



Notas de Aula - Rev 1

Arduino para Iniciantes

Prof. Rahul M. Juliato
www.rahuljuliato.com

Curso

- ❖ Foco: pessoas com pouco ou nenhum conhecimento na plataforma ou programação
- ❖ Duração: 40 horas
- ❖ Metodologia: Hands-on



Informações Importantes

- ❖ O que é Arduino?
- ❖ Quais placas de desenvolvimento Arduino existem?
- ❖ Posso ter acesso ao diagrama eletrônico?
- ❖ Posso contribuir com o projeto?
- ❖ Onde obter ajuda?
- ❖ Respostas: www.arduino.cc

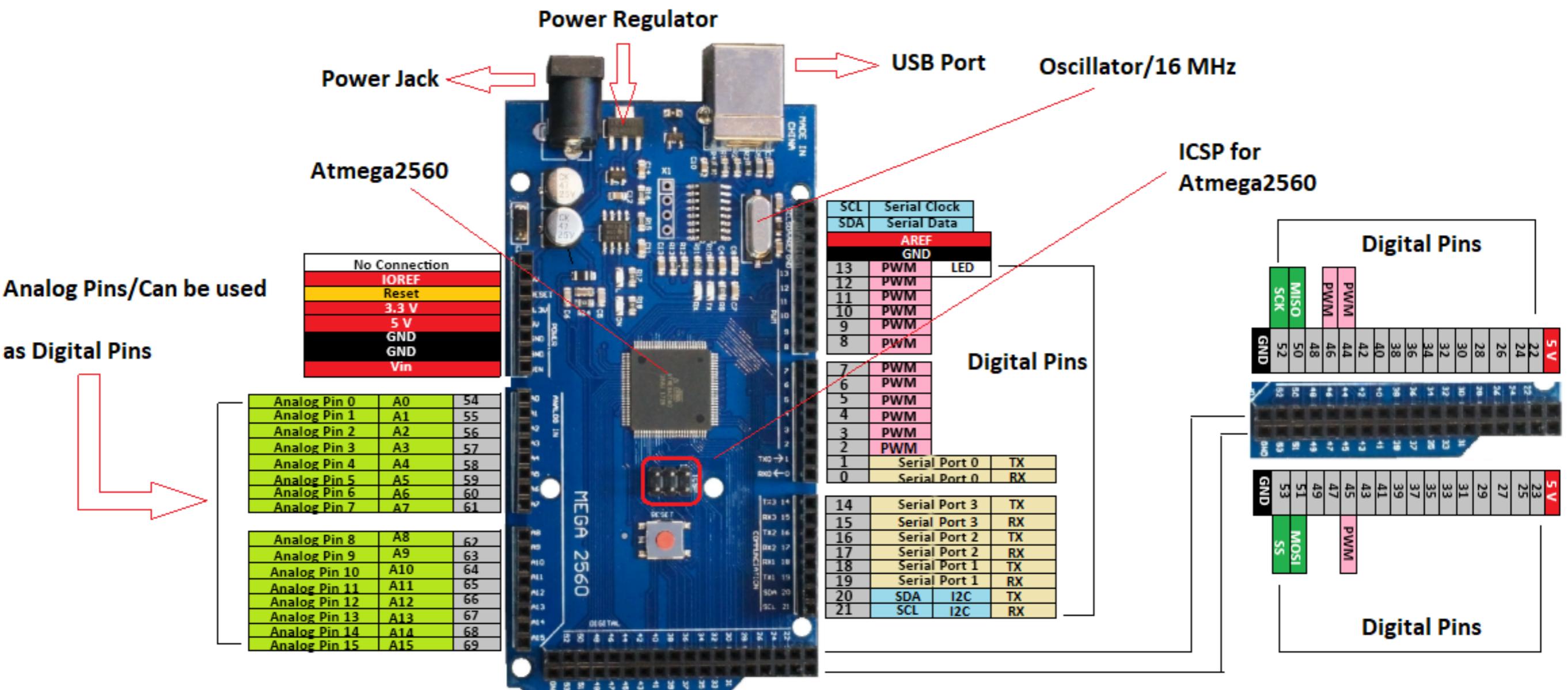


Plataforma Utilizada em nosso Curso

- ❖ Arduino Mega 2560



Pinagem (pinout)



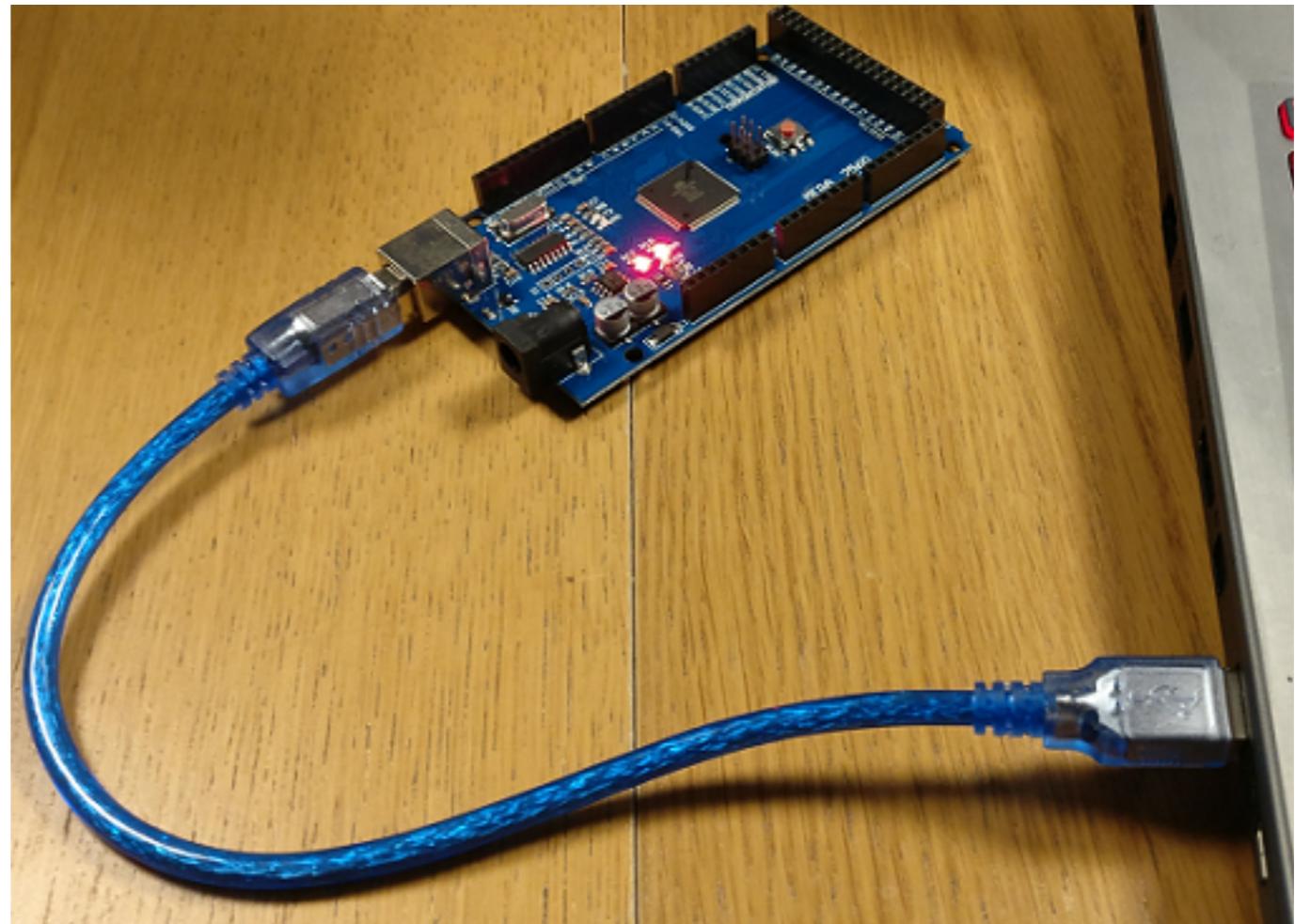
Preparação

- ❖ No computador:
 - ❖ Interface de desenvolvimento (IDE)
 - ❖ Drivers de comunicação (Windows, Linux e MacOS)
 - ❖ <https://www.arduino.cc/en/main/software>
 - ❖ Alternativamente é possível usar um editor online e programas portáteis, embora sejam menos recomendados para projetos maiores ou com potencial de expansão.



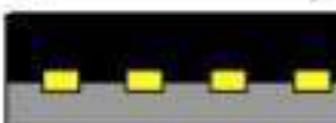
Preparação

- ❖ Hardware:
 - ❖ Com os drivers instalados, conectar o Arduino ao computador por meio do cabo USB.

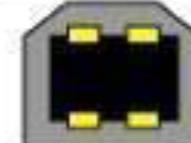


Um gostinho da complicação evitada

USB CONNECTOR TYPES



Type A



Type B



Type A Mini



Type B Mini



Type A Micro



Type B Micro



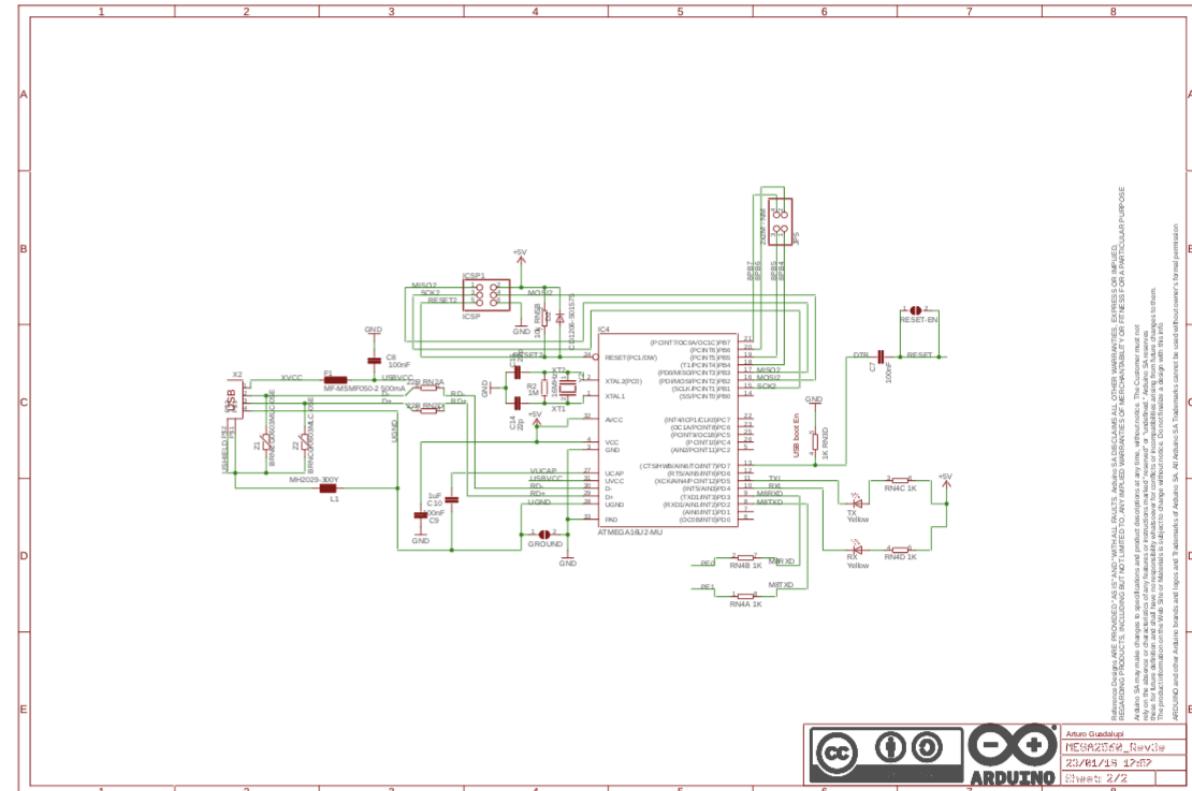
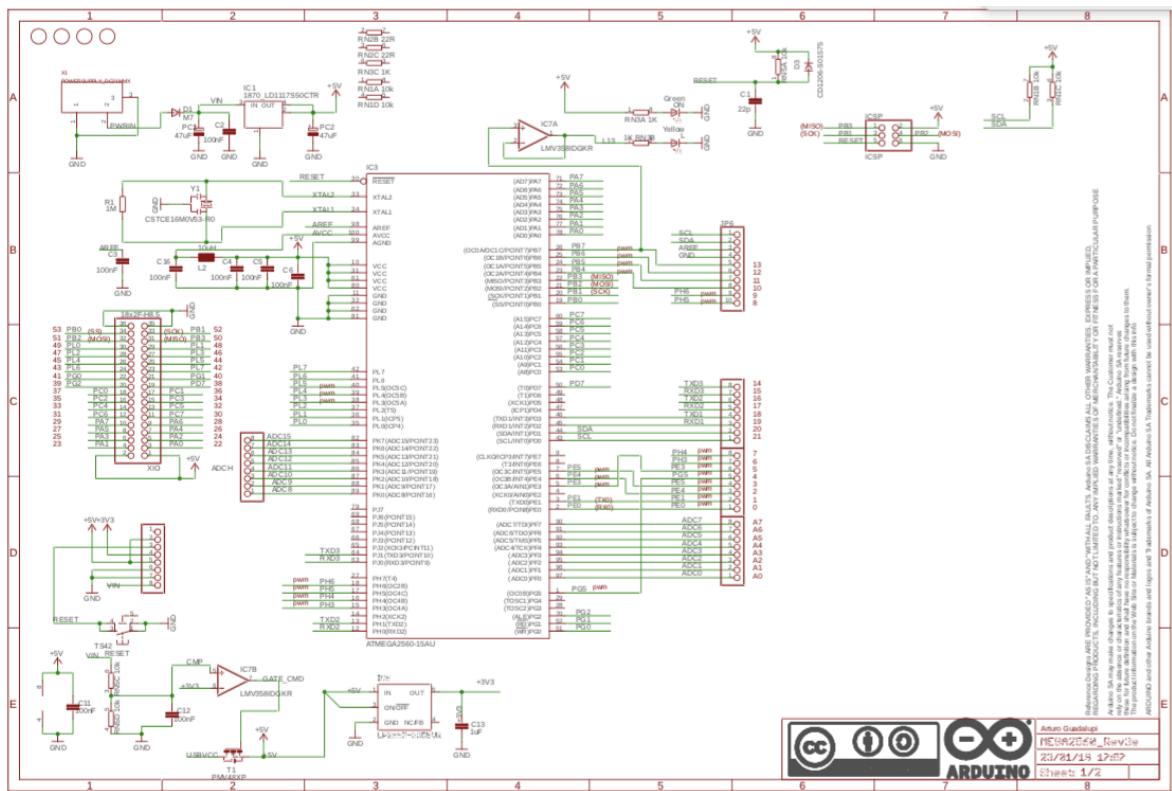
Type B Micro USB3



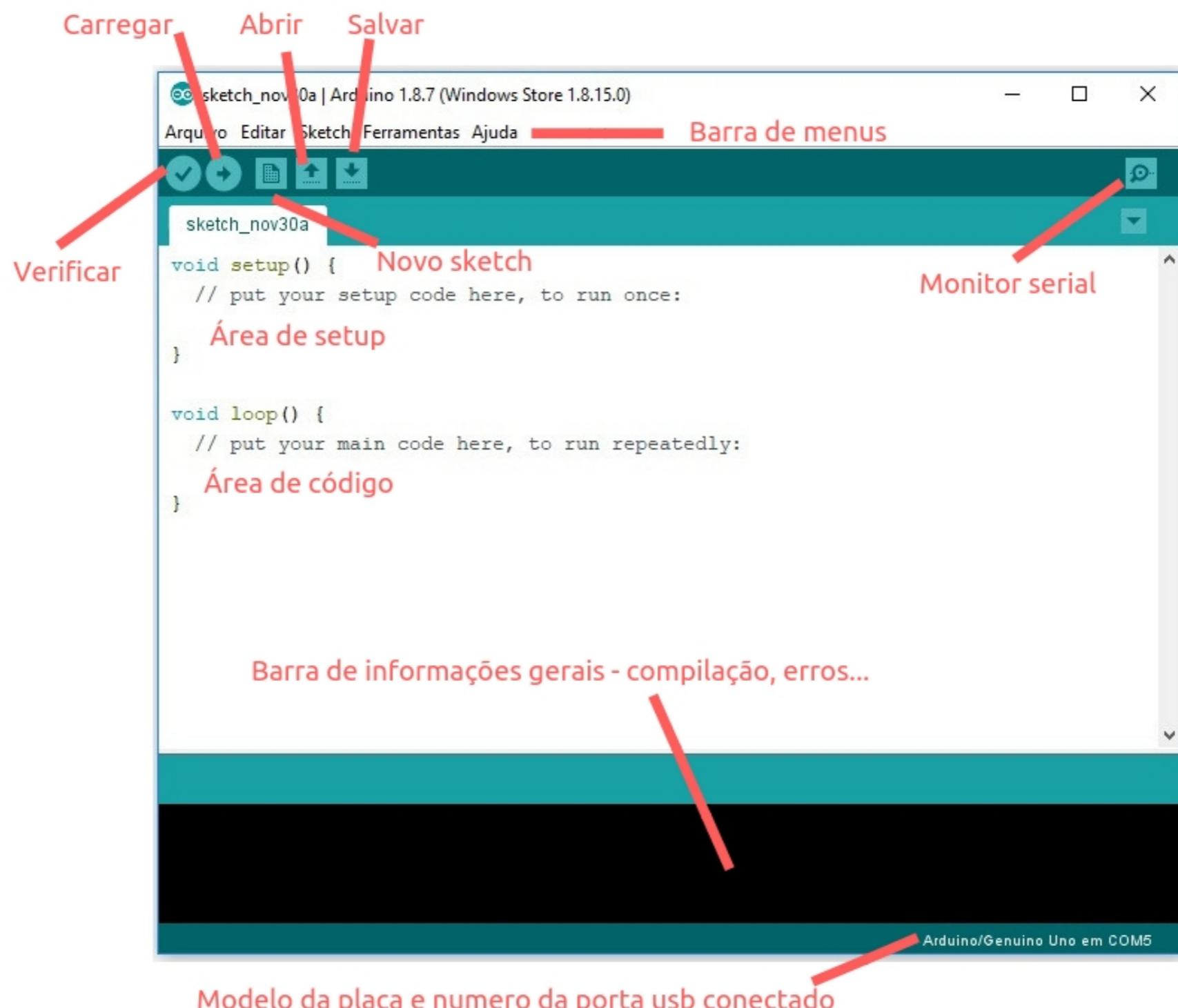
Type C

E que tal...

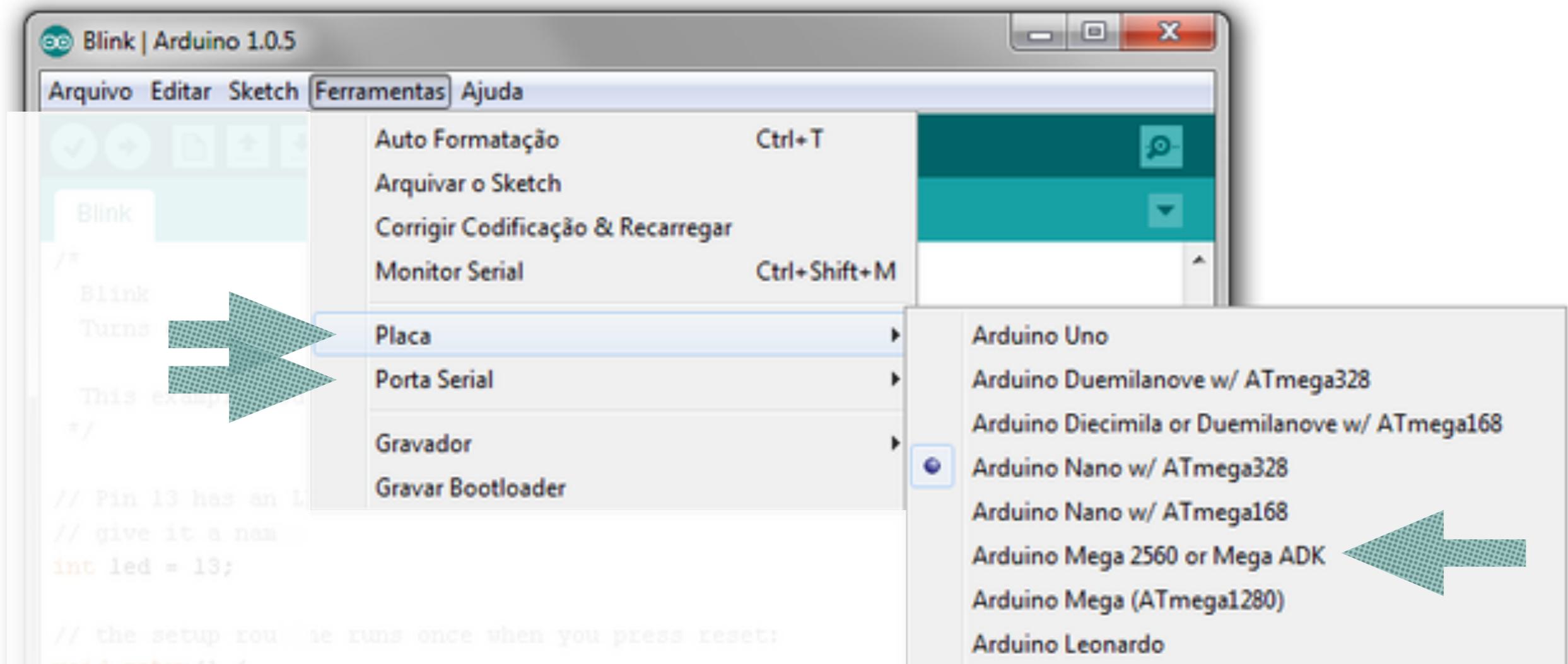
- ❖ Projetar o circuito completo toda vez que quiser utilizar um microcontrolador?



Arduino IDE



Configurar a placa e porta



Primeiro programa

- ❖ Vamos fazer juntos...



Ficou assim?

```
/*
 * Programa: Piscar o led interno da placa
 * Arduino : MEGA 2560
 * Autor   : Rahul M. Juliato
 * Data    : 16.09.2019
 */

/* Setup roda somente ao energizar o Arduino */
void setup(){
    pinMode(13, OUTPUT);           // Configura o pino 13 como saída
}

/* Loop é o laço de repetição infinito */
void loop(){
    digitalWrite(13, HIGH);        // Coloca o pino 13 em nível alto (5V)
    delay(1000);                 // Aguarda 1 segundo
    digitalWrite(13, LOW);        // Coloca o pino 13 em nível baixo (0V)
    delay(1000);                 // Aguarda 1 segundo
}
```



Conseguiu ver na prática?

Exercícios:

- ❖ 1.) Alterar o tempo para 800, 500, 200, 100, 20ms e comprovar na placa
- ❖ 2.) Criar um programa que:
 - A. Pisque o LED 4 vezes com 1 segundo de delay, em seguida...
 - B. Pisque o LED 8 vezes com 50ms de delay, e em seguida
 - C. Pisque o LED 20 vezes com 10ms de delay, retornando ao primeiro item em um laço de repetição infinito.



Lembrando...

- ❖ O que é uma função?
- ❖ Quantos argumentos têm as funções abaixo?
- ❖ Quais argumentos as funções abaixo esperam receber?

```
digitalWrite(13, HIGH);  
delay(1000);
```



Como você sabe disso?

- ❖ A linguagem de programação utilizada na IDE do Arduino é baseada em C e C++. Livros e cursos dessas linguagens podem te ajudar no futuro.
- ❖ Como saber de onde vieram esses comandos?
- ❖ <https://www.arduino.cc/reference/pt/>



Quando a mesma informação se repete...

- ❖ Imagine que estamos utilizando o LED interno da placa, conectado ao pino 13 do Arduino. Contudo, devido a uma necessidade do projeto, o pino 22 estaria mais próximo do Led. Basta substituir todos os 13 por 22, certo?
- ❖ Será que você lembraria de todos em um programa grande?



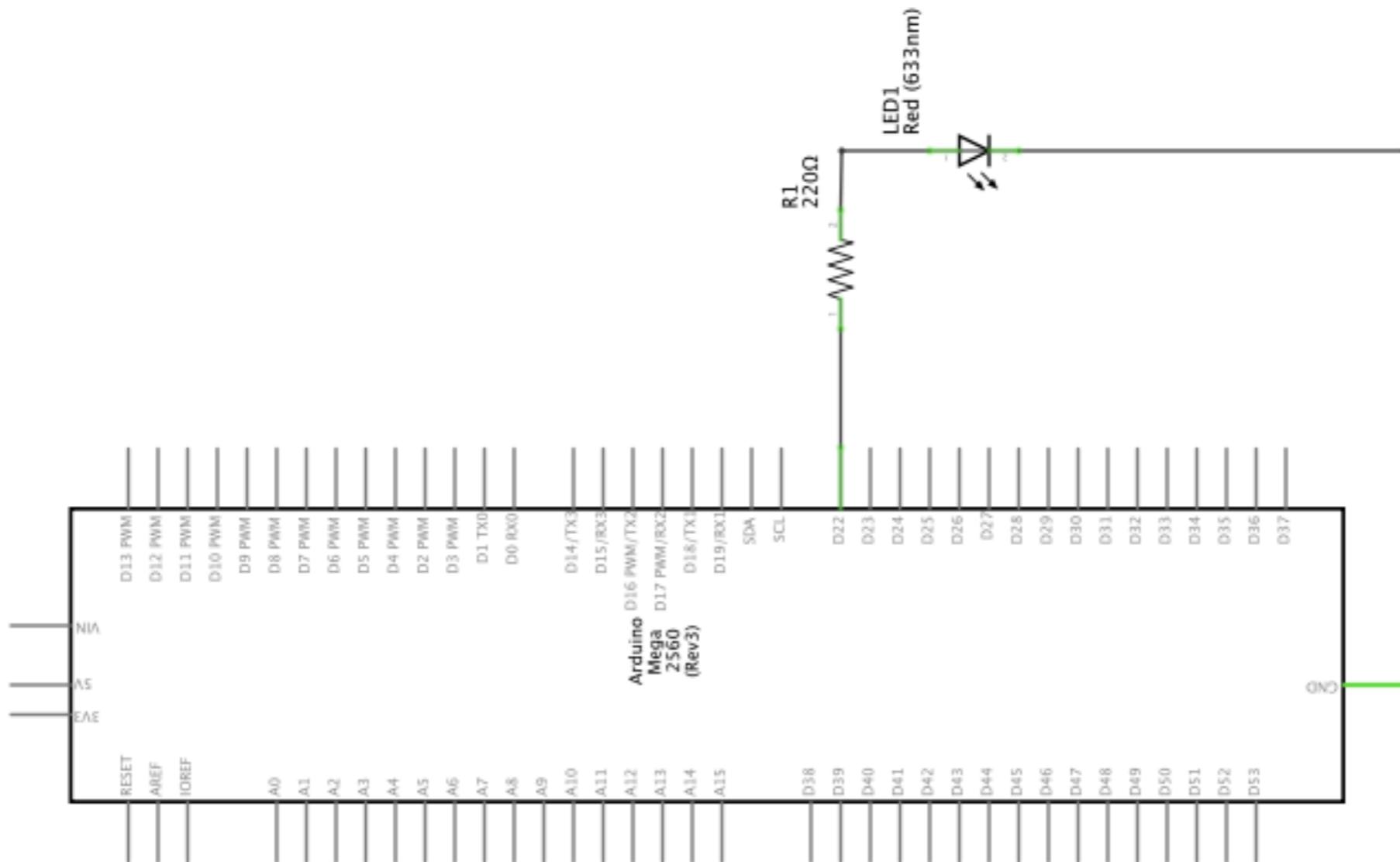
Substituir 13 por 22...

- ❖ Sua vez!
- ❖ Me mostre esse programa funcionando!

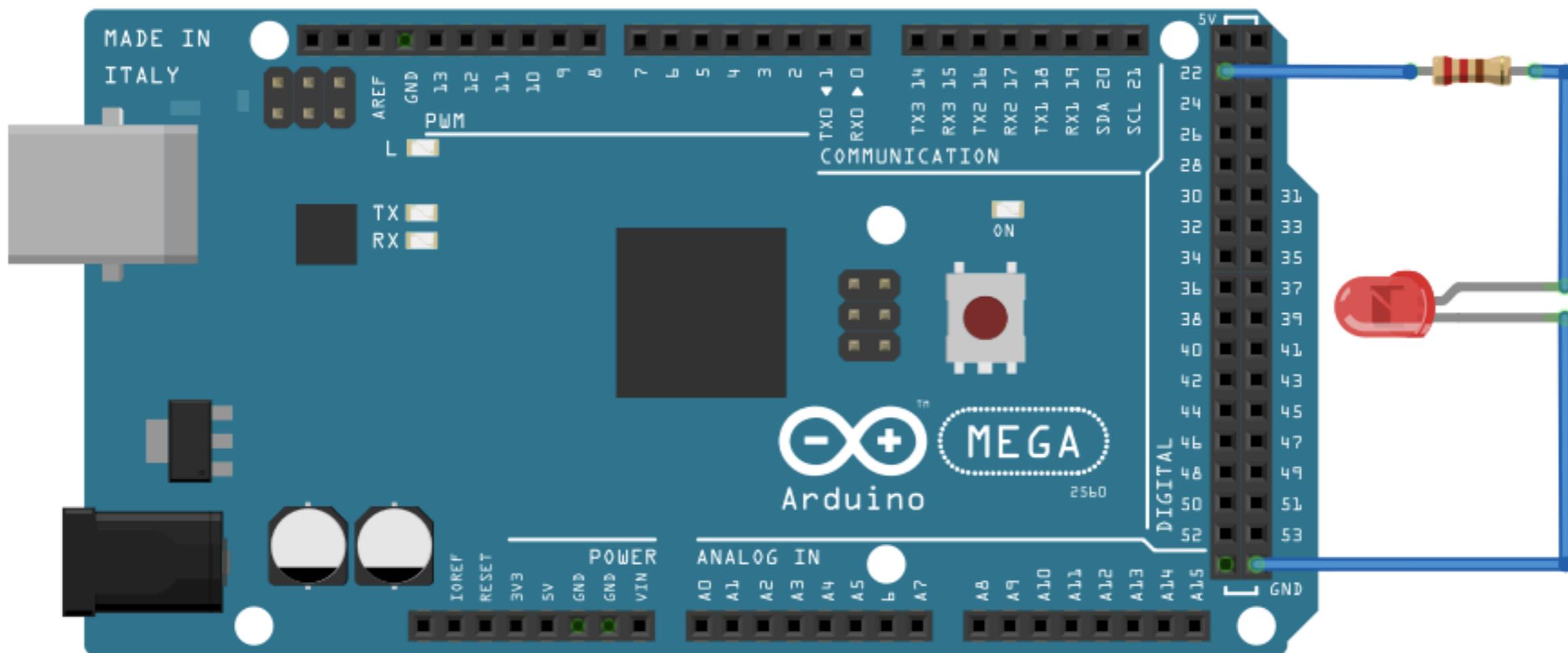


Ops... faltou alguma coisa...

- ❖ Vamos conectar algo ao pino 22:
- ❖ Realize a seguinte montagem:



Simplifiquemos...



Voltando ao nosso programa...

```
/*
 * Programa: Piscar o led interno da placa - versão 2
 * Arduino : MEGA 2560
 * Autor   : Rahul M. Juliato
 * Data    : 18.09.2019
 */

#define LED 22

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

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

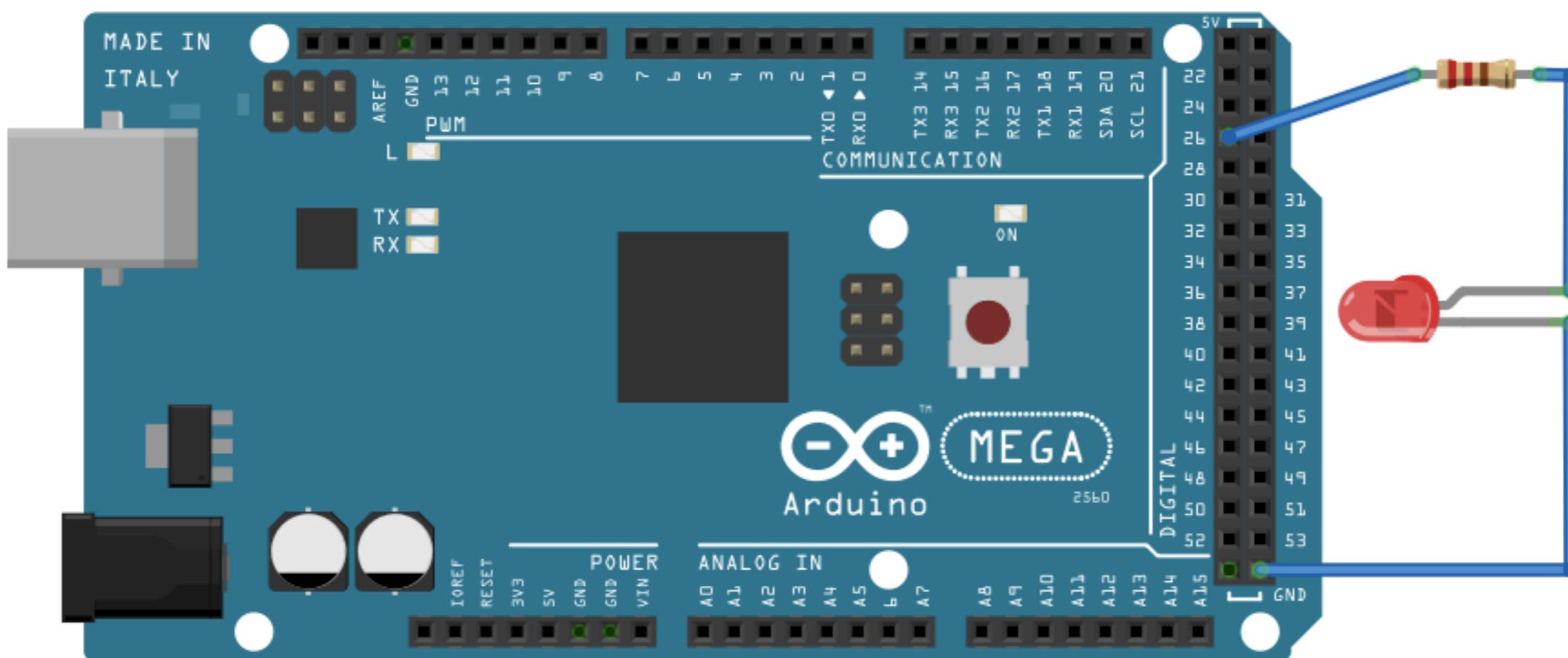


Exercício...

- ❖ Agora vamos testar o LED no pino 26, altere:
 - ❖ Hardware
 - ❖ Software



Resposta



Resposta

```
/*
 * Programa: Piscar o led interno da placa - versão 2
 * Arduino : MEGA 2560
 * Autor   : Rahul M. Juliato
 * Data    : 18.09.2019
 */

#define LED 26

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

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



Exercícios

- ❖ A) Altere o software para definir o tempo no início do programa. Utilize 200 milissegundos.
- ❖ B) Altere o software para que o tempo ligado fique em 50ms e desligado em 20ms.
- ❖ C) Encontre o limite máximo de tempo delay chutando valores.
- ❖ D) Tente inserir o tempo de delay de 2,5 segundos.
- ❖ E) Tente inserir o tempo de delay de 100,5ms.



Outra forma de definir constantes

```
/*
 * Programa: Piscar Led externo a placa no pino 26
 * Arduino : MEGA 2560
 * Autor   : Rahul M. Juliato
 * Data    : 18.09.2019
 */

const int led = 26;
const int ton = 100;
const int tof = 50;

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

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

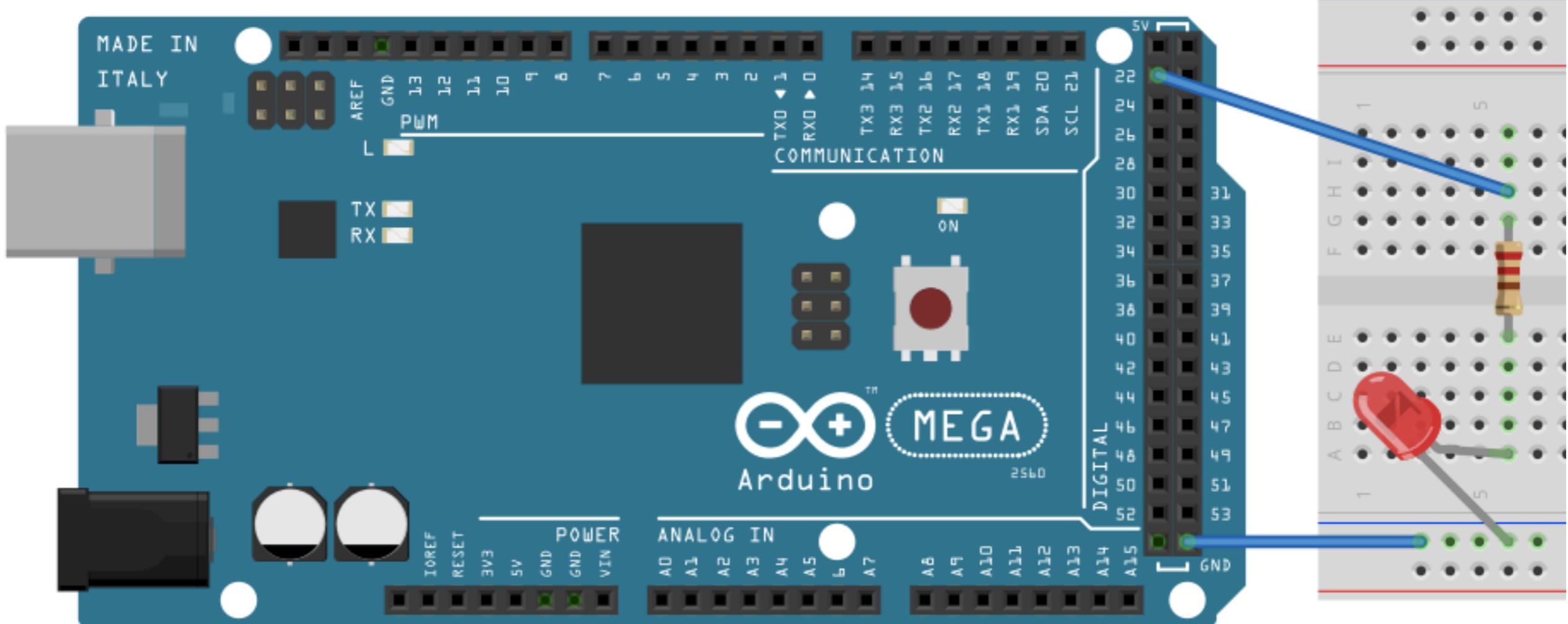


Se eu ainda não tenho um Arduino...

- ❖ Onde posso treinar?
- ❖ <https://create.arduino.cc> (sem o Arduino, somente é possível escrever os programas... tem planos pagos mais poderosos...)
- ❖ <https://www.tinkercad.com> (Simula o Arduino Uno e eletrônica online)
- ❖ <https://www.sites.google.com/site/unoardusim/services> (ArduSIM - Simulador de UNO com componentes para Windows)

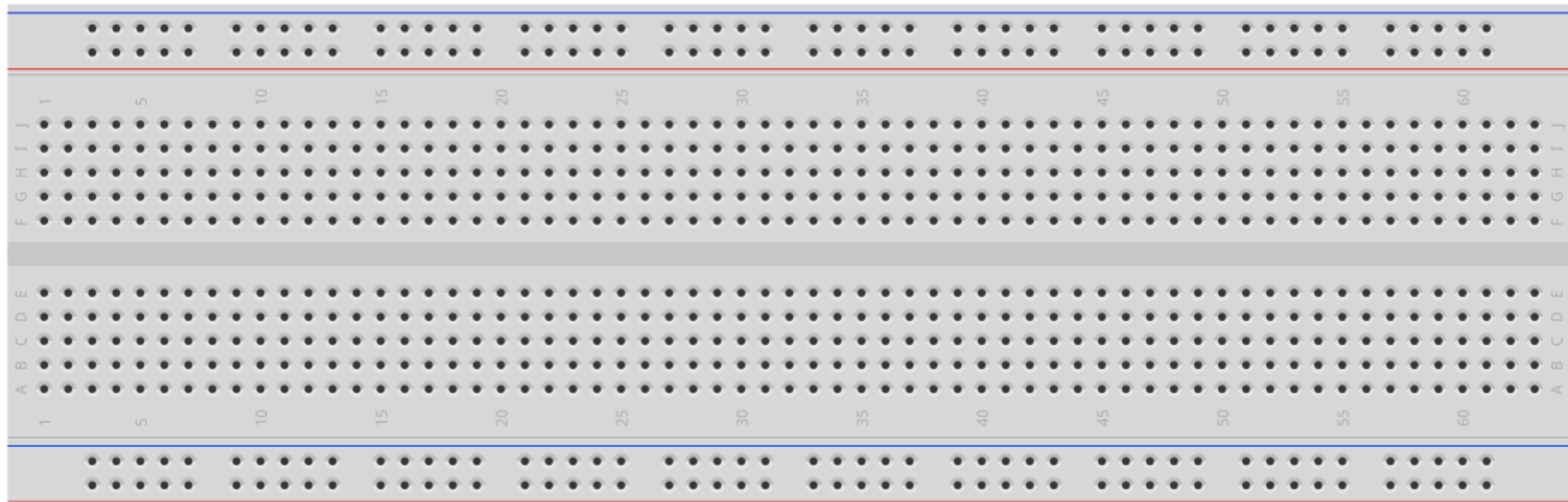


Monte o circuito...

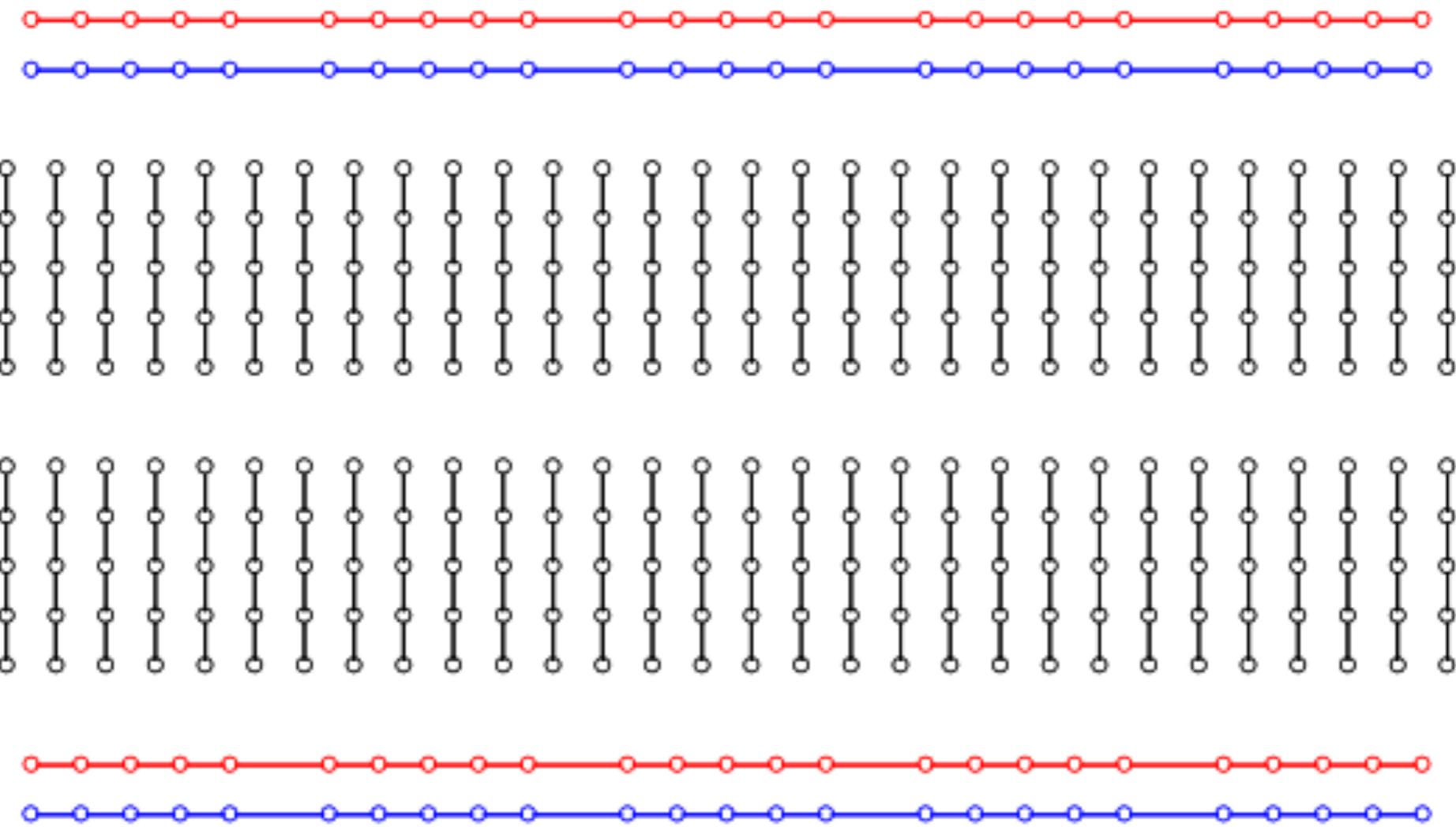


Como funciona um protoboard...

- ❖ Soldar? Encaixar? O que é isso?



Internamente

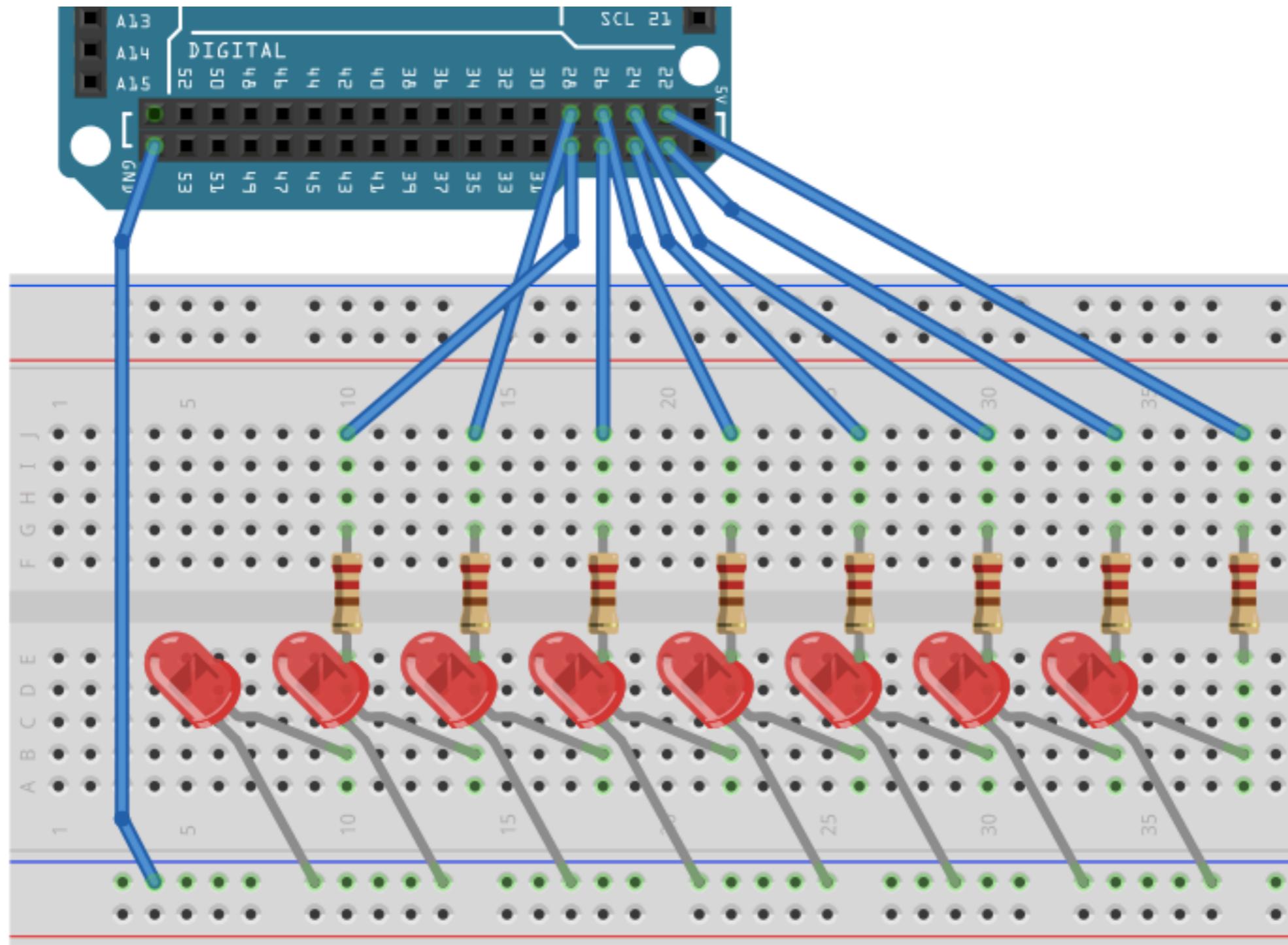


Funcionou?

- ❖ Pronto para o próximo?



Montemos o seguinte circuito...



Para esse hardware...

- ❖ Programar:
- ❖ A) Um pisca-pisca de 8 leds com 250ms de delay
- ❖ B) Um pisca-pisca do tipo 1 sim, 1 não de 250ms de delay
- ❖ C) Varredura da direita para a esquerda com 20ms de delay
- ❖ D) Varredura da esquerda para a direita com 20ms de delay
- ❖ E) Varredura C e D em sequência



Resposta A

```
/*
 * Programa: Piscar 8 leds
 * Arduino : MEGA 2560
 * Autor   : Rahul M. Juliato
 * Data    : 18.09.2019
 */
#define tempo 250

void setup(){
    pinMode(22, OUTPUT);
    pinMode(23, OUTPUT);
    pinMode(24, OUTPUT);
    pinMode(25, OUTPUT);
    pinMode(26, OUTPUT);
    pinMode(27, OUTPUT);
    pinMode(28, OUTPUT);
    pinMode(29, OUTPUT);
}

}
```

```
void loop(){
    digitalWrite(22, HIGH);
    digitalWrite(23, HIGH);
    digitalWrite(24, HIGH);
    digitalWrite(25, HIGH);
    digitalWrite(26, HIGH);
    digitalWrite(27, HIGH);
    digitalWrite(28, HIGH);
    digitalWrite(29, HIGH);

    delay(tempo);

    digitalWrite(22, LOW);
    digitalWrite(23, LOW);
    digitalWrite(24, LOW);
    digitalWrite(25, LOW);
    digitalWrite(26, LOW);
    digitalWrite(27, LOW);
    digitalWrite(28, LOW);
    digitalWrite(29, LOW);

    delay(tempo);
}

}
```



Resposta B

```
/*
 * Programa: Piscar 8 leds - 1 sim e 1 não
 * Arduino : MEGA 2560
 * Autor   : Rahul M. Juliato
 * Data    : 18.09.2019
 */
#define tempo 250

void setup(){
    pinMode(22, OUTPUT);
    pinMode(23, OUTPUT);
    pinMode(24, OUTPUT);
    pinMode(25, OUTPUT);
    pinMode(26, OUTPUT);
    pinMode(27, OUTPUT);
    pinMode(28, OUTPUT);
    pinMode(29, OUTPUT);
}

void loop(){
    digitalWrite(22, HIGH);
    digitalWrite(23, LOW);
    digitalWrite(24, HIGH);
    digitalWrite(25, LOW);
    digitalWrite(26, HIGH);
    digitalWrite(27, LOW);
    digitalWrite(28, HIGH);
    digitalWrite(29, LOW);

    delay(tempo);

    digitalWrite(22, LOW);
    digitalWrite(23, HIGH);
    digitalWrite(24, LOW);
    digitalWrite(25, HIGH);
    digitalWrite(26, LOW);
    digitalWrite(27, HIGH);
    digitalWrite(28, LOW);
    digitalWrite(29, HIGH);

    delay(tempo);
}
```



Resposta C

```
/*
 * Programa: Varredura da direita para a esquerda
 * Arduino : MEGA 2560
 * Autor   : Rahul M. Juliato
 * Data    : 18.09.2019
 */

#define tempo 20

void setup(){
    pinMode(22, OUTPUT);
    pinMode(23, OUTPUT);
    pinMode(24, OUTPUT);
    pinMode(25, OUTPUT);
    pinMode(26, OUTPUT);
    pinMode(27, OUTPUT);
    pinMode(28, OUTPUT);
    pinMode(29, OUTPUT);
}

void loop(){
    digitalWrite(22, LOW);
    digitalWrite(23, LOW);
    digitalWrite(24, LOW);
    digitalWrite(25, LOW);
    digitalWrite(26, LOW);
    digitalWrite(27, LOW);
    digitalWrite(28, LOW);
    digitalWrite(29, LOW);

    digitalWrite(22, HIGH);
    delay(tempo);

    digitalWrite(22, LOW);
    digitalWrite(23, HIGH);
    delay(tempo);

    digitalWrite(23, LOW);
    digitalWrite(24, HIGH);
    delay(tempo);

    digitalWrite(25, LOW);
    digitalWrite(26, HIGH);
    delay(tempo);

    digitalWrite(26, LOW);
    digitalWrite(27, HIGH);
    delay(tempo);

    digitalWrite(27, LOW);
    digitalWrite(28, HIGH);
    delay(tempo);

    digitalWrite(28, LOW);
    digitalWrite(29, HIGH);
    delay(tempo);
}
```



Resposta D

```
/*
 * Programa: Varredura da esquerda para a direita
 * Arduino : MEGA 2560
 * Autor   : Rahul M. Juliato
 * Data    : 18.09.2019
 */

#define tempo 20

void setup(){
    pinMode(22, OUTPUT);
    pinMode(23, OUTPUT);
    pinMode(24, OUTPUT);
    pinMode(25, OUTPUT);
    pinMode(26, OUTPUT);
    pinMode(27, OUTPUT);
    pinMode(28, OUTPUT);
    pinMode(29, OUTPUT);
}

void loop(){
    digitalWrite(22, LOW);
    digitalWrite(23, LOW);
    digitalWrite(24, LOW);
    digitalWrite(25, LOW);
    digitalWrite(26, LOW);
    digitalWrite(27, LOW);
    digitalWrite(28, LOW);
    digitalWrite(29, LOW);

    digitalWrite(29, HIGH);
    delay(tempo);

    digitalWrite(29, LOW);
    digitalWrite(28, HIGH);
    delay(tempo);

    digitalWrite(28, LOW);
    digitalWrite(27, HIGH);
    delay(tempo);

    digitalWrite(27, LOW);
    digitalWrite(26, HIGH);
    delay(tempo);

    digitalWrite(26, LOW);
    digitalWrite(25, HIGH);
    delay(tempo);

    digitalWrite(25, LOW);
    digitalWrite(24, HIGH);
    delay(tempo);

    digitalWrite(24, LOW);
    digitalWrite(23, HIGH);
    delay(tempo);

    digitalWrite(23, LOW);
    digitalWrite(22, HIGH);
    delay(tempo);
}
```



Resposta E

```
/*
 * Programa: Varredura bilateral
 * Arduino : MEGA 2560
 * Autor   : Rahul M. Juliato
 * Data    : 18.09.2019
 */

#define tempo 20

void setup(){
  pinMode(22, OUTPUT);
  pinMode(23, OUTPUT);
  pinMode(24, OUTPUT);
  pinMode(25, OUTPUT);
  pinMode(26, OUTPUT);
  pinMode(27, OUTPUT);
  pinMode(28, OUTPUT);
  pinMode(29, OUTPUT);
}

void loop(){
  digitalWrite(22, LOW);
  digitalWrite(23, LOW);
  digitalWrite(24, LOW);
  digitalWrite(25, LOW);
  digitalWrite(26, LOW);
  digitalWrite(27, LOW);
  digitalWrite(28, LOW);
  digitalWrite(29, LOW);

  digitalWrite(22, HIGH);
  delay(tempo);

  digitalWrite(22, LOW);
  digitalWrite(23, HIGH);
  delay(tempo);

  digitalWrite(23, LOW);
  digitalWrite(24, HIGH);
  delay(tempo);

  digitalWrite(24, LOW);
  digitalWrite(25, HIGH);
  delay(tempo);

  digitalWrite(25, LOW);
  digitalWrite(26, HIGH);
  delay(tempo);

  digitalWrite(26, LOW);
  digitalWrite(27, HIGH);
  delay(tempo);

  digitalWrite(27, LOW);
  digitalWrite(28, HIGH);
  delay(tempo);

  digitalWrite(28, LOW);
  digitalWrite(29, HIGH);
  delay(tempo);

  digitalWrite(29, LOW);
  digitalWrite(28, HIGH);
  delay(tempo);

  digitalWrite(28, LOW);
  digitalWrite(27, HIGH);
  delay(tempo);

  digitalWrite(27, LOW);
  digitalWrite(26, HIGH);
  delay(tempo);

  digitalWrite(26, LOW);
  digitalWrite(25, HIGH);
  delay(tempo);

  digitalWrite(25, LOW);
  digitalWrite(24, HIGH);
  delay(tempo);

  digitalWrite(24, LOW);
  digitalWrite(23, HIGH);
  delay(tempo);

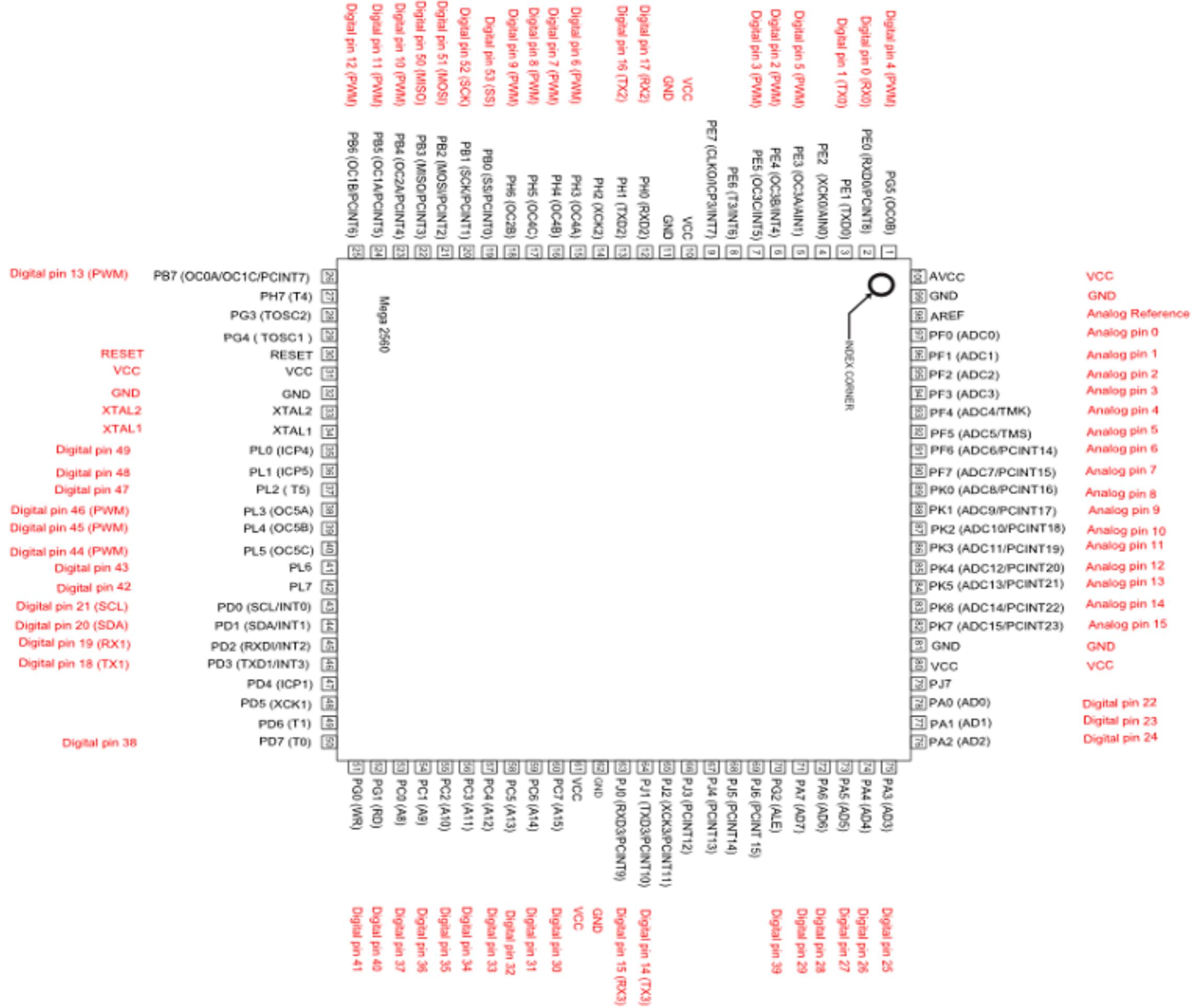
  digitalWrite(23, LOW);
  digitalWrite(22, HIGH);
  delay(tempo);
}
```



Mas porque utilizamos 22 à 29?

- ❖ Internamente, o ATMEGA (cérebro do Arduino, nosso microcontrolador), separa as portas de I/Os em blocos de 8.





Na mesma página...

71	PA7 (AD7)	Digital pin 29
72	PA6 (AD6)	Digital pin 28
73	PA5 (AD5)	Digital pin 27
74	PA4 (AD4)	Digital pin 26
75	PA3 (AD3)	Digital pin 25
76	PA2 (AD2)	Digital pin 24
77	PA1 (AD1)	Digital pin 23
78	PA0 (AD0)	Digital pin 22

- ❖ Essas portas tem 8 pinos, ou 8 bits
- ❖ Bits são dígitos binários que podem ser 0 (desligado) ou 1 (ligado). O valor de uma porta que tem o pino 22, 24 e 29 ligados e 23, 25, 26, 27 e 28 desligados é em binário: 10100001



Mas eu não poderia ter escolhido outros pinos?

- ❖ Qual a vantagem de se utilizar uma mesma porta?
- ❖ Podemos fazer operações com a porta completa em um único comando.
- ❖ Exemplo: Configurar os pinos do port A (22 à 29) todos como saída.

DDRA = B1111111;

- ❖ 1 = Saída, 0 = Entrada



O que mais podemos fazer?

- ❖ Ligar e desligar o port todo em um comando só.

POR TA = B10101010;

- ❖ 1 = Ligado (5V) 0 = Desligado (0V)



Sistemas de Numeração

- ❖ O Arduino trabalha com 4 sistemas de numeração: Binário, Octal, Hexadecimal e Decimal
- ❖ Por exemplo: O número decimal 128 pode ser representado no Arduino das formas:

Decimal	128
Binário	B10000000
Octal	0200
Hexadecimal	0x80

- ❖ Esse assunto é de grande utilidade para o futuro programador, vale a pena estudar!



Exemplo

```
/*
Programa: Pisca-pisca no Port A
Arduino : MEGA 2560
Autor   : Rahul M. Juliato
Data    : 20.09.2019
*/
const int tempo = 100;
void setup(){
    DDRA = B11111111;
}
void loop() {
    PORTA = B11111111;
    delay(tempo);
    PORTA = B00000000;
    delay(tempo);
}
```



Comparando com o programa anterior

```
/*
Programa: Pisca-pisca no Port A
Arduino : MEGA 2560
Autor   : Rahul M. Juliato
Data    : 20.09.2019
*/
const int tempo = 100;

void setup(){
  DDRA = B11111111;
}

void loop() {
  PORTA = B11111111;
  delay(tempo);
  PORTA = B00000000;
  delay(tempo);
}

/*
* Programa: Piscar 8 leds
* Arduino : MEGA 2560
* Autor   : Rahul M. Juliato
* Data    : 18.09.2019
*/
#define tempo 250

void setup(){
  pinMode(22, OUTPUT);
  pinMode(23, OUTPUT);
  pinMode(24, OUTPUT);
  pinMode(25, OUTPUT);
  pinMode(26, OUTPUT);
  pinMode(27, OUTPUT);
  pinMode(28, OUTPUT);
  pinMode(29, OUTPUT);
}

void loop(){
  digitalWrite(22, HIGH);
  digitalWrite(23, HIGH);
  digitalWrite(24, HIGH);
  digitalWrite(25, HIGH);
  digitalWrite(26, HIGH);
  digitalWrite(27, HIGH);
  digitalWrite(28, HIGH);
  digitalWrite(29, HIGH);

  delay(tempo);

  digitalWrite(22, LOW);
  digitalWrite(23, LOW);
  digitalWrite(24, LOW);
  digitalWrite(25, LOW);
  digitalWrite(26, LOW);
  digitalWrite(27, LOW);
  digitalWrite(28, LOW);
  digitalWrite(29, LOW);
}
```



Repetir os exercícios

- ❖ Programar:
- ❖ A) Um pisca-pisca de 8 leds com 250ms de delay
- ❖ B) Um pisca-pisca do tipo 1 sim, 1 não de 250ms de delay
- ❖ C) Varredura da direita para a esquerda com 20ms de delay
- ❖ D) Varredura da esquerda para a direita com 20ms de delay
- ❖ E) Varredura C e D em sequência



Resposta A

```
/*
Programa: Pisca-pisca no Port A
Arduino : MEGA 2560
Autor   : Rahul M. Juliato
Data    : 20.09.2019
*/
const int tempo = 250;

void setup(){
    DDRA = B11111111;
}

void loop() {
    PORTA = B11111111;
    delay(tempo);
    PORTA = B00000000;
    delay(tempo);
}
```



Resposta B

```
/*
Programa: Pisca-pisca no Port A - 1 sim, 1 não
Arduino : MEGA 2560
Autor   : Rahul M. Juliato
Data    : 20.09.2019
*/
const int tempo = 250;
void setup(){
  DDRA = B11111111;
}
void loop() {
  PORTA = B10101010;
  delay(tempo);
  PORTA = B01010101;
  delay(tempo);
}
```



Resposta C

```
/*
Programa: Pisca-pisca no Port A - Varredura direta para esquerda
Arduino : MEGA 2560
Autor   : Rahul M. Juliato
Data    : 20.09.2019
*/
const int tempo = 20;

void setup(){
  DDRA = B11111111;
}

void loop() {
  PORTA = B00000001;
  delay(tempo);
  PORTA = B00000010;
  delay(tempo);
  PORTA = B00000100;
  delay(tempo);
  PORTA = B00001000;
  delay(tempo);
  PORTA = B00100000;
  delay(tempo);
  PORTA = B01000000;
  delay(tempo);
  PORTA = B10000000;
  delay(tempo);
}
```



Resposta D

```
/*
Programa: Pisca-pisca no Port A - Varredura esquerda para direita
Arduino : MEGA 2560
Autor   : Rahul M. Juliato
Data    : 20.09.2019
*/
const int tempo = 20;

void setup(){
    DDRA = B11111111;
}

void loop() {
    PORTA = B10000000;
    delay(tempo);
    PORTA = B01000000;
    delay(tempo);
    PORTA = B00100000;
    delay(tempo);
    PORTA = B00010000;
    delay(tempo);
    PORTA = B00001000;
    delay(tempo);
    PORTA = B00000100;
    delay(tempo);
    PORTA = B00000010;
    delay(tempo);
    PORTA = B00000001;
    delay(tempo);
}
```



Resposta E

```
/*
Programa: Pisca-pisca no Port A - Varredura vai e vem
Arduino : MEGA 2560
Autor   : Rahul M. Juliato
Data    : 20.09.2019
*/
const int tempo = 20;
void setup(){
  DDRA = B11111111;
}
void loop() {
  PORTA = B00000001;
  delay(tempo);
  PORTA = B00000010;
  delay(tempo);
  PORTA = B00000100;
  delay(tempo);
  PORTA = B00001000;
  delay(tempo);
  PORTA = B00010000;
  delay(tempo);
  PORTA = B00100000;
  delay(tempo);
  PORTA = B01000000;
  delay(tempo);
  PORTA = B10000000;
  delay(tempo);
  PORTA = B01000000;
  delay(tempo);
  PORTA = B00100000;
  delay(tempo);
  PORTA = B00010000;
  delay(tempo);
  PORTA = B00001000;
  delay(tempo);
  PORTA = B00000100;
  delay(tempo);
  PORTA = B00000010;
  delay(tempo);
}
```



Quando repetimos uma seqüência...

- ❖ Quando repetimos várias vezes uma sequência de comandos exatos, é interessante a criação de sub-rotinas para economizar memória e organizar o fluxo de informações.
- ❖ Suponha que você queira piscar LEDs alternados 4 vezes com tempo de 1 segundo entre troca de estados e depois manter todos os LEDs acesos por 1 segundo.



Onde vão as sub-rotinas?

- ❖ Usualmente entre as rotinas de setup e loop.
- ❖ Vamos programar uma sub-rotina para pisca alternado juntos...



Ficou assim?

```
/*
Programa: Subrotinas - Piscando alternado 4 vezes e fica todo aceso por 1 segundo
Arduino : MEGA 2560
Autor   : Rahul M. Juliato
Data    : 20.09.2019
*/
```

```
const int tempo = 1000;
```

```
void setup(){
```

```
  DDRA = B11111111;
```

```
}
```

```
void piscapiscante(){
```

```
  PORTA = B10101010;
```

```
  delay(tempo);
```

```
  PORTA = B01010101;
```

```
  delay(tempo);
```

```
}
```

```
void loop() {
```

```
  piscapiscante();
```

```
  piscapiscante();
```

```
  piscapiscante();
```

```
  piscapiscante();
```

```
  PORTA = B11111111;
```

```
  delay(tempo);
```

```
}
```



Exercícios

- ❖ A) Faça um programa que pisque alternado 2 vezes e todos os LEDs 2 vezes em loop infinito
- ❖ B) Altere o programa A para que pisque 2 vezes em 100ms, 2 vezes em 50ms e 2 vezes em 25ms em loop infinito.



Resposta A

```
/*
Programa: Subrotinas - pisque alternado 2 vezes e todos os LEDs 2 vezes em loop infinito
Arduino : MEGA 2560
Autor   : Rahul M. Juliato
Data    : 20.09.2019
*/
const int tempo = 1000;
void setup(){
  DDRA = B11111111;
}
void pisca_alt(){
  PORTA = B10101010;
  delay(tempo);
  PORTA = B01010101;
  delay(tempo);
}
void pisca_tud(){
  PORTA = B11111111;
  delay(tempo);
  PORTA = B11111111;
  delay(tempo);
}
void loop() {
  pisca_alt();
  pisca_alt();
  pisca_tud();
  pisca_tud();
}
```



Resposta B

```
/*
Programa: Subrotinas - pisque alternado 2 vezes e todos os LEDs 2 vezes em loop infinito variando o tempo
Arduino : MEGA 2560
Autor   : Rahul M. Juliato
Data    : 20.09.2019
*/
int tempo = 1000;
void setup(){
    DDRA = B11111111;
}
void pisca_alt(){
    PORTA = B10101010;
    delay(tempo);
    PORTA = B01010101;
    delay(tempo);
}
void pisca_tud(){
    void pisca_tud(){
        PORTA = B11111111;
        delay(tempo);
        PORTA = B11111111;
        delay(tempo);
    }
    void loop() {
        pisca_alt();
        pisca_alt();
        pisca_tud();
        pisca_tud();
        tempo = 100;
        pisca_alt();
        pisca_alt();
        pisca_tud();
        pisca_tud();
        tempo = 50;
    }
    pisca_alt();
    pisca_alt();
    pisca_tud();
    pisca_tud();
    tempo = 25;
}
```



Ok, saídas, saídas, saídas...

- ❖ A letra O de I/O está compreendida!
- ❖ O = Output = Saída = 5V ligado = “Empurra corrente para fora”
- ❖ Mas onde está o I?

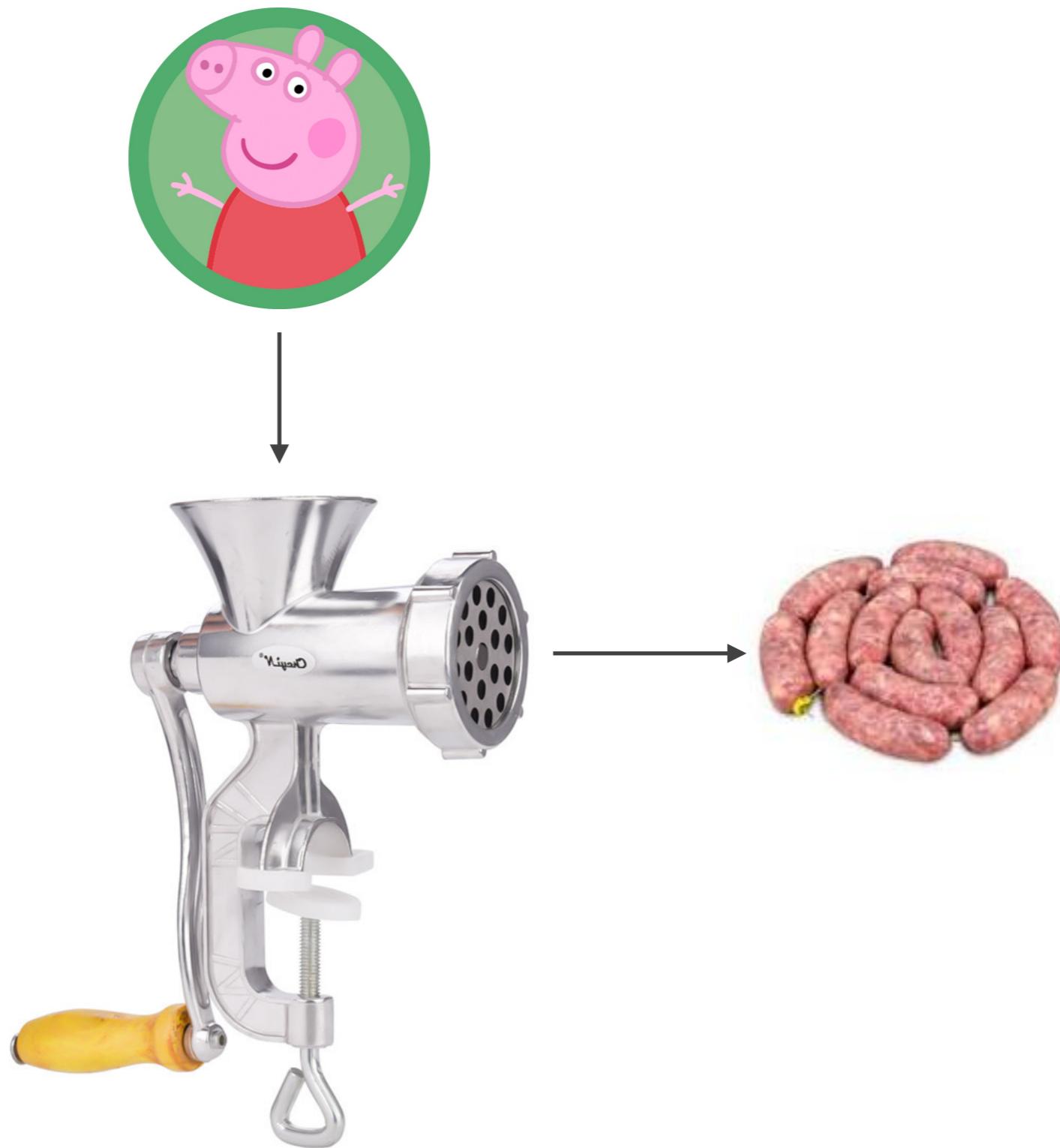


Entradas (Inputs)

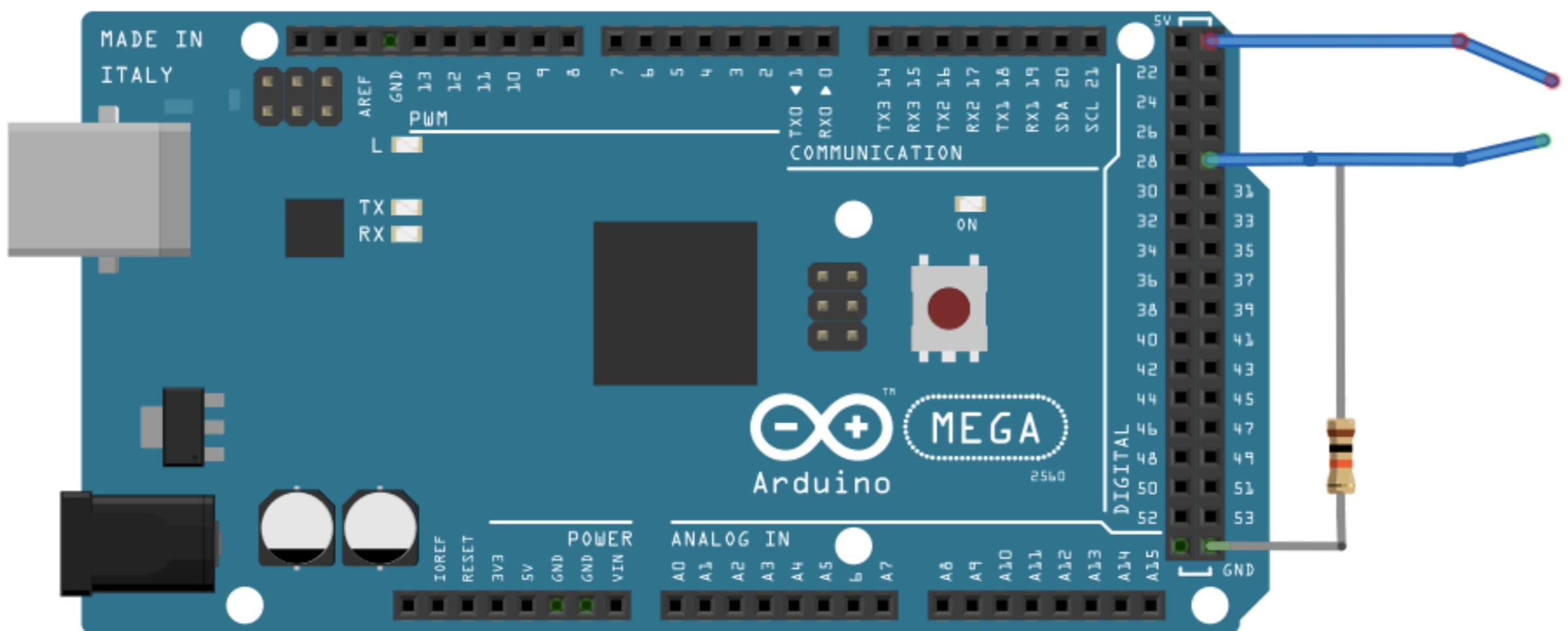
- ❖ Um processamento comum, normalmente possui 3 etapas:
- ❖ Entrada de Informação
- ❖ Processamento de Informação
- ❖ Saída da Informação processada



Em outras palavras...



Hardware de uma entrada



Programa para leitura de entrada

```
/* Programa: Básico para aquisição do estado de uma entrada
 * Arduino : MEGA 2560
 * Autor   : Rahul M. Juliato
 * Data    : 20/09/2019
 */
```

```
const int entrada_pino = 29;

int entrada_estado = 0;

void setup(){
    pinMode(entrada_pino, INPUT);
}

void loop(){
    entrada_estado = digitalRead(entrada_pino);
}
```



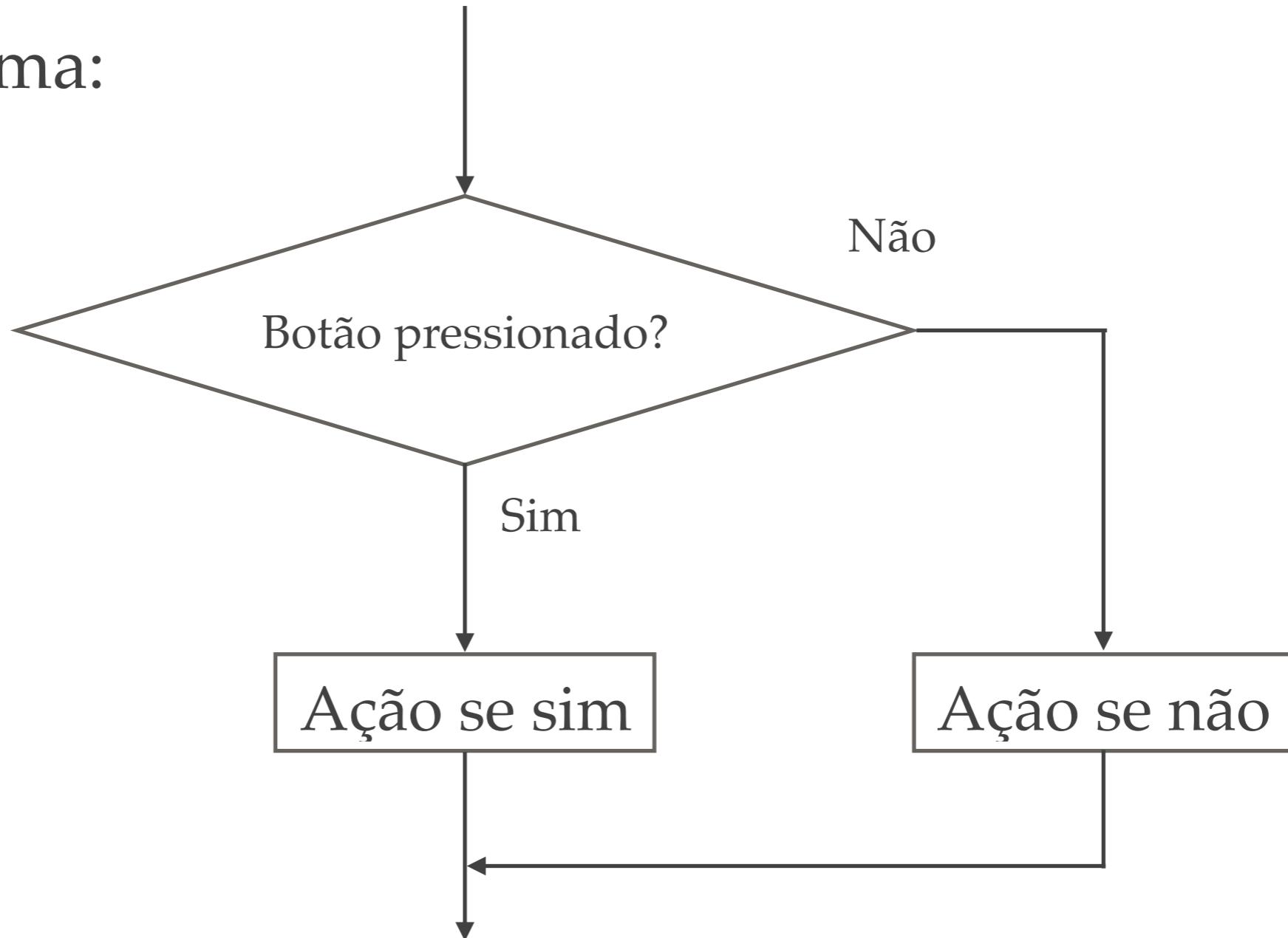
Funcionou?

- ❖ Claro que sim!
- ❖ Mas tudo o que esse programa faz é guardar o estado da sua chave (Fechada ou Aberta) na variável `estado_entrada`;
- ❖ A variável `estado_entrada` recebe HIGH se a chave estiver fechada (ou o botão estiver apertado) ou LOW se a chave estiver aberta (ou o botão estiver solto).
- ❖ O que fazer com essa entrada? Processar!

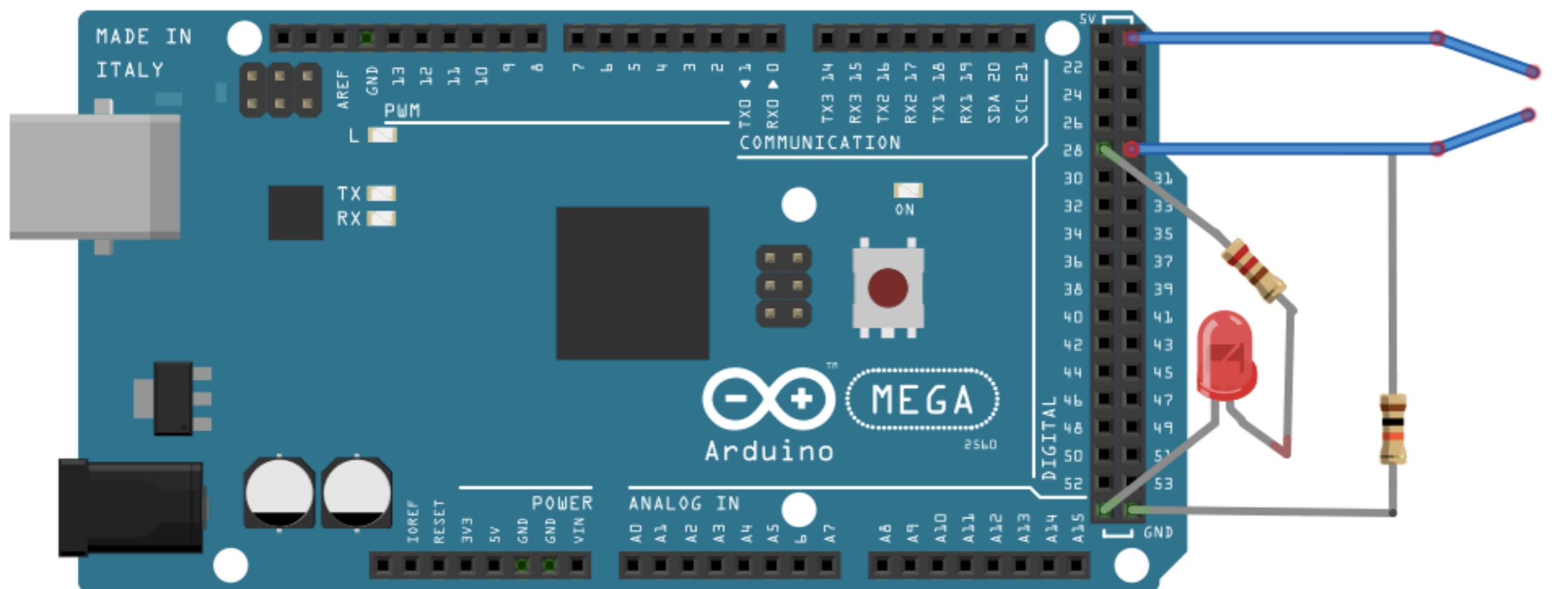


Como funcionará nosso programa...

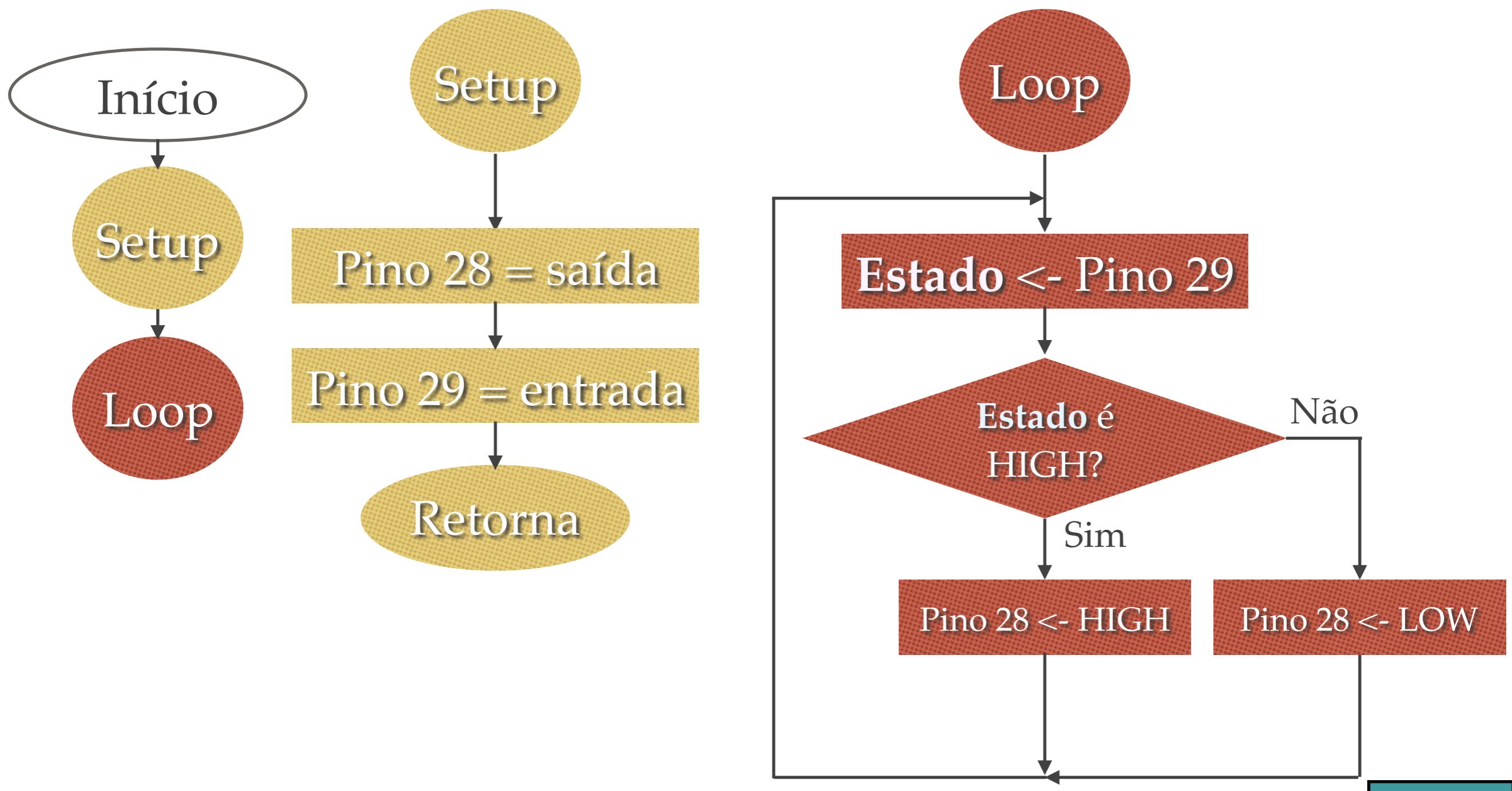
- ❖ Fluxograma:



Primeiro monte um novo hardware



O programa seguirá esse fluxo...



Você consegue elaborar o programa baseado nesse fluxograma?



Resposta

```
/* Programa: Acender LED ao verificar uma entrada
 * Arduino : MEGA 2560
 * Autor   : Rahul M. Juliato
 * Data    : 20/09/2019
 */
int estado_entrada = 0;

void setup(){
  pinMode(28, OUTPUT);
  pinMode(29, INPUT);
}

void loop(){
  estado_entrada = digitalRead(29);

  if (estado_entrada == HIGH){
    digitalWrite(28, HIGH);
  }
  else {
    digitalWrite(28, LOW);
  }
}
```



Vamos tentar o contrário?

- ❖ Com base nesse programa, faça o fluxograma:

```
/* Programa: Acender LED ao verificar uma entrada
 * Arduino : MEGA 2560
 * Autor   : Rahul M. Juliato
 * Data    : 20/09/2019
 */

#define LED 28
#define BOTAO 29

int estado_entrada = 0;

void setup(){
  pinMode(LED, OUTPUT);
  pinMode(BOTAO, INPUT);
}


```

```
void loop(){
  estado_entrada = digitalRead(BOTAO);

  if (estado_entrada == HIGH){
    digitalWrite(LED, HIGH);
  }
  else {
    digitalWrite(LED, LOW);
  }
}
```



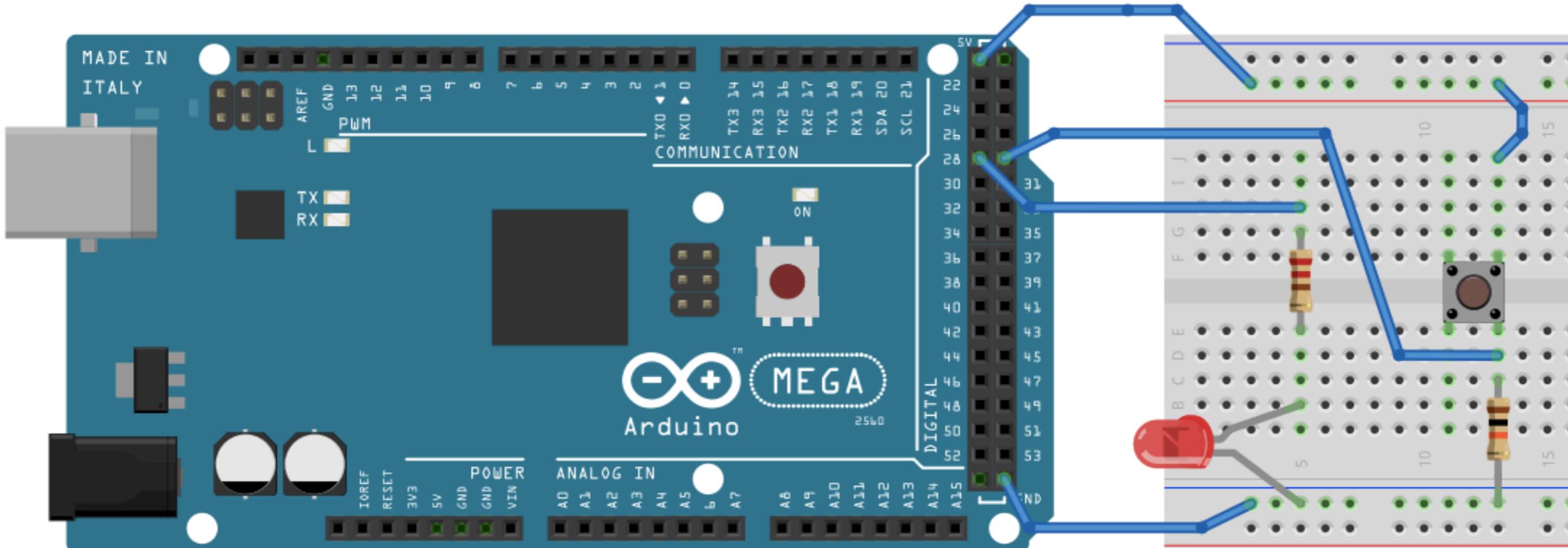
Comparem suas respostas!

- ❖ Pensar em fluxogramas pode ser muito importante em programas de lógicas complicadas. Em cursos de programação é um recurso bastante utilizado.



Antes de treinarmos...

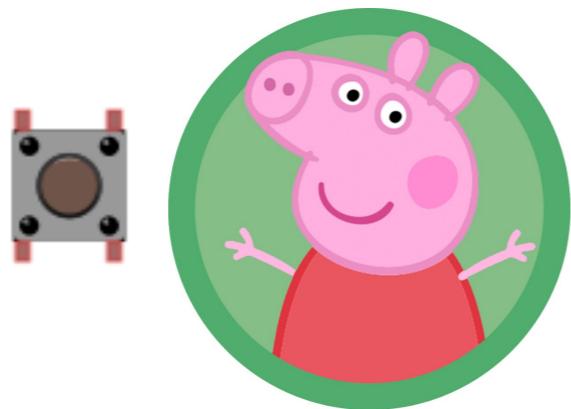
- ❖ Melharemos nossa montagem!



- ❖ Teste sua montagem com o programa anterior



Em outras palavras...



Exercícios

- ❖ Desenhe o hardware para 2 botões e 1 LED;
- ❖ Monte esse hardware;
- ❖ Faça um programa em que se um botão for pressionado. O LED ligue e permaneça aceso, quando o outro botão for pressionado o LED desliga e permanece desligado.



Resposta

```
/* Programa: Liga/Desliga LED com 2 botões
 * Arduino : MEGA 2560
 * Data    : 25/09/2019
 * Autor   : Rahul M. Juliato
 */

#define BOTA01 30
#define BOTA02 29
#define LED 28

int estado1 = 0;      //Declaração + reset da variável
int estado2 = 0;

void setup(){
  pinMode(BOTA01, INPUT);
  pinMode(BOTA02, INPUT);
  pinMode(LED, OUTPUT);
}
```

```
void loop(){
  estado1 = digitalRead(BOTA01);
  if (estado1 == HIGH) {
    digitalWrite(LED, HIGH);
  }

  estado2 = digitalRead(BOTA02);
  if (estado2 == HIGH){
    digitalWrite(LED, LOW);
  }
}
```



Exercícios... continuação

- ❖ O que acontece se pressionarmos os dois botões ao mesmo tempo?
- ❖ Faça um programa que SEMPRE DESLIGUE nessa condição.
- ❖ Faça um programa que SEMPRE LIGUE nessa condição.



Resposta

```
/* Programa: Liga/Desliga LED com 2 botões (Desliga Dominante)
 * Arduino : MEGA 2560
 * Data    : 25/09/2019
 * Autor   : Rahul M. Juliato
 */

#define BOTA01 30
#define BOTA02 29
#define LED 28

int estado1 = 0;      //Declaração + reset da variável
int estado2 = 0;

void setup(){
  pinMode(BOTA01, INPUT);
  pinMode(BOTA02, INPUT);
  pinMode(LED, OUTPUT);
}

void loop(){

  estado1 = digitalRead(BOTA01);
  estado2 = digitalRead(BOTA02);

  if (estado2 == HIGH){
    digitalWrite(LED, LOW);
  }
  else {
    if (estado1 == HIGH){
      digitalWrite(LED, HIGH);
    }
  }
}
```



Resposta

```
/* Programa: Liga/Desliga LED com 2 botões (Liga Dominante)
 * Arduino : MEGA 2560
 * Data    : 25/09/2019
 * Autor   : Rahul M. Juliato
 */

#define BOTA01 30
#define BOTA02 29
#define LED 28

int estado1 = 0;      //Declaração + reset da variável
int estado2 = 0;

void setup(){
  pinMode(BOTA01, INPUT);
  pinMode(BOTA02, INPUT);
  pinMode(LED, OUTPUT);
}

void loop(){

  estado1 = digitalRead(BOTA01);
  estado2 = digitalRead(BOTA02);

  if (estado1 == HIGH){
    digitalWrite(LED, HIGH);
  }
  else {
    if (estado2 == HIGH){
      digitalWrite(LED, LOW);
    }
  }
}
```



Exercício

- ❖ Faça um pisca pisca em que seja possível aumentar e diminuir o tempo entre estados para uma faixa entre 10ms e 100ms, aumentando de 10 em 10ms.



Resposta

```
/* Programa: Pisca-Pisca ajustável
 * Arduino : MEGA 2560
 * Data    : 25/09/2019
 * Autor   : Rahul M. Juliato
 */
#define BOTA01 30
#define BOTA02 29
#define LED 28

int estado1 = 0;
int estado2 = 0;
int tempo = 100;

void setup(){
  pinMode(BOTA01, INPUT);
  pinMode(BOTA02, INPUT);
  pinMode(LED, OUTPUT);
}
```

```
void loop(){

  digitalWrite(LED, HIGH);
  delay(tempo);
  digitalWrite(LED, LOW);
  delay(tempo);

  /* Botão 1 -> acelera o pisca-pisca */
  estado1 = digitalRead(BOTA01);
  if (estado1 == HIGH){
    if (tempo >= 20){
      tempo = tempo - 10;
    }
  }

  /* Botão 2 -> desacelera o pisca-pisca */
  estado2 = digitalRead(BOTA02);
  if (estado2 == HIGH){
    if (tempo < 100){
      tempo = tempo + 10;
    }
  }
}
```



Tente...

- ❖ Aumentar para 500ms de tempo;
- ❖ Passos de 50 em 50ms;
- ❖ O que acontece?
- ❖ Interrupção é a solução... (mas não nesse curso...)

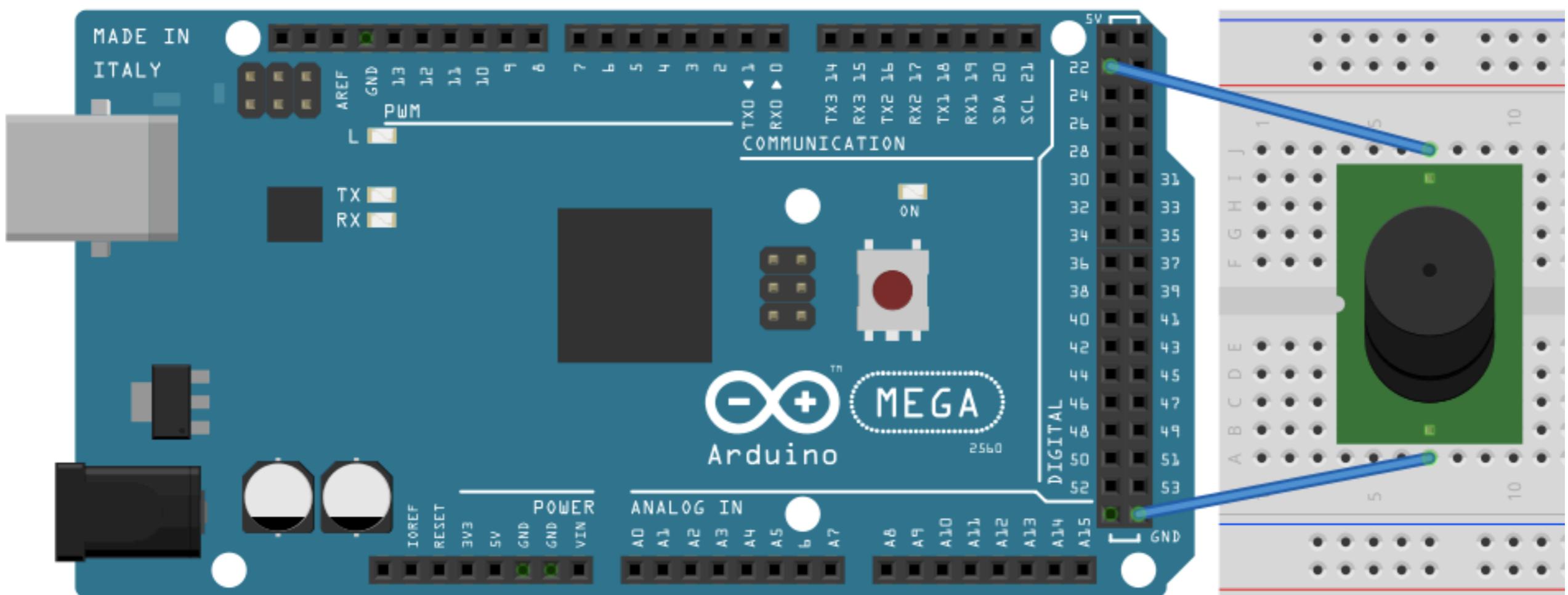


Vamos trabalhar com som

- ❖ Se você quer que o Arduino envie sinais para uma buzina do tipo buzzer, o pino responsável por isso deve ser uma entrada ou saída?



Hardware



Programa

```
/* Programa: Teste de Buzzer
 * Arduino : Mega
 * Autor   : Rahul M. Juliato
 * Data    : 26/09/2019
 */

#define buzzer 22

void setup(){

    pinMode(buzzer, OUTPUT);

}

void loop(){

    tone(buzzer, 1000);
    delay(1000);
    noTone(buzzer);
    delay(1000);

}
```



Exercício

- ❖ Varie o segundo argumento da função tone. O que acontece?
- ❖ Varie o tempo de delay. O que acontece?



Shields

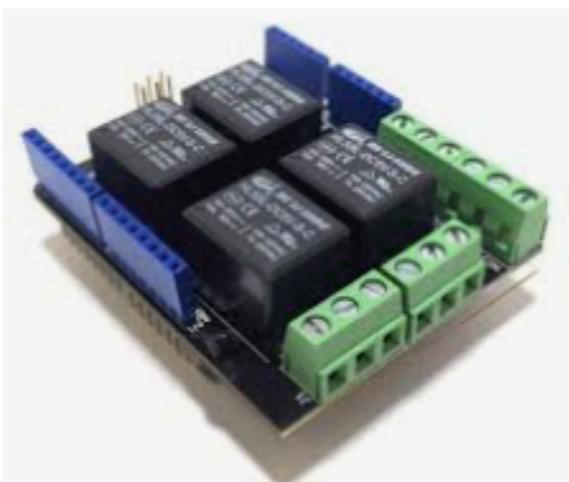
- ❖ Queremos fazer nosso Arduino ligar um pequeno motor que move as rodas de um robô.
- ❖ Como fazer?



Vamos utilizar um Shield de Relé



Arduino A000110 Arduino Quad Relay ...
uk.rs-online.com



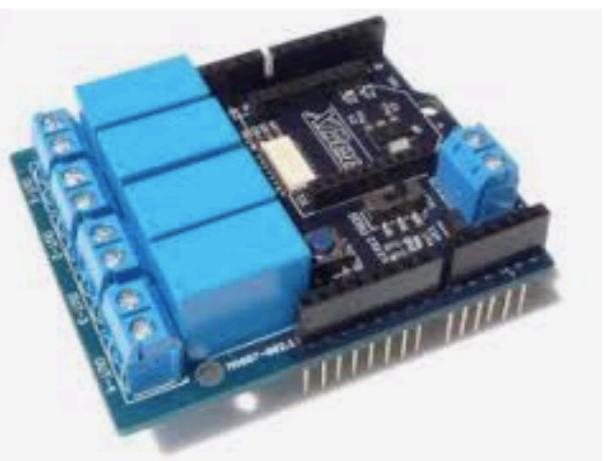
ARDUINO RELAY SHIELD V3.0 SE...
leeselectronic.com



Arduino 4 Relay Shield A000110 ...
ebay.co.uk



[Arduino I/O Shield relay won't switch ...](#)
arduino.stackexchange.com



Relay Shield for Arduino v2
robomaa.fi



4-Channel Relay Controller for Arduino Nano
shop.controleeverything.com

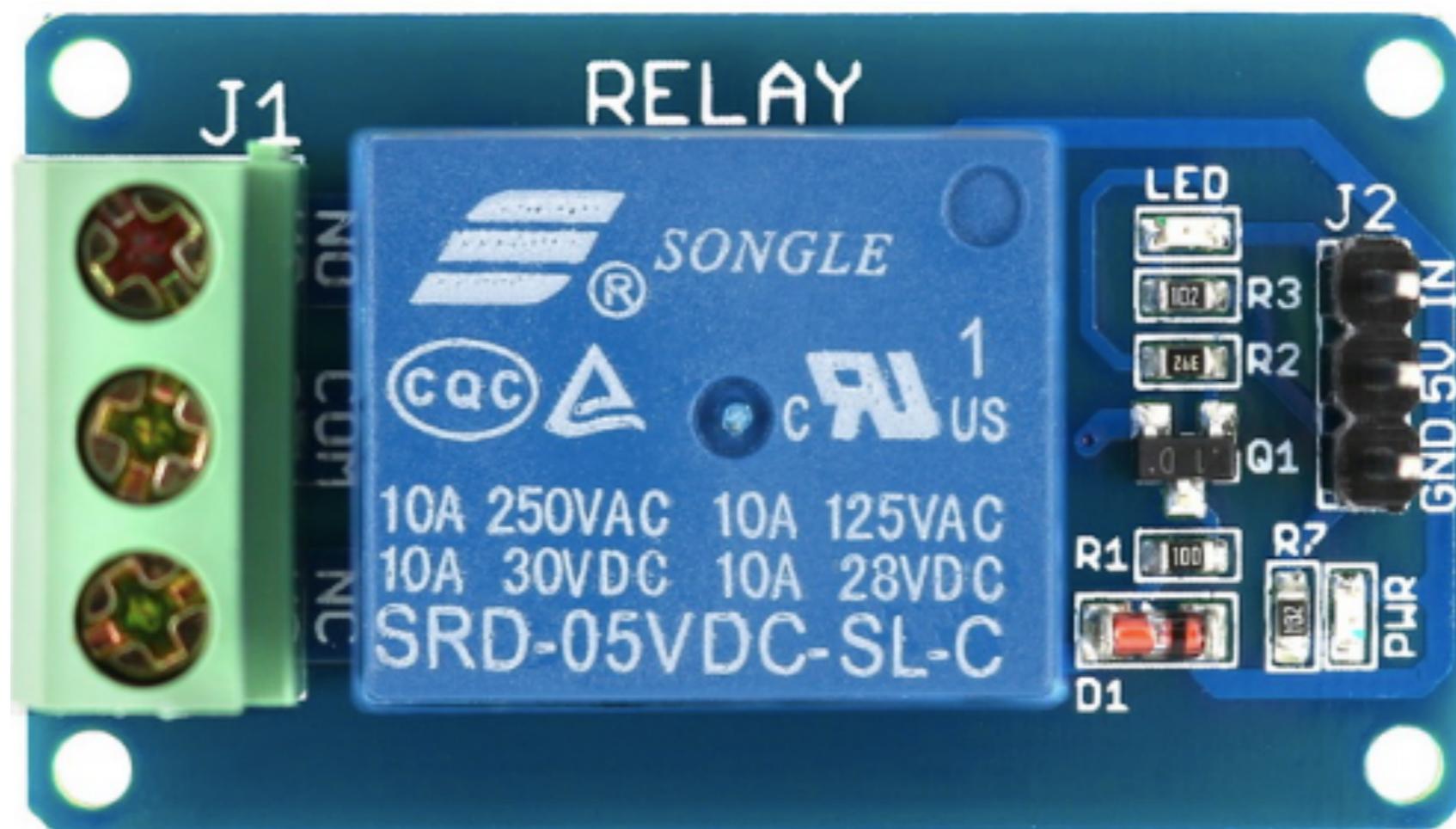


4 channel 5V Relay module exte...
xcluma.com

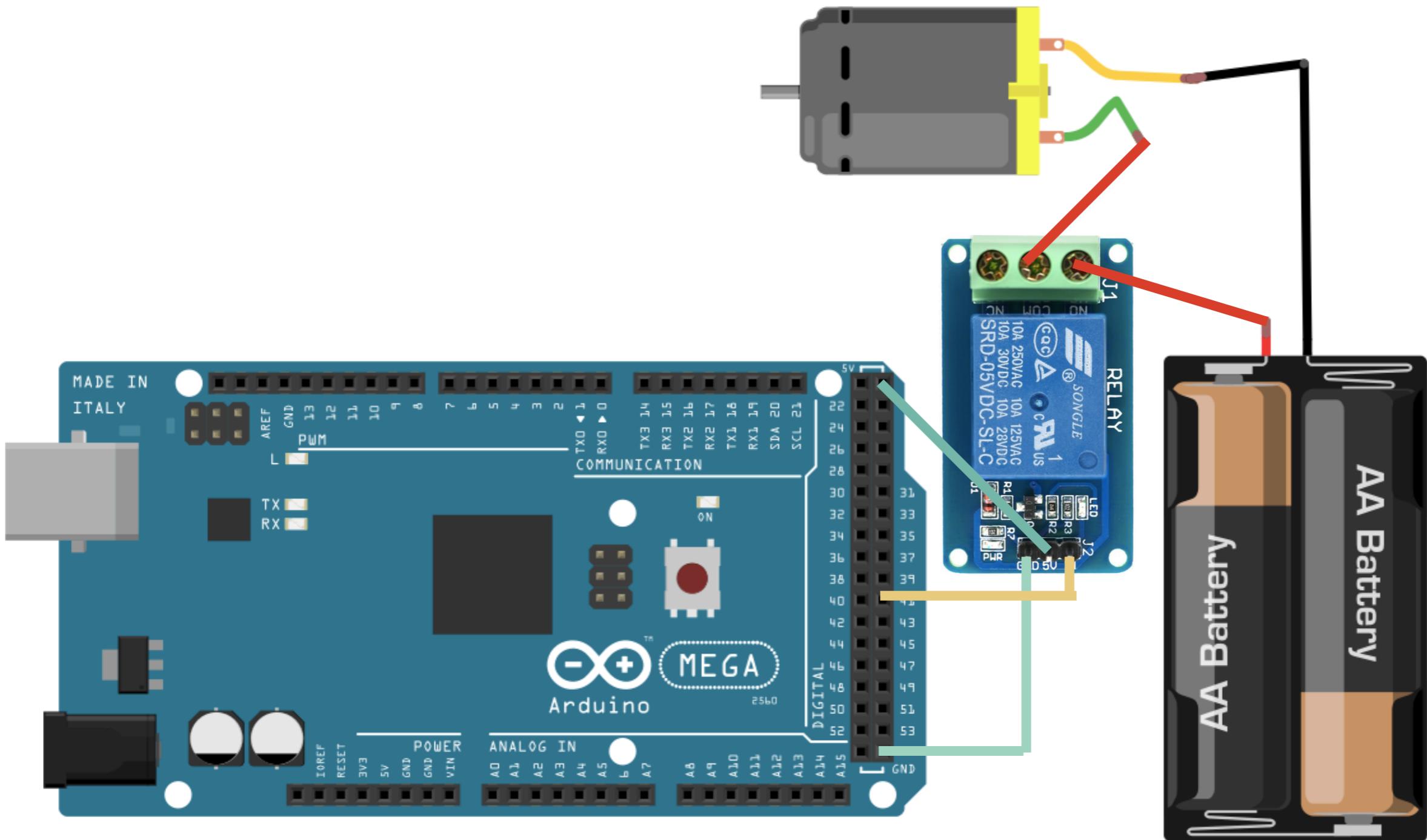


Relay Shield V2.0
seeeddoc.github.io

Começamos Simples



Hardware



Exercício

- ❖ Faça um software que liga o relé, mantém ligado por 5 segundos, desligue o relé por 5 segundos em loop;



Muito simples!

```
/* Programa: Testa o Relé na saída 41
 * Arduino : Mega
 * Shield  : Relé - Hong Wei Low level Trigger - 5V
 * Autor   : Rahul M. Juliato
 * Data    : 27/09/2019
 */

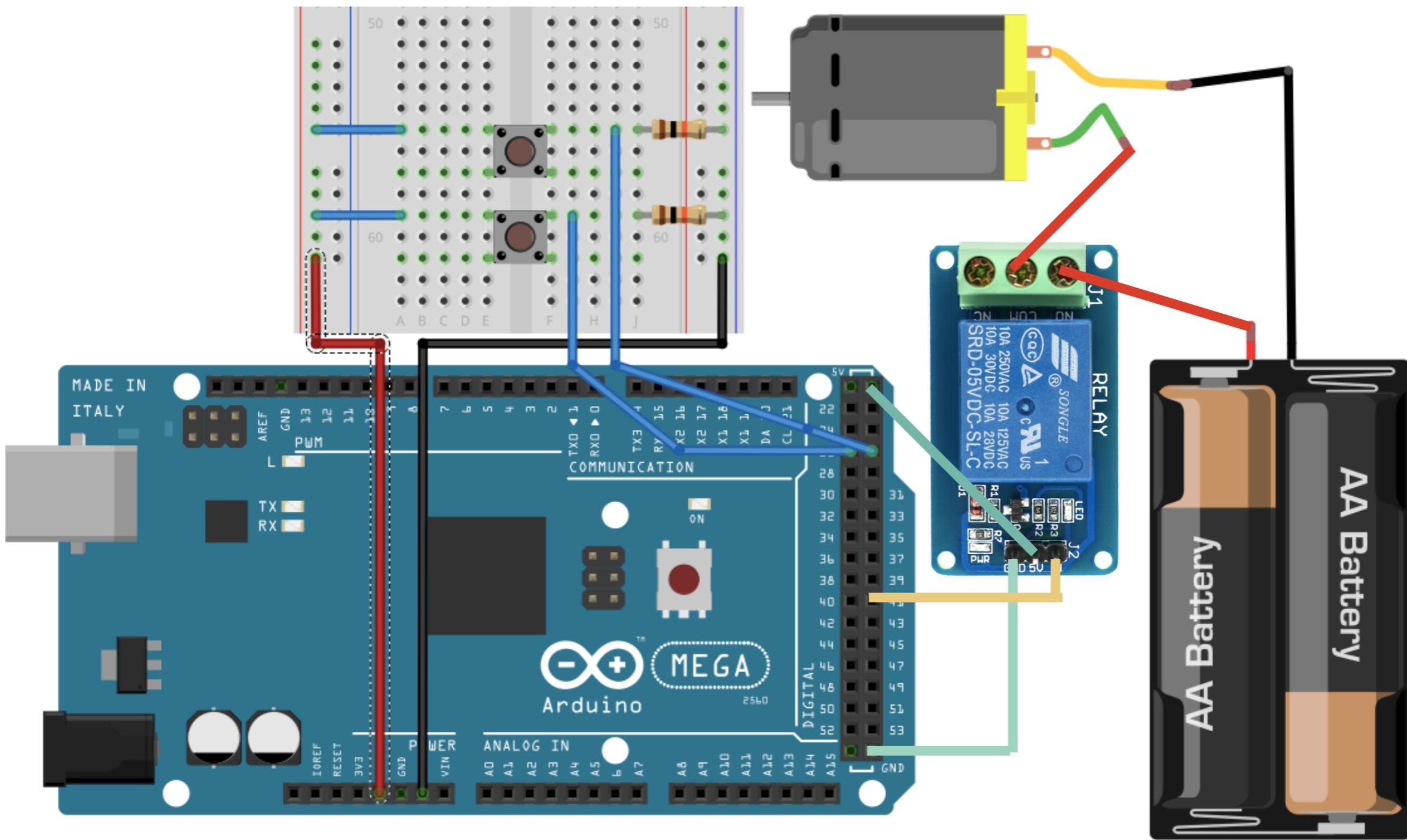
#define RELE 41
#define TEMPO 5000

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

void loop(){
    digitalWrite(RELE, HIGH);
    delay(TEMPO);
    digitalWrite(RELE, LOW);
    delay(TEMPO);
}
```



Mais um Hardware...



Exercício

- ❖ Faça um software que ao pressionar um botão ligue o relé e ao pressionar outro botão desligue o mesmo;
- ❖ As condições ligado e desligado devem ser memorizadas (não há necessidade de permanecer com o botão pressionado);
- ❖ Ao pressionar os dois botões ao mesmo tempo o relé deve permanecer desligado.



Resposta

```
/* Programa: Testa o Relé na saída 41 com botões
 * Arduino : Mega
 * Shield  : Relé - Hong Wei Low level Trigger - 5V
 * Autor   : Rahul M. Juliato
 * Data    : 27/09/2019
 */
#define LIGA 27
#define DESL 26
#define RELE 41

int estado_liga = 0;
int estado_desl = 0;

void setup(){
  pinMode(RELE, OUTPUT);
  pinMode(LIGA, INPUT);
  pinMode(DESL, INPUT);
}

void loop(){
  estado_liga = digitalRead(LIGA);
  estado_desl = digitalRead(DESL);

  if (estado_desliga == HIGH){
    digitalWrite(RELE, LOW);
  }
  else {
    if (estado_liga == HIGH){
      digitalWrite(RELE, HIGH);
    }
  }
}
```



Ok... o eixo do motor gira infinitamente...

- ❖ Mas há alguns casos onde eu preciso de um motor que gire somente um ângulo específico e pare nessa posição.
- ❖ Para esses casos utilizamos servo-motores.



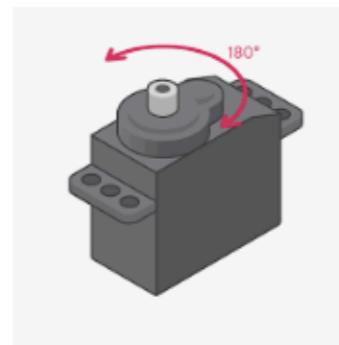
Arduino Compatible 9G Micr...
jaycar.co.nz



Mitsubishi , 50W To 1KW A...
indiamart.com



[NTE-207-LONS-0000 Nidec Control ...](http://variablefrequencydrive.com)
variablefrequencydrive.com



Strawbees | Dream Big, Bui...
strawbees.com



Where to Buy Servo Motor...
electronicsurplus.ca



AC Servo Motor 400W with...
damencnc.com

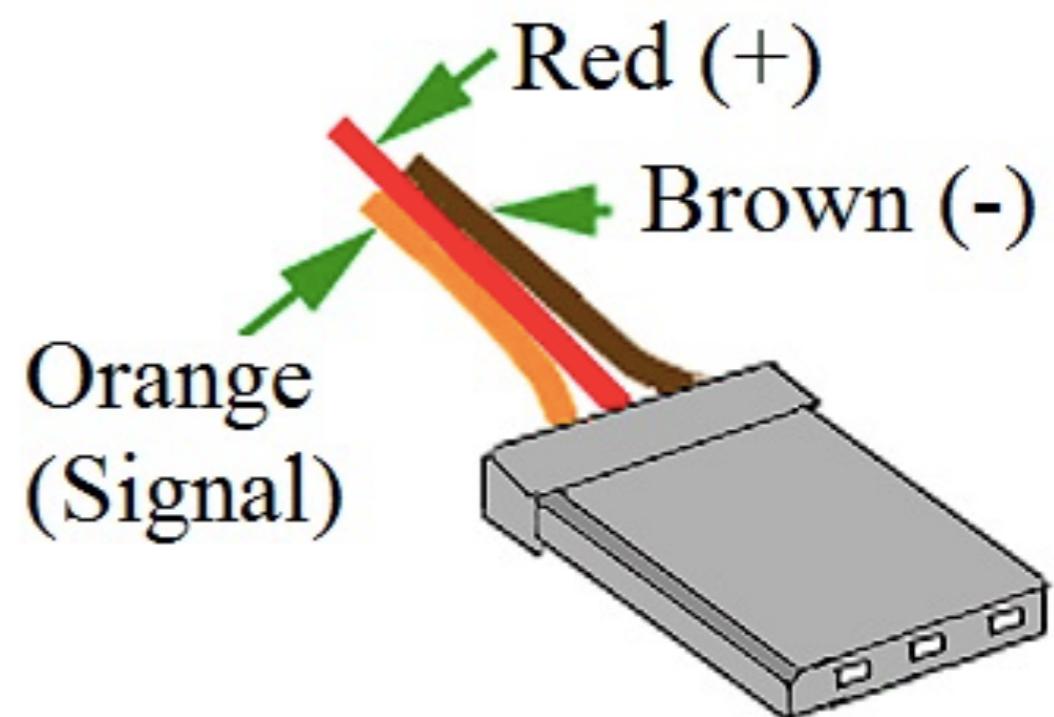


20kg waterproof high torq...
banggood.com

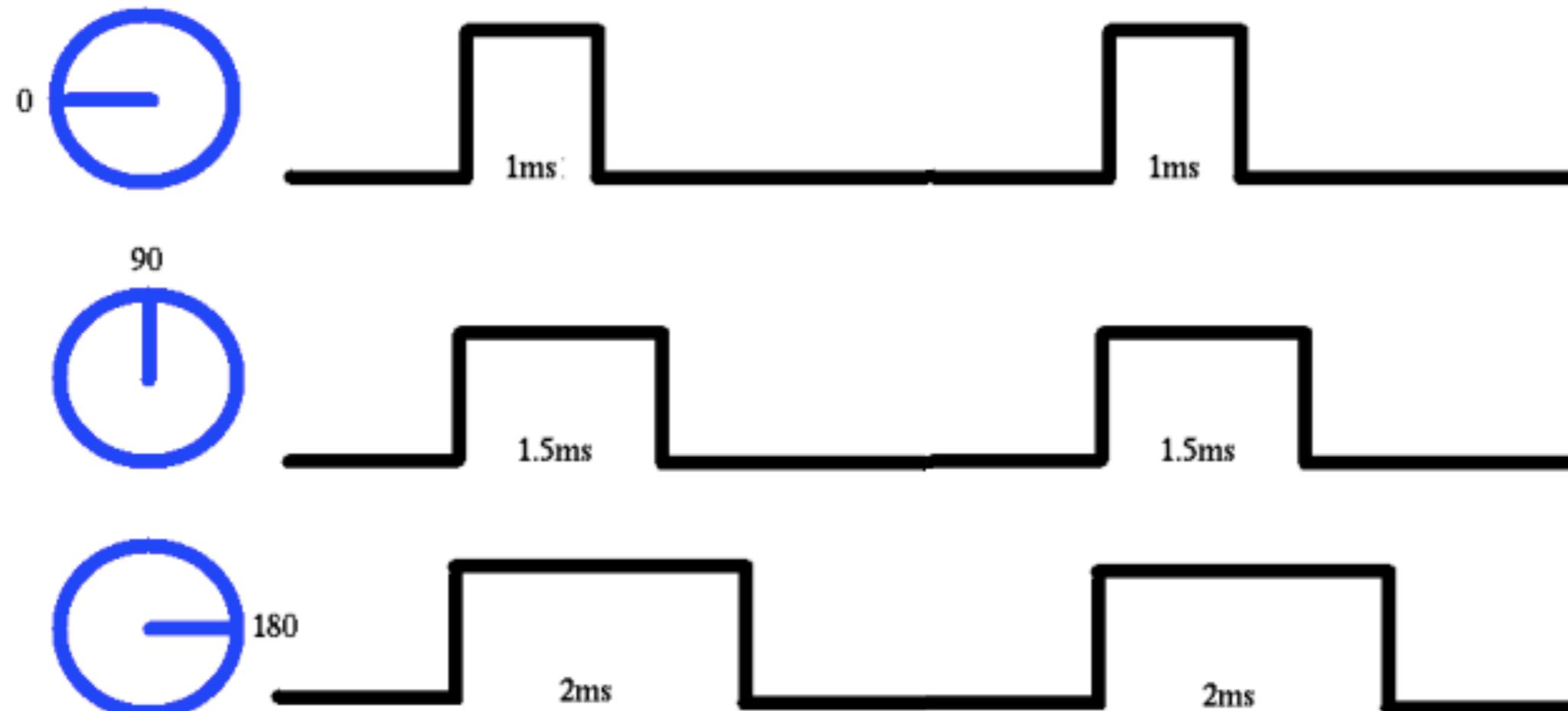


Servo Motor Technology
orientalmotor.com

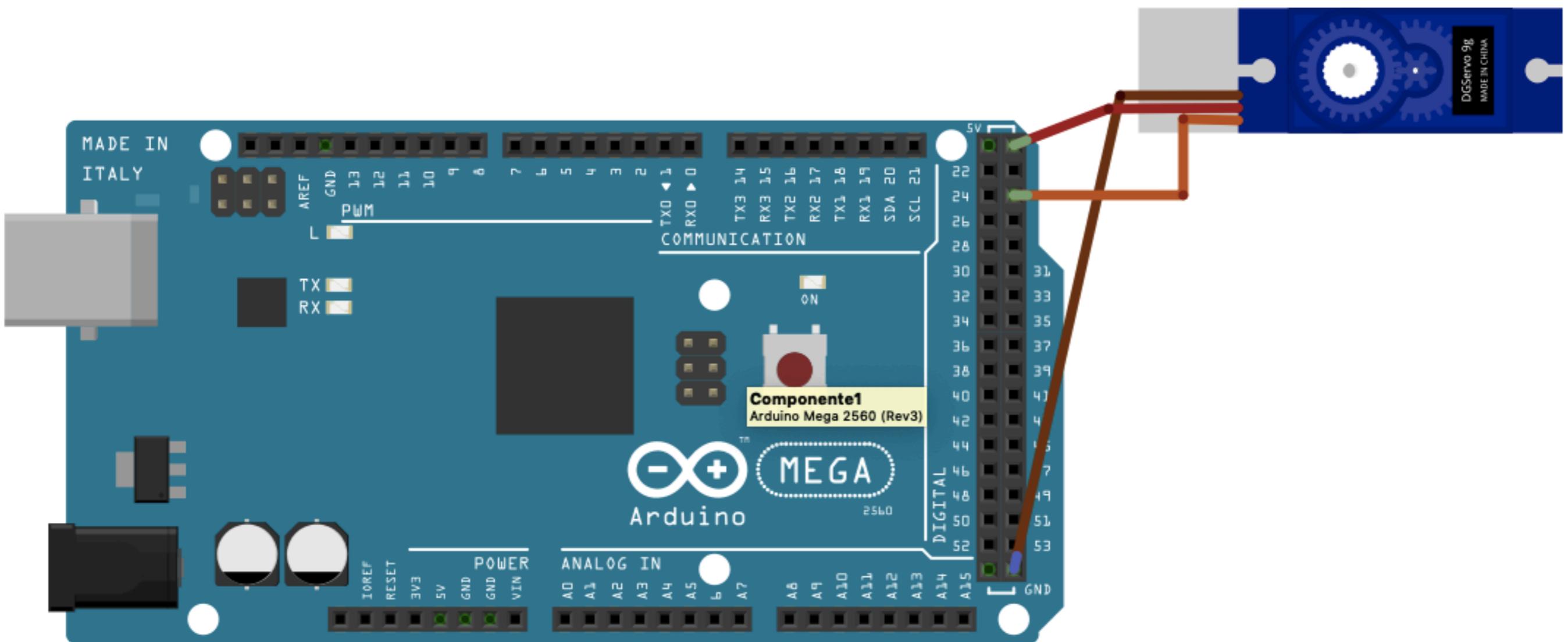
Vamos utilizar...



Como controlá-lo?



Hardware



Software

```
/* Programa: Testa o Servo Motor
 * Arduino : Mega
 * Shield  : Servo 9g
 * Autor   : Rahul M. Juliato
 * Data    : 27/09/2019
 */

#include <Servo.h>
Servo s;

void setup(){
  s.attach(25);
  s.write(10);
}

void loop(){
  s.write(90);
  delay(1000);
  s.write(45);
  delay(1000);
  s.write(15);
  delay(1000);
}
```



Exercício

- ❖ Desenvolva um software que faça o servo motor andar de 1 em 1 grau, começando com 5 graus e terminando em 105, com um tempo de 10ms entre cada mudança de posição.



Ficou assim?

```
/* Programa: Testa o Servo Motor
 * Arduino : Mega
 * Shield  : Servo 9g
 * Autor   : Rahul M. Juliato
 * Data    : 27/09/2019
 */
#include <Servo.h>
Servo s;
int i;
int angulo=5;

void setup(){
  s.attach(25);
  s.write(angulo);
}

void loop(){
  for (i=1;i<=100;i++){
    s.write(angulo);
    angulo = angulo + 1;
    delay(10);
  }
  angulo = 5;
}
```



Dá para usar for com outros programas?

- ❖ Sim!
- ❖ Para todos os programas em que você quer que sequências de comando se repitam!
- ❖ Incluindo os anteriores!
- ❖ (Buzzer de 1000 até 2000?)

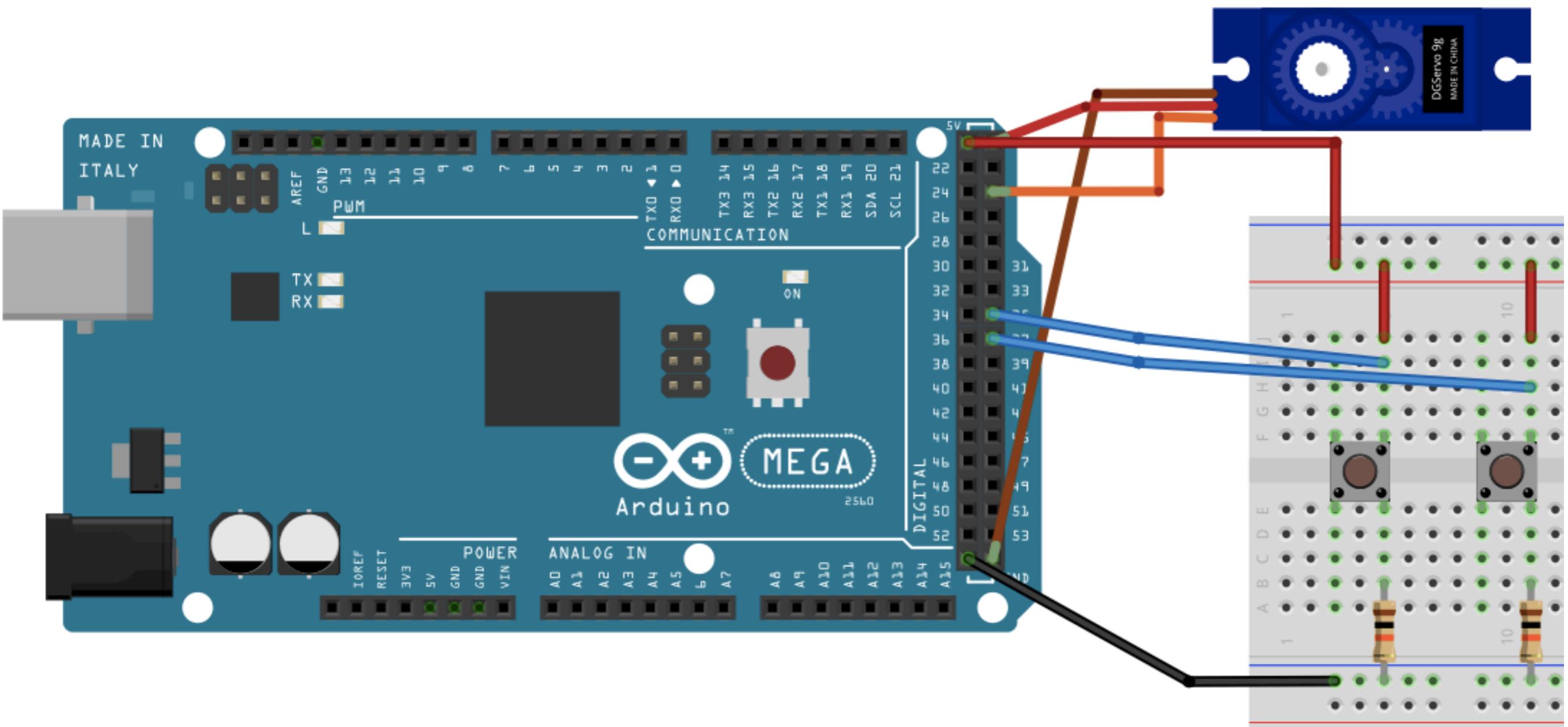


Monte um hardware...

- ❖ Para controlar a posição do servo à partir de dois botões, um para incrementar outro para decrementar o ângulo;



Ficou assim?



Software

- ❖ Agora elabore o software para:
- ❖ Incrementar de 10 em 10 graus;
- ❖ Mínimo de 10 graus;
- ❖ Máximo de 190 graus;
- ❖ Inicia no mínimo e vai até o máximo, repetindo o loop.



Resposta

```
/* Programa: Testa o Servo Motor com botões
 * Arduino : Mega
 * Shield  : Servo 9g
 * Autor   : Rahul M. Juliato
 * Data    : 27/09/2019
 */
#include <Servo.h>
Servo s;

#define MAIS 37
#define MENOS 35

int i;
int angulo = 10;
int estado_mais = 0;
int estado_menos = 0;

void setup(){
  pinMode(MAIS, INPUT);
  pinMode(MENOS, INPUT);
  s.attach(25);
  s.write(angulo);
}

void loop(){
  estado_mais = digitalRead(MAIS);
  estado_menos = digitalRead(MENOS);

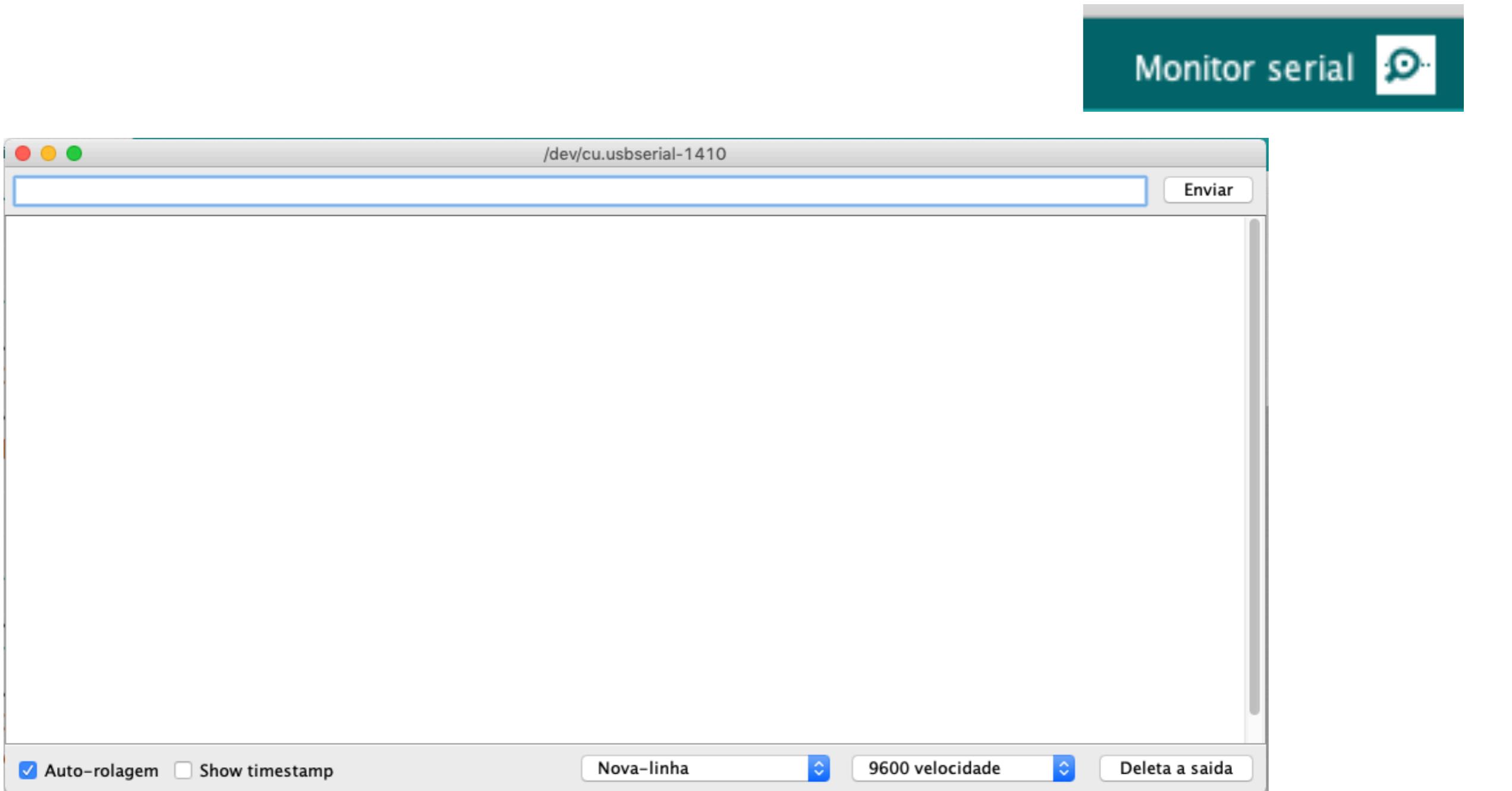
  if (estado_mais == HIGH){
    if (angulo < 190){
      angulo = angulo + 10;
    }
  }

  if (estado_menos == HIGH){
    if (angulo > 10){
      angulo = angulo - 10;
    }
  }

  s.write(angulo);
  delay(10);
}
```



Monitor Serial



Para que serve?

- ❖ Comunicação entre Arduinos
- ❖ Comunicação com o ser humano (através do terminal)
- ❖ Comunicação entre Arduino e um Shield (ex.: Bluetooth, Wi-Fi, etc.)



Vamos fazer o básico juntos!

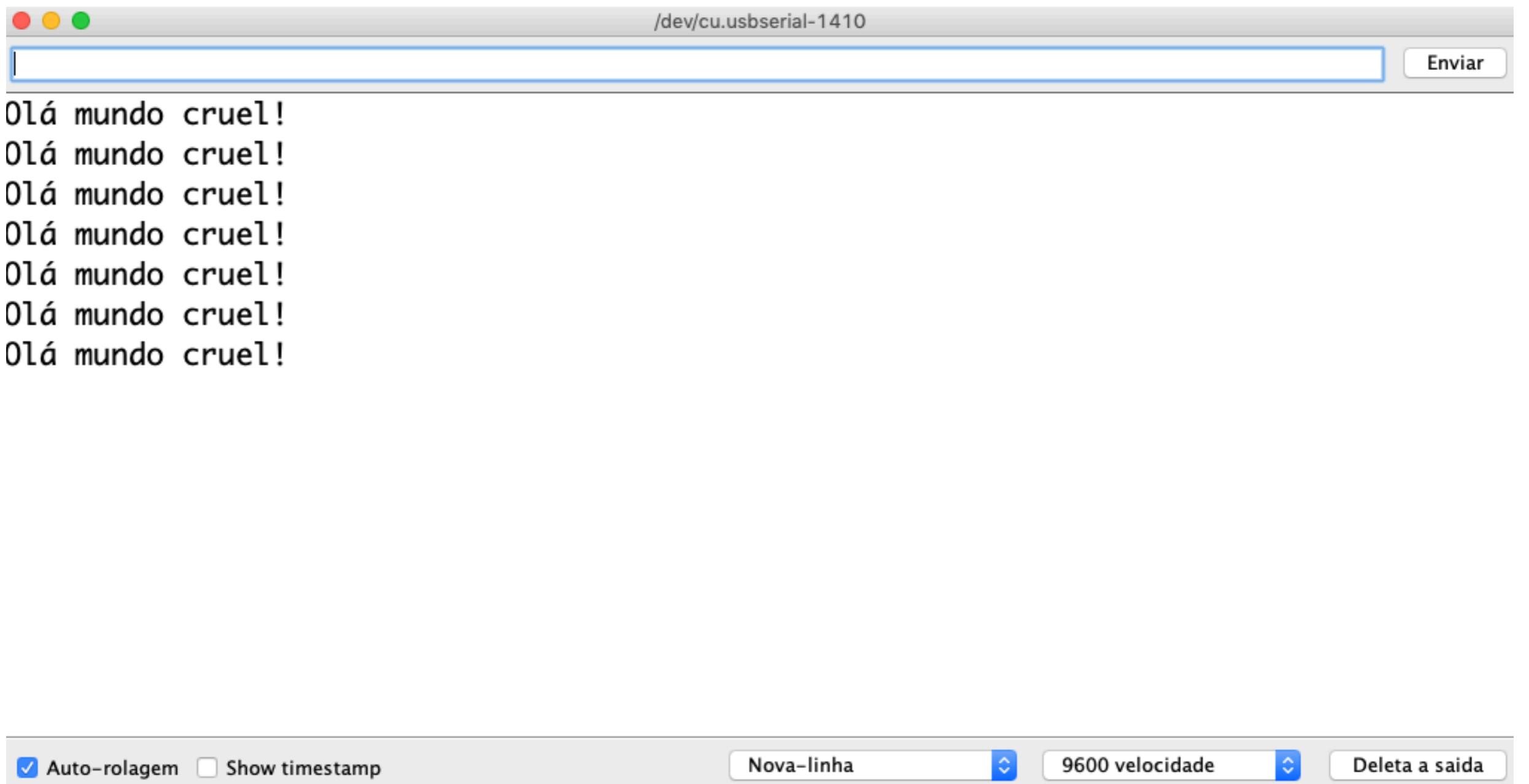
```
/* Programa: Teste de comunicação do Arduino
 * Arduino : MEGA
 * Data    : 30/09/2019
 * Autor   : Rahul M. Juliato
 */

void setup(){
    Serial.begin(9600);
}

void loop(){
    Serial.print("Olá mundo cruel! \n");
    delay(1000);
}
```



Resultado



A screenshot of a terminal window titled "/dev/cu.usbserial-1410". The window contains the following text:

```
Olá mundo cruel!
```

The terminal has the following settings at the bottom:

- Auto-rolagem (checked)
- Show timestamp (unchecked)
- Nova-linha (Newline)
- 9600 velocidade (9600 baud)
- Deleta a saída (Delete output)



Mostrando o valor de uma variável...

```
/* Programa: Teste de comunicação do Arduino
 * Arduino : MEGA
 * Data    : 30/09/2019
 * Autor   : Rahul M. Juliato
 */

int tempo = 129;

void setup(){
    Serial.begin(9600);
}

void loop(){
    Serial.print("O valor de tempo é: ");
    Serial.print(tempo);
    Serial.print("\n\n");

    delay(1000);
}
```



Altere o valor de tempo para 3.1415

- ❖ O que ocorre no monitor serial?



Exemplos

`Serial.print(78)` gives "78"

`Serial.print(1.23456)` gives "1.23"

`Serial.print('N')` gives "N"

`Serial.print("Hello world.")` gives "Hello world."

`Serial.print(78, BIN)` gives "1001110"

`Serial.print(78, OCT)` gives "116"

`Serial.print(78, DEC)` gives "78"

`Serial.print(78, HEX)` gives "4E"

`Serial.print(1.23456, 0)` gives "1"

`Serial.print(1.23456, 2)` gives "1.23"

`Serial.print(1.23456, 4)` gives "1.2346"



Nossa variável não pode ser int!

```
/* Programa: Teste de comunicação do Arduino
 * Arduino : MEGA
 * Data    : 30/09/2019
 * Autor   : Rahul M. Juliato
 */
float tempo = 3.1415;

void setup(){
    Serial.begin(9600);
}

void loop(){
    Serial.print("O valor de tempo é: ");
    Serial.print(tempo, 4);
    Serial.print("\n\n");
    delay(1000);
}
```



E se a variável for texto?

```
/* Programa: Teste de comunicação do Arduino
 * Arduino : MEGA
 * Data    : 30/09/2019
 * Autor   : Rahul M. Juliato
 */
float tempo = 3.1415;
char texto[] = "Coisinha mahr linda de meu Deus";

void setup(){
    Serial.begin(9600);
}

void loop(){
    Serial.print("O valor de tempo é: ");
    Serial.print(tempo, 4);
    Serial.print("\n");
    Serial.print("E o valor de texto é: ");
    Serial.print(texto);
    Serial.print("\n \n");
    delay(1000);
}
```



Enxergando o for funcionando

```
/* Programa: Teste de comunicação do Arduino
 * Arduino : MEGA
 * Data    : 30/09/2019
 * Autor   : Rahul M. Juliato
 */
int i;
void setup(){
    Serial.begin(9600);
}
void loop(){
    for(i=1;i<=50;i++){
        Serial.print("O valor de i é: ");
        Serial.print(i);
        Serial.print("\n");
        delay(250);
    }
}
```



Pisca-Pisca com “Report”

```
/* Programa: Pisca-Pisca com supervisão
 * Arduino : MEGA
 * Data    : 30/09/2019
 * Autor   : Rahul M. Juliato
 */
int quantidade=0;

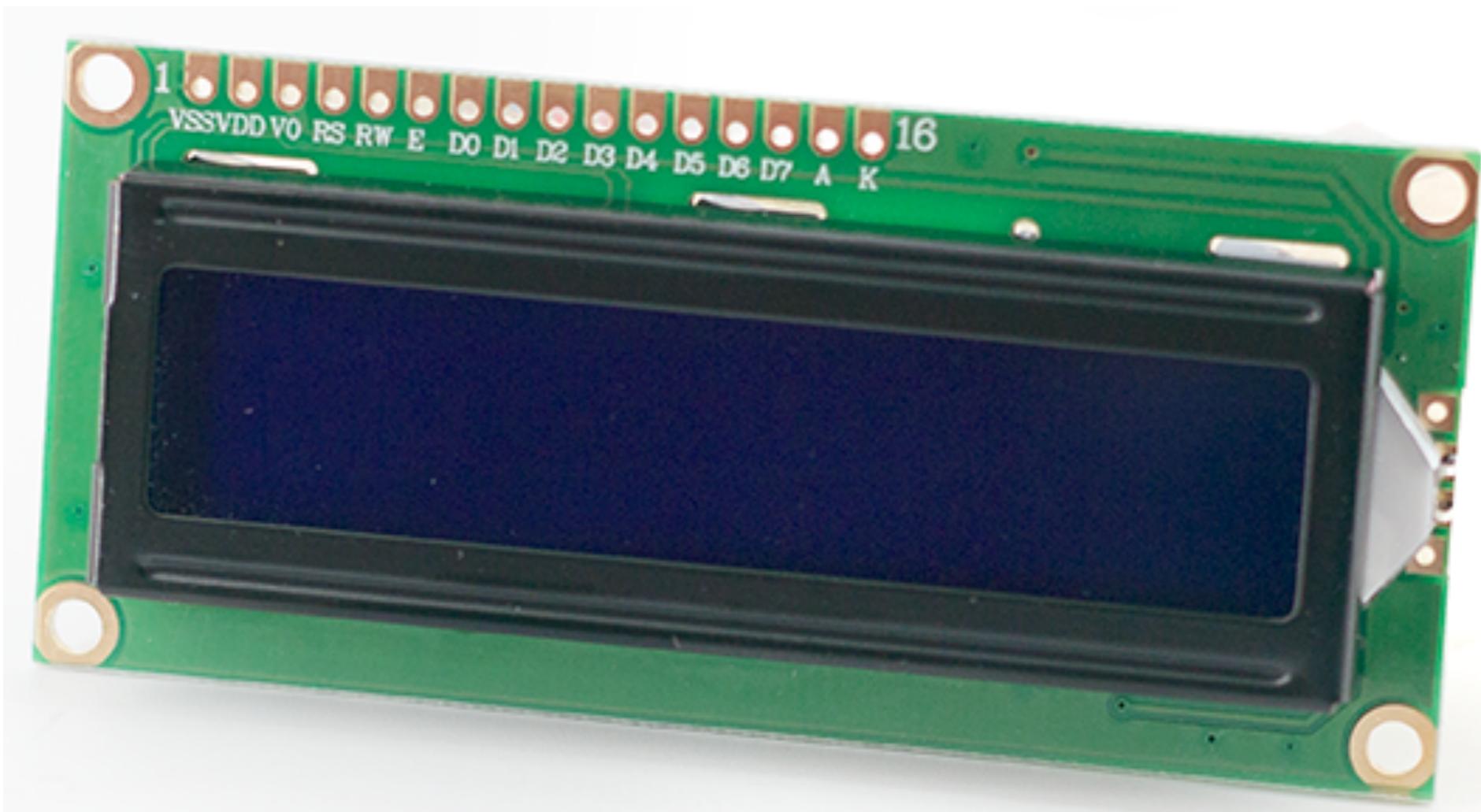
void setup(){
    pinMode(13, OUTPUT);
    Serial.begin(9600);
}

void loop(){
    Serial.print("O pisca-pisca já piscou: ");
    Serial.print(quantidade);
    Serial.print(" vezes \n");
    quantidade = quantidade + 1;
    digitalWrite(13, HIGH);
    delay(100);
    digitalWrite(13, LOW);
    delay(100);
}
```



Melhorando...

- ❖ Que tal colocar um display para mostrar informações ao usuário diretamente pelo Arduino?
- ❖ LCD 16x2 HITACHI (HD44780)



Pinagem do LCD

Pino LCD	Função
1	Vss
2	Vdd
3	V0
4	RS
5	RW
6	E
7	D0
8	D1
9	D2
10	D3
11	D4
12	D5
13	D6
14	D7
15	A
16	K

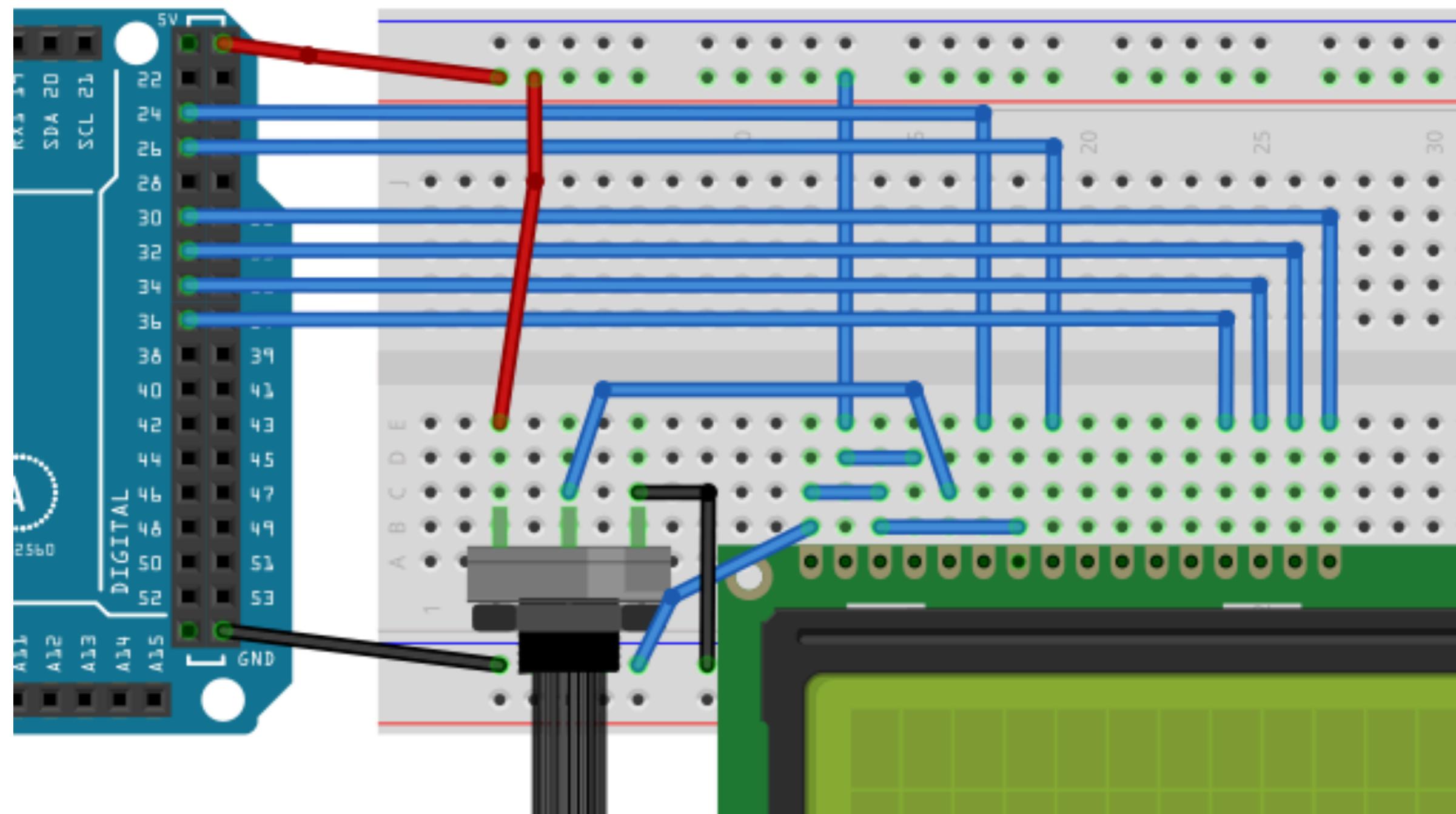


Hardware

- ❖ Vamos fazer juntos no Fritzing!



Ficou assim?



Programa de Teste

```
/* Programa: Teste do Display de LCD
 * Arduino : MEGA
 * Data    : 30/09/2019
 * Autor   : Rahul M. Juliato
 */

#include <LiquidCrystal.h>

const int rs = 34, en = 35, d4 = 33, d5 = 32, d6 = 31, d7 = 30;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  lcd.begin(16, 2);
  lcd.print("Olá mundo!");
}

void loop() {
  lcd.noDisplay();
  delay(500);
  lcd.display();
  delay(500);
}
```



Manipulando o LCD

```
/* Programa: Teste do Display de LCD
 * Arduino : MEGA
 * Data    : 30/09/2019
 * Autor   : Rahul M. Juliato
 */

#include <LiquidCrystal.h>

const int rs = 24, en = 26, d4 = 36, d5 = 34, d6 = 32, d7 = 30;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
void setup(){
    lcd.begin(16, 2);
    lcd.display();
}

void loop() {
    lcd.setCursor(0, 0);
    lcd.print("SENAI - Arduino");
    lcd.setCursor(3, 1);
    lcd.print("!Funciona!");

    delay(200);
}
```



Cursor Piscante

```
/* Programa: Teste do Display de LCD - Cursor
 * Arduino : MEGA
 * Data    : 30/09/2019
 * Autor   : Rahul M. Juliato
 */

#include <LiquidCrystal.h>

const int rs = 24, en = 26, d4 = 36, d5 = 34, d6 = 32, d7 = 30;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
void setup(){

lcd.begin(16, 2);
lcd.display();

}

void loop() {

lcd.setCursor(0, 0);
lcd.print("Cursor -> ");
lcd.noBlink();
delay(3000);
lcd.blink();
delay(3000);

}
```



Scrolling

```
/* Programa: Teste do Display de LCD - Scroll
 * Arduino : MEGA
 * Data    : 30/09/2019
 * Autor   : Rahul M. Juliato
 */
#include <LiquidCrystal.h>
int i;
const int rs = 24, en = 26, d4 = 36, d5 = 34, d6 = 32, d7 = 30;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
void setup(){
    lcd.begin(16, 2);
    lcd.display();
    lcd.print("Viva o Arduino!");
}
```

```
void loop() {
    for(i=1;i<=16;i++){
        lcd.scrollDisplayLeft();
        delay(150);
    }
    for(i=1;i<=32;i++){
        lcd.scrollDisplayRight();
        delay(150);
    }
    for(i=1;i<=16;i++){
        lcd.scrollDisplayLeft();
        delay(150);
    }
}
```



Valores analógicos

- ❖ Até o momento, nossos valores foram todos digitais, ou seja, HIGH ou LOW;
- ❖ Esses valores representam 5V (HIGH) ou 0V (LOW) e são muito úteis para chaves e cargas elétricas que funcionam simplesmente com funções de liga/desliga;
- ❖ Suponha que temos um sensor de temperatura. Esse sensor não entrega 5V ou 0V, mas uma infinidade de valores intermediários entre 0 e 5V que representam uma temperatura em graus Celsius.

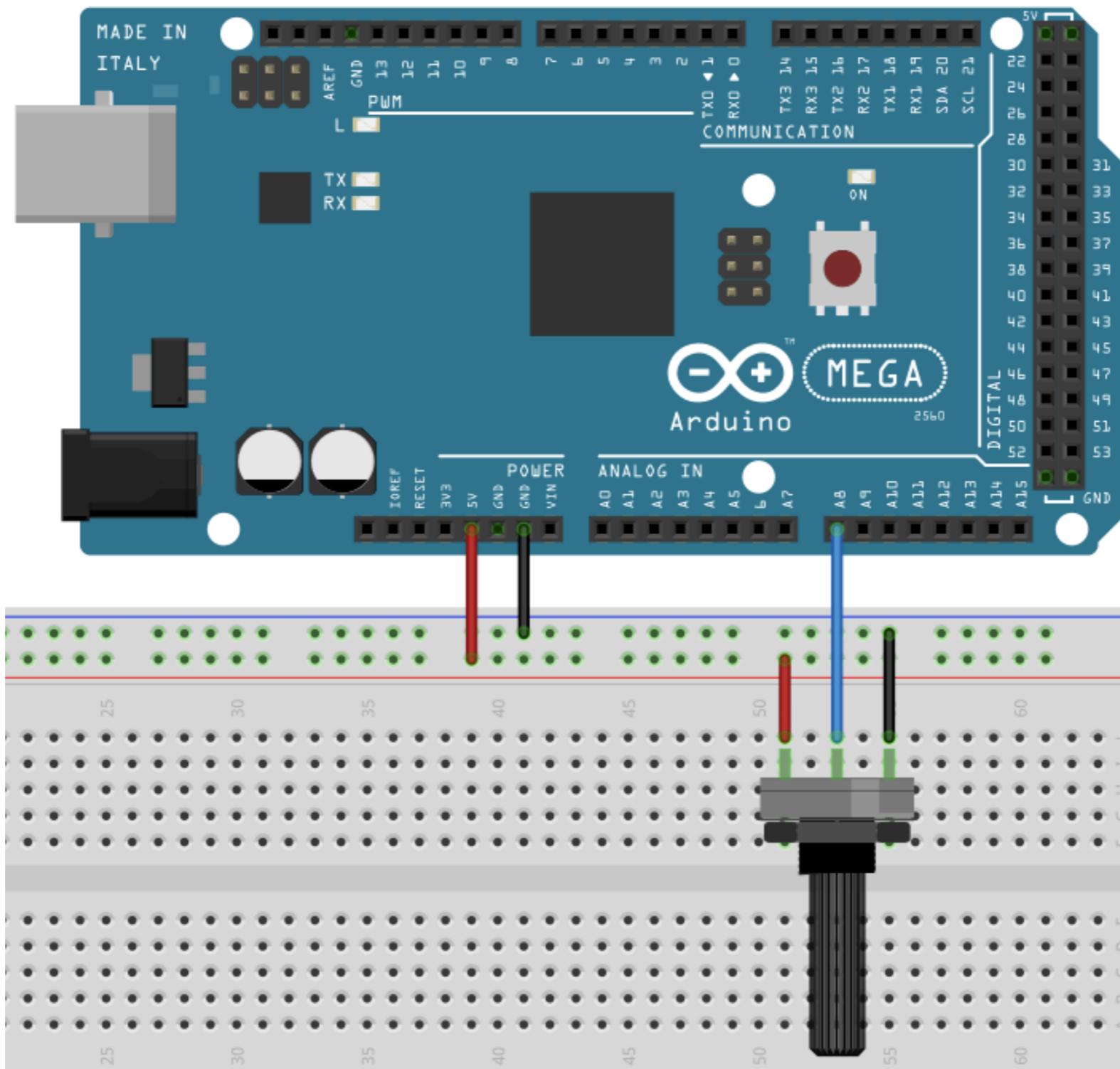


Leitura de valores analógicos

- ❖ Para iniciarmos a leitura de valores analógicos, precisamos utilizar um pino com essa função. No MEGA são os pinos de A0 até A15;
- ❖ Iniciemos com a montagem do hardware básico;

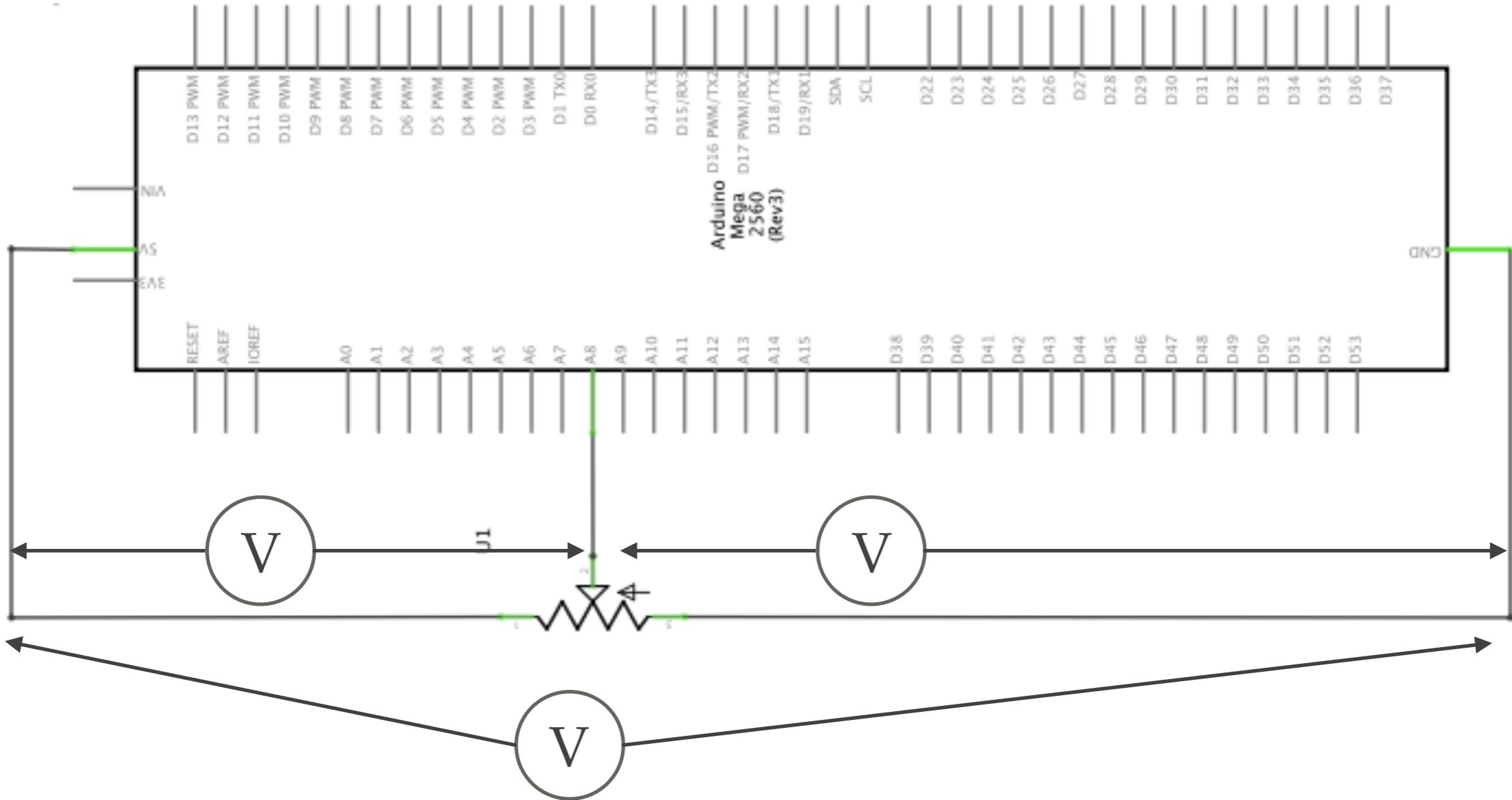


Hardware



Electricamente

❖ Circuito divisor de tensão:



Software

```
/* Programa: Teste da Entrada Analógica
 * Arduino : MEGA
 * Data    : 30/09/2019
 * Autor   : Rahul M. Juliato
 */

int leitura;

void setup(){
    Serial.begin(9600);
}

void loop() {
    leitura = analogRead(A8);
    Serial.println(leitura);
}
```



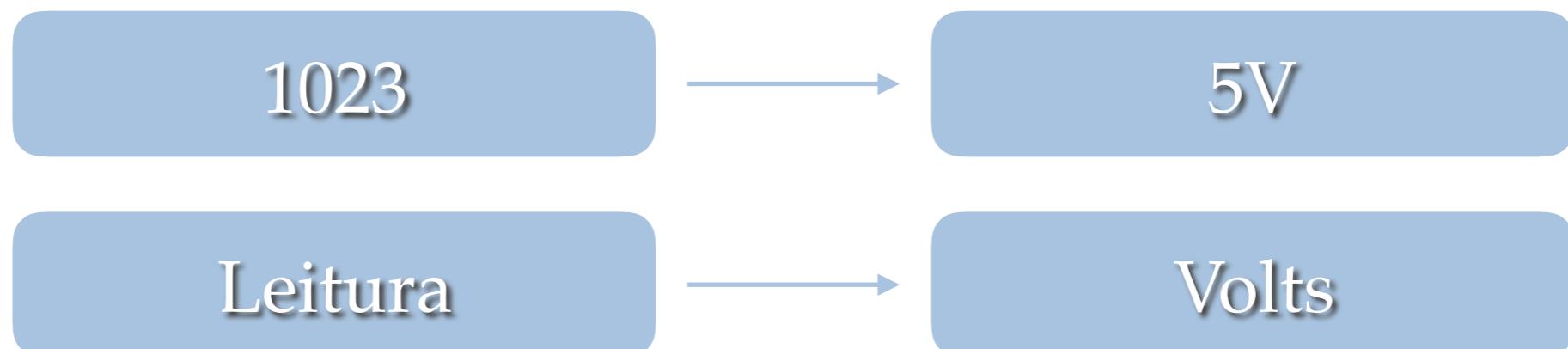
Quais números são esses?

- ❖ Nosso conversor analógico para digital converte o sinal dividindo 0-5V em 1024 partes, ou seja:
- ❖ $0V = 0$
- ❖ $5V = 1023$
- ❖ Podemos dizer que temos uma resolução de 10 bits de conversão.



E se eu quiser que o Arduino me mostre a tensão?

- ❖ Utilizamos uma “regra de três”:



- ❖ Multiplicando em cruz e isolando “Volts”, temos:
- ❖ $\text{Volts} = (\text{Leitura} \times 5) / 1023$
- ❖ Repare que o Valor de “Volts” pode ser quebrado, e portanto, será do tipo float

Programa

```
/* Programa: Teste da Entrada Analógica - Converte em Volts
 * Arduino : MEGA
 * Data    : 30/09/2019
 * Autor   : Rahul M. Juliato
 */
int leitura;
float volts;

void setup(){
  Serial.begin(9600);
}

void loop() {
  leitura = analogRead(A8);
  volts = (5 * leitura) / 1023;
  Serial.print(leitura);
  Serial.print(" = ");
  Serial.println(volts);

  delay(100);
}
```



Casas decimais...

```
void loop() {  
  
    leitura = analogRead(A8);  
    volts = (float)(5 * leitura) / 1023;  
    Serial.print(leitura);  
    Serial.print(" = ");  
    Serial.println(volts);  
  
    delay(100);  
  
}
```



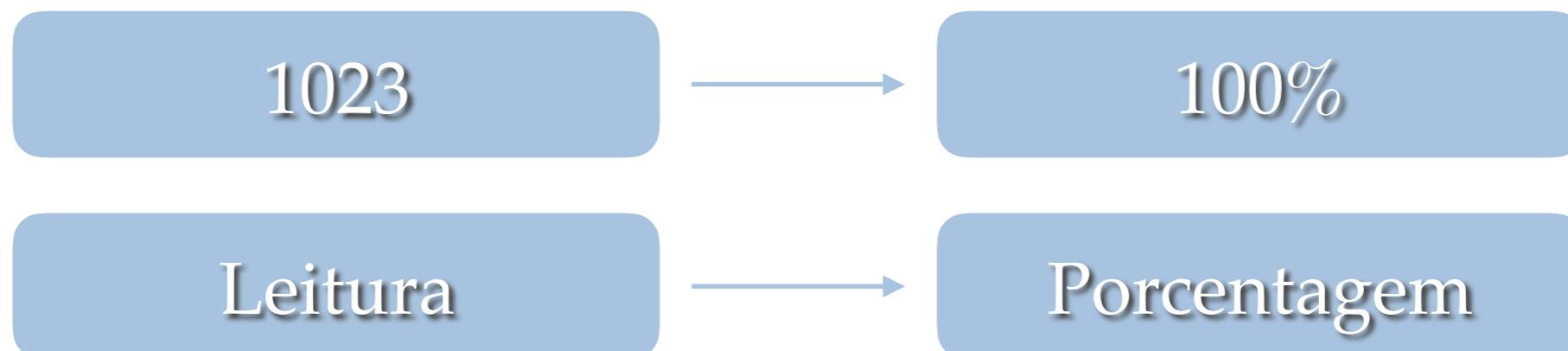
Que tal...

- ❖ Fazer com que o valor mostrado fique entre 0% e 100%?
- ❖ Faça a regra de três e o programa.



Porcentagem

- ❖ Utilizamos uma “regra de três”:



- ❖ Multiplicando em cruz e isolando “Porcentagem”, temos:
- ❖ $\text{Porcentagem} = (\text{Leitura} \times 100) / 1023$
- ❖ Repare que o Valor de “Porcentagem” pode ser quebrado, e portanto, será do tipo float

Programa

```
/* Programa: Teste da Entrada Analógica - Converte em %
 * Arduino : MEGA
 * Data    : 30/09/2019
 * Autor   : Rahul M. Juliato
 */
int leitura;
float porcentagem;
void setup(){
    Serial.begin(9600);
}
void loop() {
    leitura = analogRead(A8);
    porcentagem = (float)(100 * leitura) / 1023;
    Serial.print(leitura);
    Serial.print(" = ");
    Serial.println(porcentagem);
    delay(100);
}
```



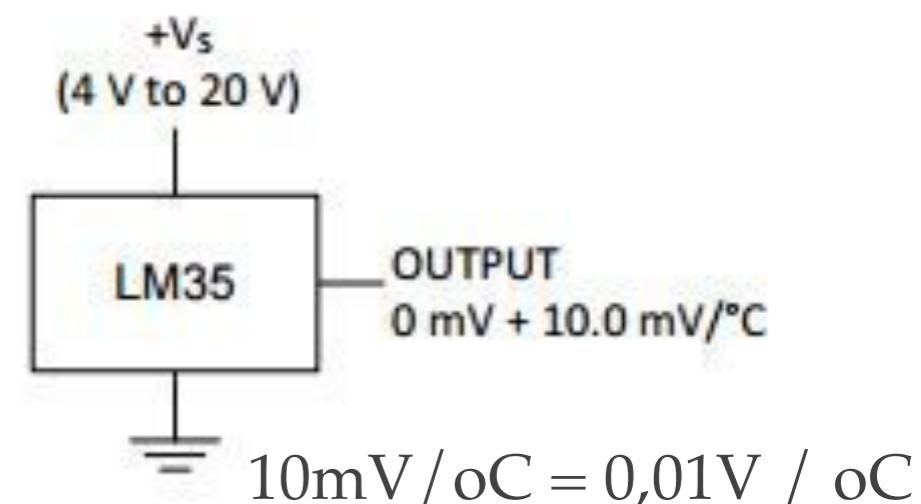
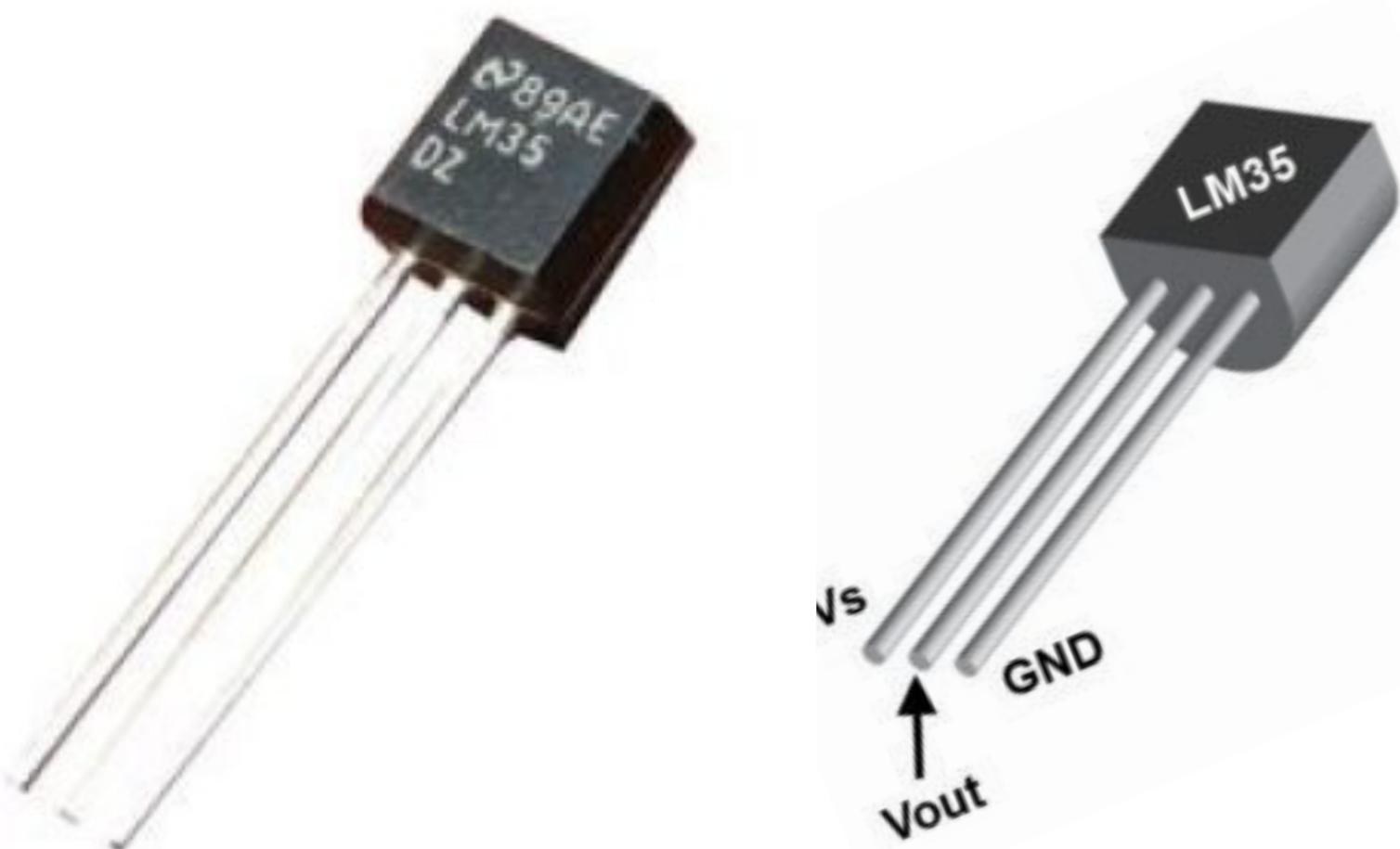
Fix

```
void loop() {  
  
    leitura = analogRead(A8);  
    porcentagem = (100.0 * leitura) / 1023.0;  
    Serial.print(leitura);  
    Serial.print(" = ");  
    Serial.println(porcentagem);  
  
    delay(100);  
  
}
```

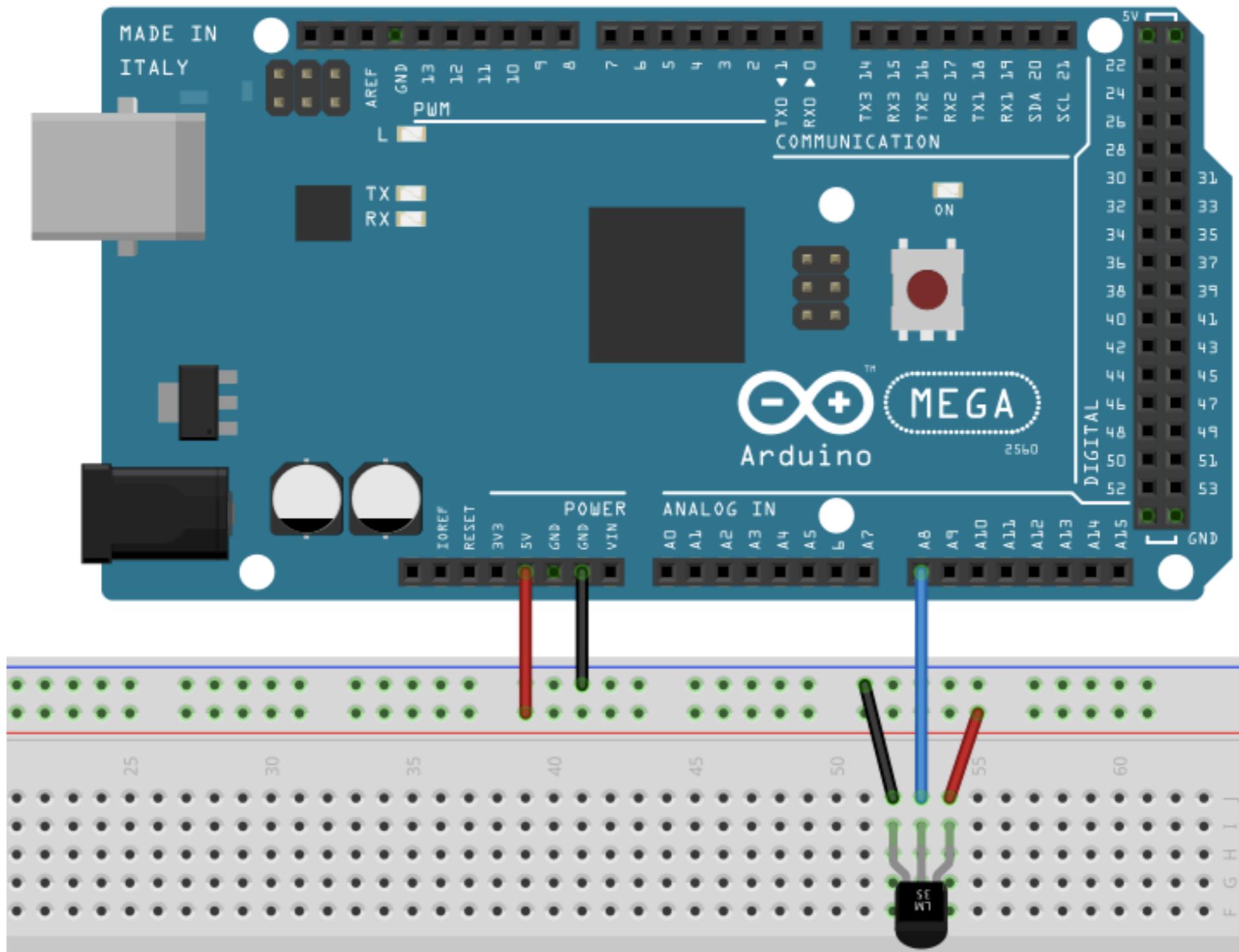


Medidor de Temperatura

- ❖ Utilizamos o circuito integrado LM 35 como sensor de temperatura:



Hardware



Software

```
/* Programa: Medidor de Temperatura
 * Arduino : MEGA
 * Data    : 30/09/2019
 * Autor   : Rahul M. Juliato
 */

int leitura;
float temperatura;

void setup(){
    Serial.begin(9600);
}

void loop() {
    leitura = analogRead(A8);
    temperatura = ((5.0 * leitura) / 1023.0 * 100.0);
    Serial.println(temperatura);
    delay(500);
}
```



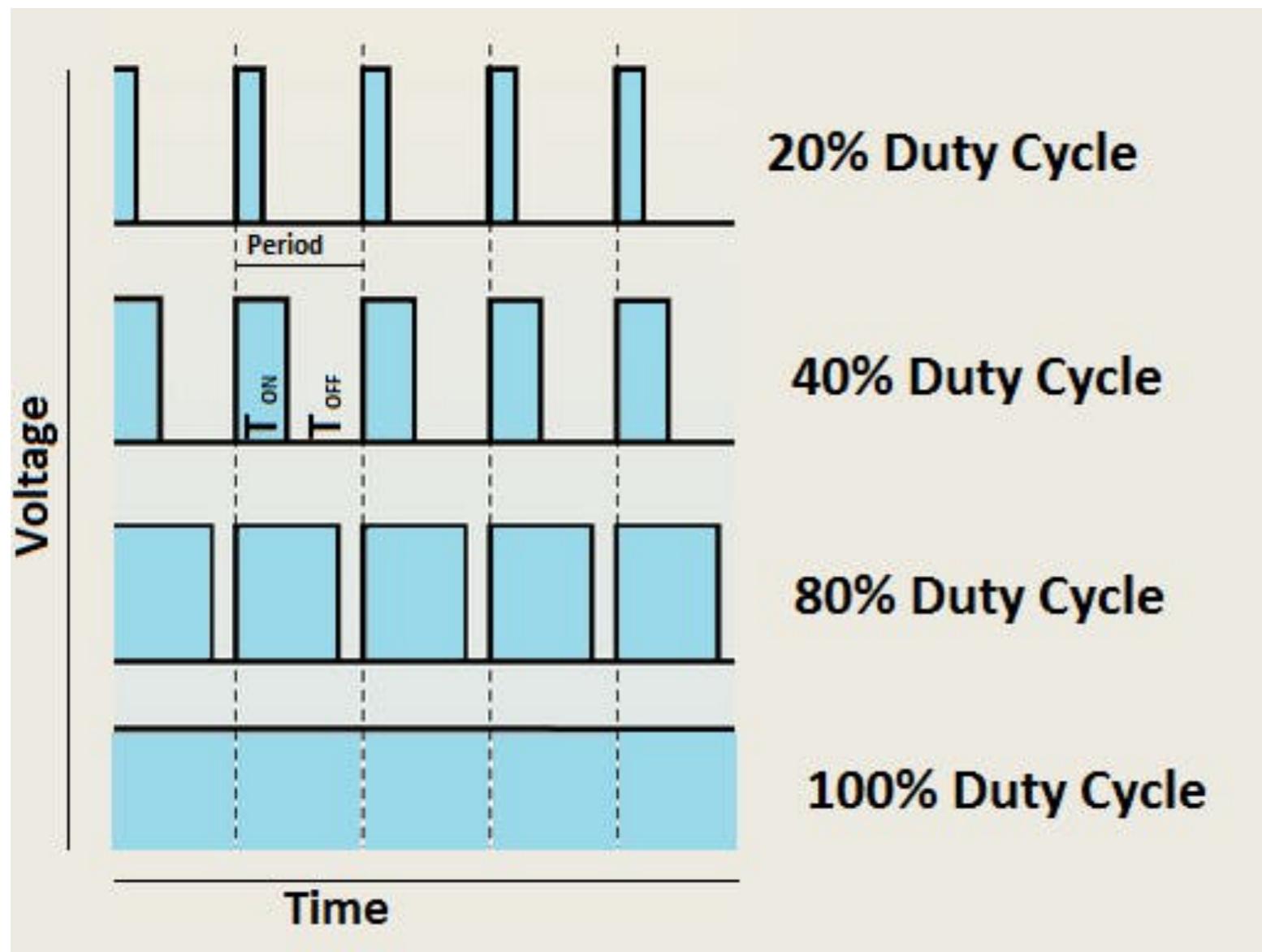
Exercício

- ❖ Faça um hardware para leitura do sensor LM35 e que mostre o valor da temperatura em um display LCD;
- ❖ Faça o Software para esse hardware considerando:
- ❖ Uma *splashscreen*;
- ❖ Tirar uma medição a cada 2 segundos;



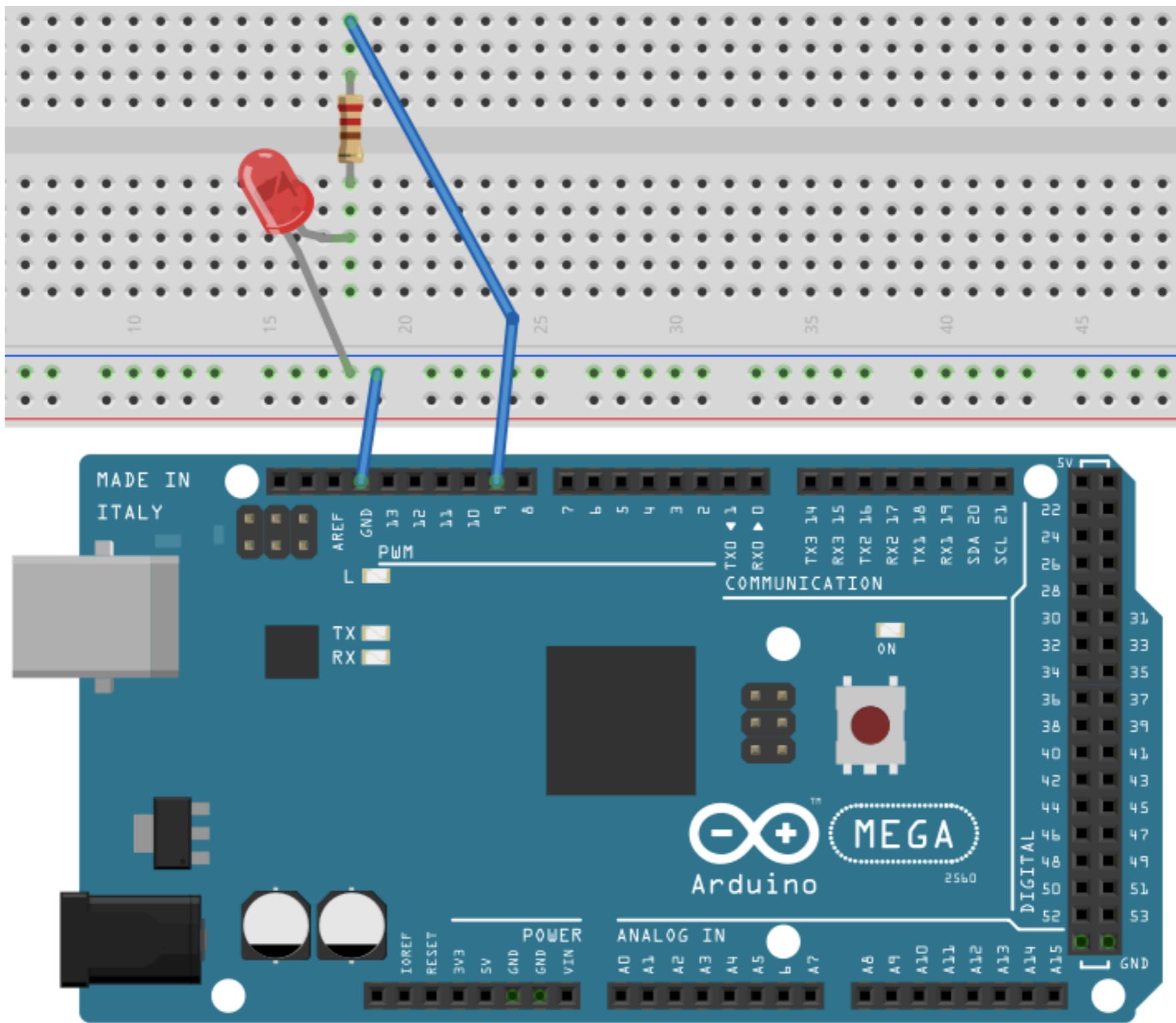
E uma saída analógica?

- ❖ O Arduino utiliza uma técnica chamada PWM (pulse width modulation) para geração de um sinal médio que pode variar entre 0 e 5V;



$$U = 5 * T_{on} / (T_{on} + T_{off})$$

Hardware



Software

```
/* Programa: Saída analógica (PWM)
 * Arduino : MEGA
 * Data    : 02/10/2019
 * Autor   : Rahul M. Juliato
 */
```

```
#define LED 9
```

```
void setup(){
```

```
  pinMode(LED, OUTPUT);
```

```
}
```

```
void loop() {
```

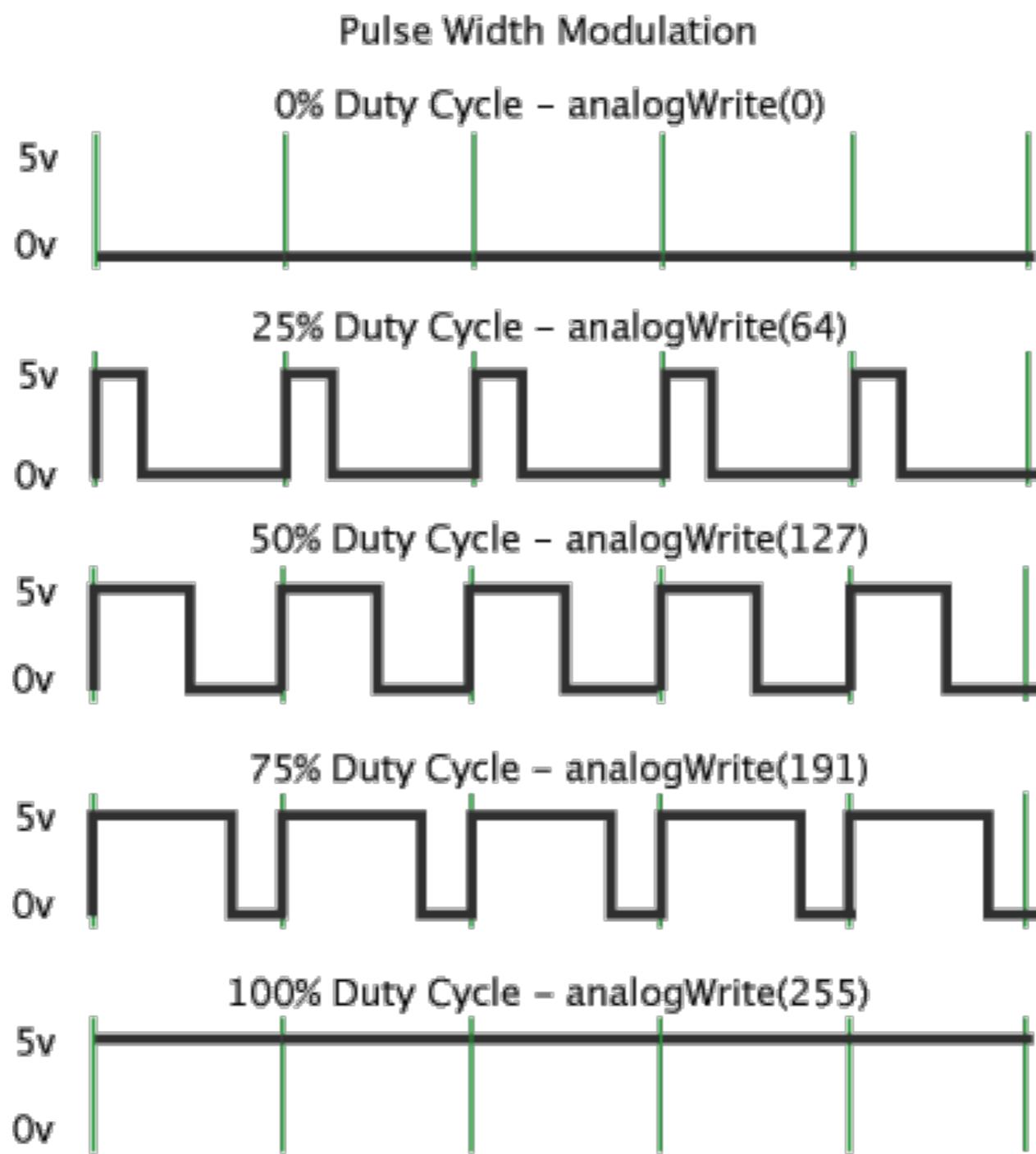
```
  analogWrite(LED, 127);
```

```
}
```

- ❖ Teste este programa variando o segundo argumento de analogWrite para:
 - ❖ 0
 - ❖ 64
 - ❖ 127
 - ❖ 191
 - ❖ 255



Explicação



Para o mesmo hardware...

- ❖ Vamos desenvolver um software que aumente a saída analógica de 0 até 5V (0 para 255) e em seguida diminua a saída analógica de 5V para 0V (255 para 0).



Software

```
/* Programa: Saída analógica (PWM)
 * Arduino : MEGA
 * Data    : 02/10/2019
 * Autor   : Rahul M. Juliato
 */

#define LED 9

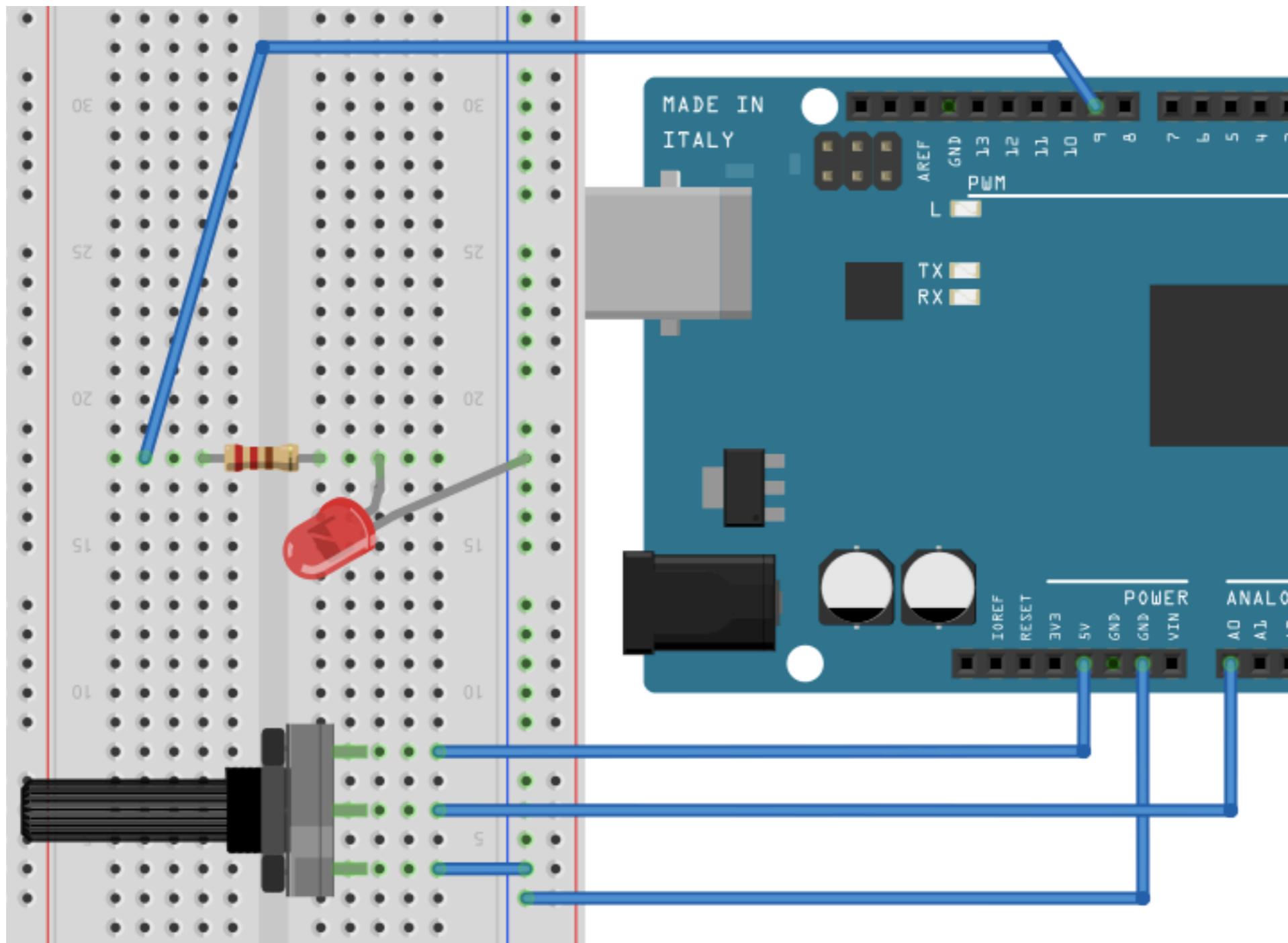
int i;

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

void loop() {
    for(i=0;i<255;i++){
        analogWrite(LED, i);
        delay(5);
    }
    for(i=255;i>0;i--){
        analogWrite(LED, i);
        delay(5);
    }
}
```



Hardware



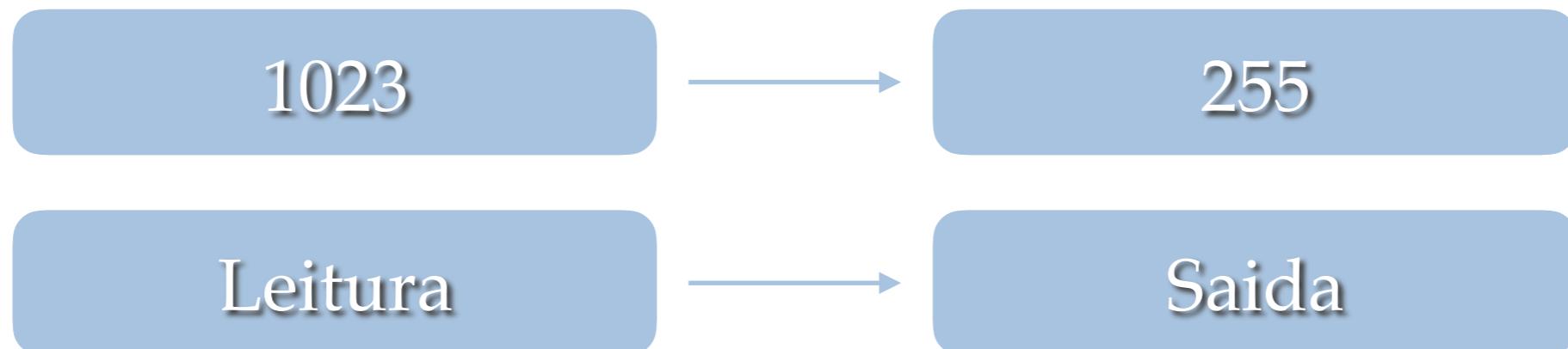
Sua vez!

- ❖ Faça um software que controle a saída no led entre 0V e 5V (0 à 255) com base na leitura da entrada analógica 0V a 5V (0 à 1023);



Mais uma vez

- ❖ Utilizamos uma “regra de três”:



- ❖ Multiplicando em cruz e isolando “Saida”, temos:
- ❖ $\text{Saida} = (\text{Leitura} \times 255) / 1023$
- ❖ Repare que o Valor de “Saida” será utilizando na função analogWrite, que só aceita int!

Software

```
/* Programa: Leitura para saída analógica
 * Arduino : MEGA
 * Data    : 02/10/2019
 * Autor   : Rahul M. Juliato
 */

int led = 9;
int pot = A0;

int leitura;
int saida;

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

void loop() {
    leitura = analogRead(pot);
    saida = (leitura * 255) / 1023;
    analogWrite(led, saida);
    delay(1);
}
```



Fix

- ❖ A conta deve ser feita como float para não estourar os limites da variável tipo int.
- ❖ Tente novamente com 1023.0 e 255.0



Dica...

- ❖ O Arduino tem uma função pronta que faz a tal “regra de três” para situações simples.
- ❖ É a função map
- ❖ Ela mapeia valores dessa forma:
- ❖ `map(variável, menor_1, maior_1, menor_2, maior_2);`



No nosso caso:

```
/* Programa: Leitura para saída analógica
 * Arduino : MEGA
 * Data    : 02/10/2019
 * Autor   : Rahul M. Juliato
 */

int led = 9;
int pot = A0;

int leitura;
int saida;

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

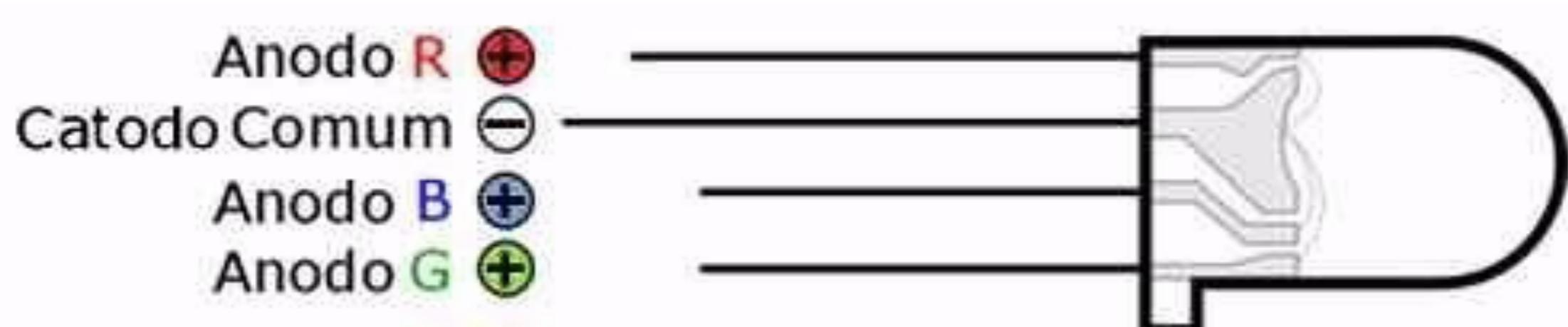
void loop() {
    leitura = analogRead(pot);
    saida = map(leitura, 0, 1023, 0, 255);
    analogWrite(led, saida);
    delay(1);
}
```



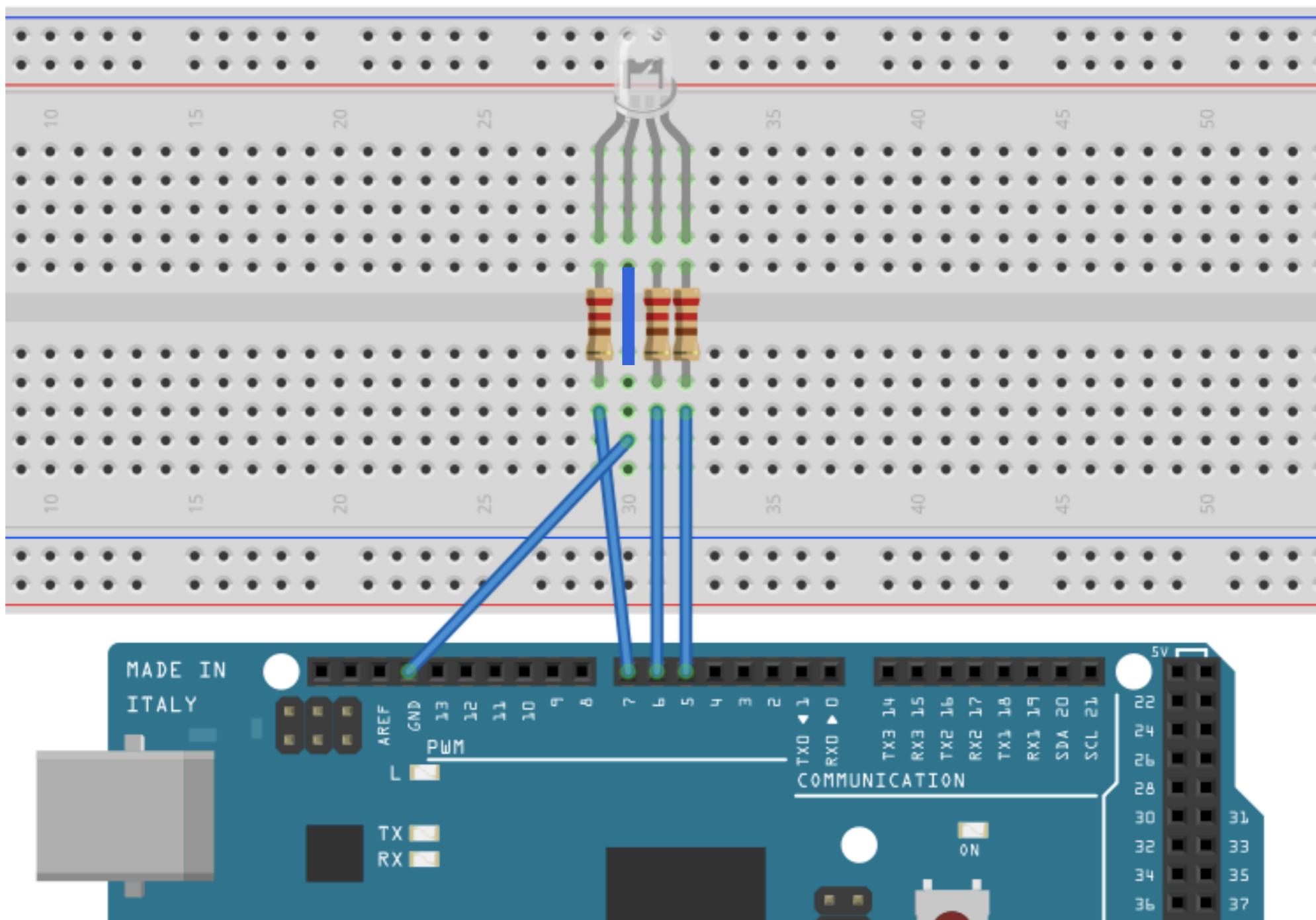
Exercício

- ❖ Você conseguiria fazer o mesmo hardware para controlar um servo-motor entre 0 e 180 graus?
- ❖ E o programa?
- ❖ Treine no simulador!
- ❖ E faça na prática!

Um pouco mais de PWM



Hardware



Software

- ❖ Faça o software levando em consideração:
- ❖ Cada LED pode receber de 0V a 5V;
- ❖ Acender só Vermelho por 2 segundos, depois Verde por 2 segundos depois Azul por 2 segundos, depois Amarelo por 2 segundos e Roxo por 2 segundos em loop.



Cores RGB

- ❖ Coincidentemente... 0 -> 255

Red:
Green:
Blue:



Red:
Green:
Blue:



Red:
Green:
Blue:



Red:
Green:
Blue:



Red:
Green:
Blue:



Software

```
/* Programa: Led RGB
 * Arduino : MEGA
 * Data    : 03/10/2019
 * Autor   : Rahul M. Juliato
 */
int ledr = 7;
int ledg = 5;
int ledb = 6;

void setup(){
    pinMode(ledr, OUTPUT);
    pinMode(ledg, OUTPUT);
    pinMode(ledb, OUTPUT);
}
```

```
void loop() {
    digitalWrite(ledr, 255);
    digitalWrite(ledg, 0);
    digitalWrite(ledb, 0);
    delay(2000);

    digitalWrite(ledr, 0);
    digitalWrite(ledg, 255);
    digitalWrite(ledb, 0);
    delay(2000);

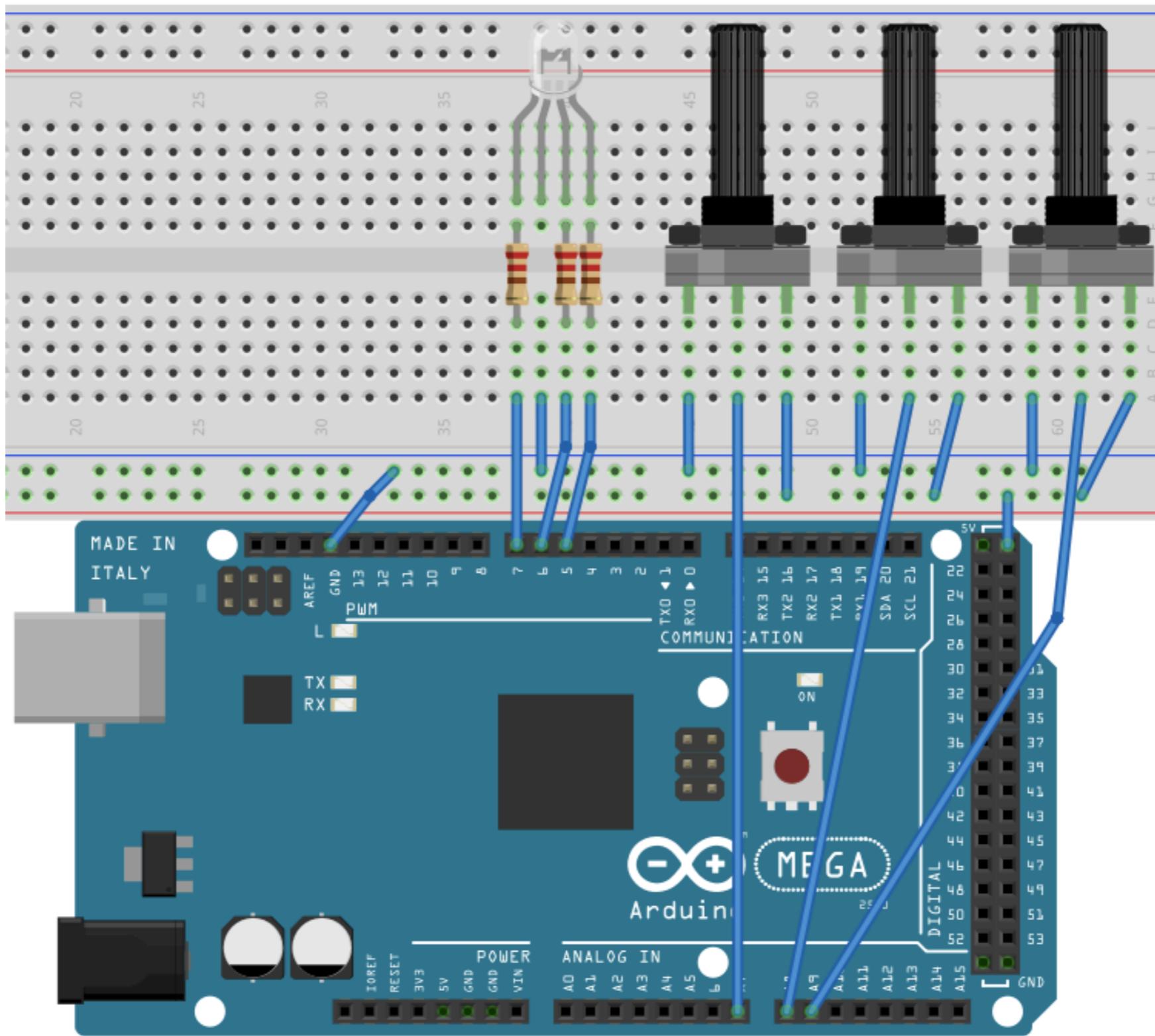
    digitalWrite(ledr, 0);
    digitalWrite(ledg, 0);
    digitalWrite(ledb, 255);
    delay(2000);
}
```

```
analogWrite(ledr, 255);
analogWrite(ledg, 255);
analogWrite(ledb, 0);
delay(2000);

analogWrite(ledr, 255);
analogWrite(ledg, 0);
analogWrite(ledb, 255);
delay(2000);
```



Hardware



Software

```
/* Programa: Led RGB controlado por potenciômetros
 * Arduino : MEGA
 * Data    : 03/10/2019
 * Autor   : Rahul M. Juliato
 */
int potr = A9;
int potg = A8;
int potb = A7;

int ledr = 7;
int ledg = 5;
int ledb = 6;

int leiturar = 0;
int leiturag = 0;
int leiturab = 0;

int saidar = 0;
int saidag = 0;
int saidab = 0;

void loop() {

    leiturar = analogRead(potr);
    leiturag = analogRead(potg);
    leiturab = analogRead(potb);

    saidar = map(leiturar, 0, 1023, 0, 255);
    saidag = map(leiturag, 0, 1023, 0, 255);
    saidab = map(leiturab, 0, 1023, 0, 255);

    analogWrite(ledr, saidar);
    analogWrite(ledg, saidag);
    analogWrite(ledb, saidab);

}
```



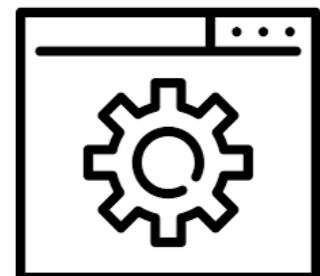
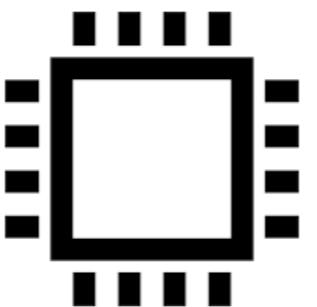
Desafios

- ❖ Que tal incluir uma mensagem na porta serial?
- ❖ Como faríamos para incluir um display LCD que mostra o ajustado?
- ❖ Que tal incluir botões para selecionar cores pré-configuradas?
- ❖ Que tal incluir uma opção nas seleções com o nome “ajustável”, que lê as entradas analógicas somente nesse estado?



Não acaba por aqui!

- ❖ Estude eletrônica!
- ❖ Estude linguagens de programação!
- ❖ Treine, treine, treine e treine!



fin