

ADA

PROJETOS EM ENGENHARIA  
DE COMPUTAÇÃO

# Arduino Básico

Hello World

# Arduino?

- O que é isso?
- Por que usar?



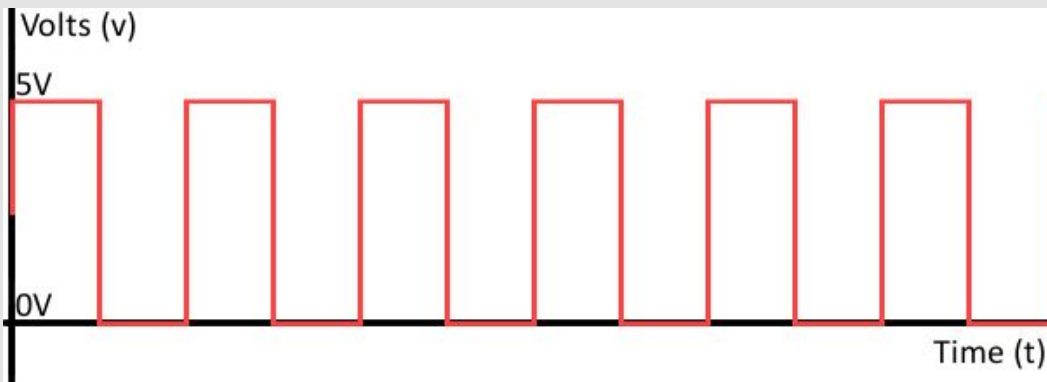
# Como essa placa funciona?

- Pinos controlados por código
  - Saídas digitais
  - Entradas digitais
  - Saídas analógicas
  - Entradas analógicas



# Eletrônica Digital

- Apenas dois valores possíveis
  - 0 e 1 (Low e High)
  - Saída digital
    - Saída é 5V (1) ou 0V (0)
  - Entrada digital
    - Há tensão na porta (1) ou não há (0)





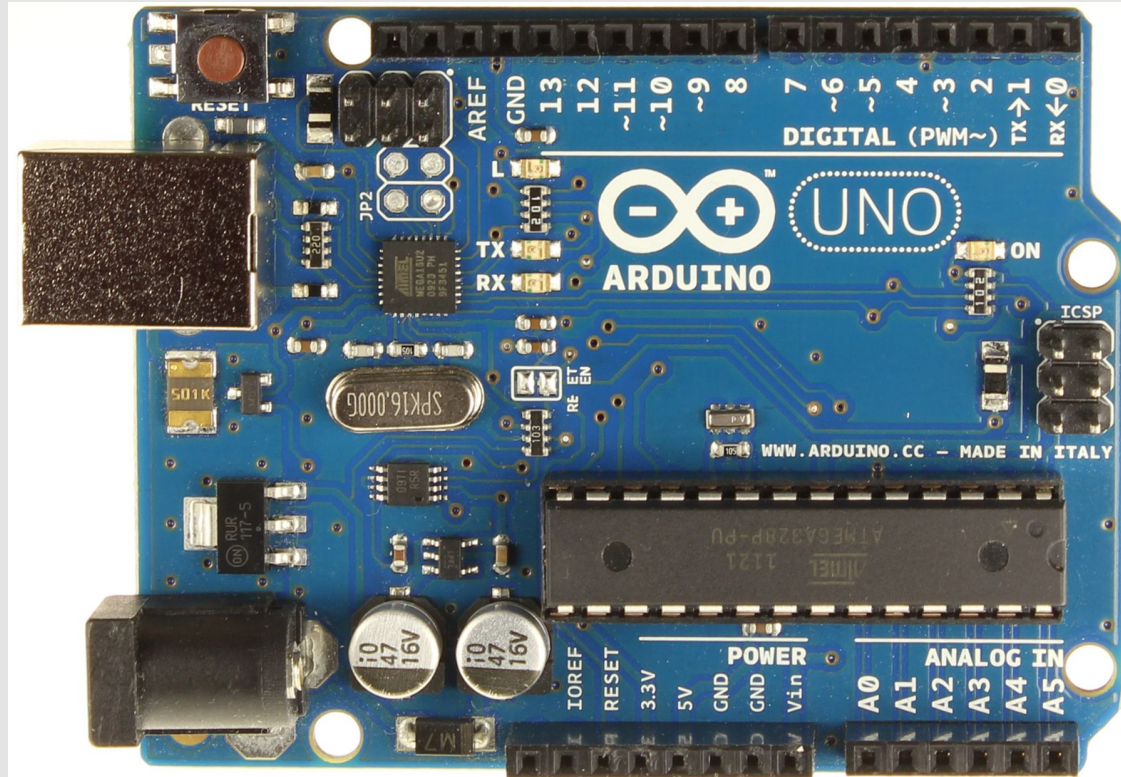
# Eletrônica Analógica

- Sinais contínuos
  - Infinitos valores possíveis (ou quase)
  - Entrada analógica
    - Valor da tensão na porta
  - Saída analógica
    - Valor entre 0V e 5V





# Conhecendo a placa





# Conhecendo a IDE

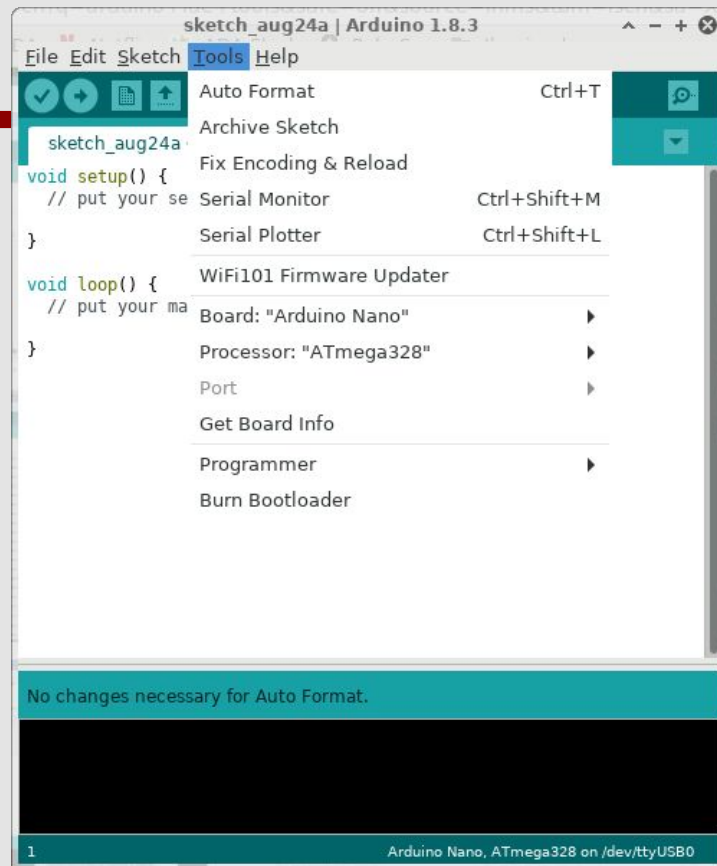
- Compilar
- Upload
- Novo
- Abrir
- Salvar





# Conhecendo a IDE

- Board
- Processor
- Port





# Vamos começar

- Programa blink - o Hello World dos microcontroladores

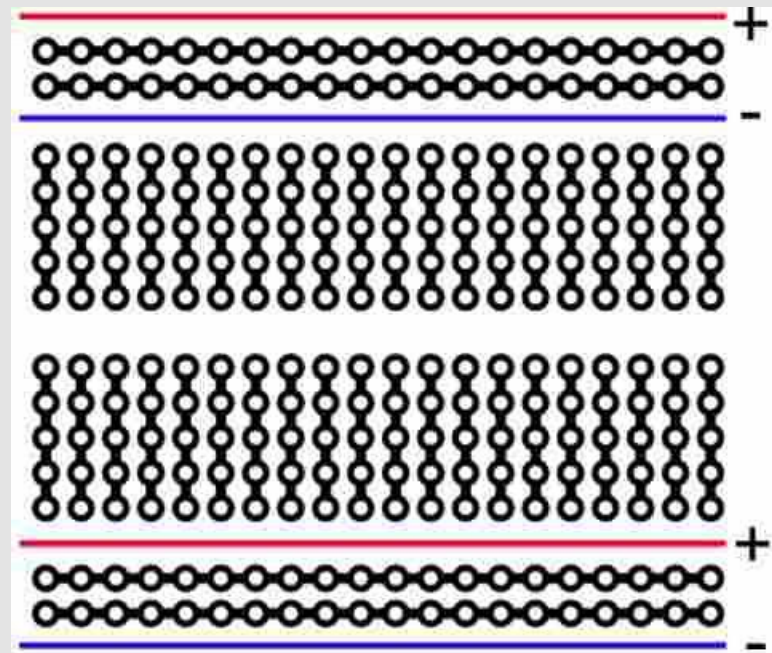


```
void setup() {  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);  
  delay(1000);  
  digitalWrite(13, LOW);  
  delay(1000);  
}
```



# Conectando um LED externo

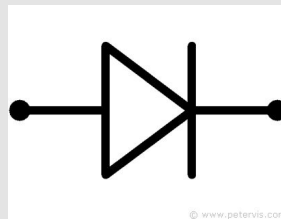
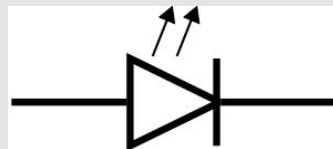
- Iremos precisar de uma protoboard!
  - É uma placa para prototipagem de circuitos
  - Os furos do meio são interligados na vertical
  - Os furos nas extremidades são ligados na vertical





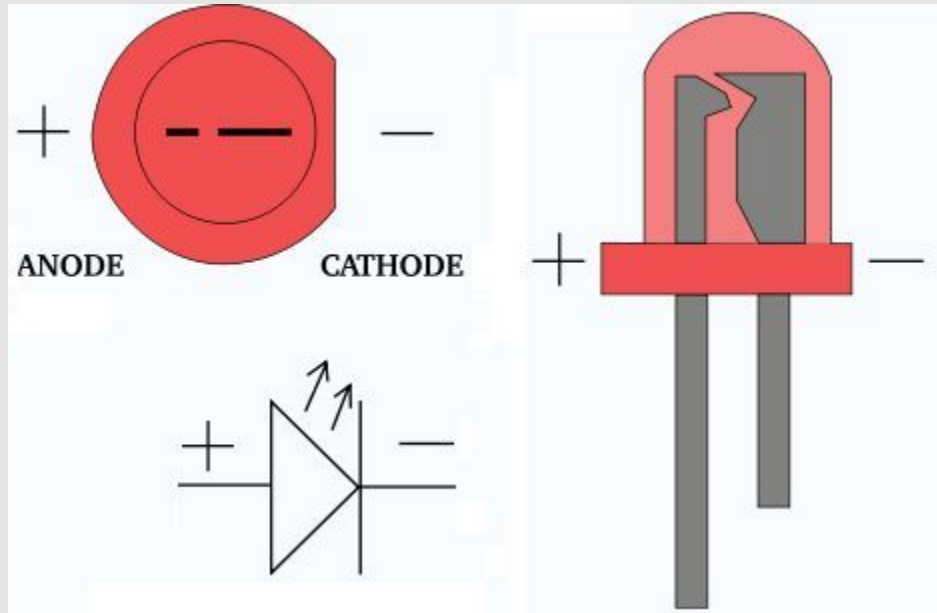
# Conectando um LED externo

- O que é um LED?
  - Um diodo que emite luz
- Mas o que é um diodo?
  - É um componente que permite a passagem de corrente apenas em um sentido
- Como saber qual é o lado certo?





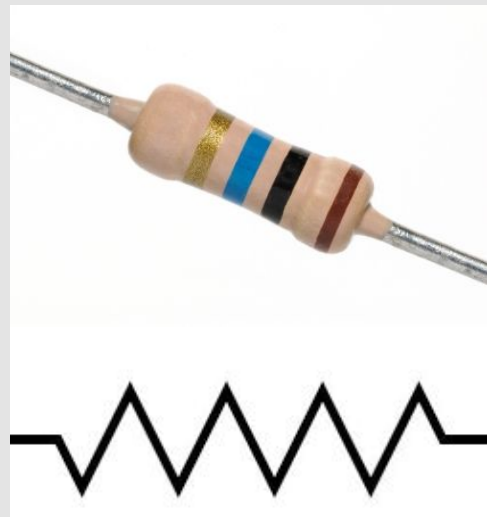
# Conectando um LED externo





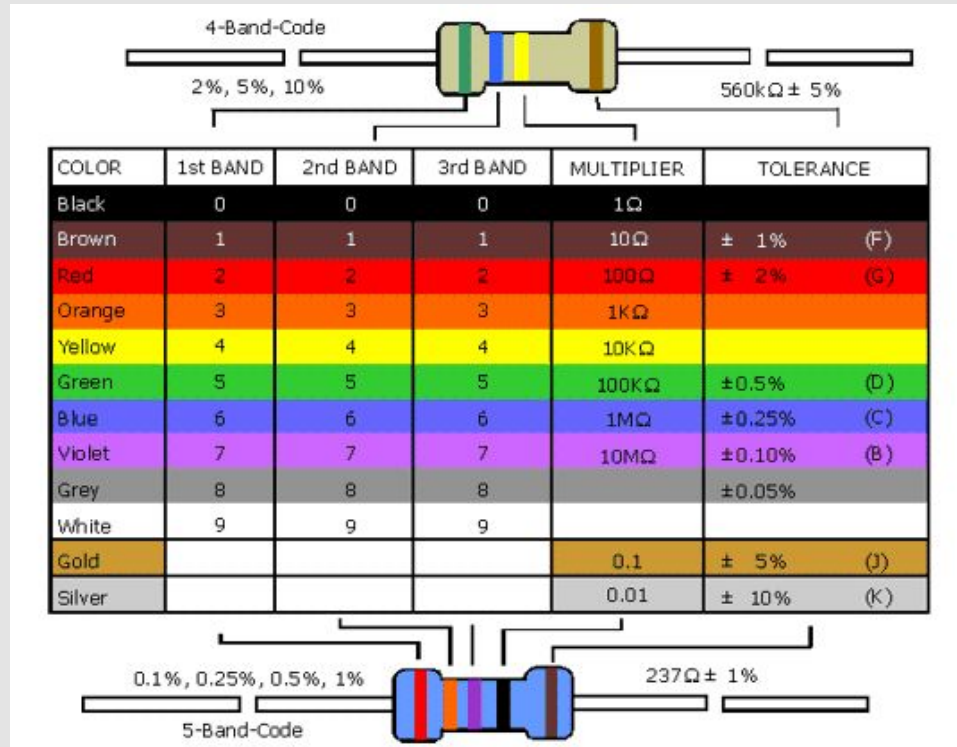
## Conectando um LED externo

- Também precisamos limitar a corrente que passa pelo LED com um resistor
- As faixas coloridas nele indicam o valor (em Ohm) e a precisão





# Conectando um LED externo



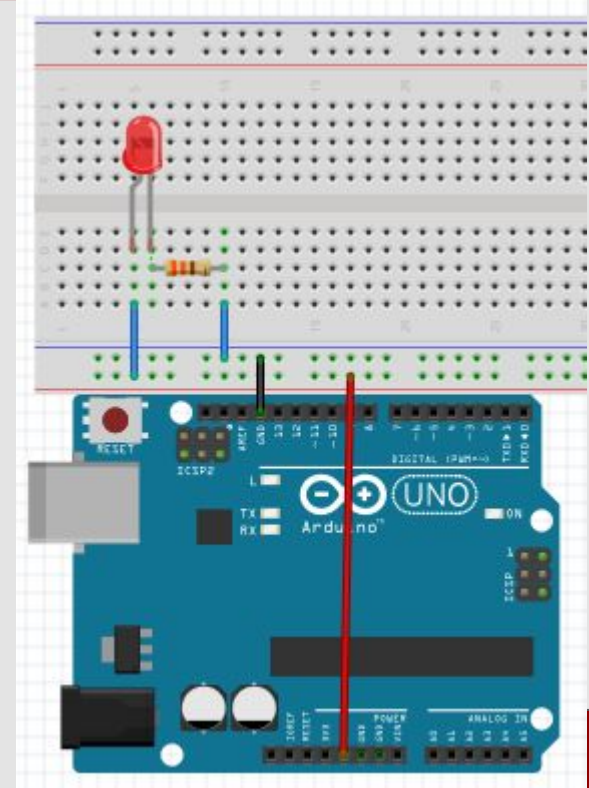
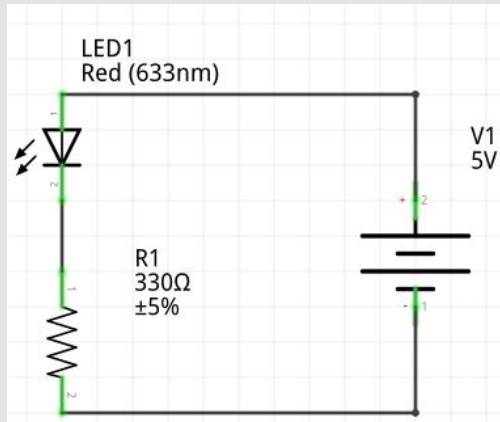
# Conectando um LED externo

- E como calcular o resistor ideal para o LED?
  - A conta é um pouquinho complicada, mas é mais ou menos assim:
  - Deve passar por volta de 20mA no LED
  - Com esta corrente passando, ele dissipa uma tensão de aproximadamente 2V
  - Alimentamos o circuito com 5V
  - Então  $(5V - 2V) / 20mA$  deve dar o valor da resistência



# Conectando um LED externo

- Mas não precisamos da potência toda do LED
  - Podemos colocar um resistor entre 200 e 400 $\Omega$





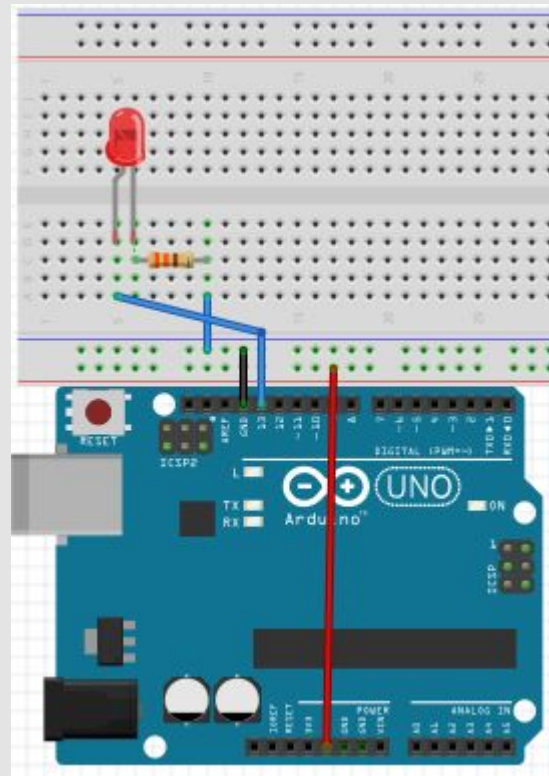
# Conectando um LED externo

- Ok. Desse jeito o Led fica sempre aceso
  - Como podemos controlar?



# Conectando um LED externo

- Ok. Desse jeito o Led fica sempre aceso
  - Como podemos controlar?
- Ligando o LED em uma porta digital podemos controlar quando ele fica aceso ou apagado



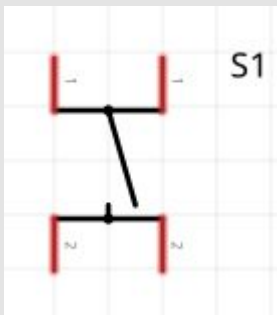
# Desafio 01

- Ligar 3 LEDs em 3 portas digitais diferentes
- Acender o primeiro, depois o segundo, o terceiro e assim sucessivamente
- Apenas um aceso por vez



# Entrada digital

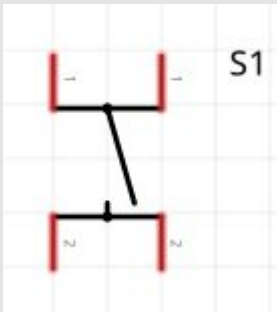
- Já aprendemos como controlar uma saída digital
- Agora vamos ver como ler uma entrada
- Vamos utilizar um *push button*





# Entrada digital

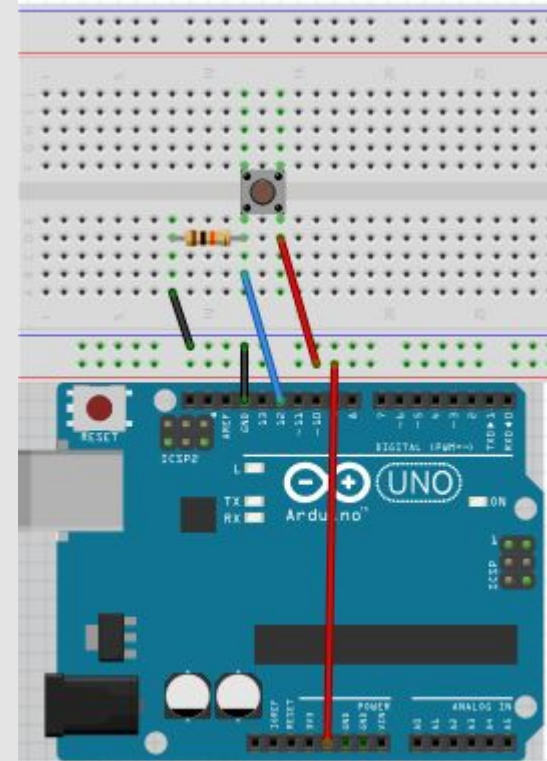
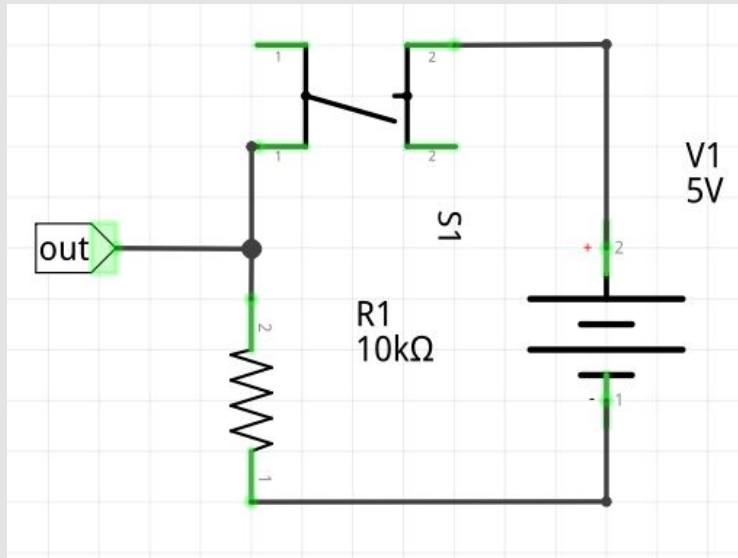
- Como funciona?
  - É bem simples:
  - Enquanto o botão não está sendo apertado não há contato entre os pares de terminais (perninhas)
  - Enquanto pressionamos o contato é fechado





# Entrada digital

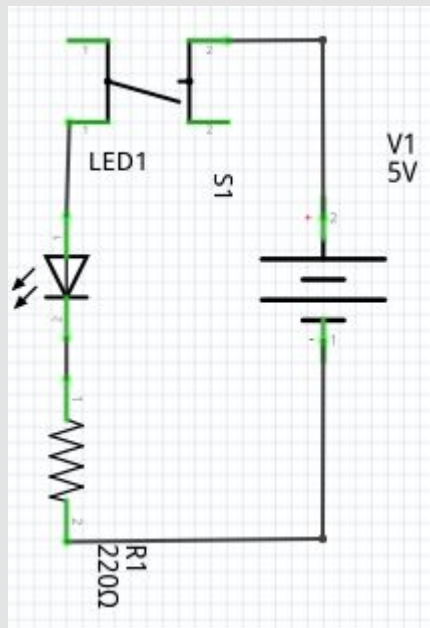
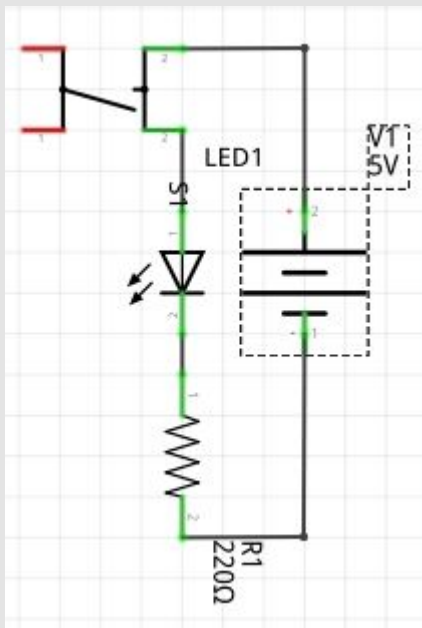
- E como ligamos ele no Arduino?
  - (Para que serve este resistor?)





# Entrada digital

- Como saber qual par de terminais é conectado?





# Entrada digital

- Como ler a entrada pelo Arduino?

```
void setup() {  
    pinMode(12, INPUT) ;  
    pinMode(13, OUTPUT) ;  
}  
  
void loop() {  
    if (digitalRead(12)) {  
        digitalWrite(13, HIGH) ;  
    } else {  
        digitalWrite(13, LOW) ;  
    }  
}
```



# Exercício 01

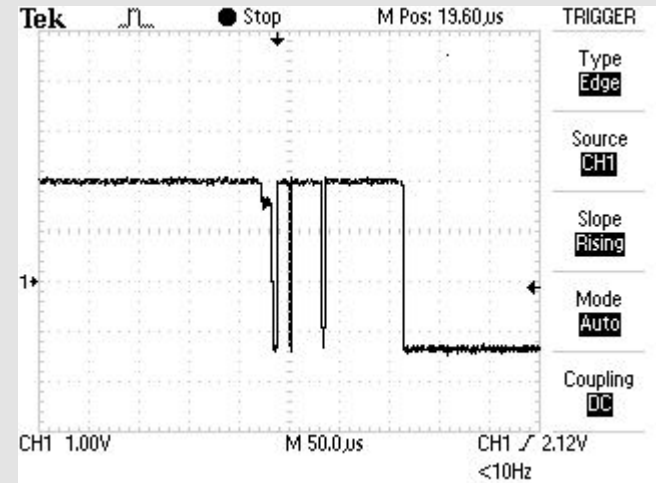
- Montar um LED e um botão no Arduino (pode-se utilizar o LED da própria placa)
- Ao apertar o botão o LED muda de estado (aceso → apaga; apagado → acende)



# Entrada digital

- Debouncing
  - Pelo botão usar um processo mecânico é gerado ruído
  - Este ruído pode ser interpretado de maneira errada pelo Arduino

- Sugestões?



# Desafio 02

- Ligar três LEDs e um botão ao Arduino
- O primeiro LED começa aceso
- Ao pressionar o botão, apaga-se o LED aceso e acende o próximo (ciclicamente)

## Função millis()

- Retorna a quantidade de milisegundos passados desde o início do programa
- Consegue guardar até 50 dias

```
unsigned int tempo = millis() ;
```

# Exercício 02

- Faça o programa blink utilizando a função `millis()`

# Extra 01

Levando em consideração o código e a montagem do desafio 02:

- Agora configure uma porta como OUTPUT
- Faça a saída da porta alternar entre 0 e 1 cada 1 segundo
- Ligue essa porta no lugar do botão



# Comunicação Serial

- Comunicação com o computador
- Envio de dados
- Recebimento de dados/comandos



# Comunicação Serial

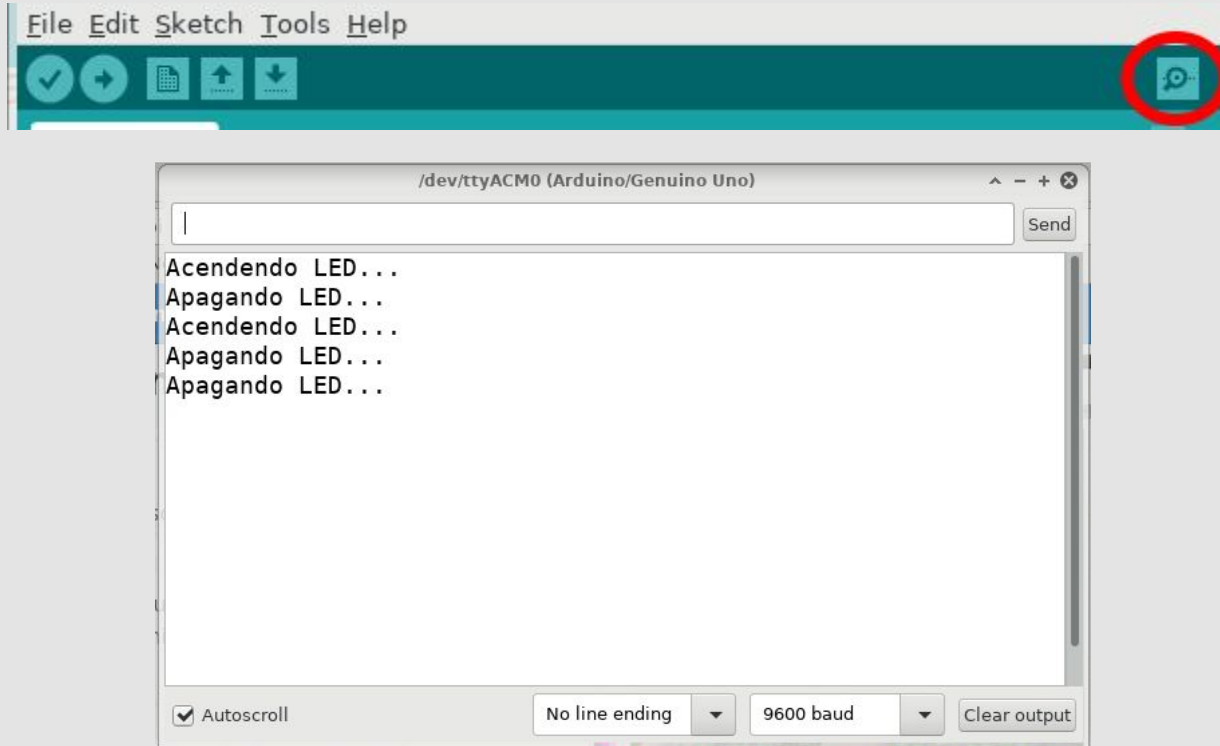
```
void setup() {  
  Serial.begin(9600) ;  
}  
  
void loop() {  
  Serial.println("Hello World") ;  
  delay(500) ;  
}
```

```
int byteEntrada = 0 ;  
  
void setup() {  
  pinMode(13, OUTPUT) ;  
  Serial.begin(9600) ;  
}  
  
void loop() {  
  if (Serial.available() > 0) {  
    byteEntrada = Serial.read() ;  
    if (byteEntrada == '0') {  
      digitalWrite(13, LOW) ;  
      Serial.println("Apagando LED...") ;  
    } else if (byteEntrada == '1') {  
      digitalWrite(13, HIGH) ;  
      Serial.println("Acendendo LED...") ;  
    }  
  }  
}
```





# Comunicação Serial



# Exercício 03

- Ligue três LEDs ao Arduino
- Receba da entrada serial um número N entre 1 e 3
- O número indicará que deve-se mudar o estado do LED N

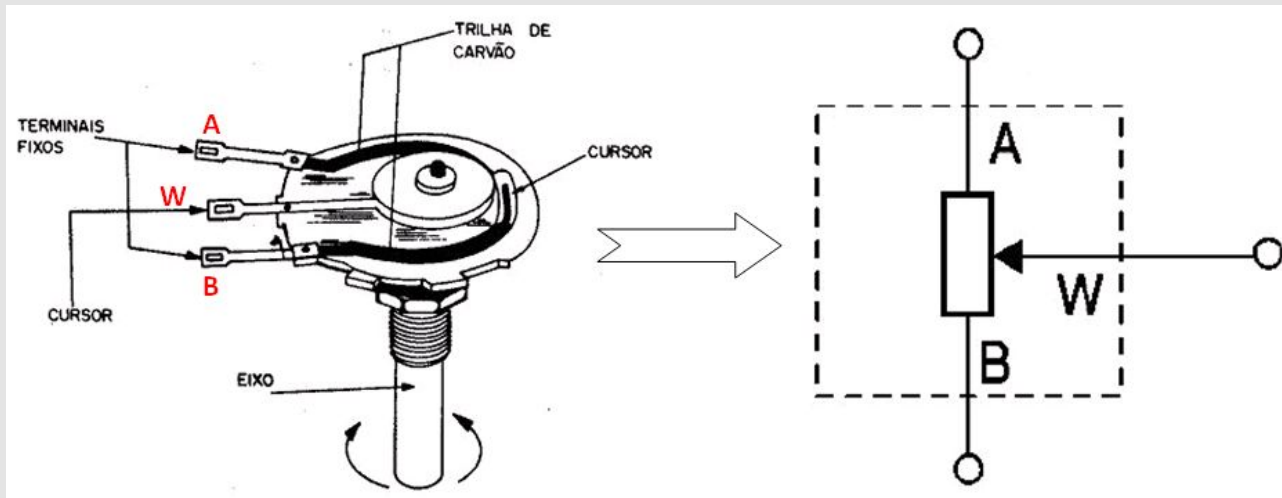
# Entrada analógica

- Converte a tensão no pino para um valor entre 0 (0V) e 1023 (5V)
- Usamos para obter valores de sensores como:
  - Temperatura
  - Luminosidade
  - Nível de determinado gás
  - Leitura de um potenciômetro



# Entrada analógica

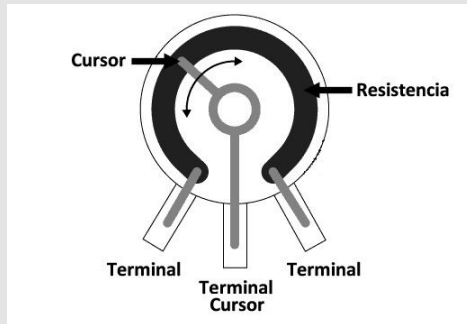
- O que é um potenciômetro?
  - Componente que permite variação na resistência





# Entrada analógica

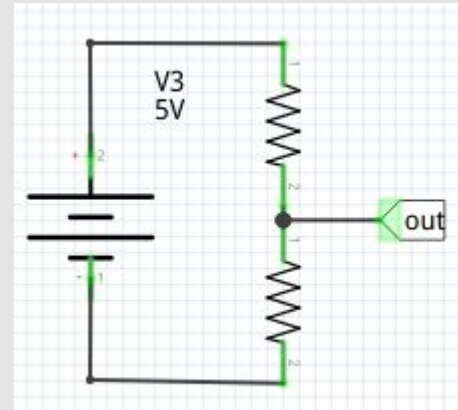
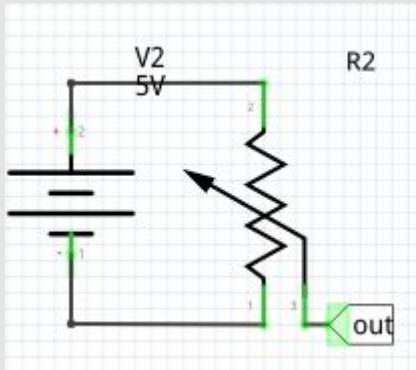
- Como utilizar:
  - O potenciômetro tem três terminais
  - A resistência entre as duas extremidades é sempre fixa
  - A resistência entre o pino do meio e qualquer uma das extremidades é variável
  - Conseguimos variar essa resistência girando o pino central





# Entrada analógica

- Utilizamos o potenciômetro como um divisor de tensão
  - Conforme variamos a resistência do terminal central, a tensão neste terminal varia

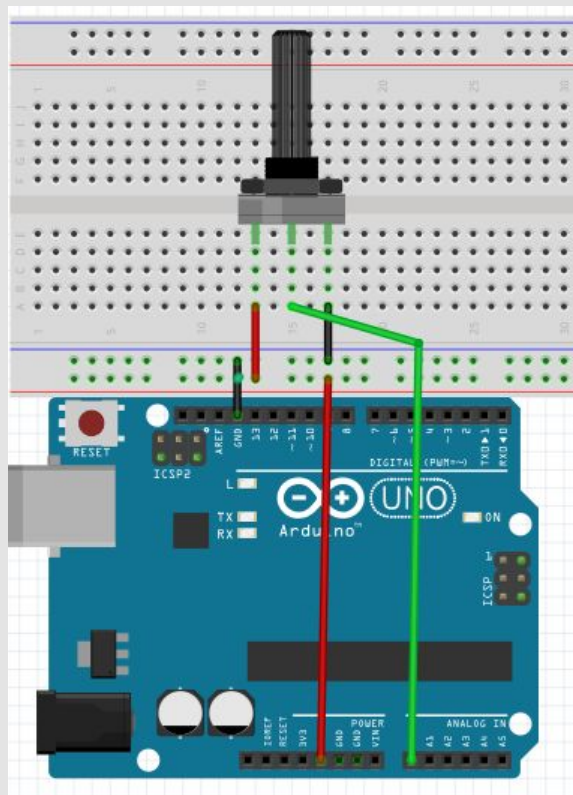




# Entrada analógica

- Conectamos o terminal central a uma das portas analógicas (A0 ~ A1)

```
void setup() {  
  Serial.begin(9600) ;  
}  
  
void loop() {  
  Serial.println(analogRead(0)) ;  
  delay(500) ;  
}
```



# Desafio 03

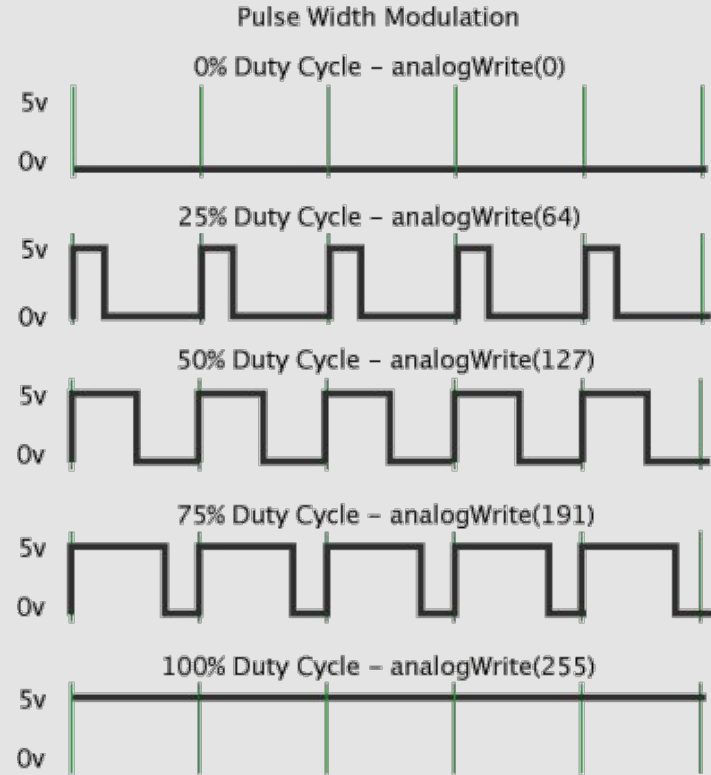
- Ligue 4 LEDs ao Arduino
- Leia o valor de tensão do potenciômetro
- Considerando o valor lido faça:
  - Até 200: nenhum LED aceso
  - 201 ~ 400: LED 1 aceso
  - 401 ~ 600: LEDs 1 e 2 acesos
  - 601 ~ 800: LEDs 1, 2 e 3 acesos
  - Maior que 800: todos LEDs acesos





# Saída analógica

- É feita através de PWM (Pulse Width Modulation)
- Simula valores entre 0V e 5V
- Apenas em portas digitais com til (~)





# Saída analógica

- Qual é a utilidade?
  - Geralmente se usa PWM para controlar a potência transmitida a uma carga
  - Isso nos permite controlar por exemplo a intensidade de um LED ou a velocidade de um motor



# Saída analógica

- No Arduino utilizamos a função `analogWrite`

```
int val = 0 ;

void setup() {
  pinMode(11, OUTPUT) ;
}

void loop() {
  if (val >= 255) {
    val = 0 ;
  }
  analogWrite(11, val) ;
  val += 10 ;
  delay(200) ;
}
```

# Exercício 04

- Ler o valor de tensão do potenciômetro
- Deixar este valor entre 0 e 255
- Mudar a intensidade de um LED conforme o valor lido

# Desafio 04

- Implementar novamente o desafio 03
- Mudar a intensidade do último LED aceso conforme o valor lido
- Exemplo:
  - O LED 2 ficará mais aceso quando o valor lido for 550 do que quando for 450.