OFICINA DE ROBÓTICA
INVENTAR E RECICLAR PARA EDUCAR
oficinaderobotica.ufsc.br

Oficina de Robótica

Programação Básica em Arduino – Aula 2

Execução:

UNIVERSIDADE FEDERAL
DE SANTA CATARINA

LARM
Laboratório de Automação
e Robótica Móvel

Site e Canal

- Site: http://oficinaderobotica.ufsc.br/

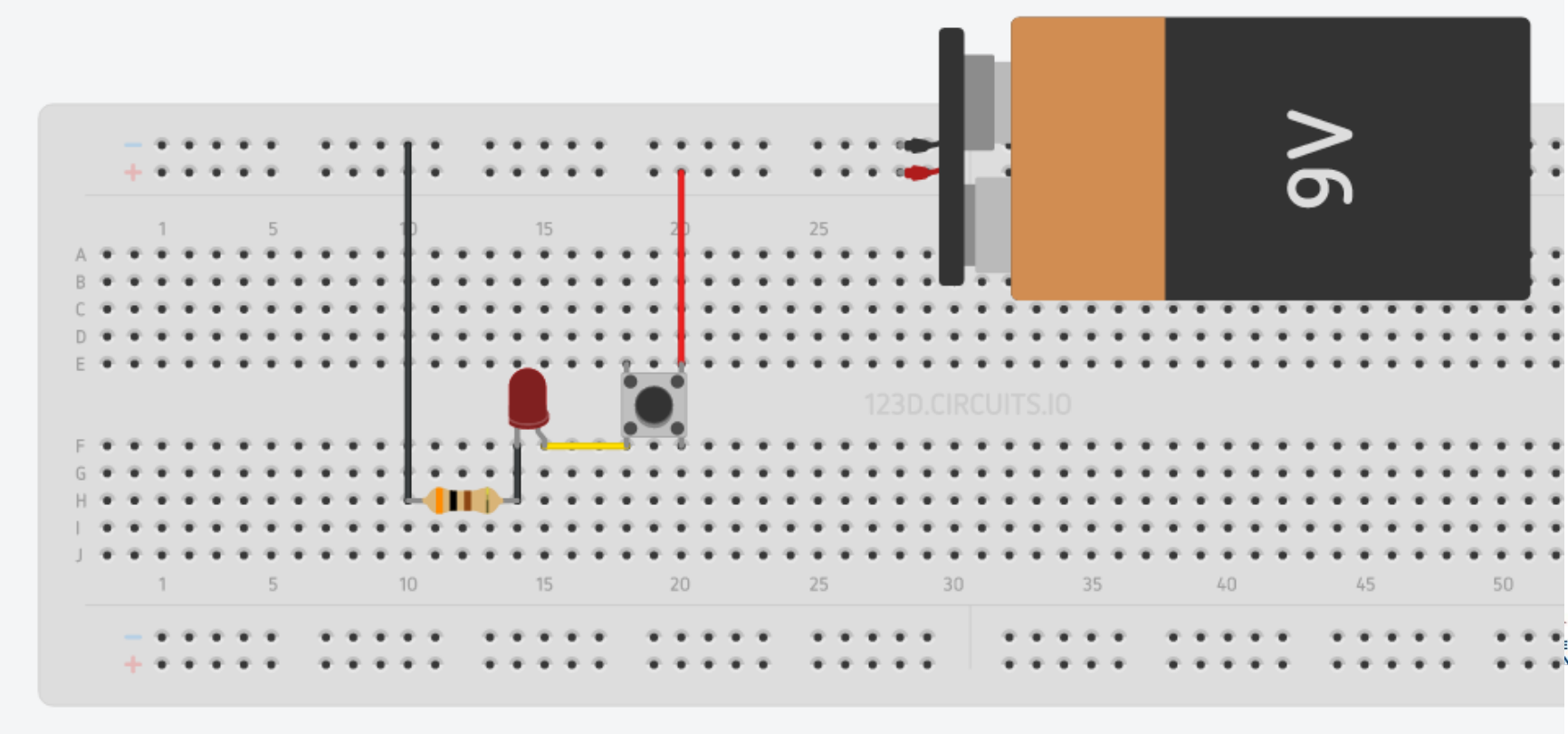
- Canal: Oficina de Robótica UFSC

https://www.youtube.com/channel/UC4OOjsP2FHfkdRnj0Wd7Iag

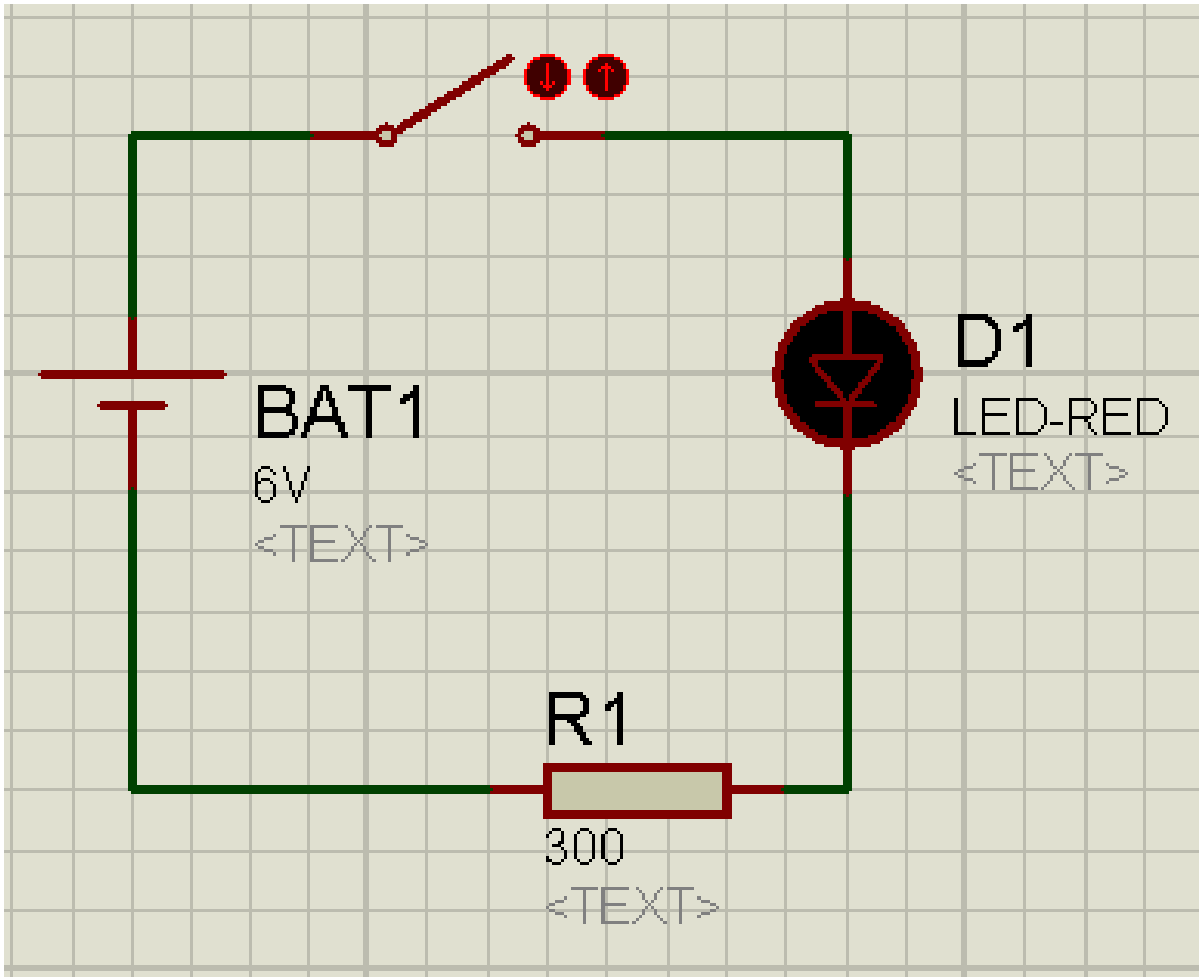
UNIVERSIDADE FEDERAL
DE SANTA CATARINA

LARM

Circuito



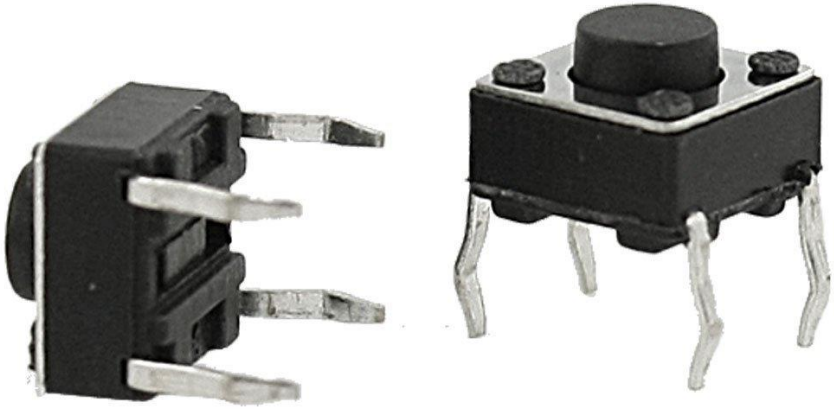
Esquemático



UNIVERSIDADE FEDERAL
DE SANTA CATARINA

LARM

Botão

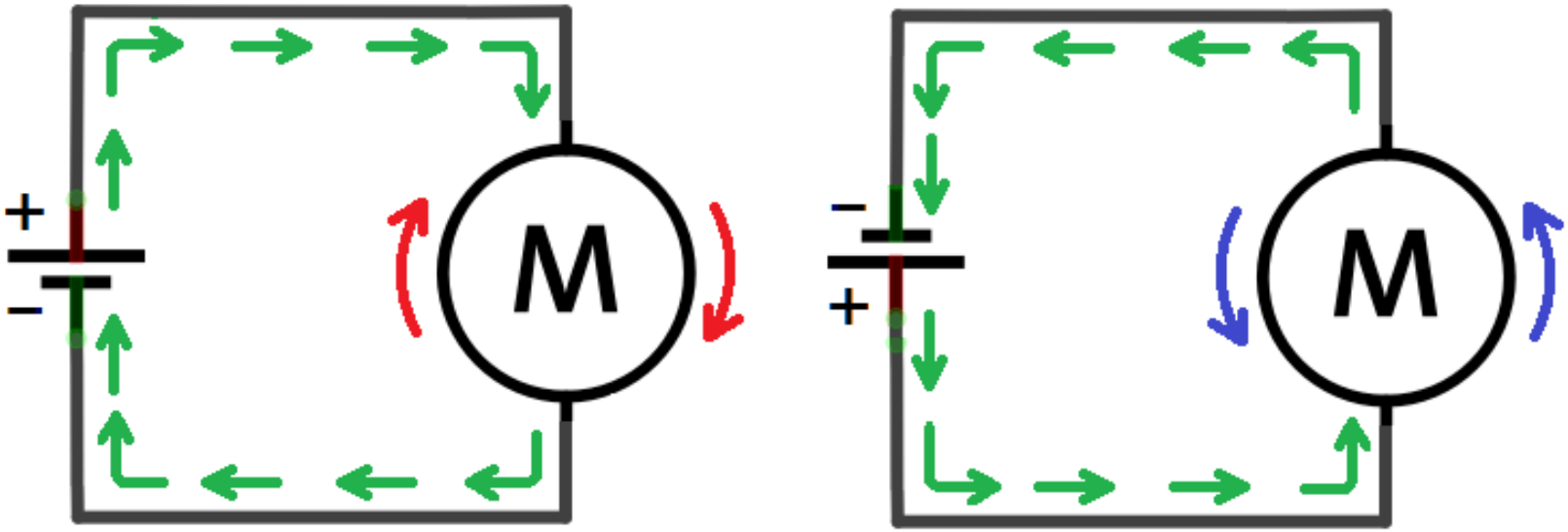


UNIVERSIDADE FEDERAL
DE SANTA CATARINA

LARM

Motor DC (Motor de Corrente Contínua)

- Inversão do sentido de giro:

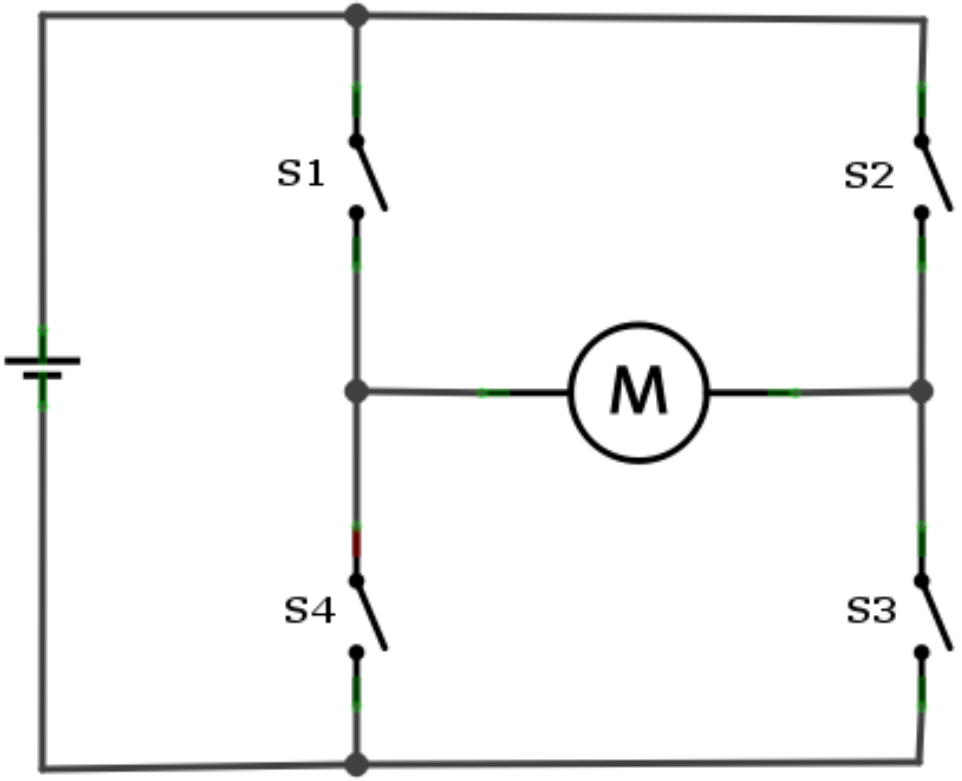


Ponte H

- A Ponte H é um circuito que permite a inversão do sentido de giro de um motor DC através da comutação de chaves eletrônicas.
- Pode ser implementada com chaves de contato, como push-buttons, ou transistores, que permitem o acionamento e inversão do sentido de giro de um motor através de sinais elétricos, sem a intervenção humana.

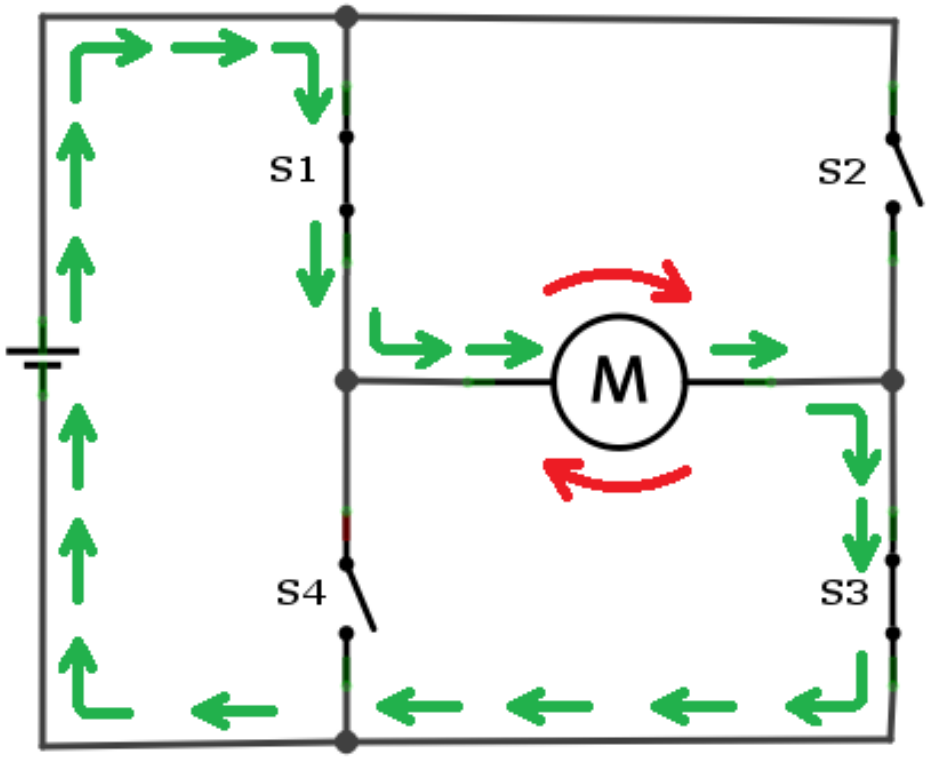
Ponte H

- Todas as chaves abertas – Motor parado.



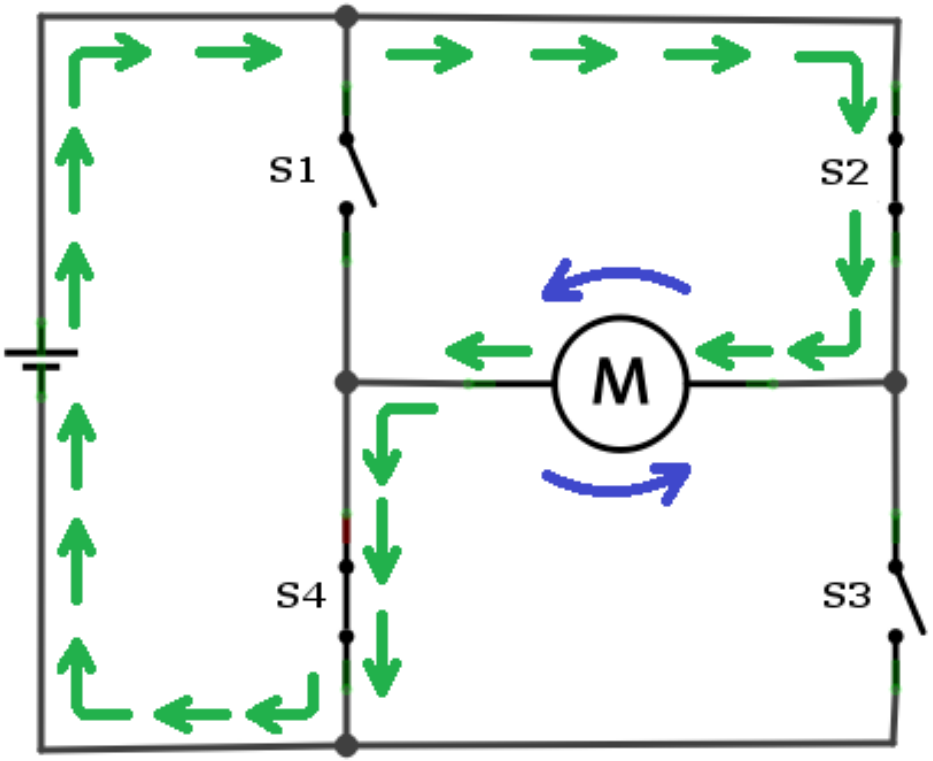
Ponte H

- S1-S3 fechadas e S2-S4 abertas - Rotor gira em um sentido.



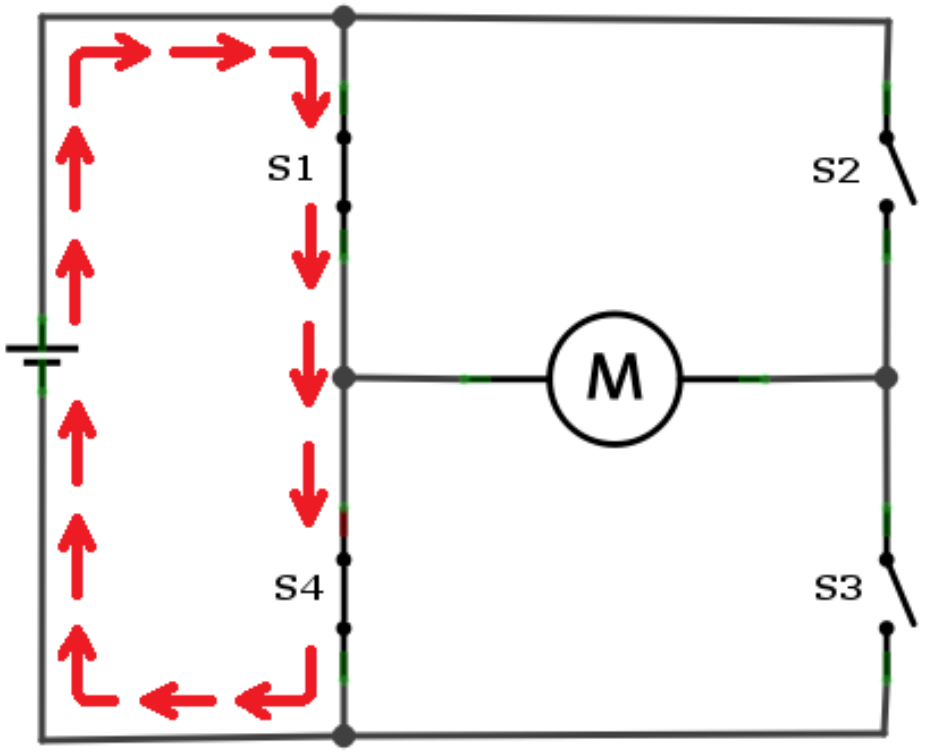
Ponte H

- S2-S4 fechadas e S1-S3 abertas - Rotor gira no sentido oposto ao anterior.



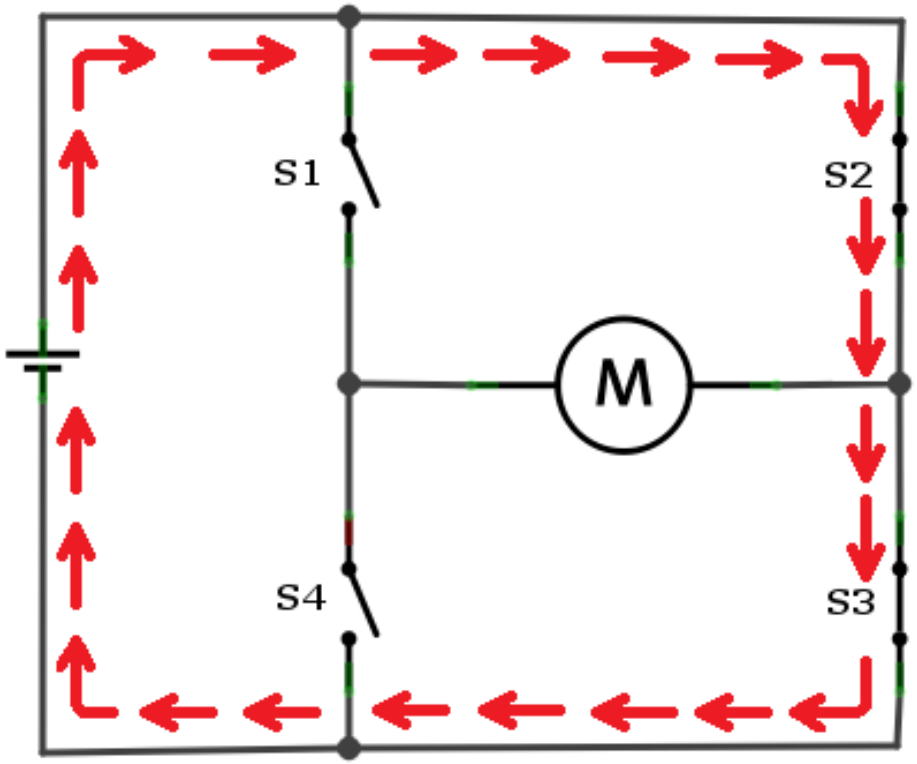
Ponte H

- S1-S4 fechadas – Essa configuração não deve ocorrer. *** CURTO CIRCUITO ***



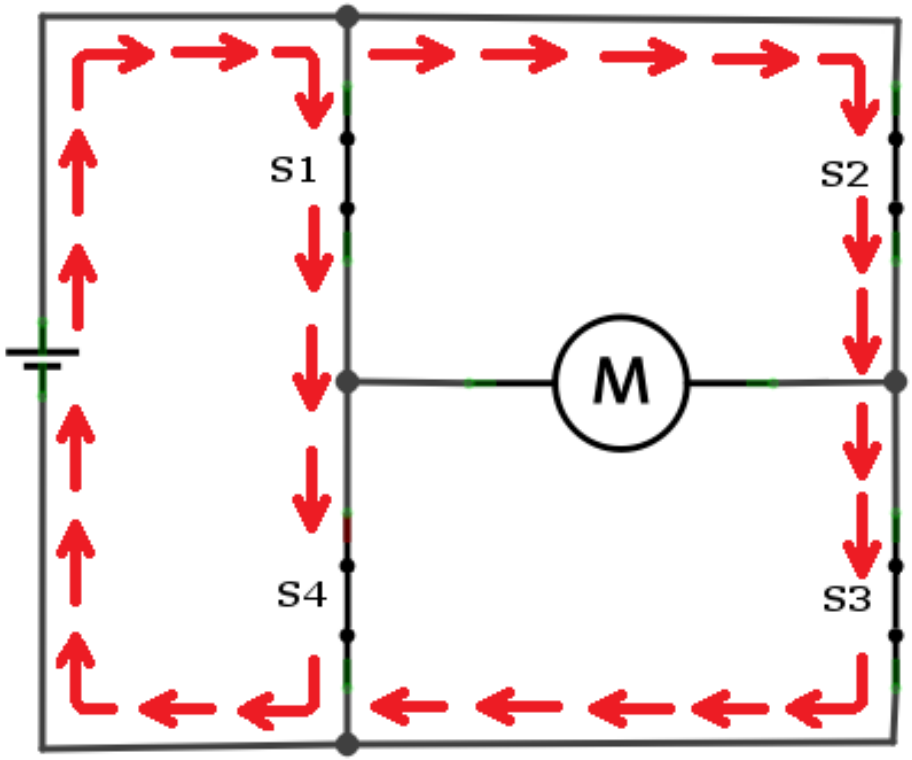
Ponte H

- S2-S3 fechadas – Também não deve ocorrer. *** CURTO CIRCUITO ***



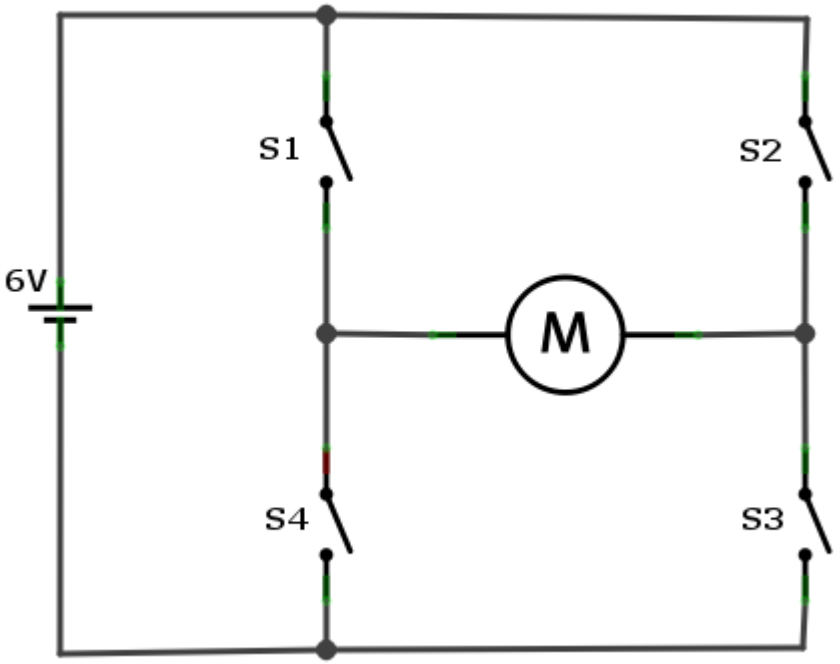
Ponte H

- Todas as chaves fechadas – Também não deve ocorrer. *** CURTO CIRCUITO ***



Exercício

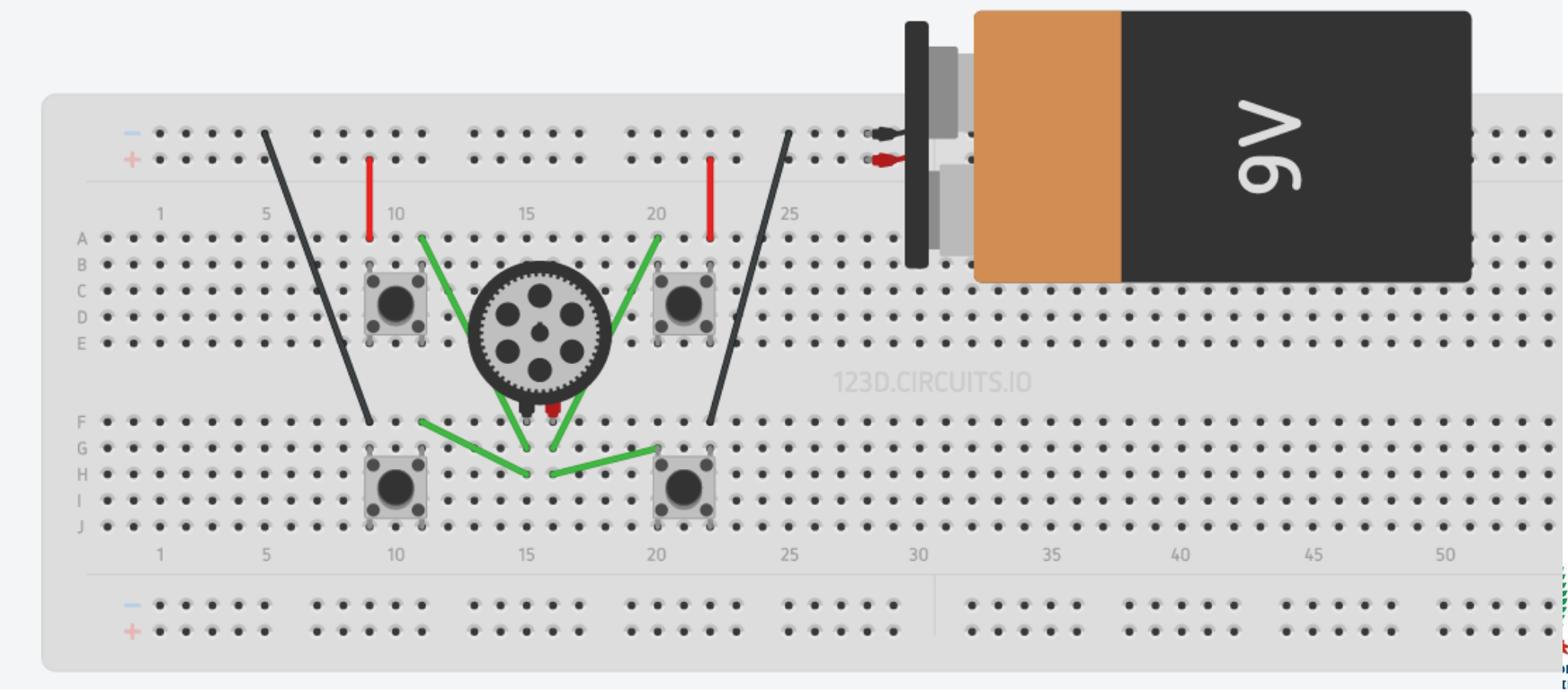
- Exemplo
  - Circuito de controle de sentido de giro de um motor DC com botões – circuito Ponte H



UNIVERSIDADE FEDERAL DE SANTA CATARINA

LARM

Exercício



Variáveis

- Variáveis são lugares (posições) na memória principal que servem para armazenar dados.
- As variáveis são acessadas através de um identificador único.
- O conteúdo de uma variável pode variar ao longo do tempo durante a execução de um programa.
- Uma variável só pode armazenar um valor a cada instante.
- Um identificador para uma variável é formado por um ou mais caracteres, obedecendo a seguinte regra: o primeiro caractere deve, obrigatoriamente, ser uma letra.

UNIVERSIDADE FEDERAL DE SANTA CATARINA

LARM

Variáveis

- ATENÇÃO!!!
  - **Um identificador de uma variável ou constante não pode ser formado por caracteres especiais ou palavras reservadas da linguagem.**

UNIVERSIDADE FEDERAL
DE SANTA CATARINA

LARM

Tipos de Dados

- Tipos de dados definem:
  - A quantidade de memória que uma variável ou constante irá ocupar;
  - As operações que podem ser executadas sobre uma variável ou constante de determinado tipo;
  - A faixa de valores que uma variável ou constante pode armazenar;
  - O modo como o valor armazenado será interpretado.

UNIVERSIDADE FEDERAL DE SANTA CATARINA

LARM

Tipos de Dados

- Tipos de Variáveis no Arduino

Tipo	Definição
void	Indica tipo indefinido. Usado geralmente para informar que uma função não retorna nenhum valor.
boolean	Os valores possíveis são true (1) e false (0). Ocupa um byte de memória.
char	Ocupa um byte de memória. Pode ser uma letra ou um número. A faixa de valores válidos é de -128 a 127.
unsigned char	O mesmo que o **char**, porém a faixa de valores válidos é de 0 a 255.
byte	Ocupa 8 bits de memória. A faixa de valores é de 0 a 255.
int	Armazena números inteiros e ocupa 16 bits de memória (2bytes). A faixa de valores é de -32.768 a 32.767.
unsigned int	O mesmo que o **int**, porém a faixa de valores válidos é de 0 a 65.535.
word	O mesmo que um **unsigned int**.

UNIVERSIDADE FEDERAL DE SANTA CATARINA

LARM

Tipos de Dados

- Tipos de Variáveis no Arduino

Tipo	Definição
long	Armazena números de até 32 bits (4 bytes). A faixa de valores é de -2.147.483.648 até 2.147.483.647.
unsigned long	O mesmo que o **long**, porém a faixa de valores é de 0 até 4.294.967.295.
short	Armazena número de até 16 bits (2 bytes). A faixa de valores é de -32.768 até 32.767.
float	Armazena valores de ponto flutuante (com vírgula) e ocupa 32 bits (4 bytes) de memória. A faixa de valores é de -3.4028235E+38 até 3.4028235E+38
double	O mesmo que o **float**.

UNIVERSIDADE FEDERAL DE SANTA CATARINA

LARM

Exemplo

```
int led = 13;

void setup() {

  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

UNIVERSIDADE FEDERAL DE SANTA CATARINA

LARM

Operadores

- Em uma linguagem de programação existem vários operadores que permitem operações do tipo:
  - Aritmética
  - Relacional
  - Lógica
  - Composta

UNIVERSIDADE FEDERAL
DE SANTA CATARINA

LARM

Operadores

- Operadores aritméticos

Símbolo	Função
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo (resto da divisão inteira)

UNIVERSIDADE FEDERAL
DE SANTA CATARINA

LARM

Operadores

- Operadores relacionais

Símbolo	Função
>	Maior
<	Menor
>=	Maior ou igual
<=	Menor ou igual
==	Igual
!=	Diferente

UNIVERSIDADE FEDERAL
DE SANTA CATARINA

LARM

Operadores

- Operadores lógicos

Símbolo	Função
&&	E (and)
||	OU (or)
!	Não (not)

UNIVERSIDADE FEDERAL
DE SANTA CATARINA

LARM

Operadores

- Operadores compostos

Símbolo	Função
++	Incremento
--	Decremento
+=	Adição com atribuição
-=	Subtração com atribuição
*=	Multiplicação com atribuição
/=	Divisão com atribuição

UNIVERSIDADE FEDERAL DE SANTA CATARINA

LARM

Operadores

- Operador de Atribuição
  - A atribuição de valores a variáveis e constantes é feita com o uso do operador de atribuição (=).
  - O operador de atribuição coloca o valor situado à sua direita dentro do objeto localizado à sua esquerda.
  - Exemplos:
    - int valor = 100;
    - const float pi = 3.14;
  - **Atenção!!!**
    - O operador de atribuição não vale para o comando ***#define***.

UNIVERSIDADE FEDERAL DE SANTA CATARINA

LARM

Comandos de Seleção

- Em vários momentos em um programa precisamos verificar uma determinada condição afim de selecionar uma ação ou ações que serão executadas.
- Um comando de seleção também é conhecido por desvio condicional, ou seja, dada um condição, uma parte do programa é executada.
- Os comandos de seleção podem ser do tipo:
  - Seleção simples
  - Seleção composta
  - Seleção de múltipla escolha

UNIVERSIDADE FEDERAL DE SANTA CATARINA

LARM

Comandos de Seleção

- Seleção simples
  - Um comando de seleção simples avalia uma condição, ou expressão, para executar uma ação ou conjunto de ações.
  - No Arduino o comando de seleção simples é:

```
if (expr) {
   cmd
}
```

  - onde:
    - ***expr*** – representa uma expressão a ser avaliada que poc ser do tipo lógica, relacional ou aritmética. O resultado avaliação de uma expressão é sempre um valor lógico.
    - ***cmd*** – comando(s) a ser executado.

UNIVERSIDADE FEDERAL DE SANTA CATARINA

LARM

Comandos de Seleção

- Seleção composta
  - Um comando de seleção composta é complementar ao comando de seleção simples.
  - O objetivo é executar um comando mesmo que a expressão avaliada pelo comando *if (expr)* retorne um valor falso.
  - No Arduino o comando de seleção composta é:

```
if (expr) {
    cmd;
}
else {
    cmd;
}
```

  - onde:
    - ***expr*** – representa uma expressão a ser avaliada que pode ser do tipo lógica, relacional ou aritmética. O resultado da avaliação de uma expressão é sempre um valor lógico.
    - ***cmd*** – comando(s) a ser executado.

LARM

Comandos de Seleção

- Para a entrada de dados é utilizado:

```
pinMode(pino, INPUT);
```

- E para a leitura é utilizado:

```
digitalRead(pino);
```

Onde essa função irá retornar um valor de 0 ou 1 que pode ser testado diretamente no if ou então salvo em uma variável e depois testada variável.

UNIVERSIDADE FEDERAL DE SANTA CATARINA

LARM

Exemplo – Opção 1

```
int botao = 12;
int led = 13;
int opcao = 0;
void setup()
{
  pinMode(led, OUTPUT);
  pinMode(botao, INPUT);
}

void loop()
{
  opcao = digitalRead(botao);
  if(opcao == 1)
  {
    digitalWrite(led, HIGH);
    delay(200);
  }
  else
  {
    digitalWrite(led, LOW);
    delay(200);
  }
}
```

UNIVERSIDADE FEDERAL DE SANTA CATARINA

LARM

Exemplo – Opção 2

```
int botao = 12;
int led = 13;

void setup()
{
  pinMode(led, OUTPUT);
  pinMode(botao, INPUT);
}

void loop()
{
  if(digitalRead(botao) == 1)
  {
    digitalWrite(led, HIGH);
    delay(200);
  }
  else
  {
    digitalWrite(led, LOW);
    delay(200);
  }
}
```

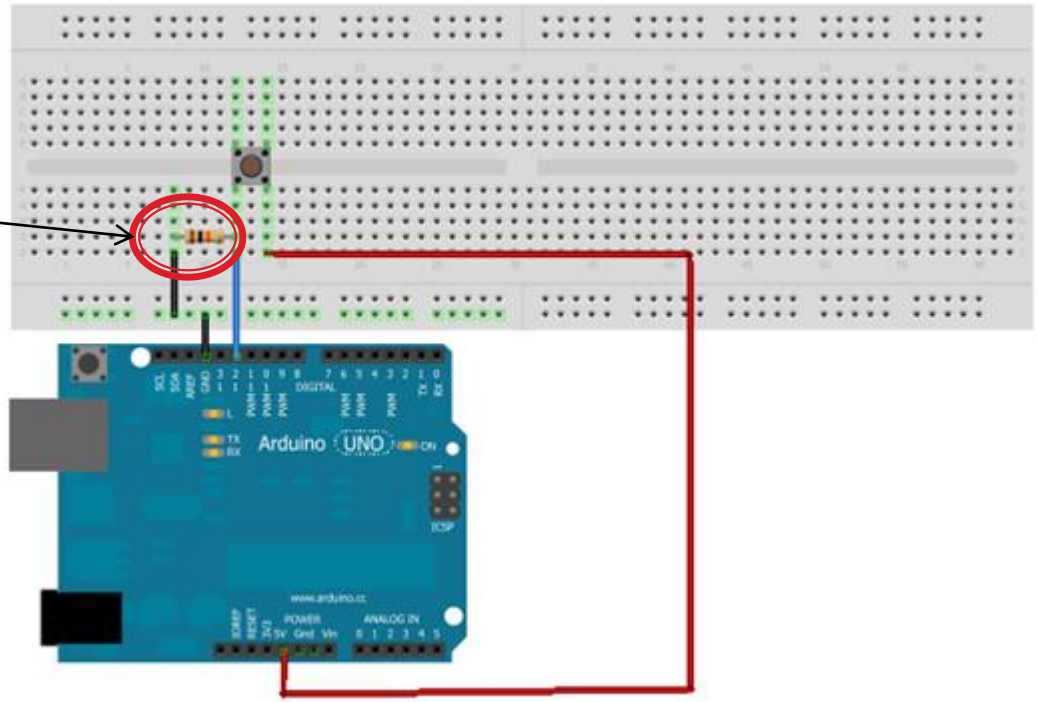
UNIVERSIDADE FEDERAL DE SANTA CATARINA

LARM

Entrada Digital de Dados

- Leitura de um botão com resistor *pull-down*
  - Ligação na protoboard

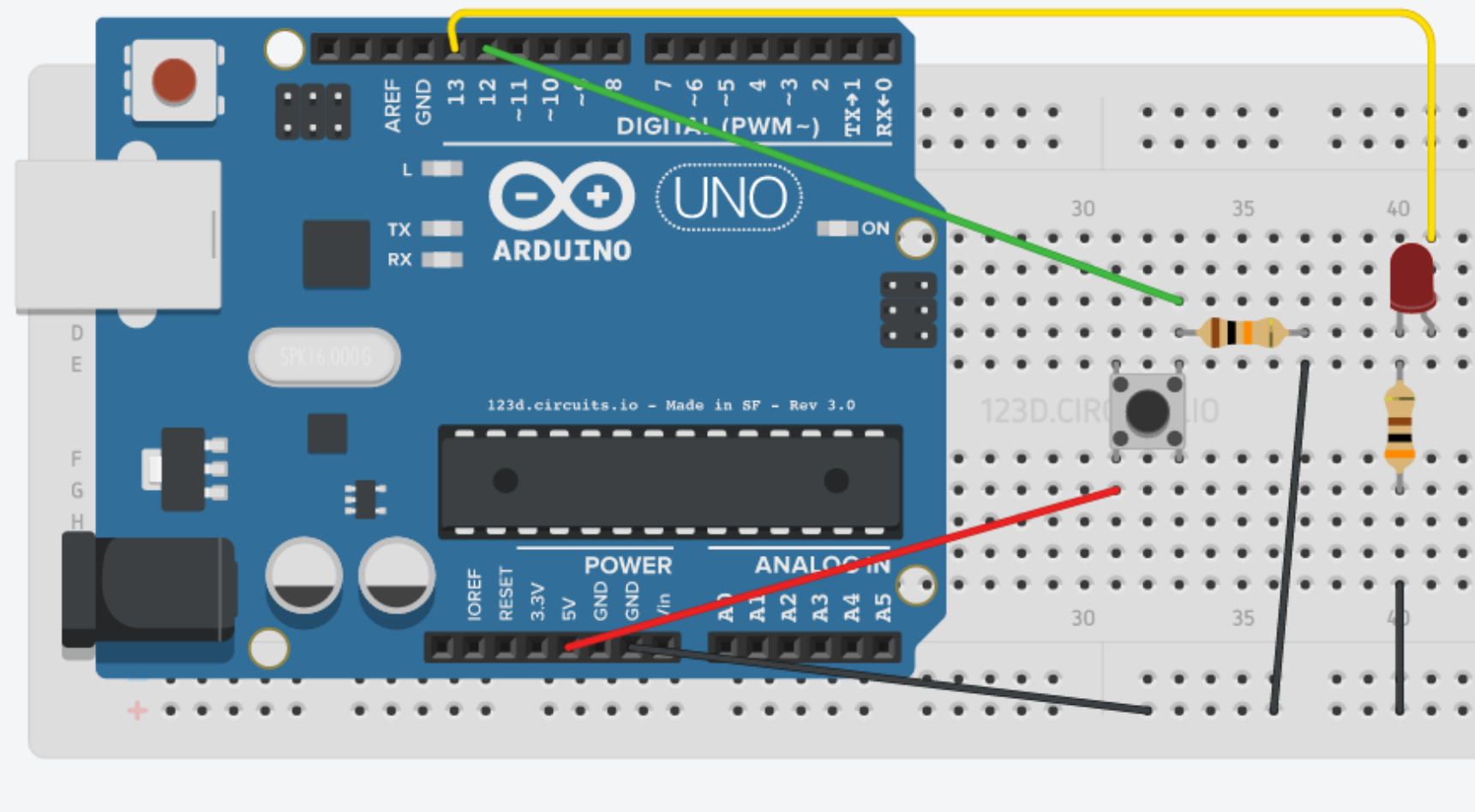
Resistor pull-down 10kΩ



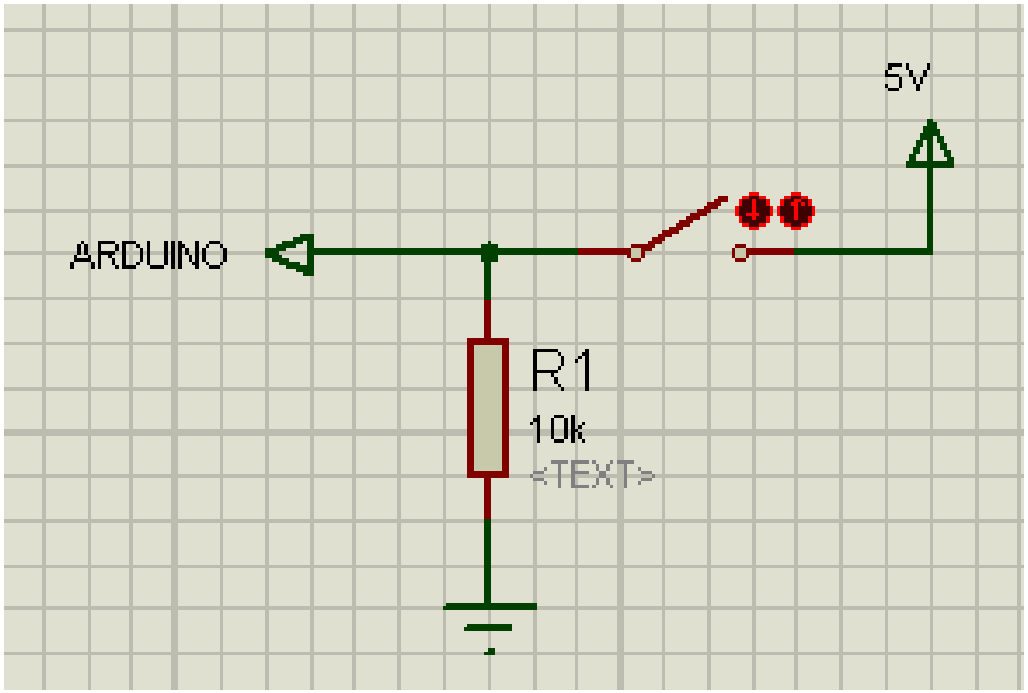
UNIVERSIDADE FEDERAL DE SANTA CATARINA

LARM

Exercício


AREF
GND
13
12
~11
~10
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
123d.circuits.io - Made in SF - Rev 3.0
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
30
35
40

Esquemático Pulldown



UNIVERSIDADE FEDERAL
DE SANTA CATARINA

LARM

Entrada Digital de Dados

- **Nota**
  - O Arduino possui resistores *pull-up* nas portas digitais.
  - Para ativar os resistores *pull-up* de uma porta digital basta defini-la como entrada e colocá-la em nível alto (HIGH).
    - pinMode(*pin, INPUT*)
    - digitalWrite(*pin, HIGH*)
  - Para desativar os resistores *pull-up* de uma por digital basta colocá-la em nível baixo.
    - digitalWrite(*pin, LOW*)

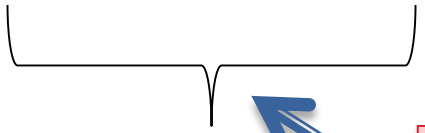
UNIVERSIDADE FEDERAL DE SANTA CATARINA

LARM

Entrada Digital de Dados

- **Nota**
  - O Arduino possui uma constante chamada *INPUT_PULLUP* que define que a porta será de entrada e o resistor *pull-up* da mesma será ativado.
  - Exemplo:

```
void setup()
{
  pinMode(10, INPUT_PULLUP);
}
```

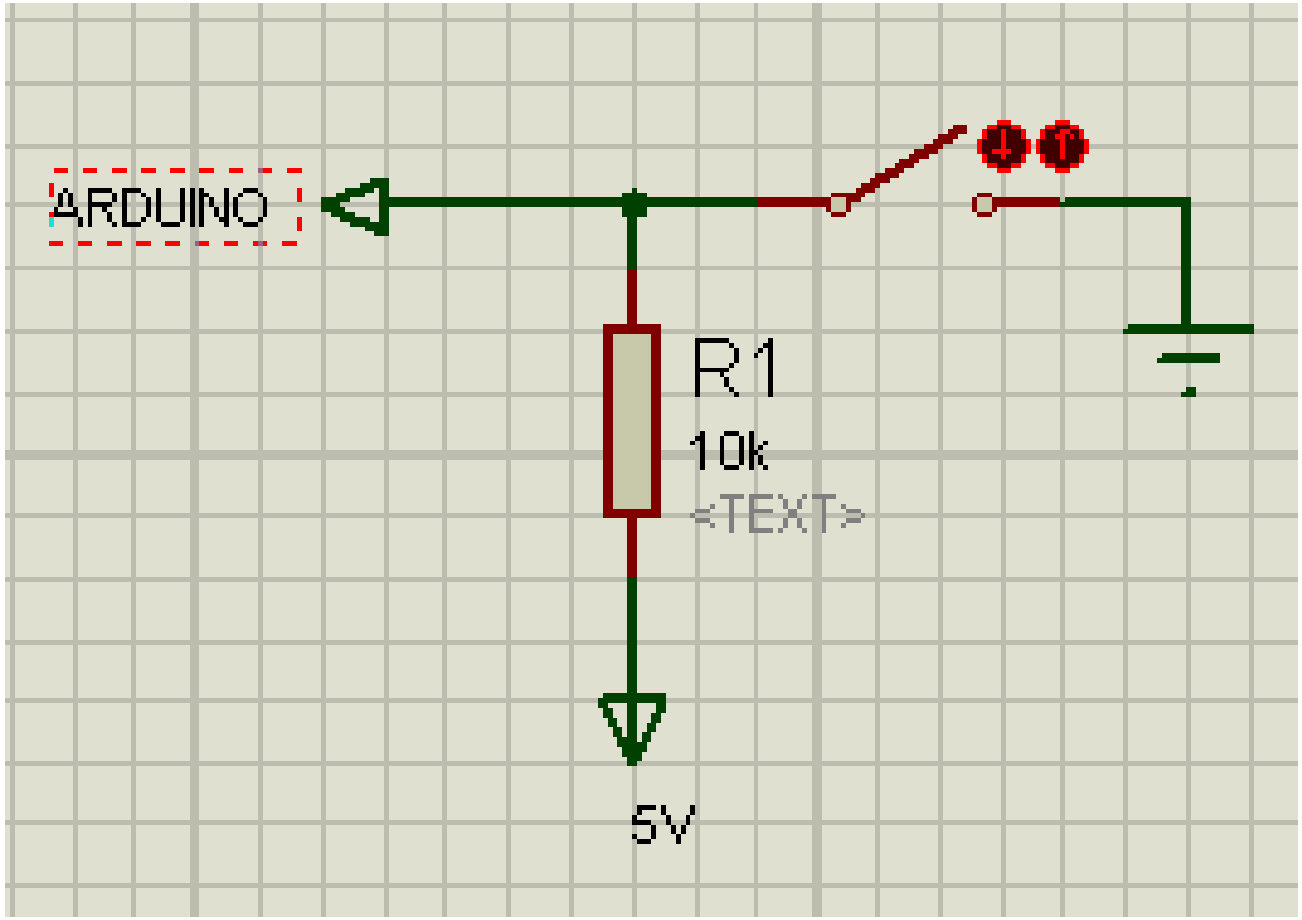


Define a porta 10 como entrada de dados e ativa o resistor pull-up.

UNIVERSIDADE FEDERAL DE SANTA CATARINA

LARM

Esquemático PULLUP

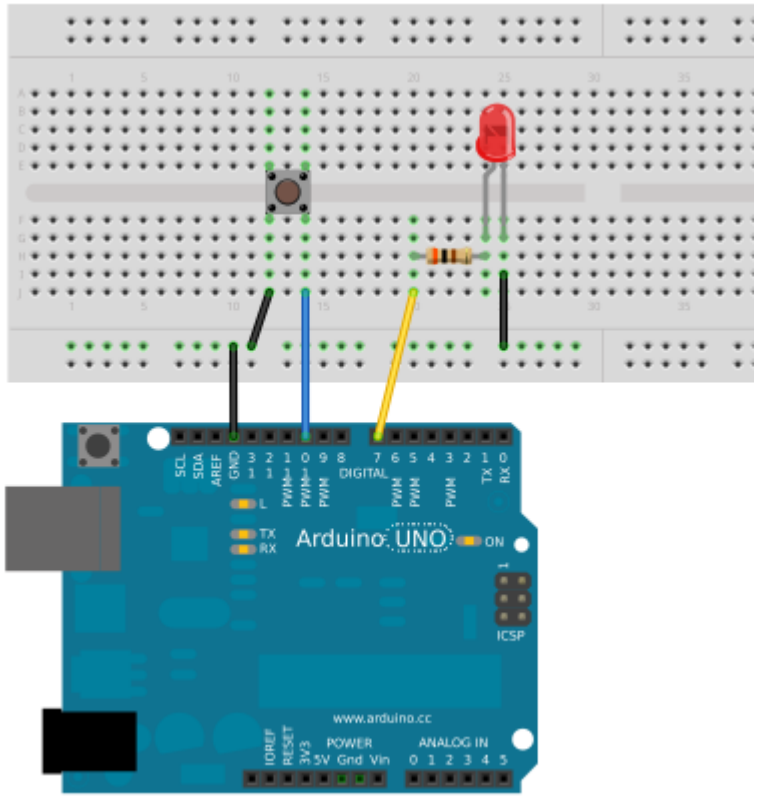


UNIVERSIDADE FEDERAL
DE SANTA CATARINA

LARM

Entrada Digital de Dados

- Ativando o *pull-up* de uma porta digital
  - Ligação na protoboard



UNIVERSIDADE FEDERAL DE SANTA CATARINA

LARM

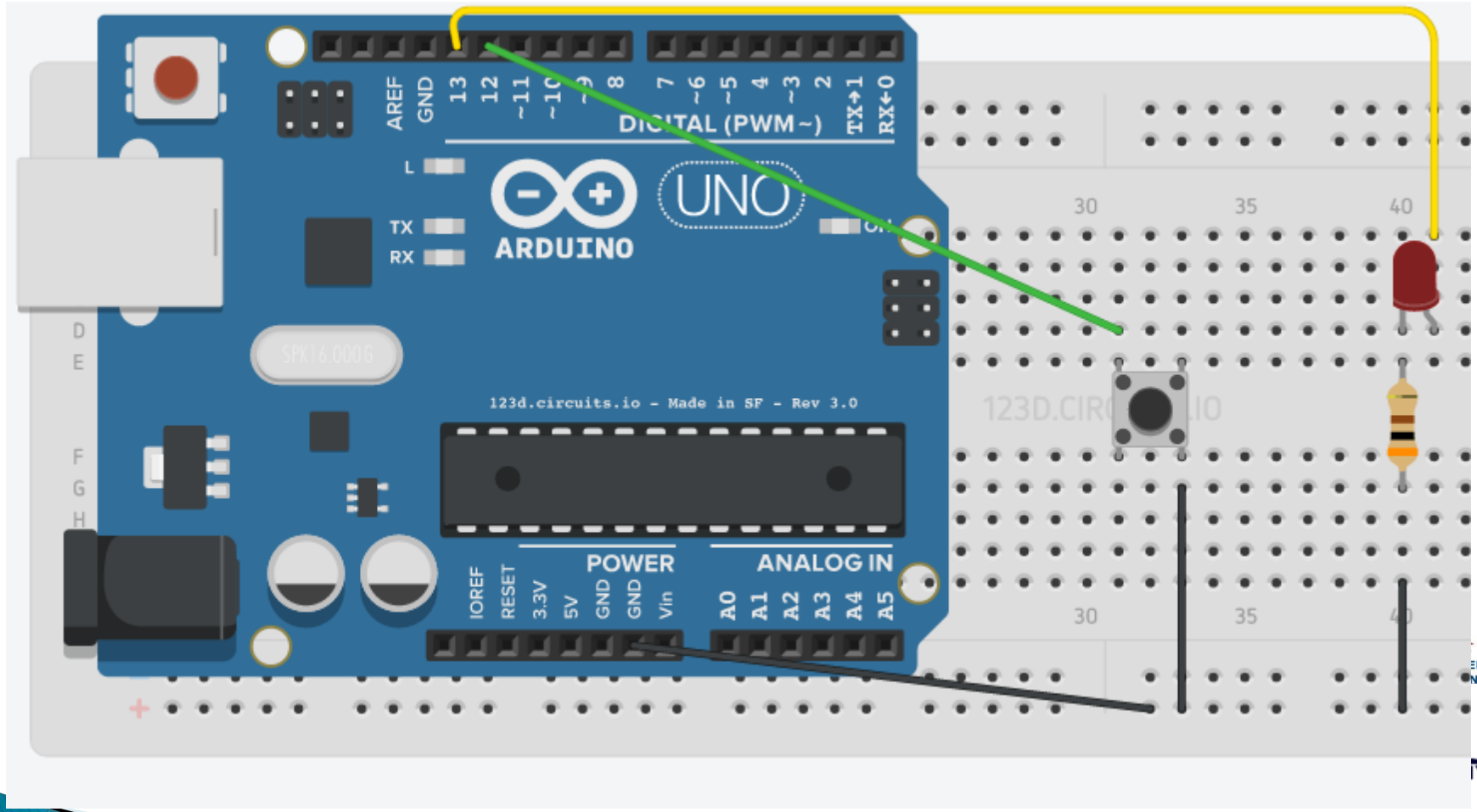
Código PULLUP

```
int botao = 12;
int led = 13;

void setup()
{
  pinMode(led, OUTPUT);
  pinMode(botao, INPUT_PULLUP);
}

void loop()
{
  if(digitalRead(botao) == 0)
  {
    digitalWrite(led, HIGH);
    delay(200);
  }
  else
  {
    digitalWrite(led, LOW);
    delay(200);
  }
}
```

UNIVERSIDADE FEDERAL DE SANTA CATARINA

LARM

Exemplo PULLUP



Exercícios

- Semáforo para pedestres. Um semafóro comum que está sempre no verde. Ele terá mais 2 LEDs para o pedestre e um botão. Quando um pedestre pressionar o botão, o semáforo deve esperar 500 ms e depois deve passar do amarelo para o vermelho. O semáforo de pedestre deve ficar verde, contar 2 segundos e depois voltar a vermelho. Quando ficar vermelho, o semáforo da estrada deve apagar o vermelho e acender o verde.

UNIVERSIDADE FEDERAL DE SANTA CATARINA

LARM