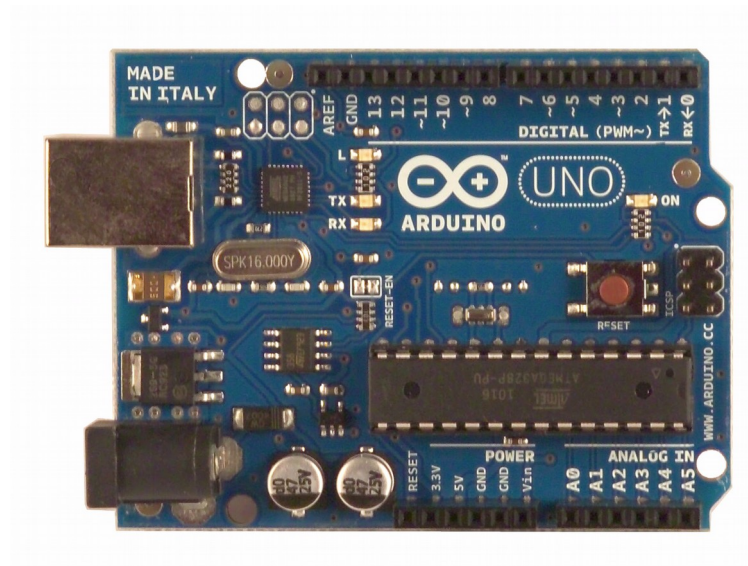


# Introdução do Arduino para aquisição de dados e controle de instrumentos



Rafael Pezzi

Centro de Tecnologia Acadêmica  
Instituto de Física - UFRGS



# Roteiro

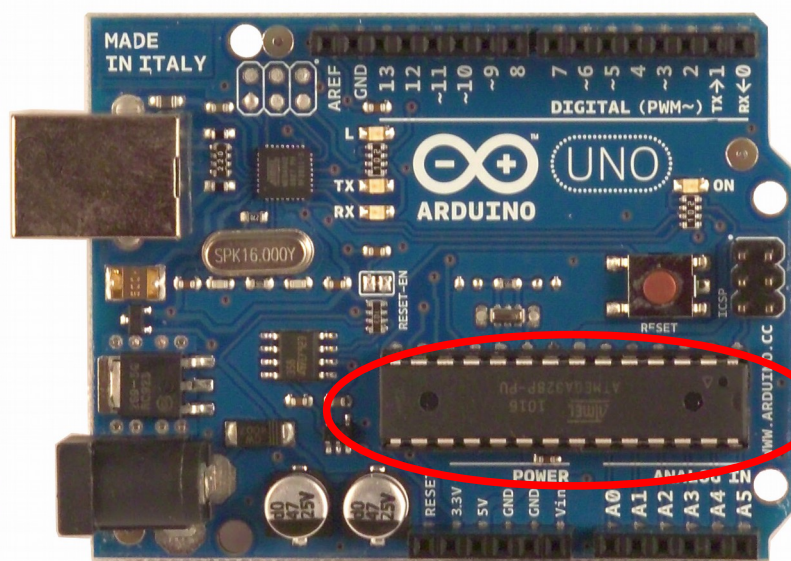
- O que é o Arduino?
  - Especificações
- Entradas e saídas digitais
  - Funções Setup() e Loop()
- Comunicação serial
- Entradas analógicas
- Saídas analógicas PWM
- Programação Avançada
- Guardar o material

# Materiais

- Arduino UNO ou Duemilanove
- Cabo USB
- Computador com IDE do Arduino (TropOS)
- LEDs
- LDR + Resistor
- Potenciometro
- Protoboard
- Jumpers

# O que é o Arduino?

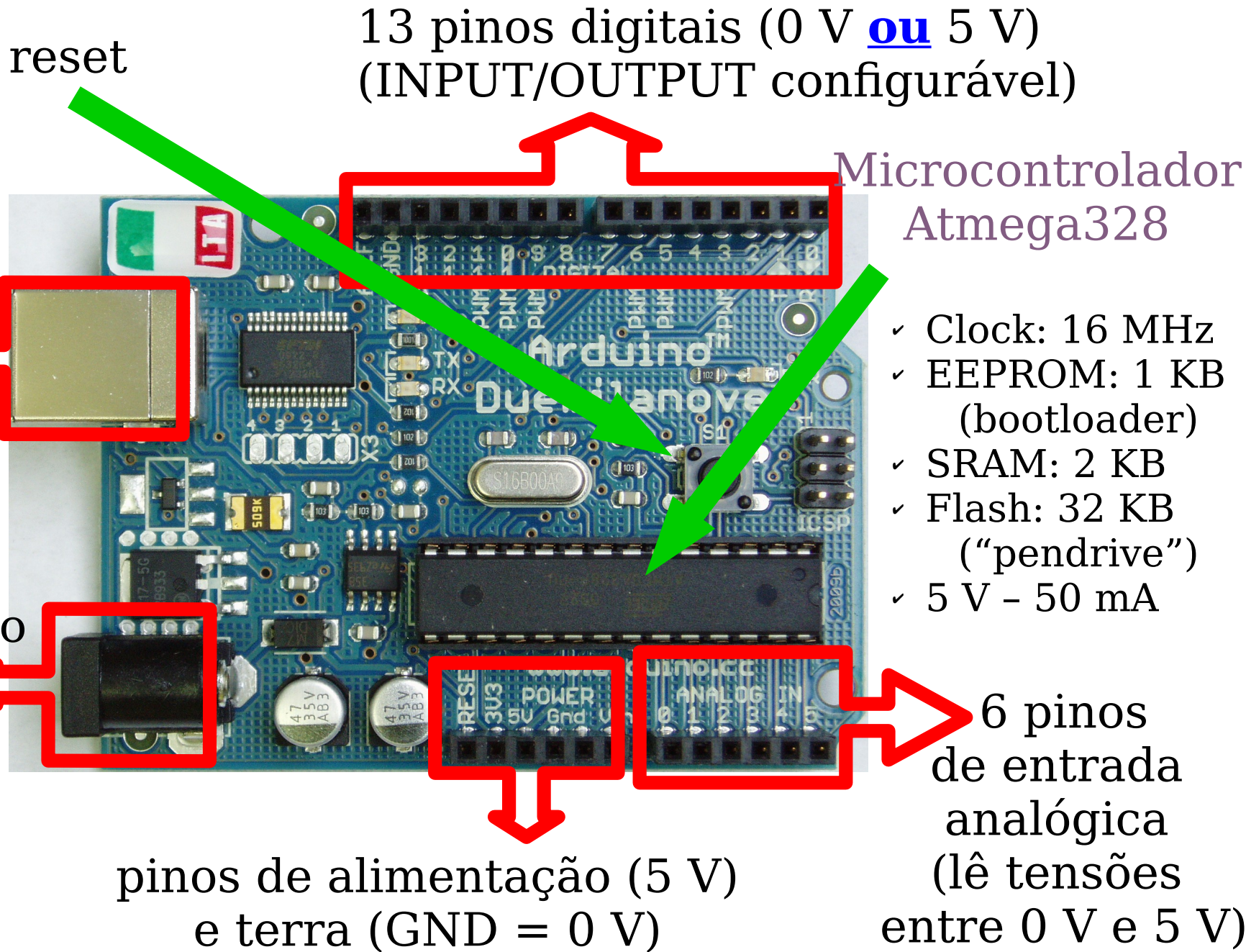
- O Arduino é uma plataforma de prototipagem de eletrônicos de código aberto
- Todos os diagramas e fontes de programação estão disponíveis sob licenças livres



Microcontrolador  
Atmega328 da Atmel

O Arduino Uno é baseado no Atmega328 e contém pinos digitais de entrada e saída, entradas analógicas. A conexão USB é realizada por um chip separado

# Pinos e conexões do Arduino





# Cuidados

- Antes de começar
  - Conheça as limitações do Arduino:
    - Corrente máxima: 500 mA total, 50 mA por porta
  - Certifique-se que seu circuito não requer mais corrente do que o arduino pode oferecer
  - Cuidado com curto-circuitos, pode queimar o microcontrolador

# Entrada e saída digital

- Entradas e saídas digitais são portas programáveis para leitura ou "gravação" de um sinal digital (0 ou 1 – ligado ou desligado – zero ou 5 volts )
  - Entrada: botão pressionado, porta aberta...
  - Saída: ligar e desligar lâmpadas, motores...

# Programação de Arduinos: o essencial

Estrutura básica de um programa Arduino as funções setup e loop

- Setup: executa uma vez (quando ligado ou resetado)
- Loop: repete em laço indefinidamente

Na IDE do Arduino: File → Examples → 01. Basics → Blink

Define o pino 13 conectado ao LED

```
int ledPin = 13; // LED conectado ao pino digital 13
```

```
// A função setup() é executada uma vez quando o Arduino é iniciado.
```

Inicializa – define pino como saída

```
void setup() {  
  pinMode(ledPin, OUTPUT); // Inicializa o pino digital como saída  
}
```

Programa que roda em loop

```
// a função loop() é executada indefinidamente enquanto o Arduino está sendo alimentado
```

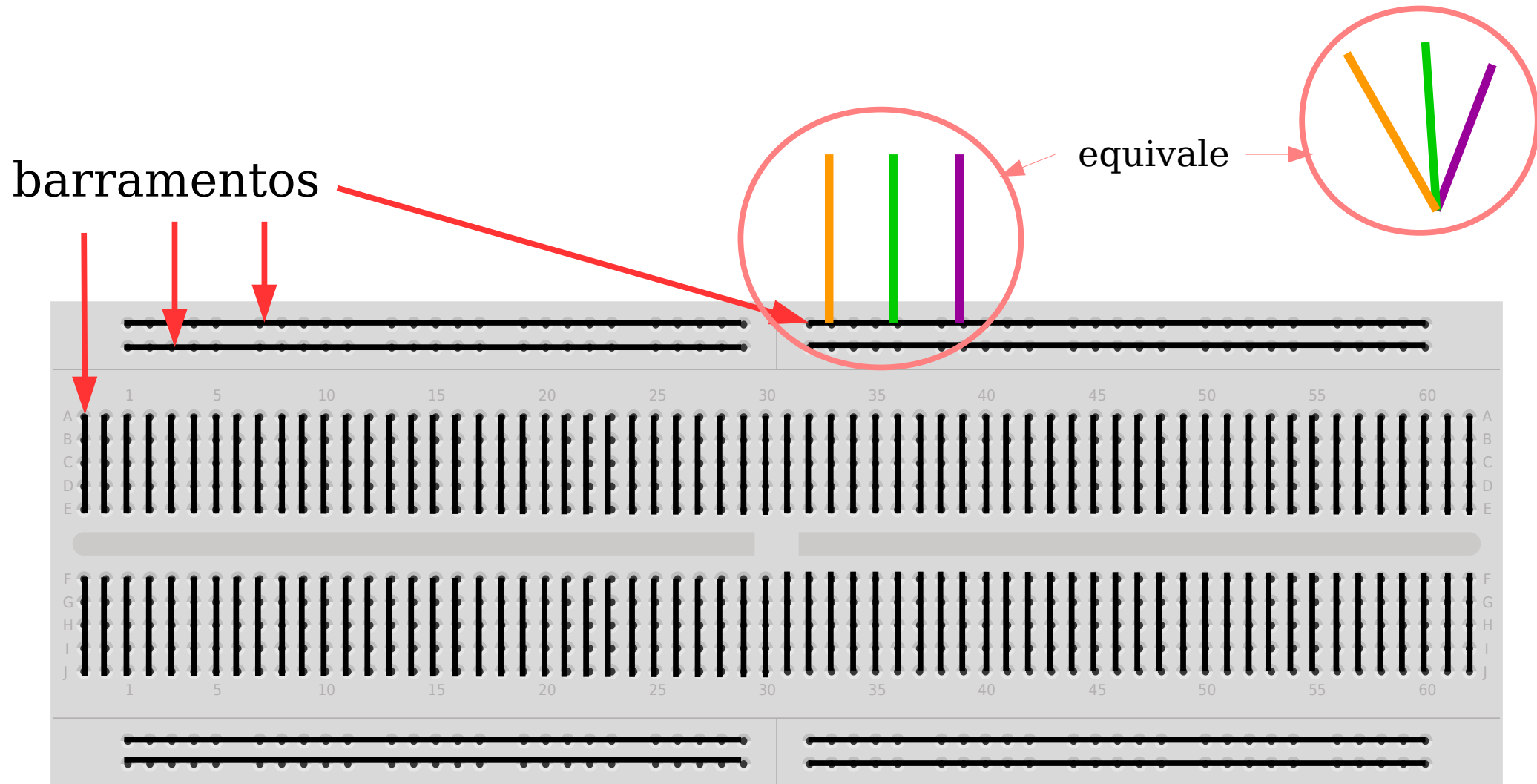
```
void loop()  
{  
  digitalWrite(ledPin, HIGH); // Acende o LED  
  delay(1000); // Aguarda um segundo  
  digitalWrite(ledPin, LOW); // Apaga o LED  
  delay(1000); // Aguarda um segundo  
}
```

Para carregar o programa, selecione a porta serial do arduino, o modelo da placa e faça o upload do programa.



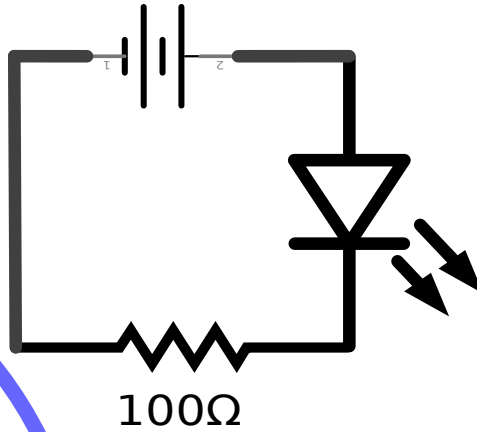
# Protoboard: Como funciona?

- Consiste num conjunto de barramentos isolados entre si;
- Um barramento equivale à uma junção de dois ou mais fios;

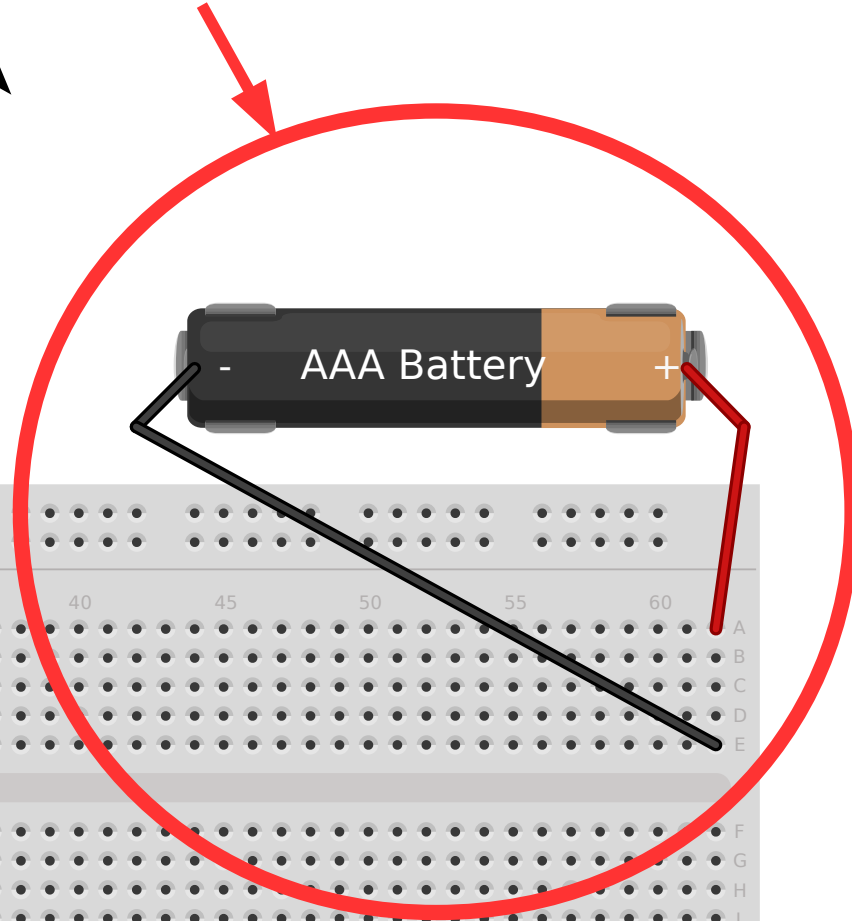
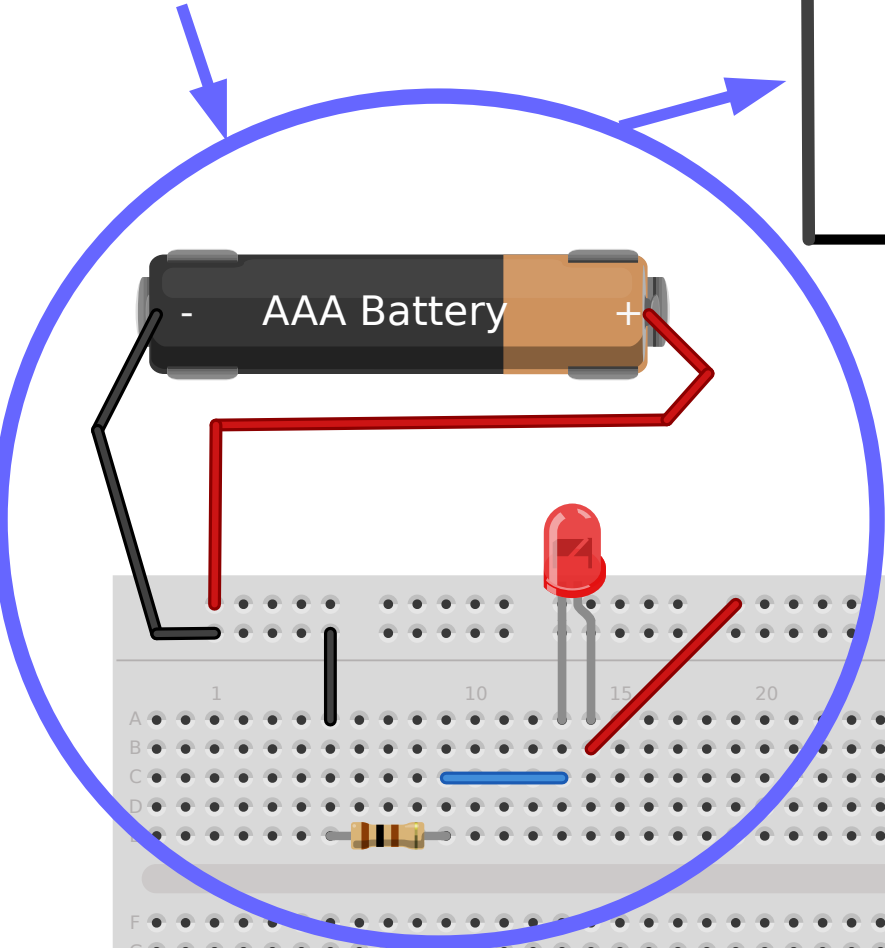


# Protoboard: Exemplos

**Circuito OK!**



**CUIDADO!!!  
CURTO CIRCUITO!!!**



# Pisca LED externo

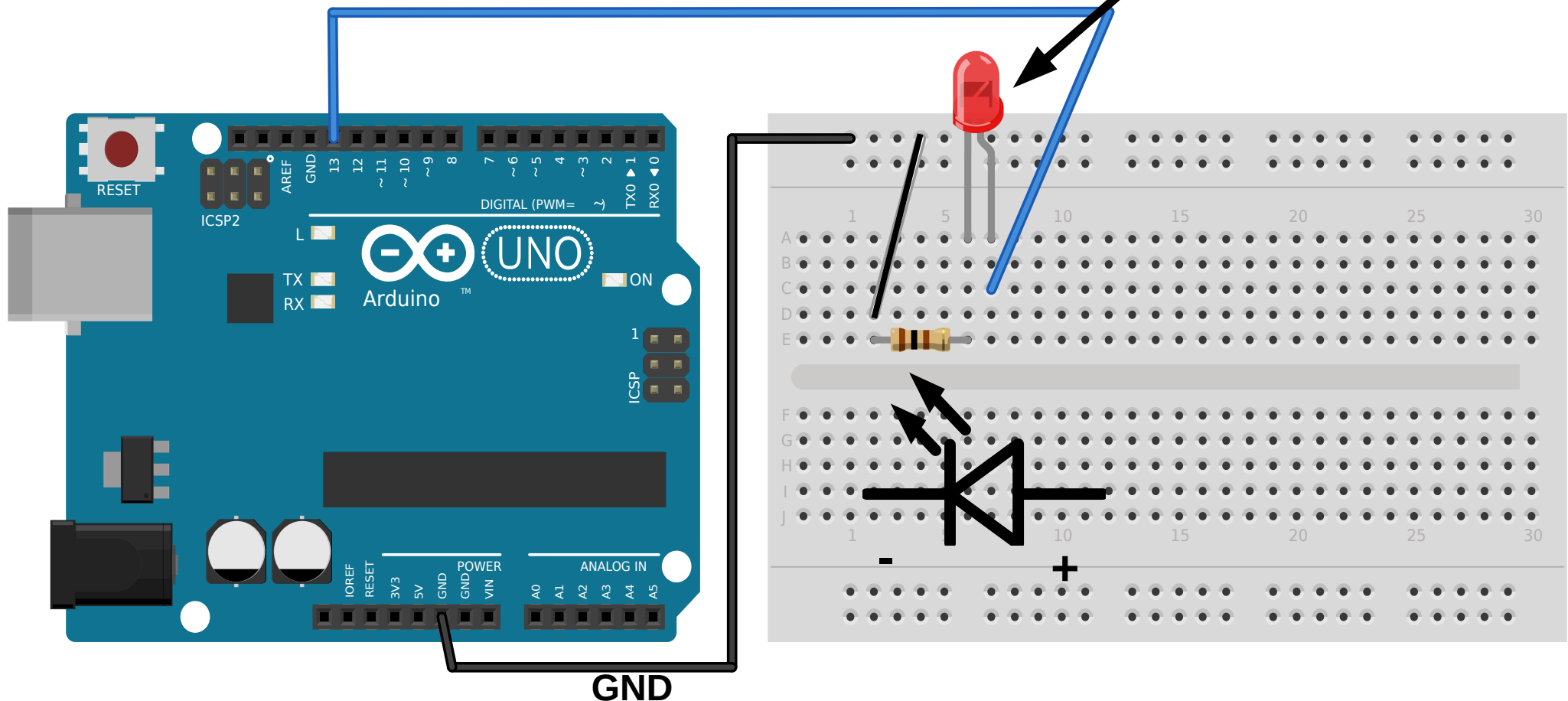
## → Materiais:

- ✓ 1 LED
- ✓ 1 resistor de 100 ohms

Componente **polarizado**:  
perna mais longa  
deve ser ligada no 5 V  
(positivo)

## → Montagem:

13 (digital)



Fonte:

Oficina de introdução ao Arduino do Centro de Tecnologia Acadêmica

# Entrada Digital

- `pinMode(CHAVE,INPUT);`
  - `bool estadopino=digitalRead(CHAVE);`
- Resistores pull-up
  - Para evitar um estado indefinido, as entradas digitais possuem um resistor chamado pull-up
  - Para ativar o resistor pull-up, escreva HIGH na porta de entrada:
    - `digitalWrite(CHAVE,HIGH)`
- Exercício: Monte um programa que lê uma entrada digital e acenda o LED de acordo com a leitura

# Comunicação serial

- Permite receber e enviar informações entre arduino e um computador
- Deve ser configurado na função setup:
  - `Serial.begin(9600); // Inicializa a porta serial para uma taxa de 9600 bits por segundo`
- Dados são enviados para o PC pelo comando
  - `Serial.println("Olá Mundo!!");`
  - Na interface IDE utilize o monitor serial
- Exercício: Altere o programa do LED para avisar ao computador quando ele é aceso e quando é apagado

# Entrada analógica

- Entradas Analógicas convertem uma tensão de 0 a 5 volts para um valor proporcional de 0 a 1023 (10 bits)
  - Sensor de temperatura, divisores de tensão, fotodiodo (sensor de luz)
  - Questão: qual a menor variação de V detectável pelo Arduino?
- Utilização das portas analógicas:
  - A leitura pode ser gravada em uma variável:  

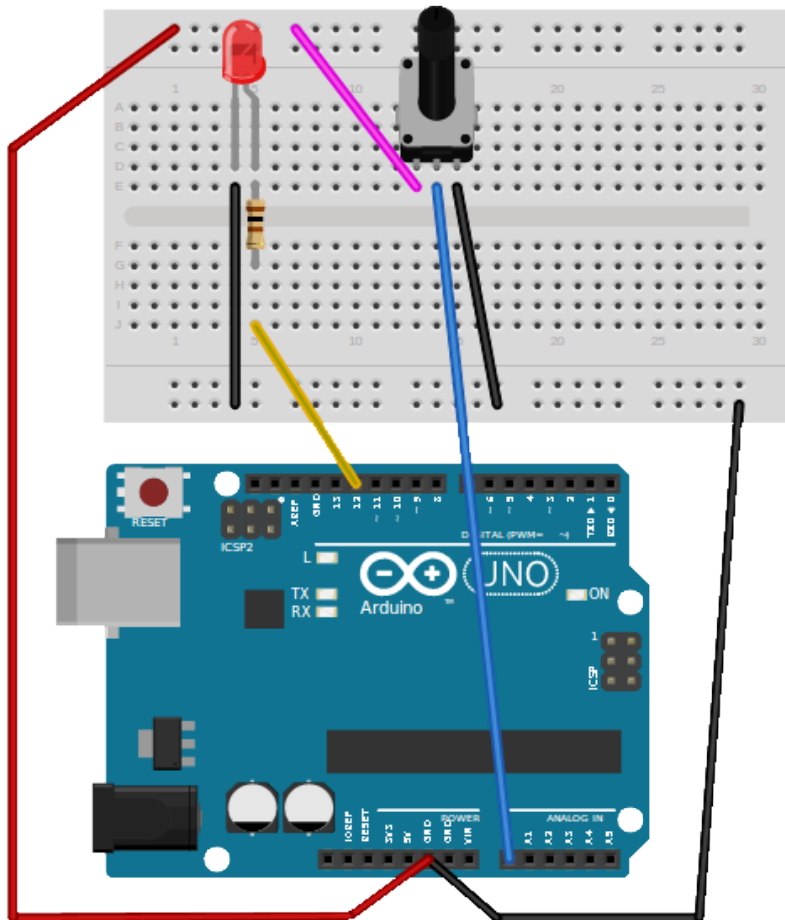
```
int valor_sensor = analogRead(A0);
```
  - Exercício: monte um circuito divisor de tensão e envie o resultado para o computador

# Projeto: Controla Pisca LED

## → Materiais:

- ✓ 1 Potenciômetro (resistor variável)

## → Montagem:



## → Código:

```
#define PINO_LED      12      // pino digital
#define PINO_POT      0      // pino analogico

int valor_pot;

void setup()
{
    // prepara uma comunicação serial
    Serial.begin(9600);
    pinMode(PINO_LED, OUTPUT);
}

void loop()
{
    // retorna um valor entre 0 e 1023
    valor_pot = analogRead(PINO_POT);
    // manda p/ USB (ver com Monitor Serial)
    Serial.println(valor_pot);

    digitalWrite(PINO_LED, HIGH);
    delay(valor_pot);
    digitalWrite(PINO_LED, LOW);
    delay(valor_pot);
}
```



# Saídas PWM (analógica?)

- PWM = Pulse Width Modulation, modulação por largura de pulso.
  - É uma maneira de gerar saídas analógicas simples a partir de portas digitais.
  - 8 bits de resolução
    - 256 possibilidades
  - Uso:  
`analogWrite(ledPin, valor)`
  - Exercício: Led com brilho variável (ajustável), baseado no exercício anterior

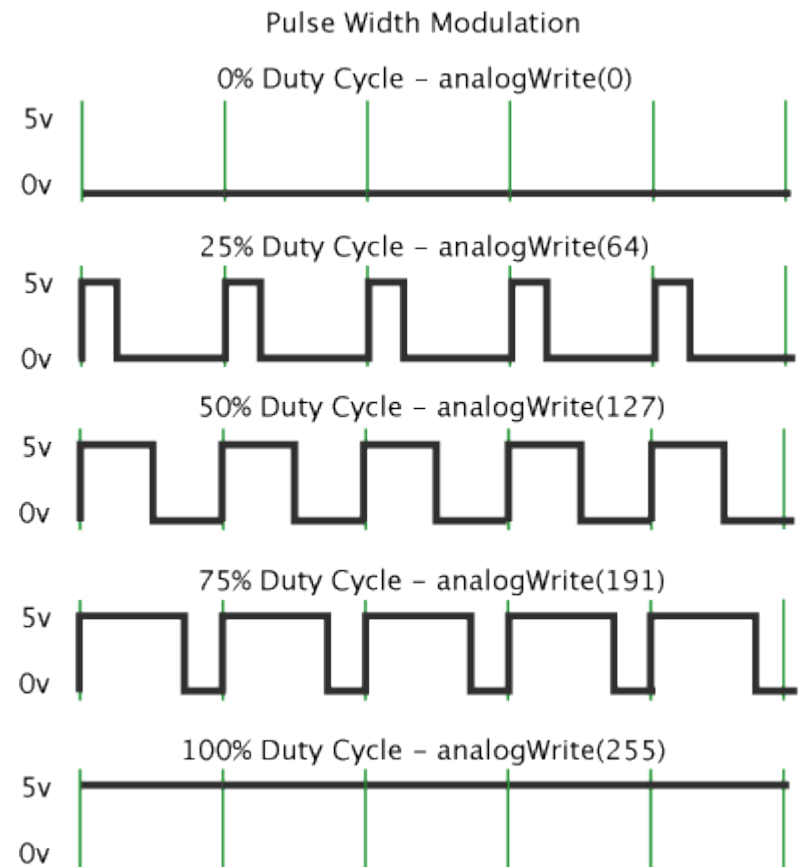


Imagem:

<http://www.arduino.cc/en/Tutorial/PW>

# Enviando comandos aos Arduino

- Exemplo de programa que permite enviar comandos ao arduino
  - Utilizado nas estação **Meteorolog**
- Exercício: Escreva um programa com pelo menos três funções que faça uso das portas digitais (entrada e saída), entrada analógica e saída PWM e forneça respostas para o computador

```
void setup() {  
  Serial.begin(115200);  
  // Configure os pinos!  
}  
void loop() {  
  if (Serial.available())  
  {  
    switch (Serial.read())  
    {  
      case 'a':  
        funcao_a();  
        break;  
      case 'b':  
        funcao_b();  
        break;  
      default:  
        break;  
    }  
  }  
}  
  
void funcao_a() {  
  //Código função_a aqui  
}  
  
void funcao_b() {  
  //Código função_b aqui  
}
```

# Automatizando a aquisição com python

- Para automatizar a aquisição utilizando este protocolo é necessário um programa que envie os comandos e salve os registros
- Veja esqueleto de [meteorolog.py](#)
- Alternativamente, o microcontrolador pode estar programado para enviar os resultados periodicamente

# Mais?

Sobrou tempo?

Faça o projeto 4: Buzzer de Luz da [oficina do CTA](#)

- Adquira um kit Arduino
- Aprenda a usar sensores e displays
- Conheça [Arduino.cc](#)
  - extensiva documentação
  - suporte da comunidade de usuários (forum)
- Saiba ler os Datasheet dos componentes!
- Busque informação em oficinas complementares
  - [Centro de Tecnologia Acadêmica IF/UFRGS](#)
  - [EITCHA!](#)