprometer o funcionamento esperado do software.

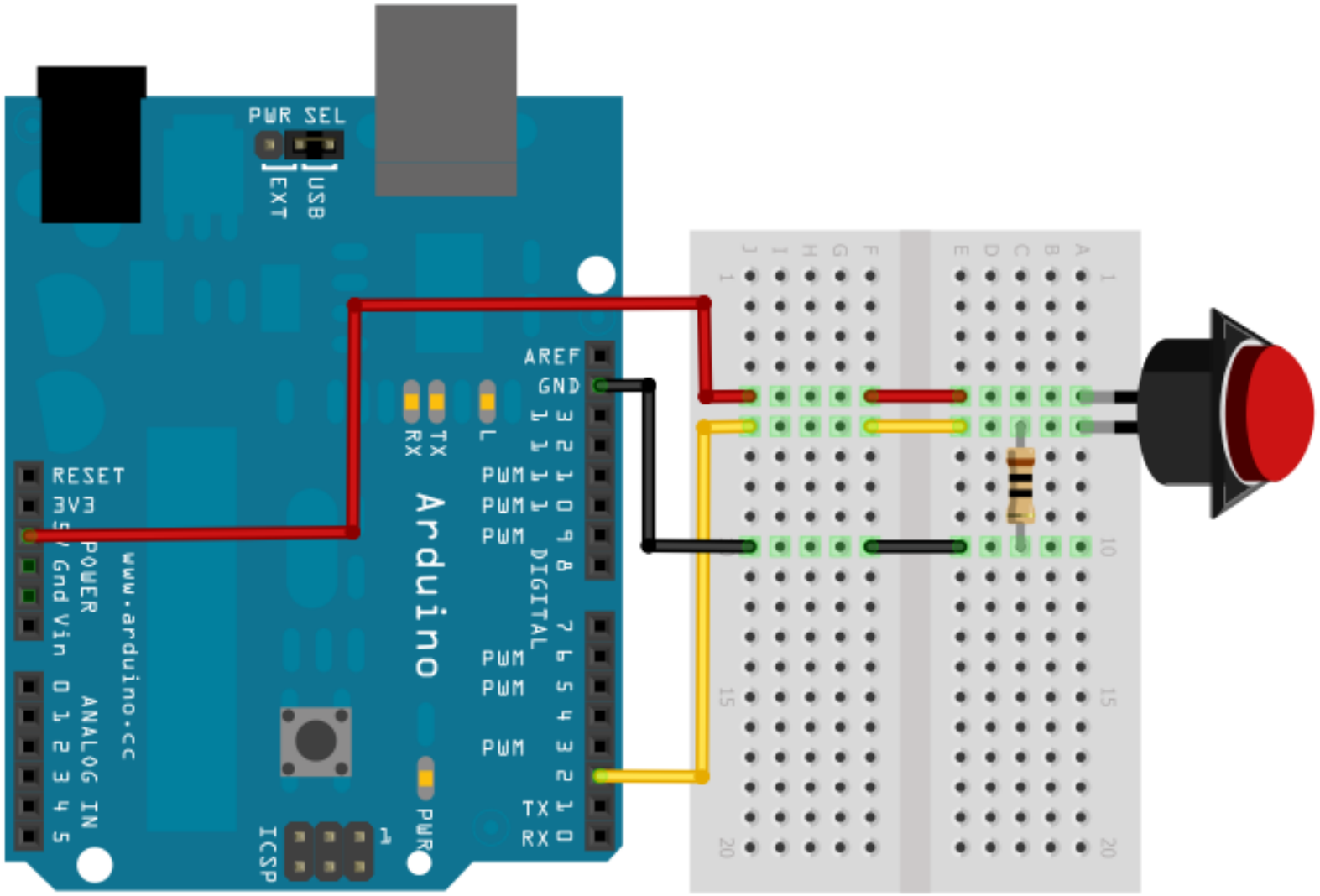
Resolvendo problema da entrada inesperada

Uma maneira de resolvermos o problema do estado indefinido, ou *flutuante*, é usarmos resistores:

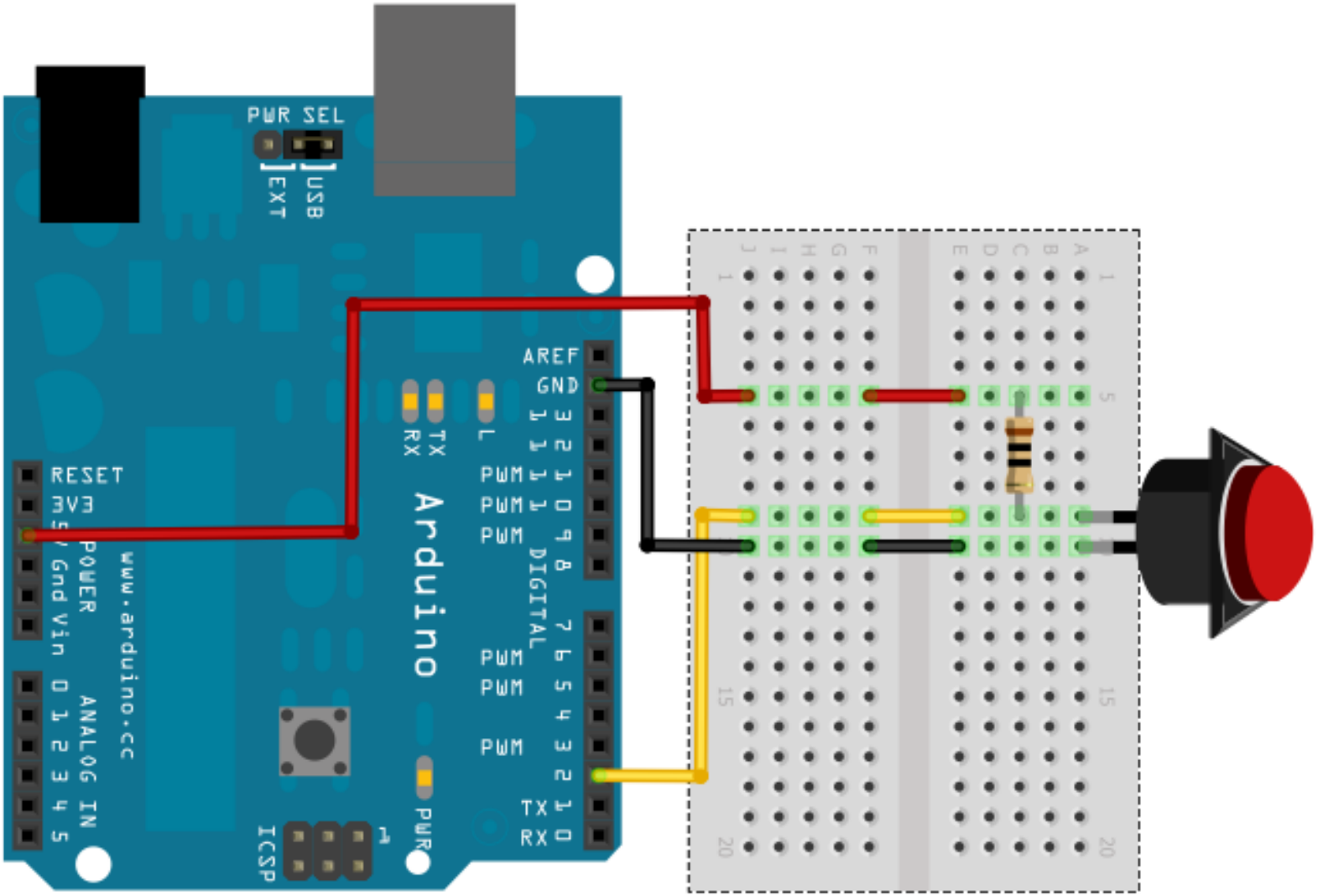
pull-down: Conectados ao **GND** (0 volts) mantém a entrada em nível lógico 0, ao estar desativada a chave

pull-up: Conectados ao VCC (5 volts), mantém a entrada em nível lógico 1, ao estar desativada a chave.

Esquema elétrico resistores pull-down



Esquema elétrico resistores pull-up



Perguntas

Qual deles é usado para garantir uma entrada alta em um determinado pino de um CI?

Qual a diferença entre um resistor pull-up e um pull-down?

Qual a configuração mais comum em um push button?

Resistor pull-up interno do Arduino

O Arduino possui um resistor interno do tipo pull-up, com valor de 20k.

Isso significa que não precisamos usar um componente externo extra para podermos usar um botão.

Porém, devemos observar que o botão usará a lógica inversa, ou seja, quando pressionado o status da porta digital será FALSE

Usando resistor pull-up do Arduino

Repare a única linha de código adicionada para fazer a ativação do resistor pull-up:

```
void setup(){
    Serial.begin(9600);
    pinMode(2, INPUT);
    digitalWrite(2, HIGH); // Ativa resistor pull-up interno
}
void loop(){
    int valor = digitalRead(2);
    Serial.print("Valor lido: ");
    Serial.println(valor);
    delay(100);
}
```

Exercícios

Separe 3 programas de efeitos diferentes para o cubo que você desenvolveu.

Faça agora um programa, que ao pressionar o botão, altere o efeito atual do cubo para o próximo efeito na sua lista.

Seu programa final deve ter ao menos 3 efeitos, alterados pelo botão externo conectado no Arduino.

Lógica Digital - Operador E lógico

O operador E é usado para verificar se todos os valores na expressão if são verdadeiros:

```
if (num == 1 && input == HIGH){

}
```

Lê-se: Se valor da variável num for igual à 1 e valor da variável input for verdadeiro, faça.

Lógica Digital - Operador OU lógico

O operador OU, é usado para testar qualquer uma dos valores na expressão. Se apenas um for satisfeito, então ela é executada:

```
if (num == 1 || input == HIGH){

}
```

Lê-se: Se valor da variável num for igual à 1 OU valor da variável input for verdadeiro, faça.

Lógica Digital - Operador ! (NÃO)

O operador !, é usado para negar um valor. Negar um valor é inverter seu estado. Se o valor for verdadeiro, e você negá-lo, irá virar falso:

```
if ( ! (digitalRead(2) ){
}
```

Lê-se: Se o valor lógico do pino 2 não for igual à 1, faça.

Lê-se também: Se o botão não estiver pressionado, faça.

Exercícios

Faça um programa que utilize os três operadores lógicos vistos anteriormente.

Use o botão e LEDs para a brincadeira.