

*Professor Marco Câmara*

---

# Sistemas Embarcados

UCSAL 2024-01

---

# Aula 01

---

# Definição

---

Um **sistema embarcado** é um sistema computadorizado construído para uma aplicação específica. Por isto mesmo, normalmente não suporta itens que não estejam relacionados à execução da aplicação. Seu *hardware* e *software* têm limitações.

O *hardware* normalmente é mais lento para reduzir o consumo de energia, e oferece apenas recursos mínimos essenciais para suporte ao seu conjunto específico de periféricos.

O *software* tipicamente precisa ter comportamento determinístico (sempre oferece o mesmo tempo de execução), e, em certos casos, em tempo real (reagindo sempre de forma imediata a eventos). Em alguns sistemas, o *software* precisa ser tolerante a falhas, com degradação controlada (considere por exemplo um satélite ou um marcador preso ao corpo de uma baleia). Em outros, o *software* precisa comunicar o erro imediatamente após o primeiro sinal de falha(considere um monitor cardíaco).

---

# Definição

---

Um **sistema embarcado** é um sistema eletrônico microprocessado que, após ser programado, possui uma função específica que geralmente não pode ser alterada. Uma impressora, por exemplo, mesmo possuindo um processador que poderia ser utilizado para qualquer tipo de atividade, tem sua funcionalidade restrita apenas à impressão de páginas. Um computador de propósito geral, no entanto, pode ser utilizado num instante como um ambiente de entretenimento, em outro como estação de trabalho, ou até mesmo um telefone.

---

# Definição

---

Um **sistema embarcado** é um sistema eletrônico microprocessado, completamente encapsulado, dedicado ao dispositivo ou sistema que ele controla. Diferentemente de computadores de propósito geral, como o computador pessoal, um sistema embarcado realiza um conjunto de tarefas predefinidas, geralmente com requisitos específicos. Já que o sistema é dedicado a tarefas específicas, através de engenharia pode-se otimizar o projeto reduzindo tamanho, recursos e custo do produto.

---

# Definição (+ algumas)

---

## Interface com o usuário

*Sistemas embarcados podem operar sem nenhuma ou muito pouca interação com seus usuários.*

## Ominipresentes

*Praticamente todo equipamento atual é um sistema embarcado;*

*Carregadores de celular, eletrodomésticos, semáforos, automóveis, aviões, motores industriais, equipamentos médicos etc.*

## Diversos níveis de complexidade

*Sistemas embarcados controlam desde um eletrodoméstico simples, como um liquidificador, até um foguete recuperável da SpaceX.*

---

# Motivação e história

---

Sempre houve demanda pela automação de tarefas repetitivas, por diversos motivos. Entre eles:

*Aumento da performance*

*Redução de Custos*

*Padronização*

*Segurança*



---

# Motivação e história

---

Sempre houve demanda pela automação de tarefas repetitivas, por diversos motivos. Entre eles:

*Aumento da performance*

*Redução de Custos*

*Padronização*

*Segurança*

A revolução industrial praticamente foi definida por esta demanda

*Autômatos complexos serviram de inspiração para as máquinas de tear*

*Os autômatos durante algum tempo eram dispositivos mecânicos: roda d'água, relógio, semáforos, e até robôs !*



---

# Motivação e história

---

Na verdade, a máquina de tear foi inventada por William Lee, na Inglaterra, em 1589. A rainha Isabel I, no entanto, negou o desenvolvimento da máquina, que segundo ela traria ruína para seus súditos.

William levou o invento para a França, onde funcionou até 1610, quando o rei Henrique IV da França, que apoiava a idéia, morreu. O projeto acabou ali, renascendo apenas em 1780, desta vez movido a vapor.

Embora inicialmente tenha gerado desemprego, a máquina de tear provocou uma revolução positiva nas áreas social, econômica e humana.

---

# Motivação e história

---

Sempre houve demanda pela automação de tarefas repetitivas, por diversos motivos. Entre eles:

*Aumento da performance*

*Redução de Custos*

*Padronização*

*Segurança*

A revolução industrial praticamente foi definida por esta demanda

*Autômatos complexos serviram de inspiração para as máquinas de tear*

*Os autômatos durante algum tempo eram dispositivos mecânicos: roda d'água, relógio, semáforos, e até robôs !*

Dispositivos mecânicos acabaram sendo substituídos por eletrônicos, e depois digitais, e finalmente microprocessados

*Redução de custos pela escala de produção;*

*Maior simplicidade de projeto, fabricação e manutenção.*

# Motivação e história

Sempre houve demanda pela automação de tarefas repetitivas, por diversos motivos. Entre eles:

*Aumento da performance*

*Padronização*

A revolução industrial pr

*Autômatos complexos servir*

*Os autômatos durante algun*

*relógio, semáforos, e até rob*

Dispositivos mecânicos aca

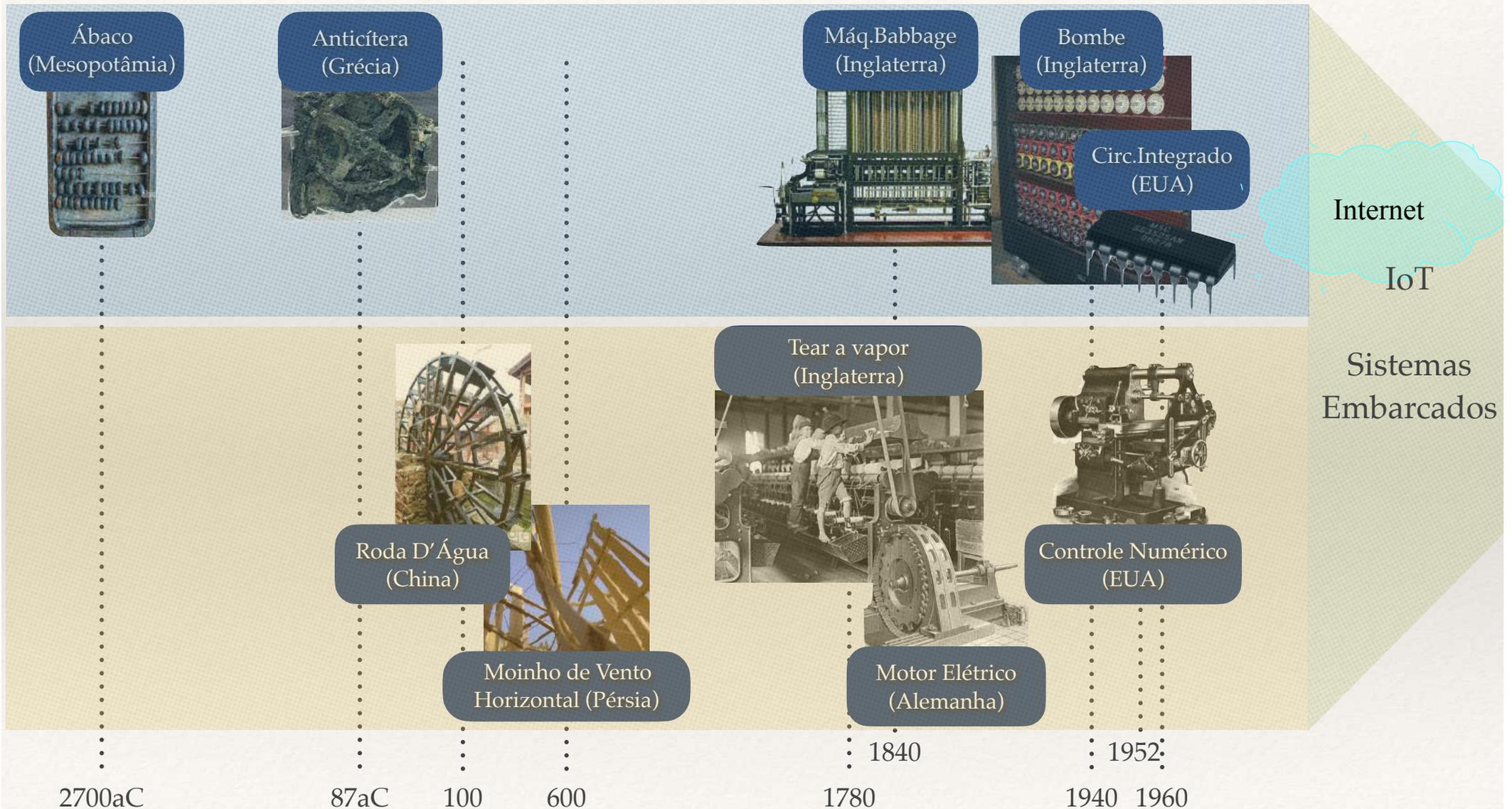
eletrônicos, e depois digitais, e finalmente **microprocessados**

*Redução de custos pela escala de produção;*

*Maior simplicidade de projeto, fabricação e manutenção.*

Alguns autores adotam uma classificação dos sistemas embarcados em função da arquitetura dos processadores (pequena ou média escala, ou "complexa). Outros usam a "geração" dos dispositivos (1ª a 4ª gerações). Como este tipo de classificação inevitavelmente ficará defasada à medida em que evoluem os dispositivos, não adotaremos esta estratégia.

# Um pouco de história



# Classificação de Sistemas embarcados

Autônomos ou “isolados”  
Em rede;  
De tempo real;  
Móveis.



\* As classificações não são mutuamente exclusivas

---

# Classificação de Sistemas embarcados

---

## Autônomos ou “isolados”

São independentes, não precisam de servidores ou acesso à internet;

Recursos de rede, quando existem, são locais, para acesso a sensores, atuadores, ou a sub-sistemas.

Em rede;

De tempo real;

Móveis.



# Classificação de Sistemas embarcados

Autônomos ou “isolados”;

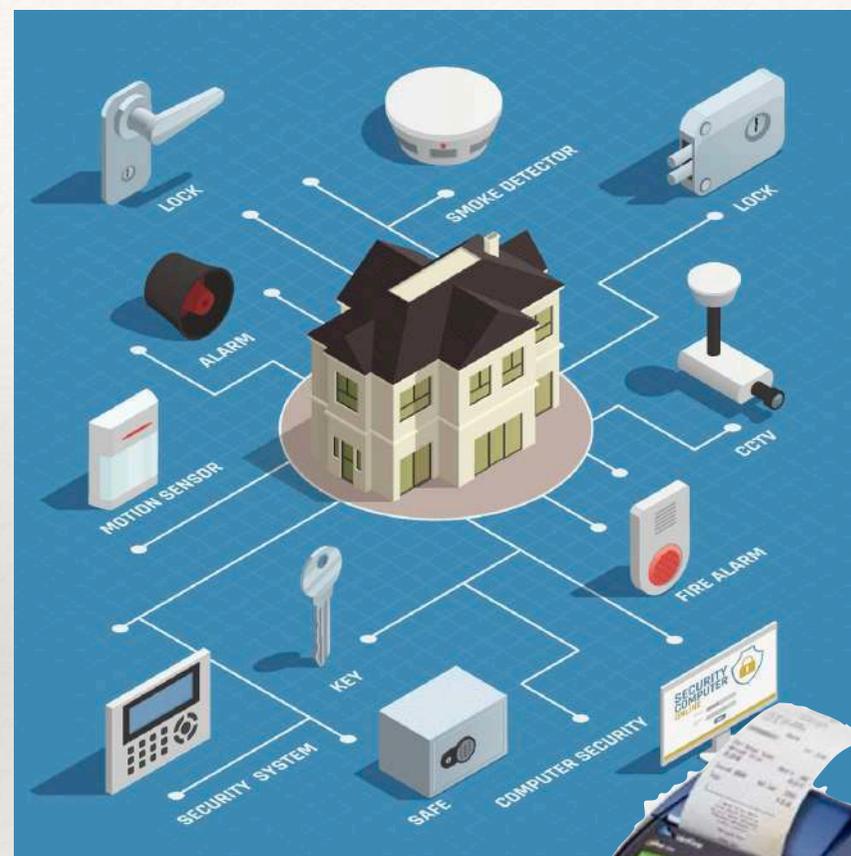
Em rede

Dependem de recursos externos, seja como servidores, ou mesmo para comunicação com outros sistemas independentes;

Incorporam a **IoT (Internet das Coisas)**, que devido à sua abrangência e diversidade, deve-se tratar como um tema à parte;

Representam um desafio para a segurança.

De tempo real;



---

# Classificação de Sistemas embarcados

---

Autônomos ou “isolados”;

Em rede;

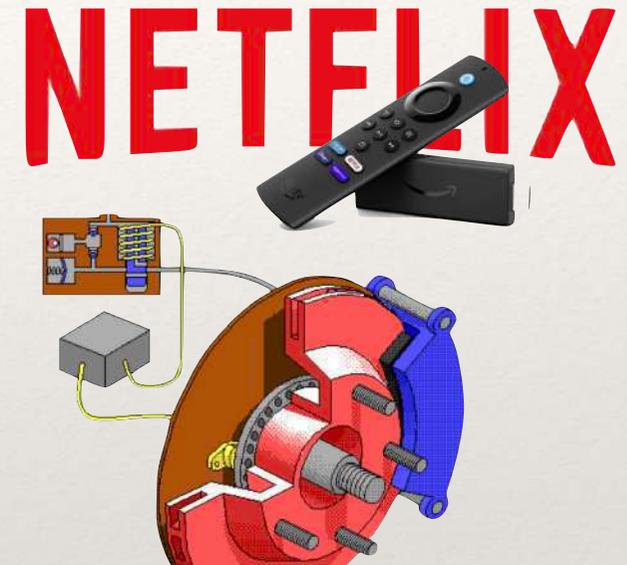
De tempo real

Precisam obedecer a requisitos de tempo de resposta:

**Sistemas *Soft Real Time*** operam mesmo diante de falhas na obediência a estes requisitos;

**Sistemas *Hard Real Time*** têm requisitos rígidos, que não podem ser desobedecidos.

Móveis.



---

# Classificação de Sistemas embarcados

---

Autônomos ou “isolados”

Em rede;

De tempo real;

Móveis

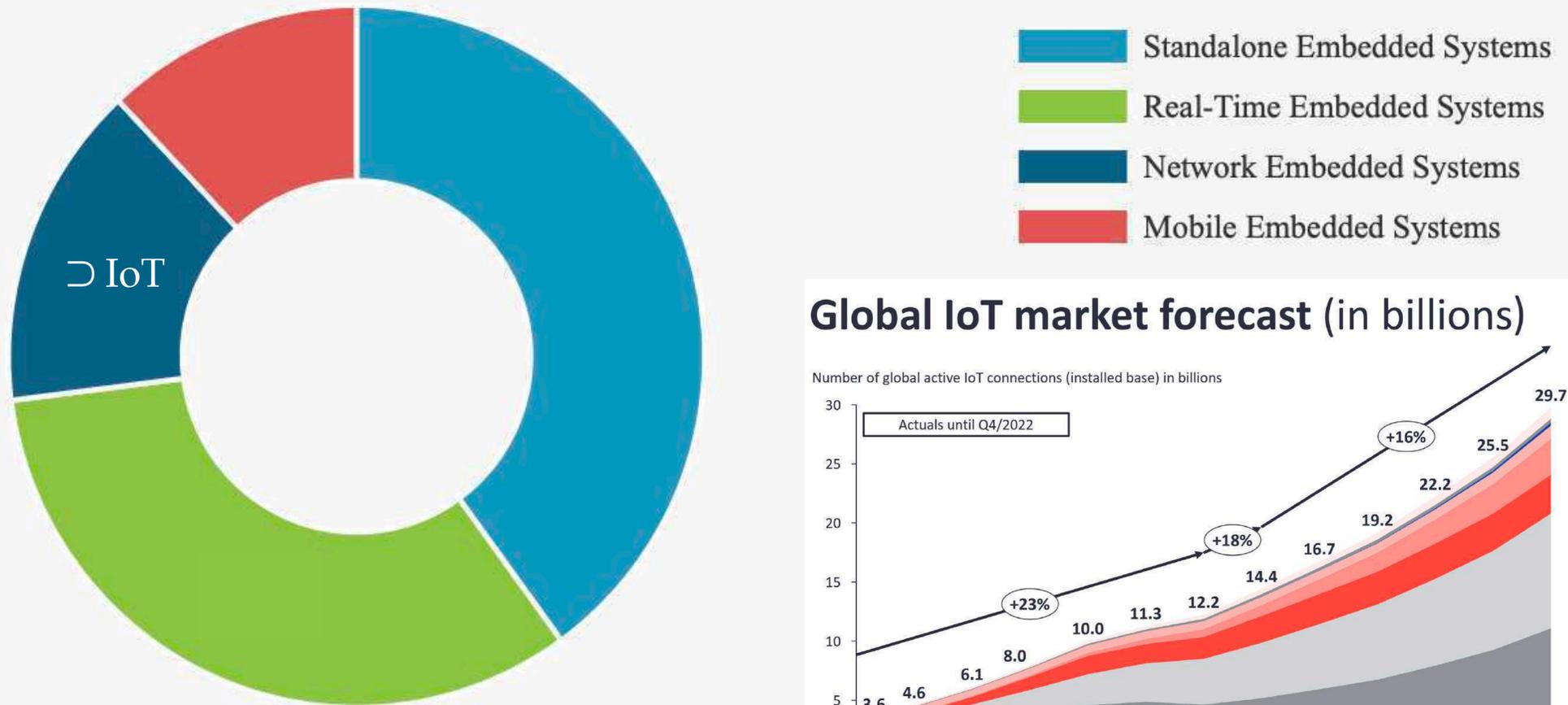
Tipicamente estão em dispositivos de pequeno porte, portáteis e com restrições severas de consumo energético;

Celular é um sistema embarcado !!?



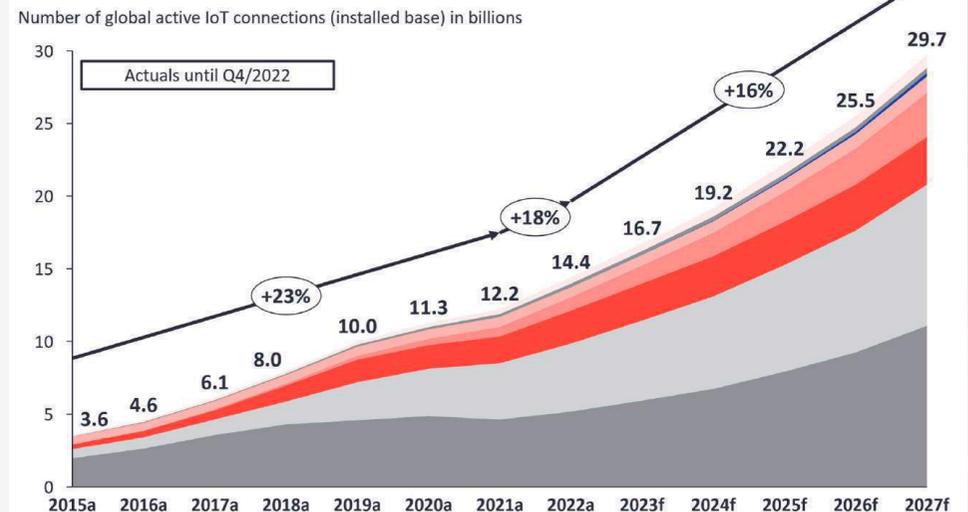
# Classificação de Sistemas embarcados

## Global Embedded Systems Market Share, By Type, 2022



[www.fortunebusinessinsights.com](http://www.fortunebusinessinsights.com)

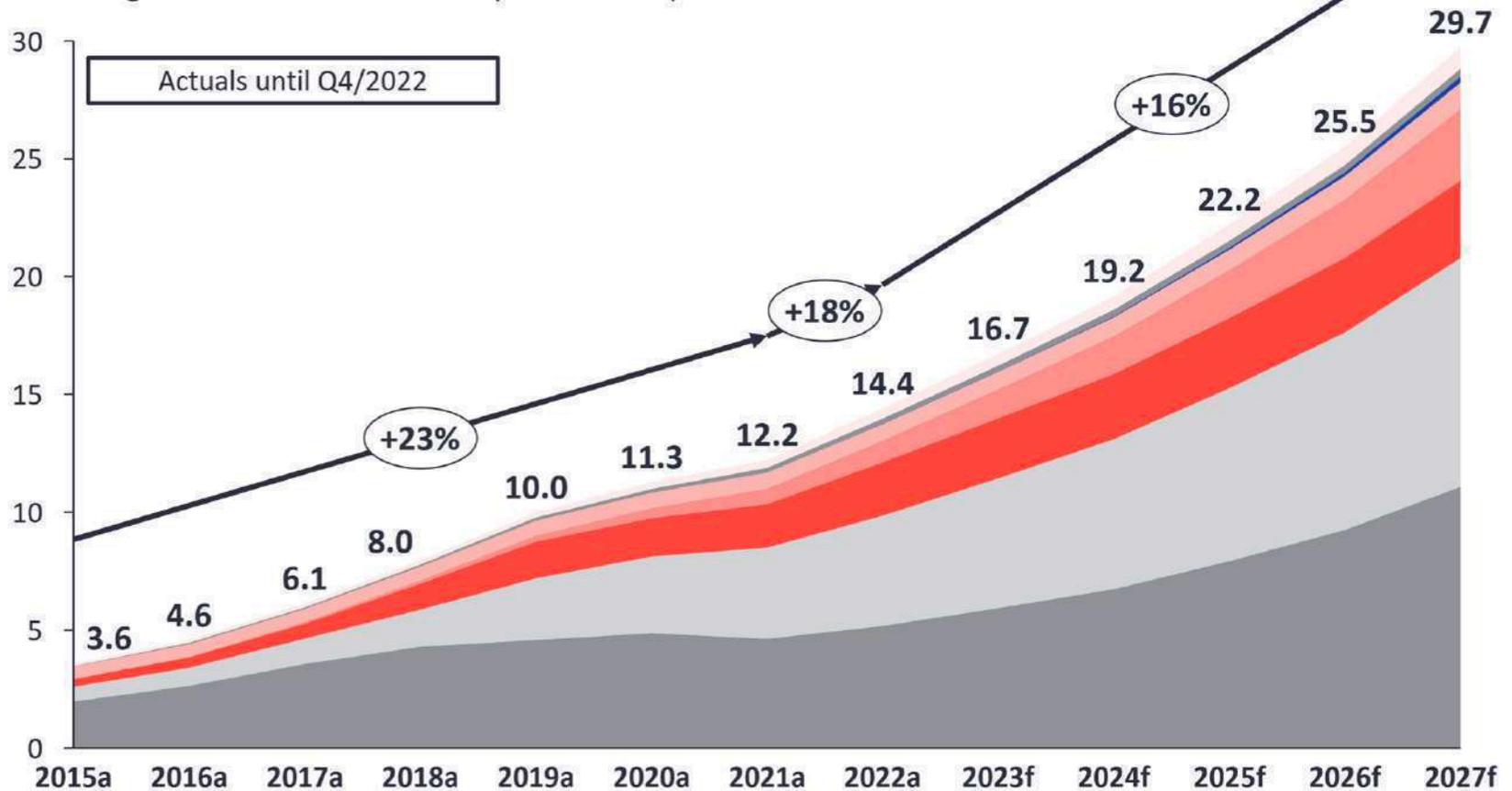
## Global IoT market forecast (in billions)



# Classificação de Sistemas embarcados

## Global IoT market forecast (in billions)

Number of global active IoT connections (installed base) in billions



www.

---

# Aplicações de Sistemas embarcados

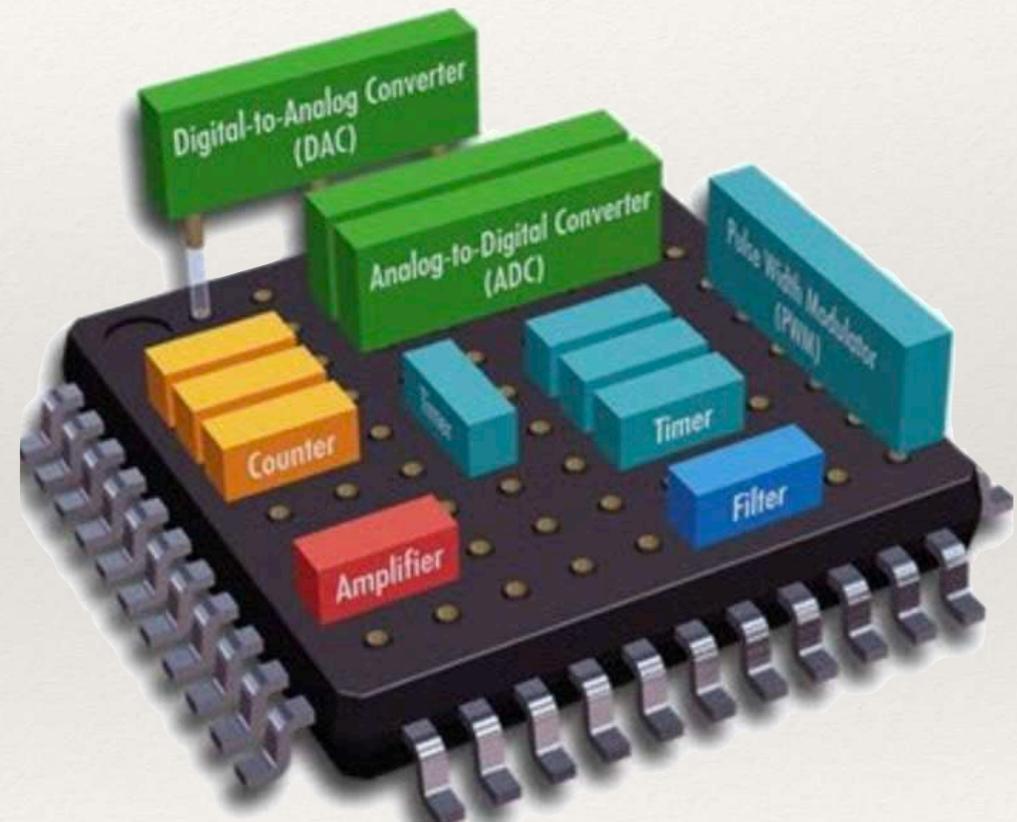
---

Automotivas

Industriais

Médicas

Outras aplicações



# Aplicações de Sistemas embarcados

## Automotivas

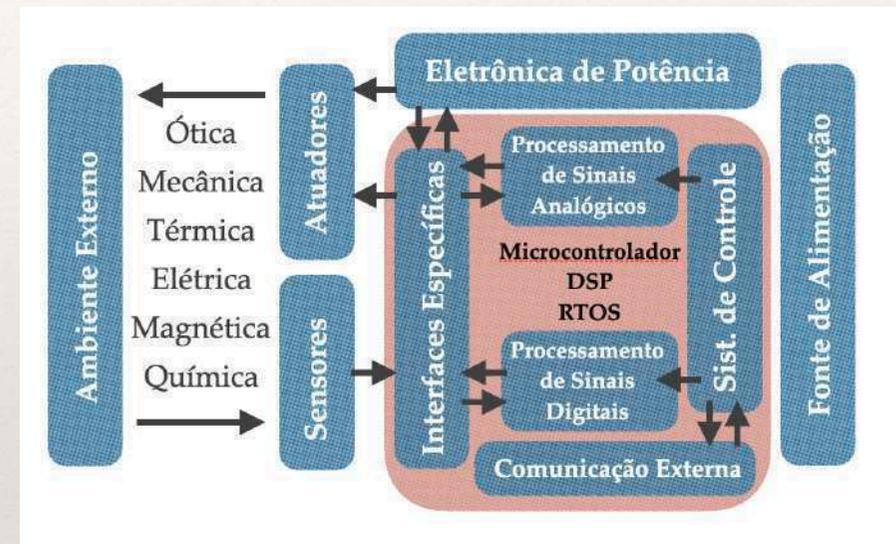
Aplicação comum dos Sistemas embarcados;

Envolve praticamente todos os componentes de um sistema completo;

Nos anos 70, o único equipamento eletrônico de um automóvel era o rádio !

Industriais

Médicas



# Aplicações de Sistemas embarcados

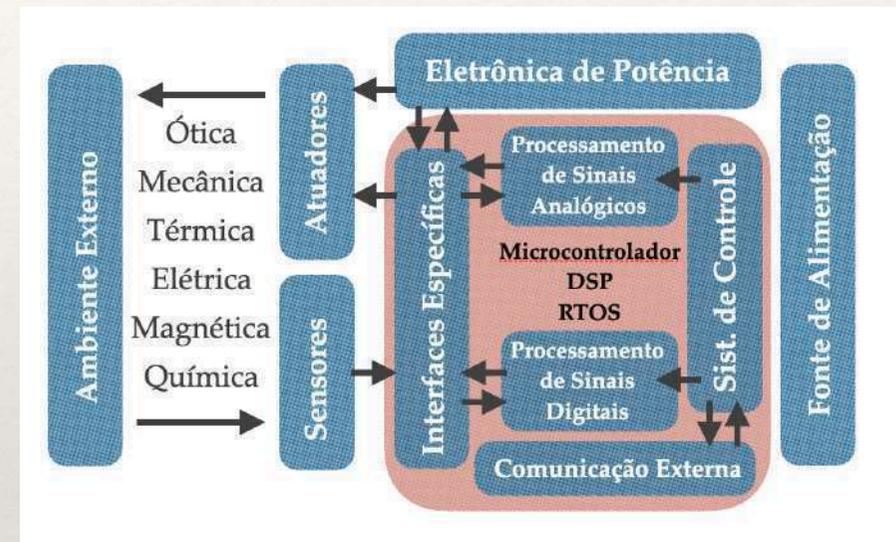
## Automotivas

Hoje qualquer automóvel possui um Sistema embarcado (o Celta já tinha !);

Temos ignição e injeção eletrônica, computador de bordo, ABS, EBD, Airbag, suspensão ativa, travas elétricas, direção elétrica assistida, travas elétricas, ar-condicionado, *cruise control* ...

Tipicamente são múltiplos sistemas de diferentes fornecedores, interligados.

## Industriais



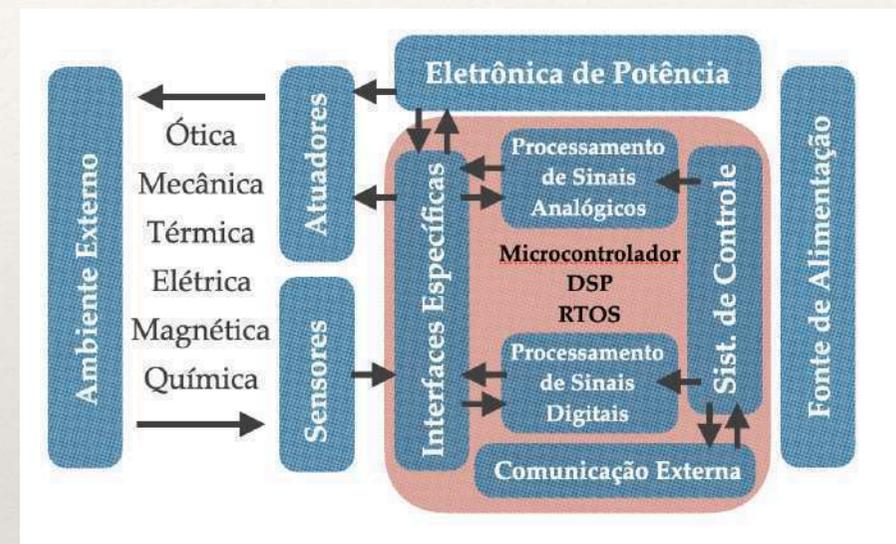
# Aplicações de Sistemas embarcados

## Automotivas

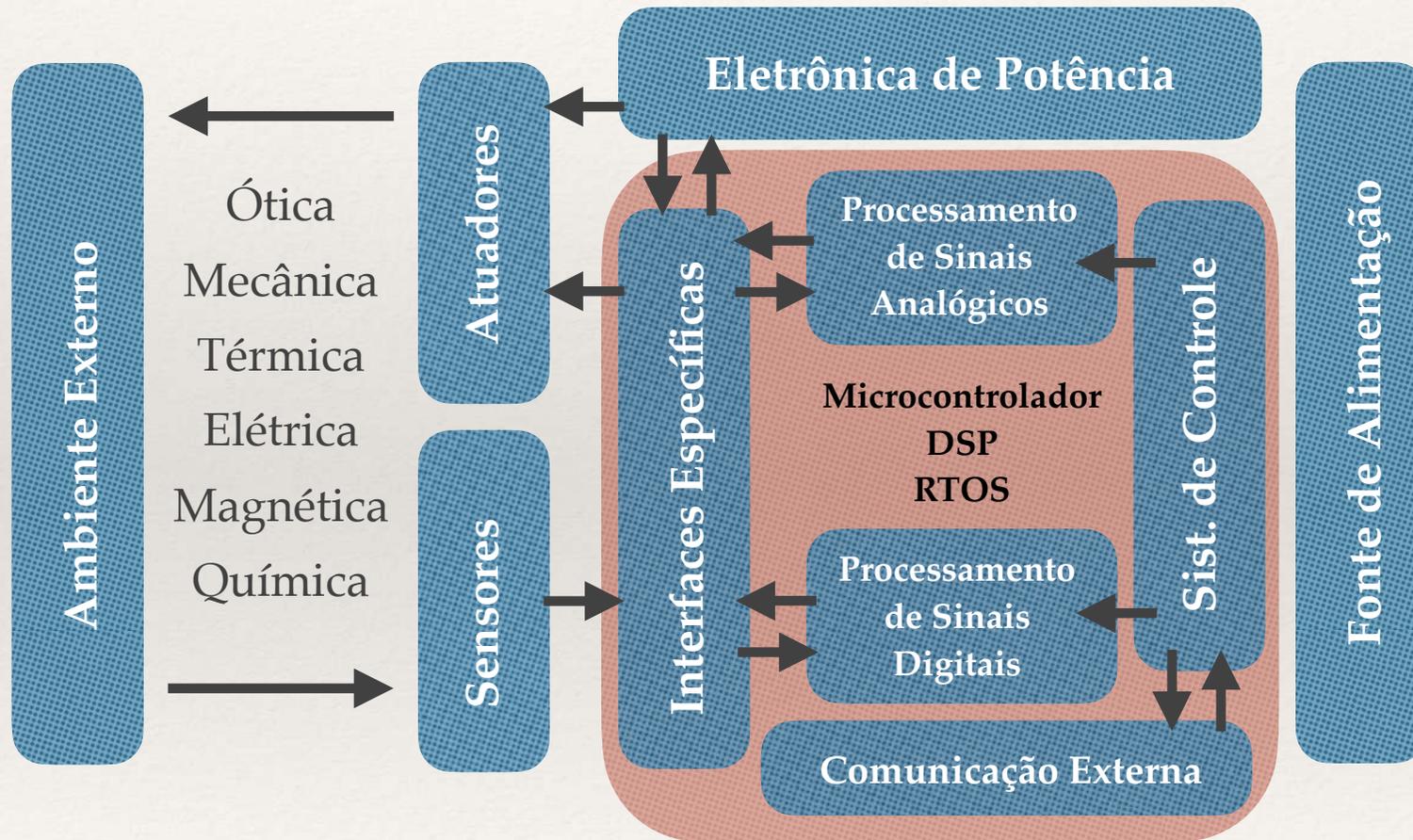
“Montadora” produz carroceria e motor. Os demais componentes são de fabricantes diversos (painel da Visteon, Freio da Bosch, Controle do Motor da Magnet Marelli, Ar-condicionado da Denso etc). Isto exige uma rede padronizada interligando os componentes do Sistema embarcado;

Protocolo CAN, porta OBD...

## Industriais



# Aplicações de Sistemas embarcados



---

# Aplicações de Sistemas embarcados

---

Automotivas

Industriais

Especialistas identificam quatro Revoluções Industriais. A primeira ocorreu no final do século 18. A segunda, no final do século 19, está relacionada à aplicação da eletricidade e a criação da linha de montagem;

É na terceira, do final do século 20, que foram adotados os Sistemas embarcados, inclusive com os primeiros esforços de desenvolvimento da **Internet das Coisas (IoT)**;

Especialistas dizem que estamos passando pela 4ª Revolução Industrial, a chamada **Indústria 4.0**. Os destaques agora são a CPS, a AI, o ML, o Big Data e a Robótica.

Médicas

---

# Aplicações de Sistemas embarcados

---

Automotivas

Industriais

Especialistas identificam quatro Revoluções Industriais. A primeira ocorreu no final do século XVIII, relacionada à aplicação da eletricidade

*Cyber-Physical Systems*: o conceito envolve a **integração** dos dispositivos e máquinas físicos com sistemas digitais "inteligentes". Esta tecnologia permite o monitoramento em tempo real, análise e controle dos processos industriais.

relacionada à aplicação da eletricidade

lotados os Sistemas embarcados, o desenvolvimento da **Internet das Coisas (IoT)**;

Em 1970, com o advento da 4ª Revolução Industrial, a chamada **Indústria 4.0**. Os destaques agora são a **CPS**, a AI, o ML, o Big Data e a Robótica.

Médicas

---

# Aplicações de Sistemas embarcados

---

Automotivas

**Industriais**

A Indústria 4.0 obviamente considera a evolução do IoT, inclusive integrada à Computação em Nuvem;

Outra tecnologia considerada em alguns casos é a *Digital Twin*, para simulação, testes e manutenção preditiva;

Aplicações Industriais possuem exigências específicas quanto à operação em ambientes ruidosos, segurança e estabilidade.

Médicas

---

# Aplicações de Sistemas embarcados

---

Automotivas

Industriais

Médicas

Requisitos críticos de estabilidade e segurança;

Utilizados em *smartwatches*, além de dispositivos para exames, monitoramento, geração de sinais vitais, próteses, robôs cirúrgicos etc.

Outras aplicações





# Aula 02

## Características

As principais características da Internet das Coisas são:

- A IoT pode ser caracterizada como uma rede mundial de coisas/objetos/dispositivos interconectados que se comportam como entidades ativas [Roman et al. 2011b];
- As coisas (dispositivos) na IoT, muitas vezes, possuem restrições de recursos como memória RAM ou ROM, poder de processamento e energia [Hummen et al. 2013];
- Mecanismos de comunicação de alguns dispositivos, na maioria das vezes sem fio, possuem baixa potência de transmissão e baixa taxa de dados [Mahalle et al. 2010];

## Características

As principais características da Internet das Coisas são:

- Integra coisas (dispositivos) heterogêneos, o que demanda uma preocupação em relação a interoperabilidade entre estes [Atzori et al. 2010, Mahalle et al. 2012];
- Pode ser caracterizada como um ambiente contendo um grande número computadores ou dispositivos invisíveis que colaboram com o usuário, ou seja, um ambiente pervasivo e ubíquo [Hanumanthappa e Singh 2012];
- Na IoT, os usuários podem interagir com as coisas em seu ambiente físico e virtual de diversas maneiras [Mahalle et al. 2012].

## Características

Informações retiradas do artigo:

<https://wiki.inf.ufpr.br/maziero/lib/exe/fetch.php?media=ceseg:2013-sbseg-mc4.pdf>

- A rede possui uma topologia dinâmica, pois muitos nós entram e saem da rede com frequência [Mahalle et al. 2012, Hanumanthappa e Singh 2012];
- Há uma grande quantidade de coisas (dispositivos) com ciclo curto de vida, o que exige uma alta capacidade de gerenciamento [Fongen 2012];

---

# IoT e as Redes

---

É inegável que a tecnologia IoT depende da qualidade das redes de comunicação, embora existam outros aspectos também importantes, como os Sistemas Operacionais e aplicações suportadas;

A tecnologia IoT adota múltiplos protocolos, alguns já existentes antes do seu surgimento, como o Wi-Fi, e outros bem ajustados às suas características:

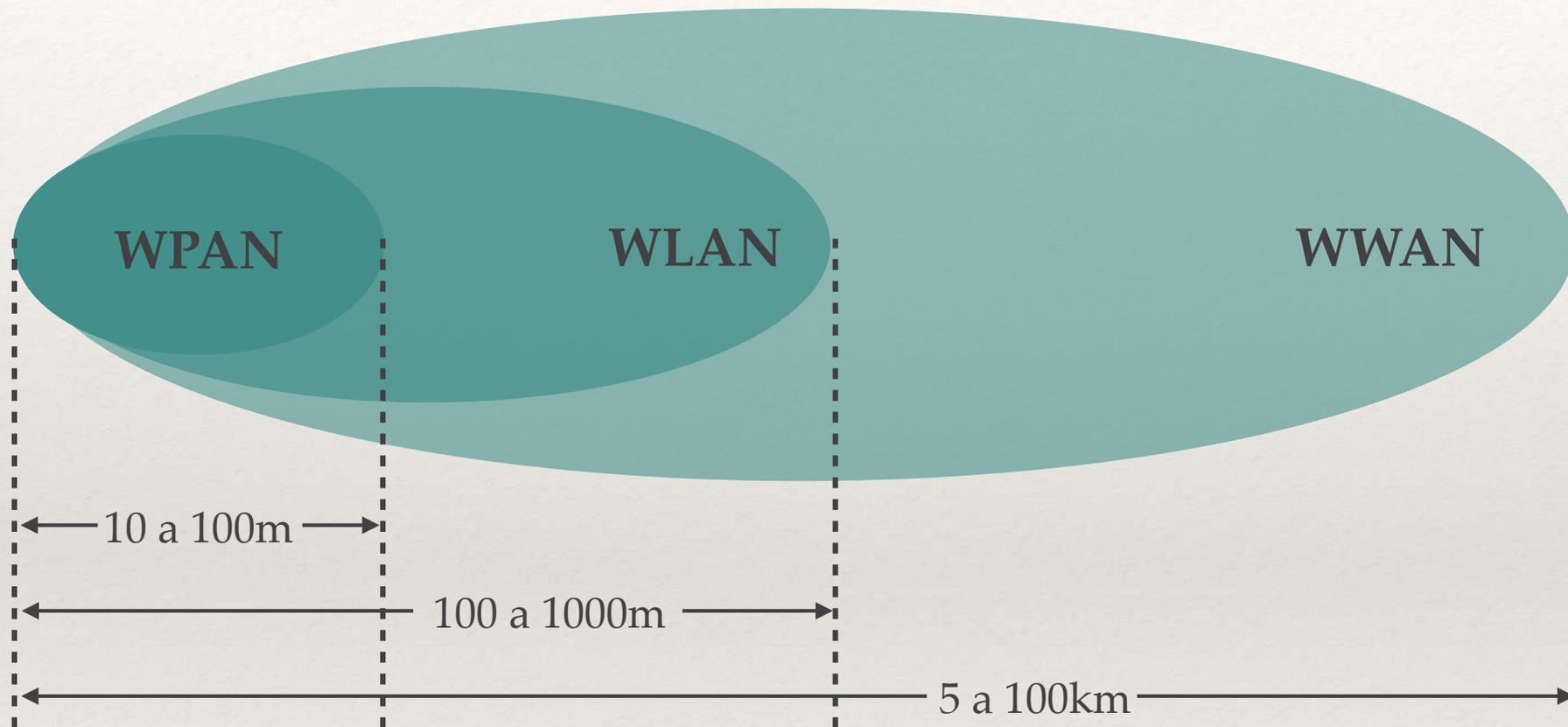
Baixa demanda de largura de banda;

Baixo consumo energético;

Acesso típico sem fio.

Tecnologias como Bluetooth, RFID, ZigBee, NFC e LoWPAN estão entre elas.

# IoT e as Redes (alcance)



A IoT normalmente está associada a redes com baixo alcance. No entanto, alguns protocolos mais recentes têm expandido seu alcance, como a LoRaWAN, por exemplo;

Novas aplicações como telemetria, medição de consumo residencial de água, luz e gás, gestão de trânsito e agropecuária podem ser possíveis.

---

# IoT X segurança

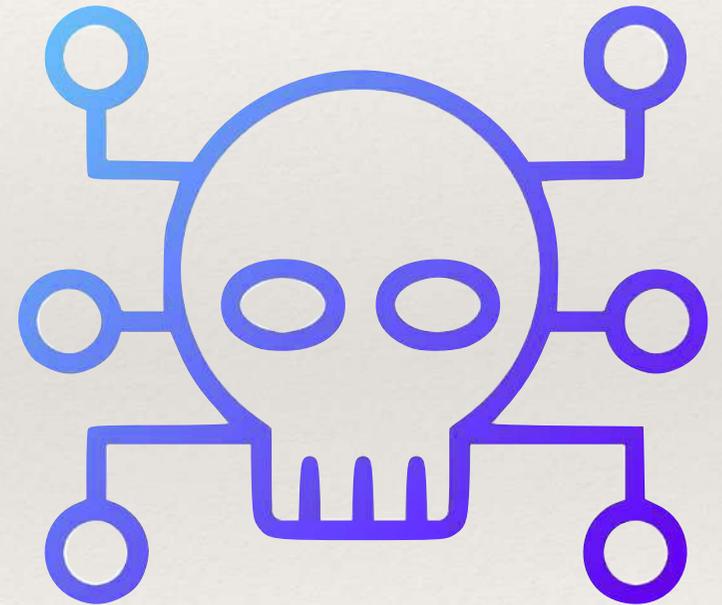
---

O conceito da IoT é conectar “tudo”, e quanto mais acessos à rede existirem, maiores são os riscos de segurança;

Métodos criptográficos seguros exigem processamento rápido e vigilância contínua, o que atenta contra as características dos dispositivos IoT !

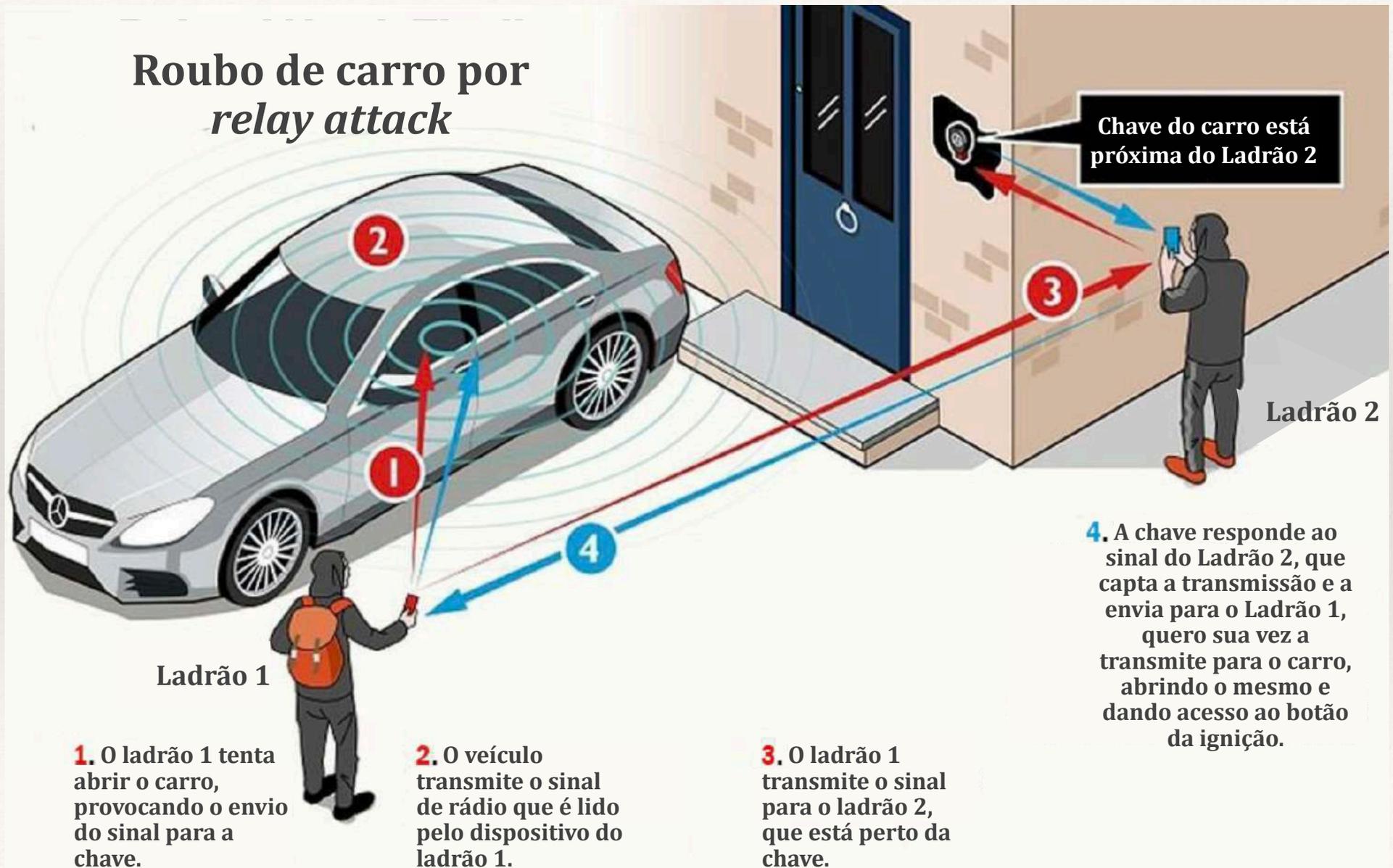
- Capacidade limitada de processamento;
- Baixo consumo energético.

São inúmeros os casos de crimes que exploram as vulnerabilidades dos sistemas IoT.



# IoT X segurança

## Roubo de carro por *relay attack*



# Furto de carros “sem chave” está se tornando comum no Brasil

Por [Felipe Demartini](#) | Editado por [Claudio Yuge](#) 08 de Abril de 2022 às 19h49



Uma brecha de segurança em modelos de veículos que utilizam a **tecnologia** keyless para permitir a abertura de portas e partida do motor sem a necessidade de uso da chave,

**1.** O ladrão 1 tenta abrir o carro, provocando o envio do sinal para a chave.



**2.** O veículo transmite o sinal de rádio que é lido pelo dispositivo do ladrão 1.

**3.** O ladrão 1 transmite o sinal para o ladrão 2, que está perto da chave.

Reserve Agora

## Voos para África

by flytap.com

Skip Ad ▶

o está  
drão 2

Ladrão 2

onde ao  
ção 2, que  
missão e a  
Ladrão 1,  
1 vez a  
ra o carro,  
mesmo e

ganho acesso ao botão  
da ignição.

# Como ladrões de carros estão virando 'hackers' para roubar modelos mais modernos

É importante manter o software do veículo atualizado, assim como já fazemos com telefones e computadores, para evitar invasões.



Por Doug Jacobson\*, BBC

18/08/2023 10h07 · Atualizado há 6 meses



abrir o carro, provocando o envio do sinal para a chave.

transmite o sinal de rádio que é lido pelo dispositivo do ladrão 1.

transmite o sinal para o ladrão 2, que está perto da chave.

Carros > Manutenção e Segurança

# 'Relay attack': em novo golpe, bandidos clonam a chave do carro a distância

Alessandro Reis • Do UOL, em São Paulo (SP)

05/10/2023 04h00



abrir o carro, provocando o envio do sinal para a chave.

transmite o sinal de rádio que é lido pelo dispositivo do ladrão 1.

transmite o sinal para o ladrão 2, que está perto da chave.

# IoT X segurança



Home > Segurança

# AirTags são 'a arma favorita de perseguidores', afirma processo contra a Apple

16/10/2023 às 10:57 • 1 min de leitura

COMPARTILHE

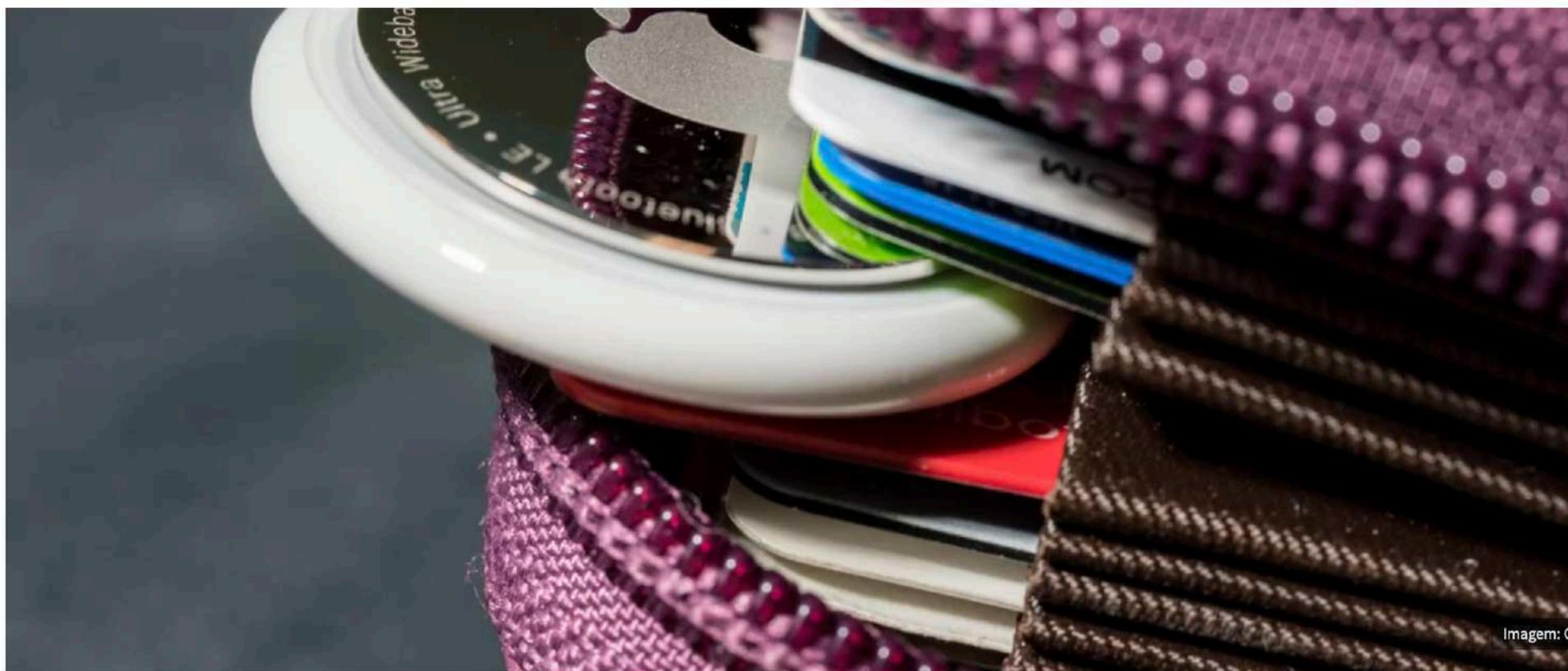


Imagem: Ge



# Atualização do Apple AirTag traz recurso 'anti-stalking' ao dispositivo de rastreamento

22 de dezembro de 2022 4

ma

COM



O **Apple Airtag** recebeu uma nova **atualização de firmware** com a chegada de um recurso anti-stalking. Ou seja, uma característica que pode ajudar as pessoas a não serem perseguidas ou vigiadas. Esse update é o **2.0.24 que chegou em novembro**, mas só em dezembro é que finalmente atingiu 100% dos aparelhos da maçã.

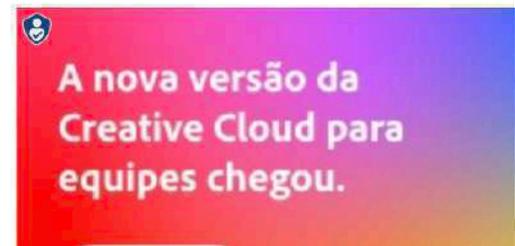


Imagem: Ge

[Notícias](#) > [Gadgets](#)

# Android ganha proteção contra stalkers que usam o Apple AirTag

Google libera ferramenta nativa que avisa sobre rastreadores por perto. Usuário também encontra dicas de como desativar o equipamento oculto.



Por [Thássius Veloso](#)  
10/08/2023 às 12:53



---

# IoT X segurança

---

Outras implementações também foram alvos de ataque, embora não sejam tão bem documentadas na imprensa.

Quais seriam os próximos alvos?

Sistemas Médicos (assassinato)?



---

# IoT X segurança

---

Outras implementações também foram alvos de ataque, embora não sejam tão bem documentadas na imprensa. Quais seriam os próximos alvos?

Sistemas Médicos (assassinato)?

Sistemas Aviônicos (terrorismo)?



# IoT X segurança

Outras implementações também foram alvos de ataque, embora não sejam tão bem documentadas na imprensa. Quais seriam os próximos alvos?

Sistemas Médicos (assassinato)?

Sistemas Aviônicos (terrorismo)?

É claro que é necessária precaução, mas também não devemos exagerar ...



---

# HW de Sistemas Embarcados

---

O HW de um sistema embarcado = Conjunto de componentes

Microcontroladores ou CLPs;      Sensores diversos;  
Atuadores;      Interfaces e protocolos de comunicação;  
Fontes de alimentação;      Componentes eletrônicos diversos.

Escolher corretamente cada um dos componentes é essencial

A portabilidade pode ser inviável;

*O hardware* determina os parâmetros de projeto, inclusive de *software*.

Outro fator importante são as condições ambientais

Acesso, proteção, temperatura, vibração, EMI etc

# Microcontroladores

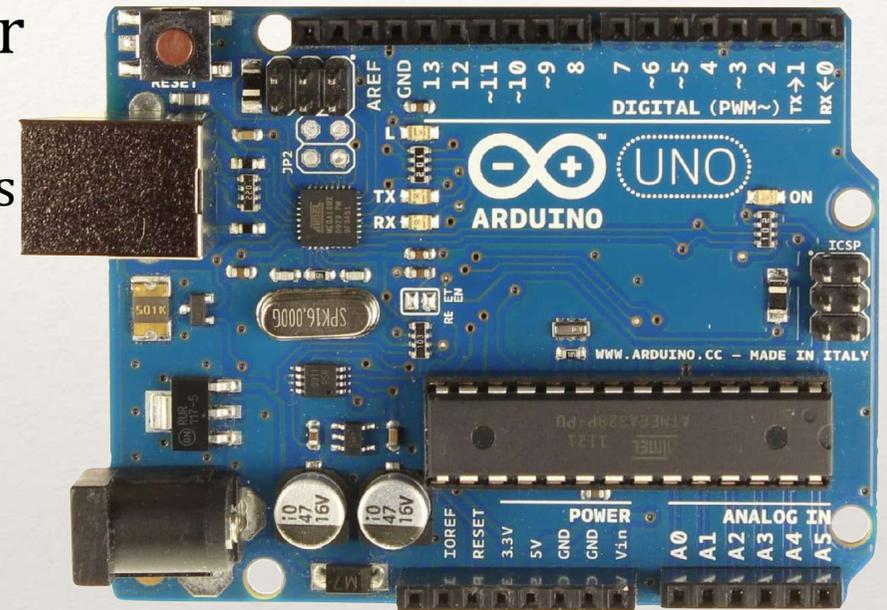
## Microcontrolador x Microprocessador

**Microprocessadores** são baseados na arquitetura von Neumann\*, enquanto que os **Microcontroladores** são baseados na arquitetura Harvard\*. Nesta última, a memória de programa está separada da memória de dados;

**Microprocessadores** tipicamente exigem diversos componentes auxiliares externos, inclusive a memória;

**Microcontroladores** tipicamente possuem características mais simples, são menores e consomem menos energia;

**Microprocessadores** possuem relógios (*clock*) muito mais rápidos e estáveis, oferecendo poder computacional mais elevado.



Placa de Protótipo Arduíno Uno

\* Diferentes Arquiteturas de Computadores

# Microcontroladores

## Tamanho e consumo

Estes são aspectos fundamentais em projetos embarcados, determinando inclusive a viabilidade, ou não, de algumas aplicações;

## Tamanho

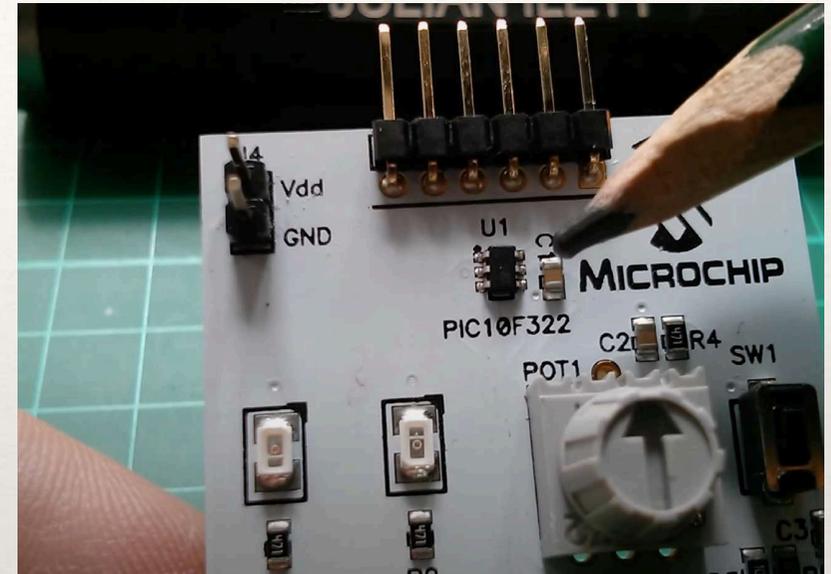
Atualmente, com a miniaturização em larga escala dos componentes, o número de componentes internos não impõe limites mínimos às dimensões do CI;

No entanto, reduzir o número de pinos afeta o tamanho significativamente, mas dificulta as interfaces, limitando as aplicações;

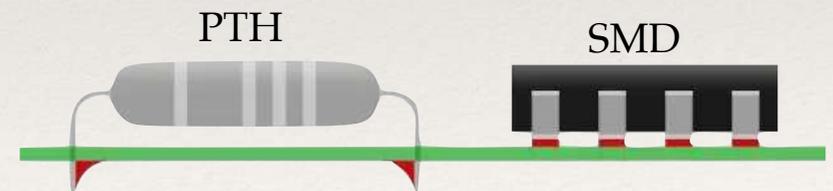
Outro fator é o tipo de pino (PTH ou SMD)

**PTH** - *Plated-Through Hole*

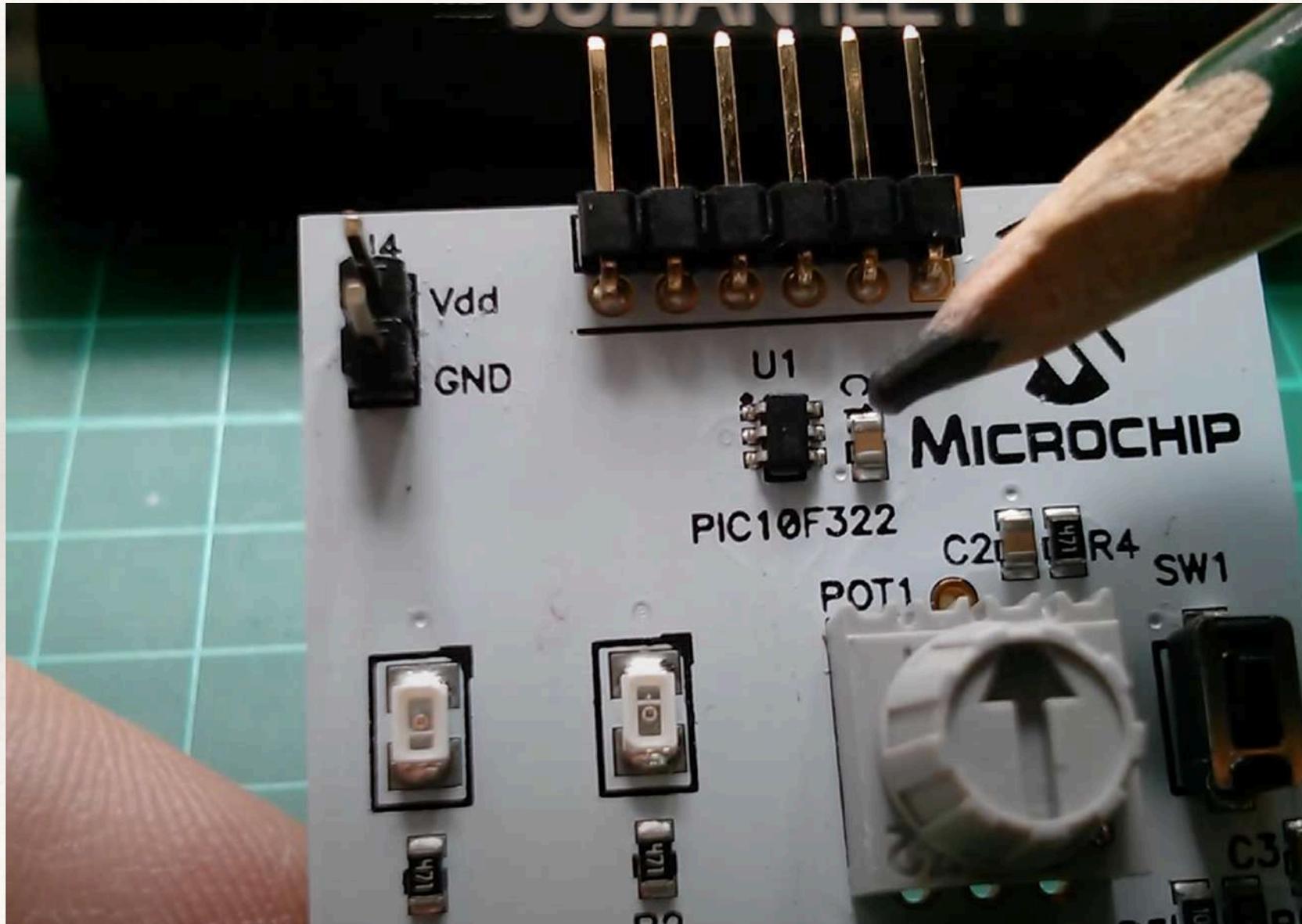
**SMD** - *Surface Mount Device*



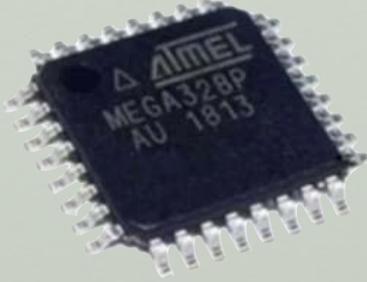
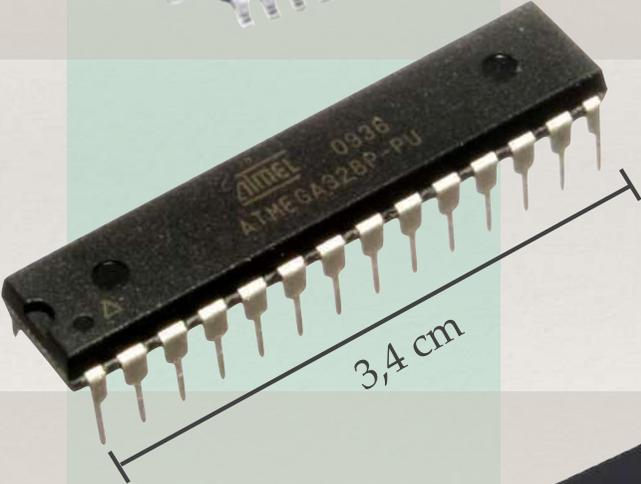
Placa de Protótipo PIC10F322



# Microcontroladores



# Comparativo Dimensional

	<u>PIC10F322</u> 6 pinos	<u>ATTiny85</u> 8 pinos	<u>ATMega328P</u> 28 pinos	<u>ESP32 SoC</u> 48 pinos
SMD				
PTH				
QFN				

---

# ATMega328P

---

## Microcontrolador do Arduíno Uno

Bastante utilizado em todo o mundo em instituições de ensino, e por hobbistas, para projetos dos mais diversos tipos;

Desenvolvido pela Atmel, comprada pela Microchip em 2016, está disponível em formatos PTH e SMP;

No projeto Arduíno, o microcontrolador é instalado em uma **placa de protótipo** que oferece recursos adicionais, facilitando a implementação de projetos sem conhecimentos muito aprofundados de eletrônica;

Arduíno, portanto, é o nome da placa de protótipo, e não do microcontrolador !



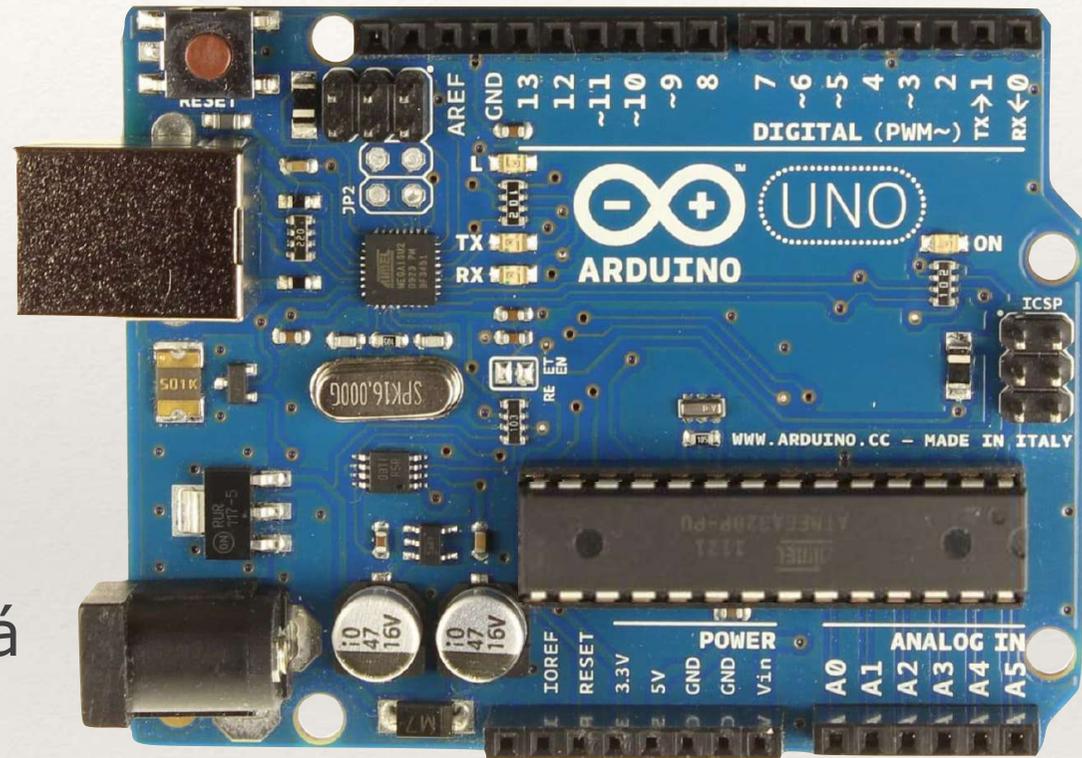
# Conhecendo o Arduíno Uno R3

Tudo começou na Itália em 2005;

Tecnologia aberta (existem diversos fabricantes produzindo produtos similares);

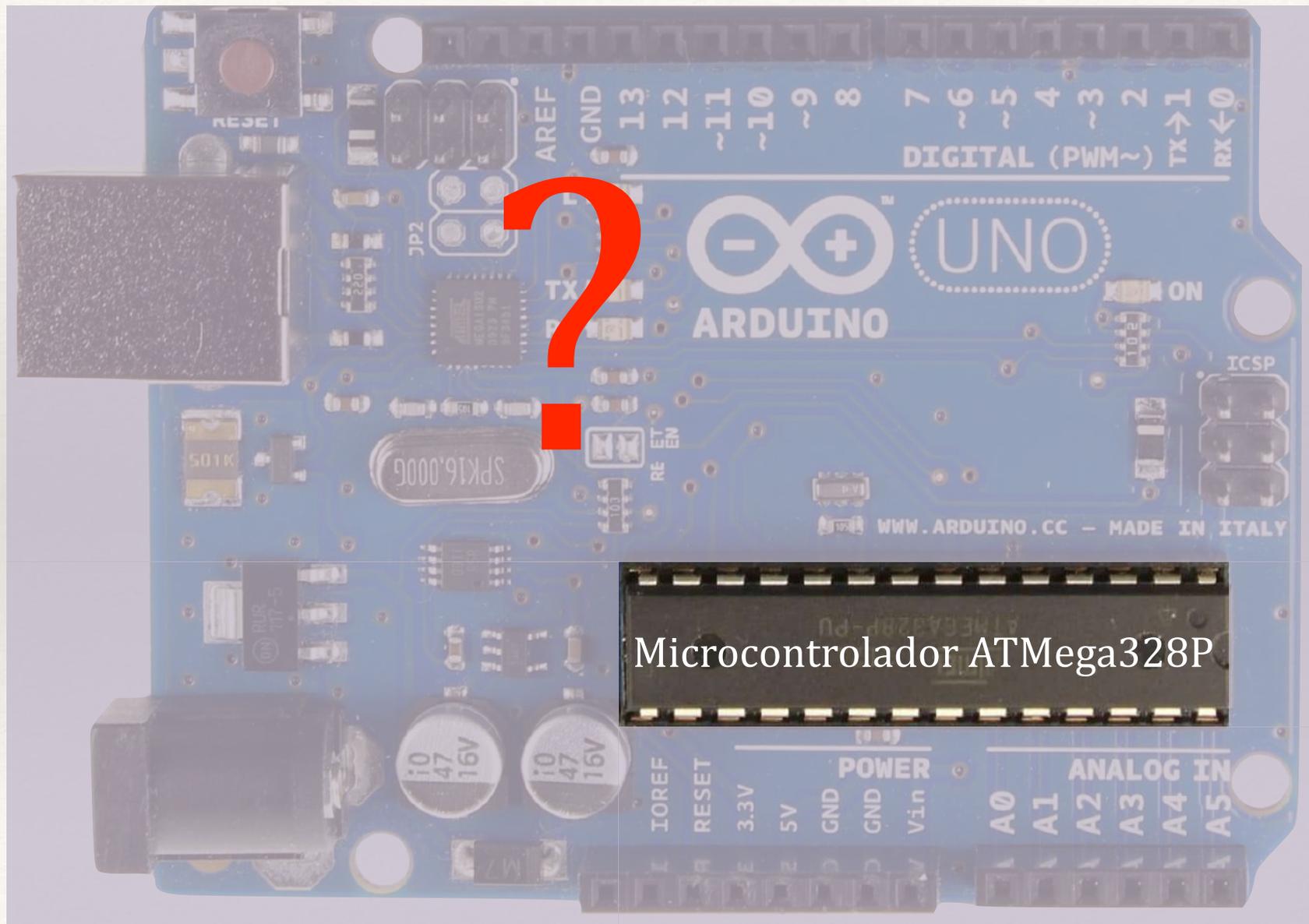
O mais correto é chamar de “**Projeto Arduíno**”, já que são diversos dispositivos;

Na foto, temos o **Arduíno Uno versão 3**. A versão mais atual desta placa é a 4. Além dela, o projeto tem diversos outros modelos com recursos variados.





# Conhecendo o Arduíno Uno R3

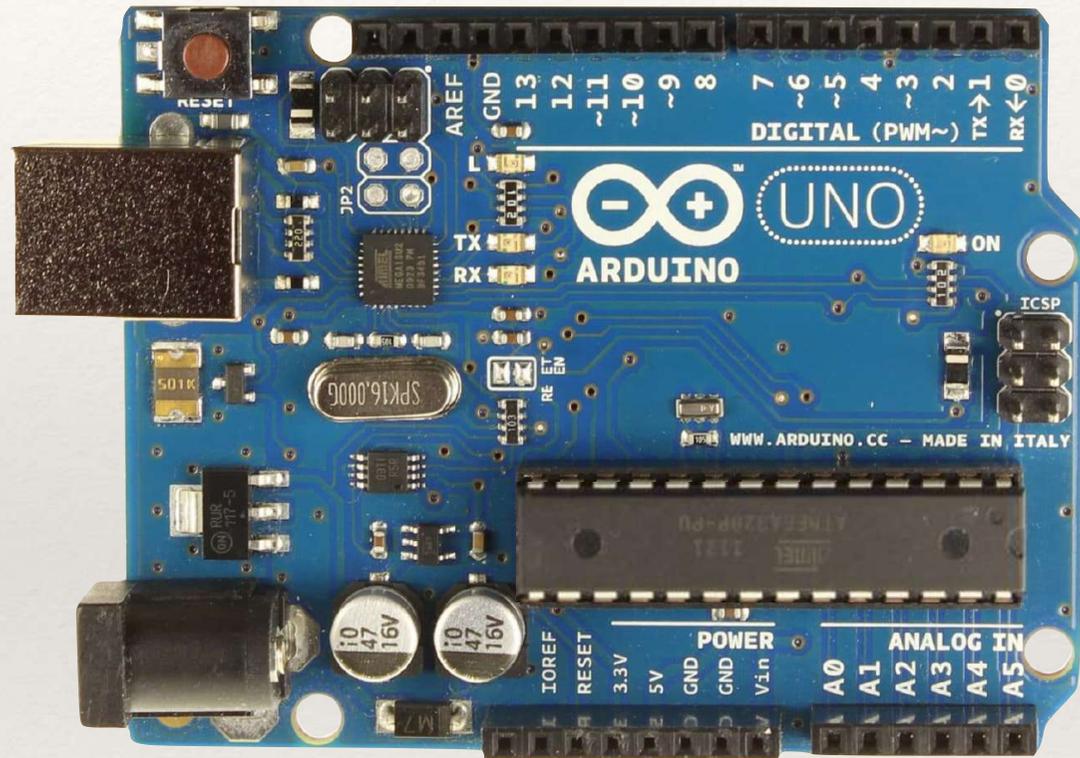


# Conhecendo o Arduino Uno R3

Uma placa de protótipo contém componentes adicionais, além do microcontrolador;

No caso do Arduino Uno, temos:

- Conexões diversas;
- Botão de Reset;
- Controladora USB;
- Clock* Externo a Quartzo;
- Regulador de Tensão (5V);
- Componentes eletrônicos diversos.



# Conexões do Arduino Uno R3

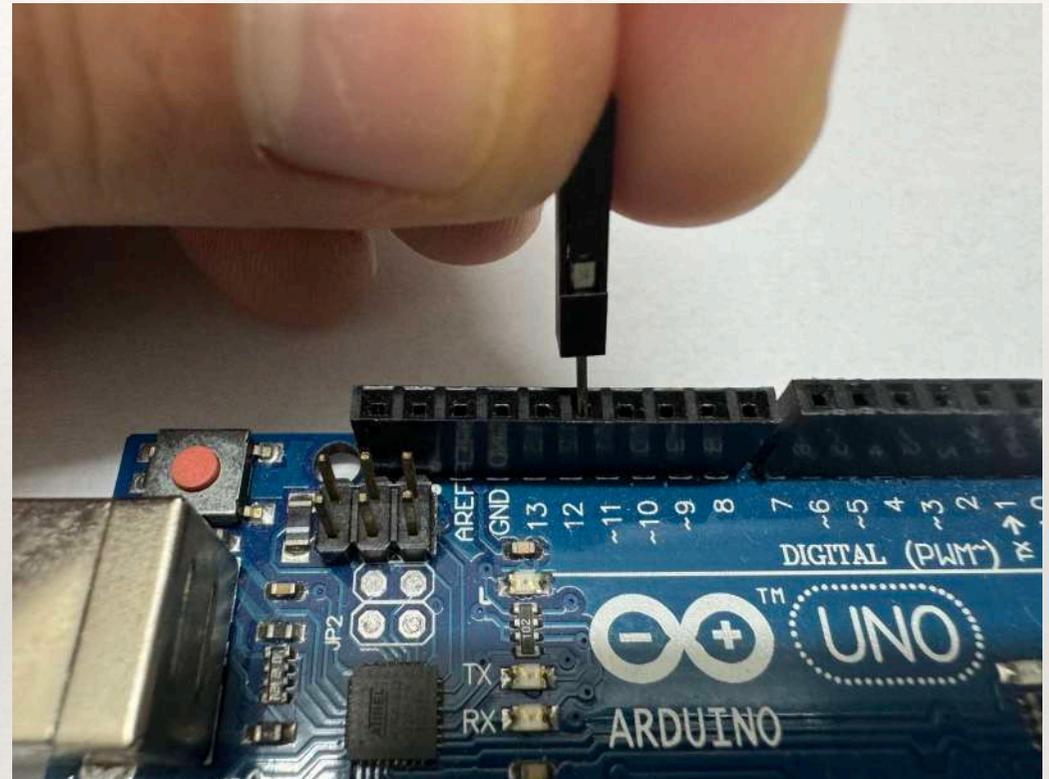
O Uno oferece conectores fêmea para fios *jumper Dupont*, sem uso de solda ou mesmo uma *protoboard*.

Embora simples, estas conexões têm alguns problemas:

- Contatos instáveis, especialmente para sistemas mais complexos;

- Baixa condutividade;

- Suportam correntes pequenas.



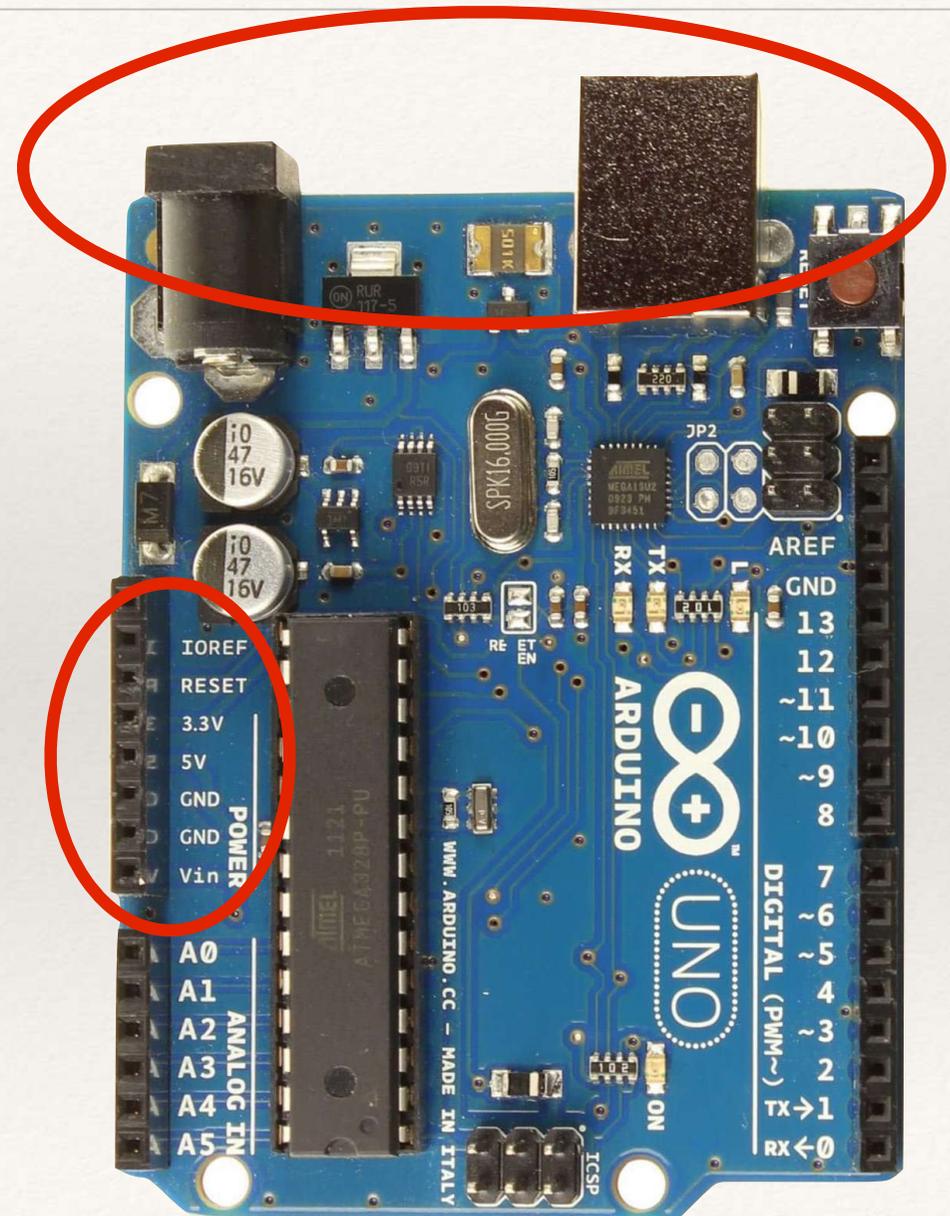
Por outro lado, outras placas de protótipo podem exigir o uso de *protoboards*. Projetos complexos podem exigir inclusive uma placa de circuito impresso, mesmo para protótipos.

# Conexões do Arduino Uno R3

Alimentação

Portas Analógicas

Portas Digitais

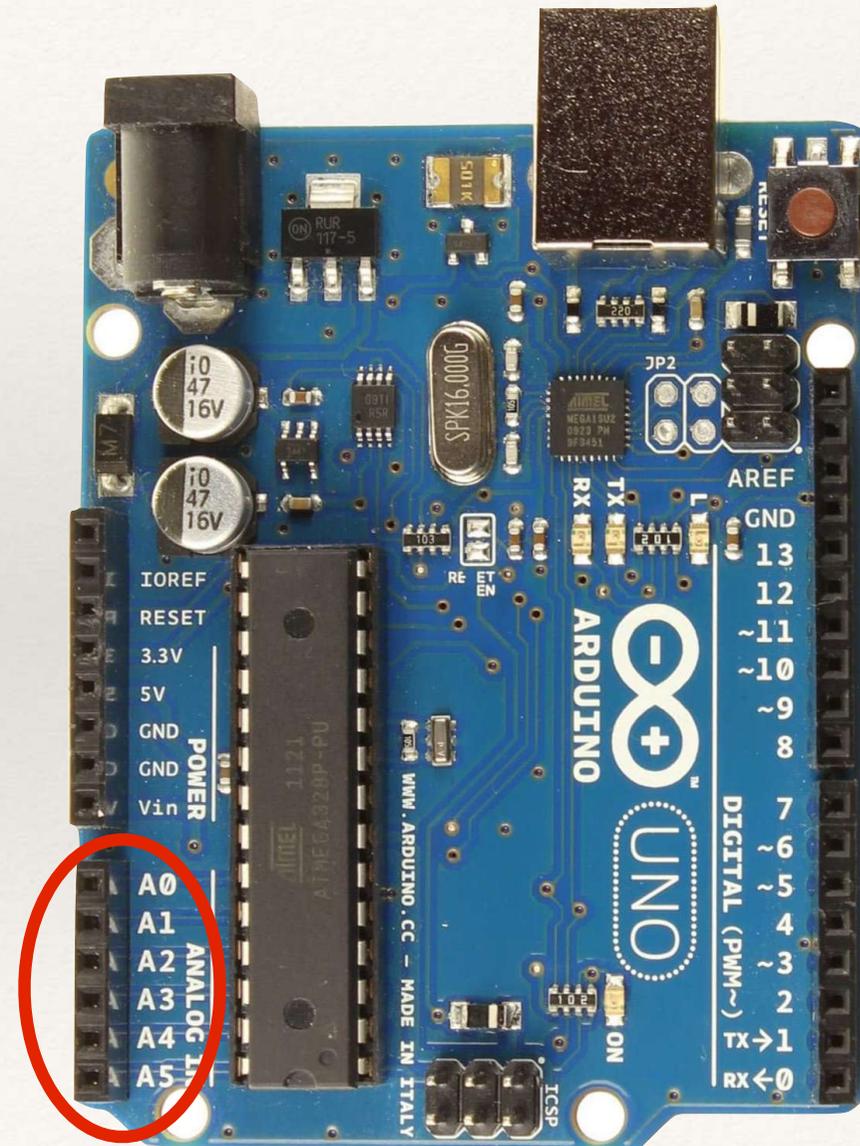


# Conexões do Arduino Uno R3

Alimentação

Portas Analógicas

Portas Digitais

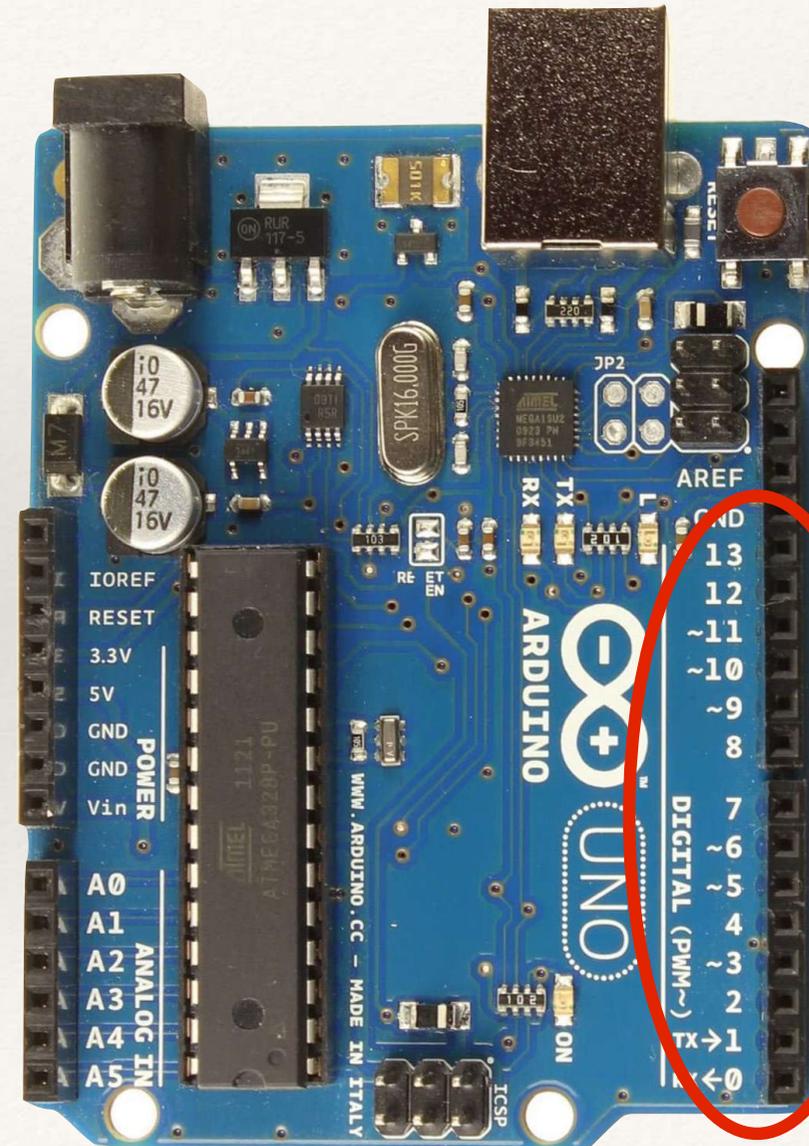


# Conexões do Arduino Uno R3

Alimentação

Portas Analógicas

Portas Digitais



# Conexões do Arduino Uno R3

## Alim. / Comunicação

### Entrada USB

Provê comunicação com o computador (e alimentação).

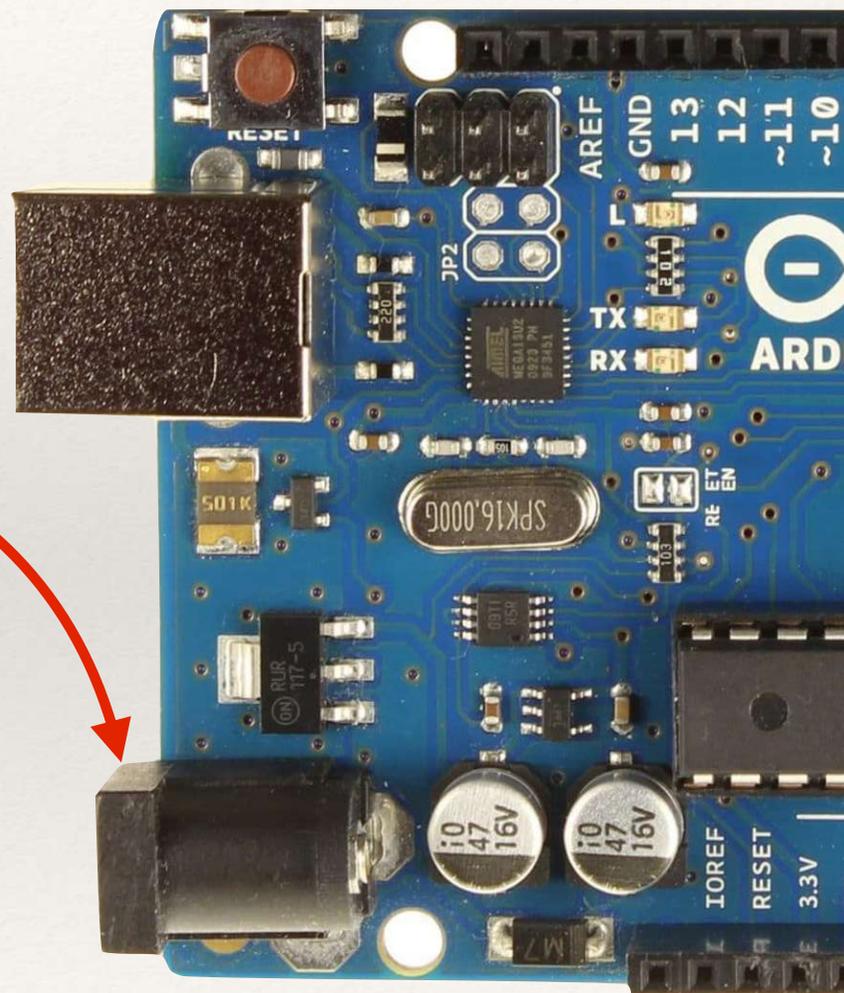
### Entrada Fonte DC Externa

Provê alimentação a partir de fonte de 5 a 17V, <100mA;

Não se recomenda >12V (aquecimento componentes).

Portas Analógicas

Portas Digitais



# Conexões do Arduino Uno R3

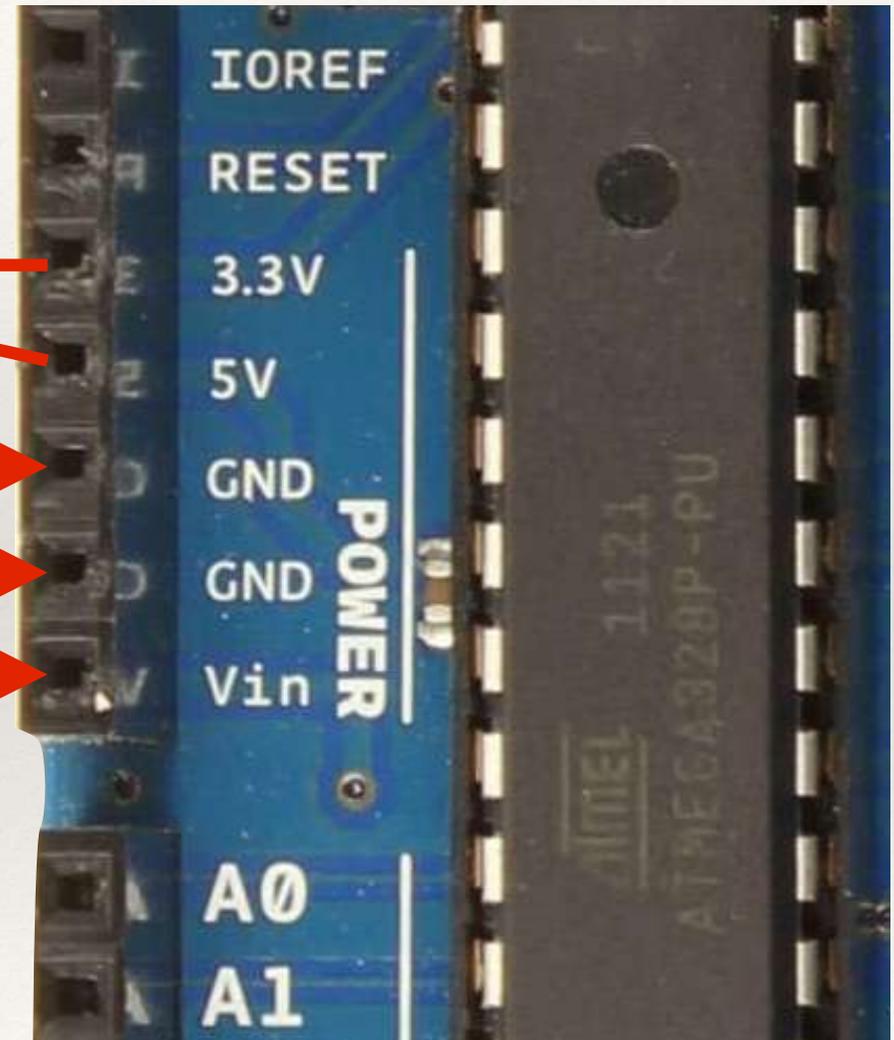
## Alimentação (pinos)

Saídas de 5V e 3,3V ←  
Corrente máxima de 40mA

Entrada Vin →  
Alimentação de 5V  $\geq$  100mA

Portas Analógicas

Portas Digitais



# Conexões do Arduino Uno R3

Alimentação

Portas Analógicas

Entradas com conversor A/D

Valores de 10 bits (0~1023)

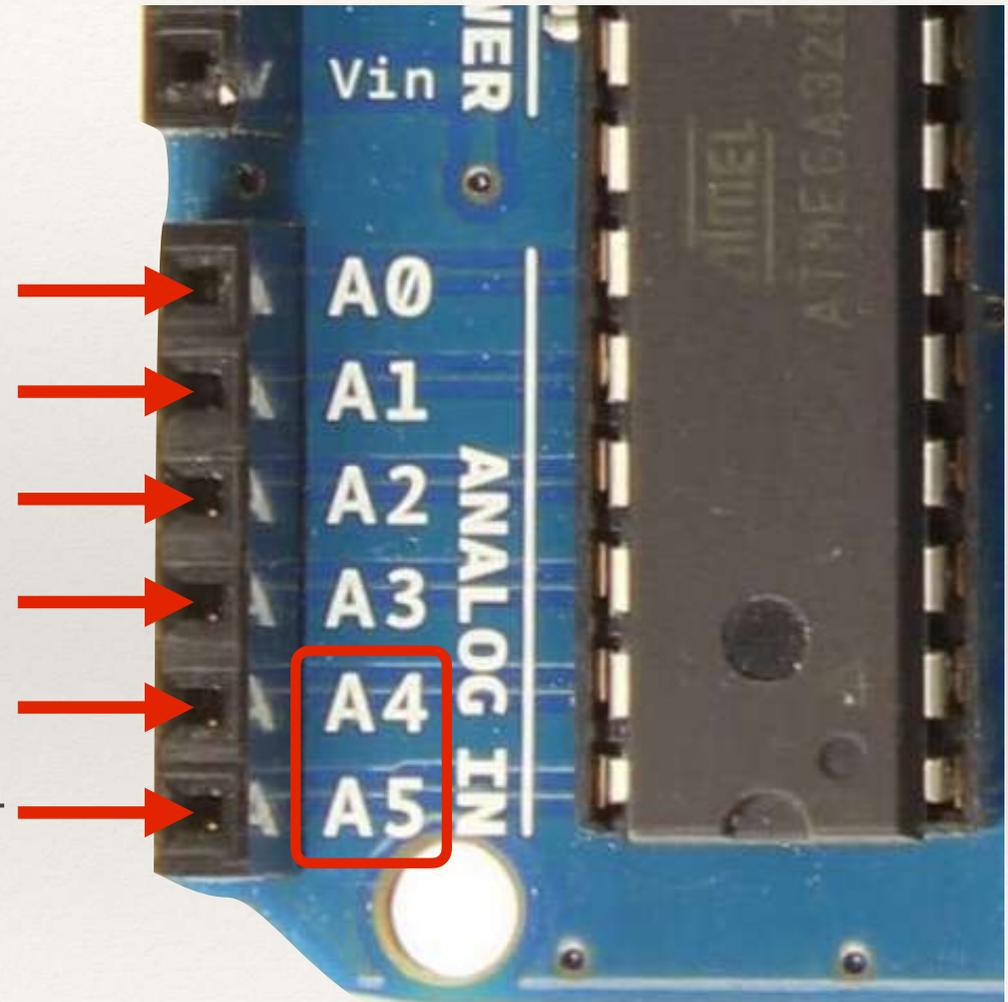
0 = **0V**

1023 = **5V** (IOREF reduz valor máximo)

Funções Especiais

I2C (SCL e SDA): portas A5 e A4

Portas Digitais



# Conexões do Arduino Uno R3

Alimentação

Portas Analógicas

Entradas com conversor A/D

Valores de 10 bits (0~1023)

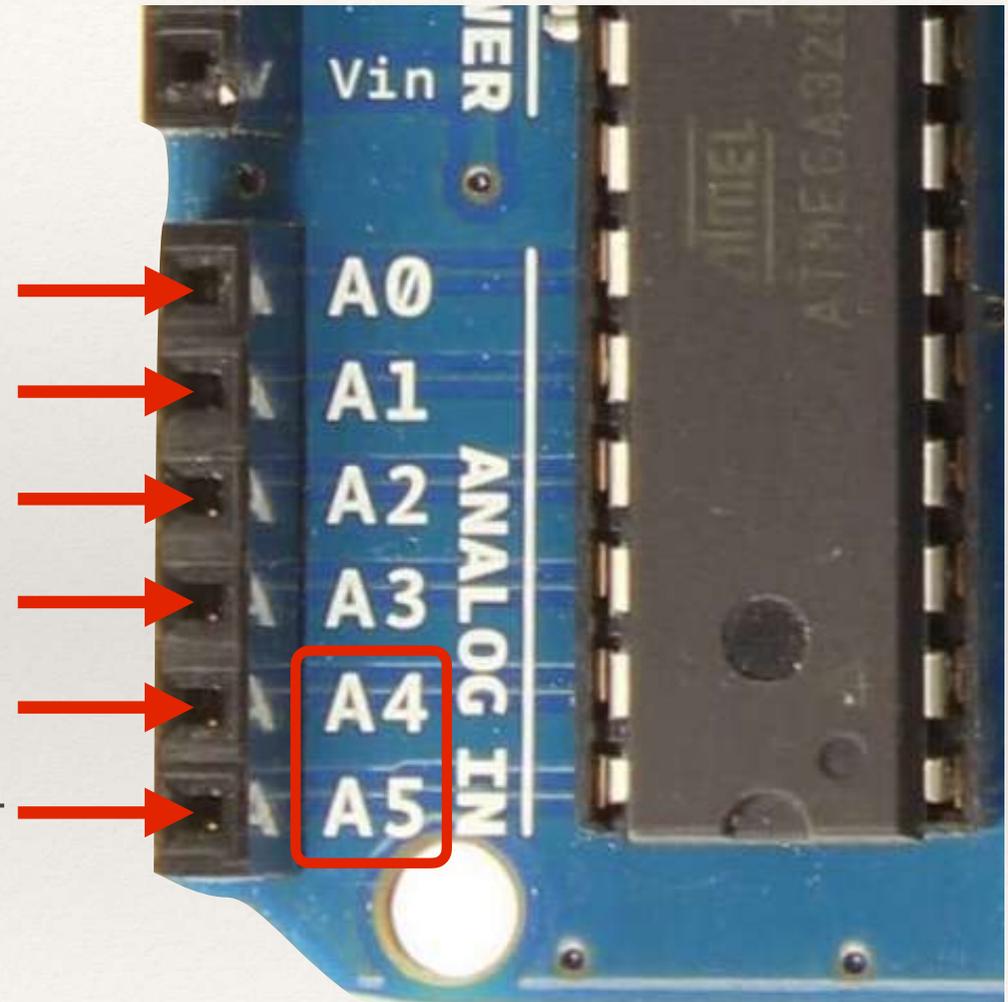
0 = **0V**

1023 = **5V** (IOREF reduz valor máximo)

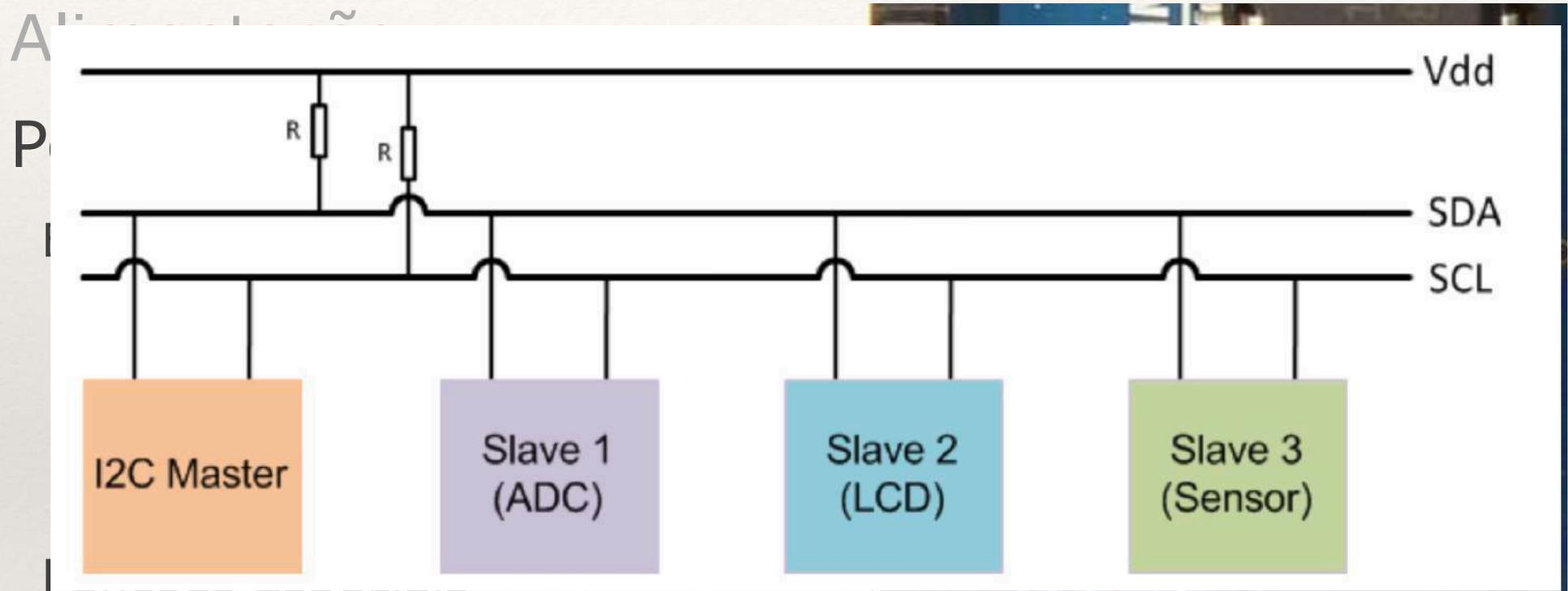
Funções Especiais

**I2C** (SCL e SDA): portas A5 e A4

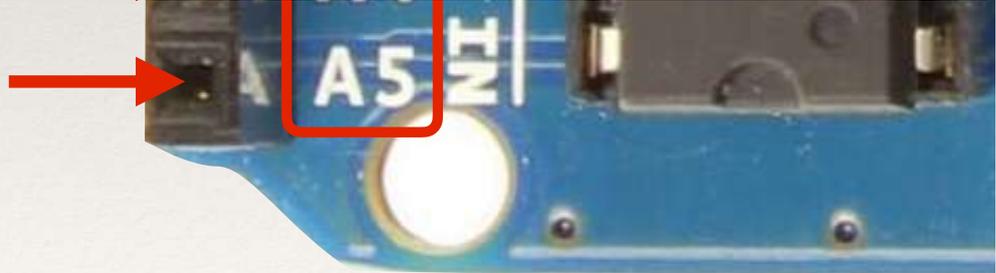
Portas Digitais



# Conexões do Arduino Uno R3



I2C (SCL e SDA): portas A5 e A4



Portas Digitais

# Conexões do Arduino Uno R3



Alimentação

Portas Analógicas

Portas Digitais

Entrada ou Saída

Configurada via SW

Saídas marcadas com ~ são **PWM**  
(*Pulse Width Modulation*)

< 20mA (máx.40 mA); total 200mA

Funções especiais

**UART:** Rx/Tx (portas 0 e 1)

**Interrupções:** INT0 e 1 (portas 2 e 3)

**SPI:** SS, MOSI, MISO, SCK (p. 10~13)

**Pino 13:** LED interno

# Conexões do Arduino Uno R3



Alimentação

Portas Analógicas

Portas Digitais

Entrada ou Saída

Configurada via SW

Saídas marcadas com ~ são PWM  
(*Pulse Width Modulation*)

< 20mA (máx.40 mA); total 200mA

Funções especiais

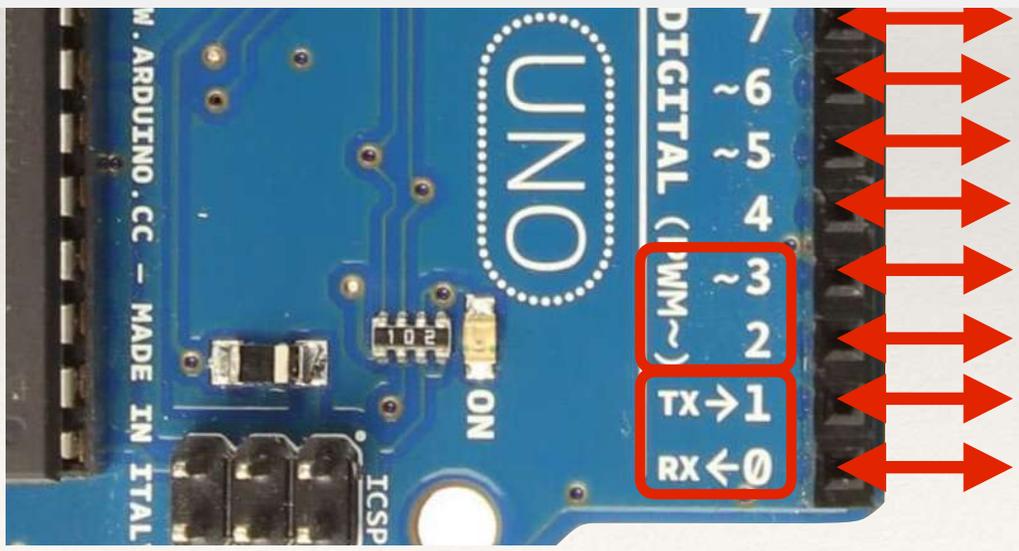
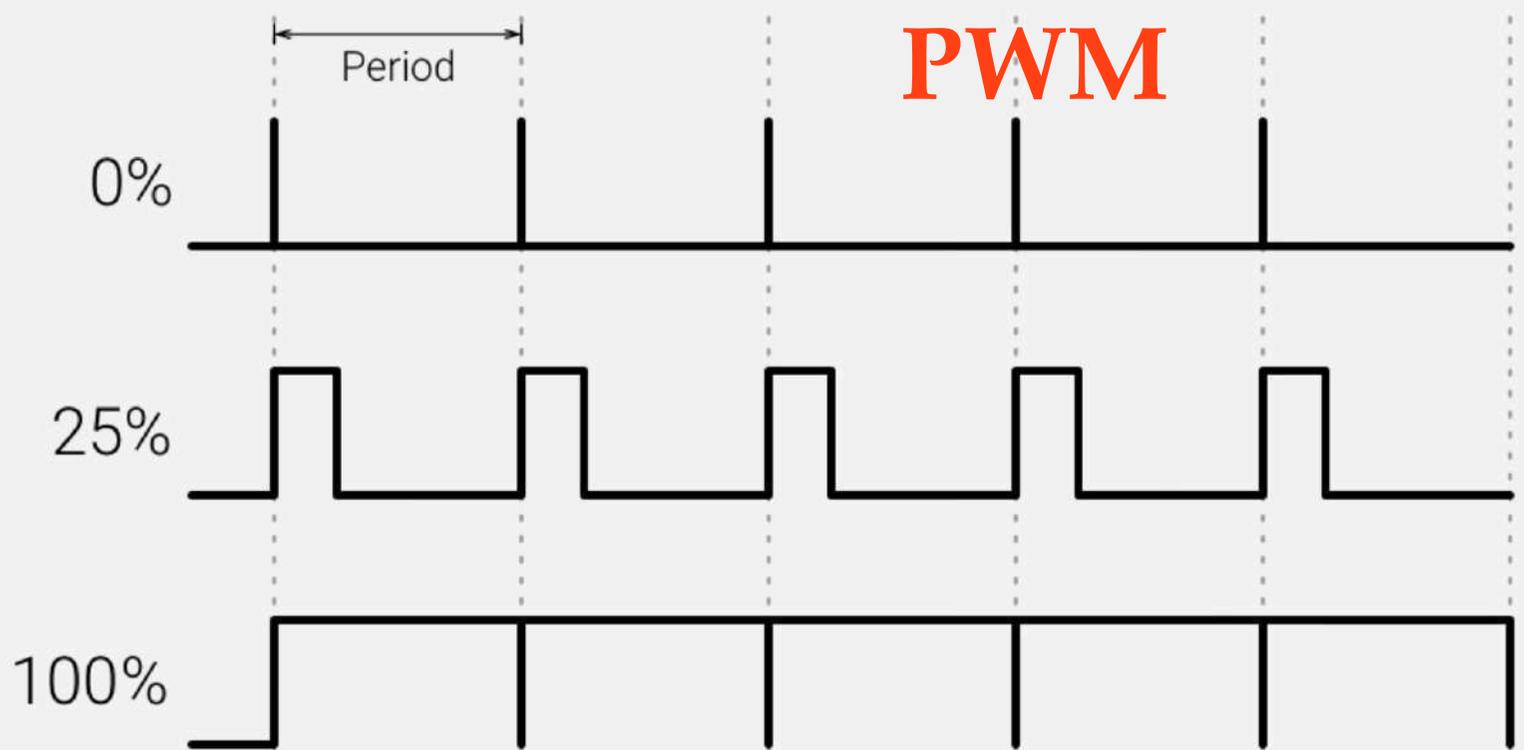
**UART:** Rx/Tx (portas 0 e 1)

**Interrupções:** INT0 e 1 (portas 2 e 3)

**SPI:** SS, MOSI, MISO, SCK (p. 10~13)

**Pino 13:** LED interno

R3



Saídas marcadas com ~ são PWM  
(Pulse Width Modulation)  
< 20mA (máx.40 mA); total 200mA

Funções especiais

- UART:** Rx/Tx (portas 0 e 1)
- Interrupções:** INT0 e 1 (portas 2 e 3)
- SPI:** SS, MOSI, MISO, SCK (p. 10~13)
- Pino 13:** LED interno

# Conexões do Arduino Uno R3



Alimentação

Portas Analógicas

Portas Digitais

Entrada ou Saída

Configurada via SW

Saídas marcadas com ~ são PWM  
(*Pulse Width Modulation*)

< 20mA (máx.40 mA); total 200mA

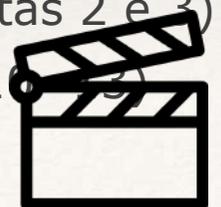
Funções especiais

**UART:** Rx/Tx (portas 0 e 1)

**Interrupções:** INT0 e 1 (portas 2 e 3)

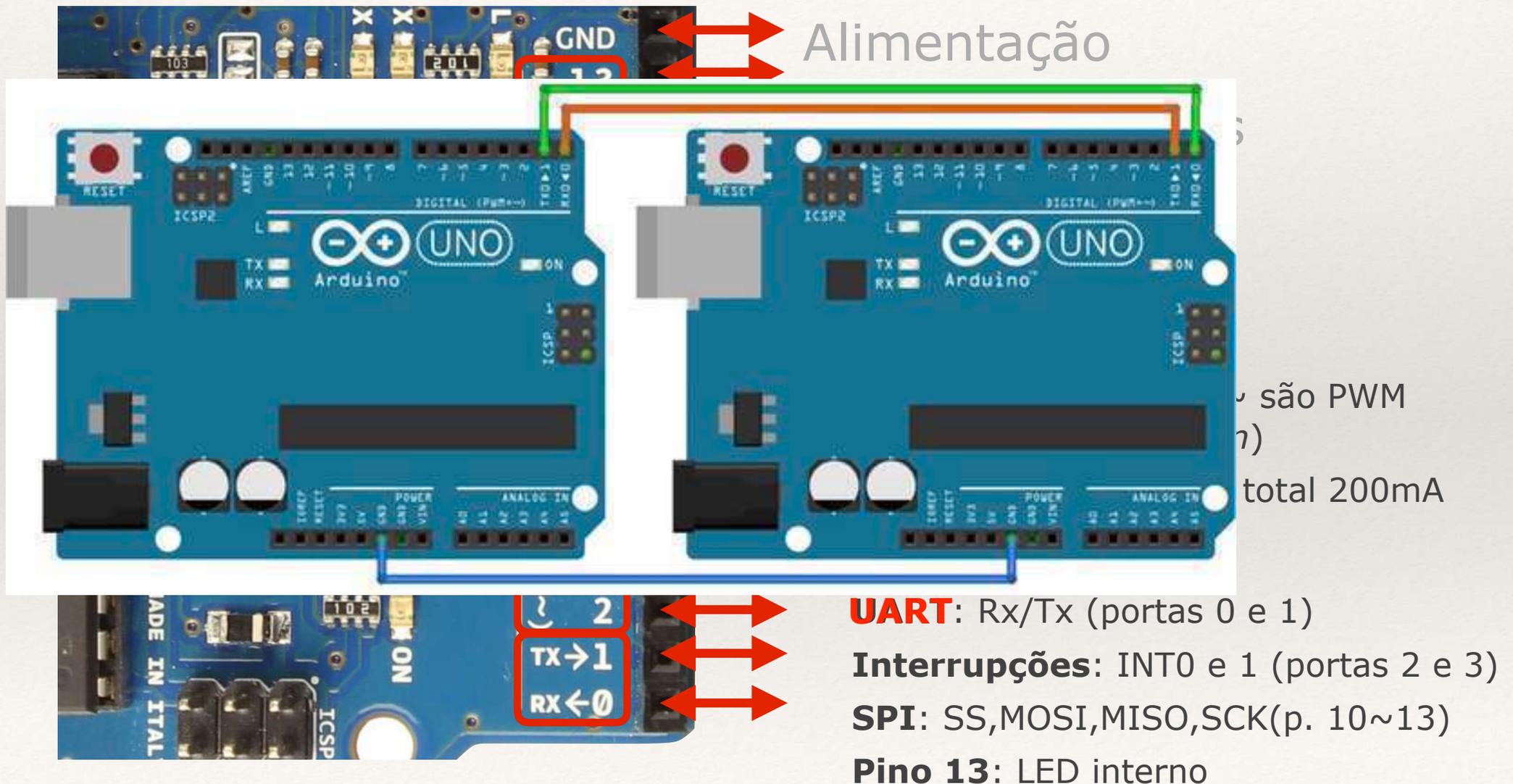
**SPI:** SS, MOSI, MISO, SCK (p. 1)

**Pino 13:** LED interno



# Aula 03

# Conexões do Arduino Uno R3

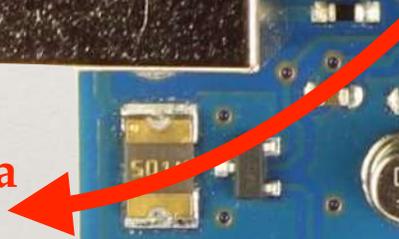


# Conhecendo o Arduino Uno R3

**Botão de  
RESET**



**Controladora  
USB**

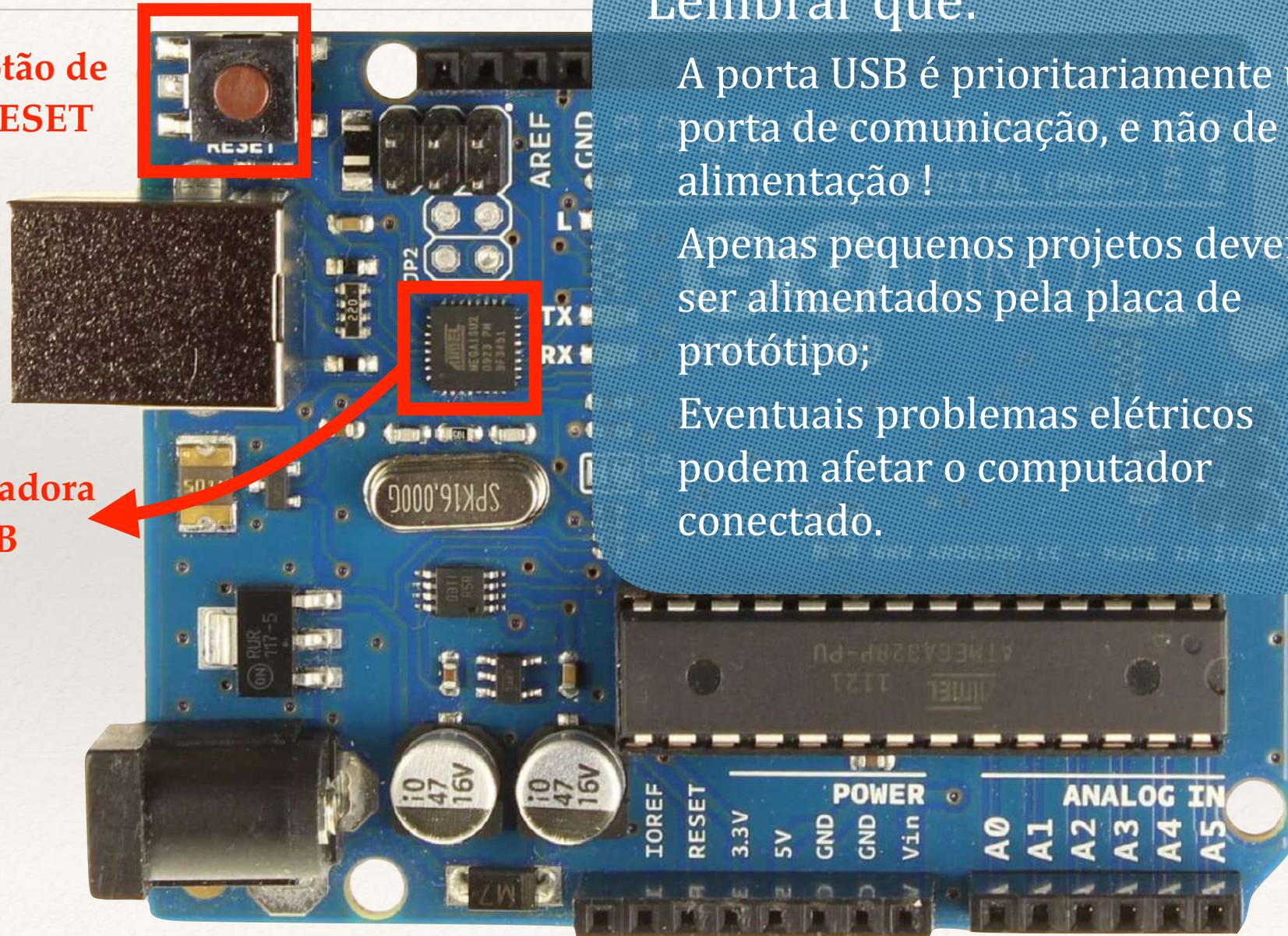


Lembrar que:

A porta USB é prioritariamente uma porta de comunicação, e não de alimentação !

Apenas pequenos projetos devem ser alimentados pela placa de protótipo;

Eventuais problemas elétricos podem afetar o computador conectado.



# Conhecendo o Arduino Uno R3



**Cristal de Clock**

O *clock* sincroniza processador, barramentos e periféricos;

**Performance:**

Se for baixa, pode tornar a operação lenta demais - porém reduz consumo;

Se for alta, pode impor tempos de resposta impossíveis para os componentes - porém é mais ágil;

**Estabilidade:**

Se não existirem operações síncronas, as exigências são baixas (1 ~ 5 %);

★ Um relógio atrasa até 14 minutos por dia com 1% de estabilidade.

Para aplicações síncronas, é um item crítico;

★ Ex: ADC, DAC e comunicação de dados.

# Conhecendo o Arduino Uno R3



Cristal de Clock

Existem algumas opções para implementação do *clock*:

- Oscilador RC interno;
- Oscilador RC externo;
- Oscilador com cristal de quartzo.

As opções diferem na estabilidade, tamanho, consumo e custo;

Fatores externos (ou internos) afetam a estabilidade:

- Temperatura, alimentação, interferências elétricas e vibração;

O Arduino Uno R3 usa um cristal de 16MHz em seu oscilador.



# Conhecendo o Arduino Uno R3

## A alimentação é crítica:

Queda de tensão por descarga da bateria, ou falhas de projeto, podem provocar alteração de comportamento;

Ruído injetado na alimentação é uma das fontes de instabilidade mais difíceis de diagnosticar. O próprio circuito digital é uma fonte de ruído;

Um surto de tensão, mesmo pequeno, pode danificar diversos componentes.

## Como evitar problemas?

Escolha cuidadosa das fontes de alimentação;

Capacitores de desacoplamento individuais para cada CI, instalados bem próximos dos mesmos.

Regulador  
de Tensão  
(5V)



Capacitor de  
Desacoplamento

# Microcontroladores

Microcontrolador	PIC10F322	ATTiny85	ATMega328P	ESP32 SoC
Arquitetura	8 bits	8 bits	8 bits	32 bits Dual-Core
GPIO	4	6	23	34
Program Memory (Flash)	512 B	8 KB	32 KB	16 MB
Data Memory (SRAM)	64 B	512 B	2 KB	520 KB
EEPROM	128 B	512 B	1 KB	4MB
Clock Máximo	16 MHz	20 MHz	16 MHz	240 MHz
UART	Não	Não	1	3
i2C	Não	Não	1	2
SPI	Não	1	1	4
PWM	2	2	6	16
ADC	3 x 8-bits	4 x 10-bits	8 x 10-bits	18 x 12-bits
Contadores	2 x 8-bits	2 x 8-bits	2 x 8-bits + 1 x 16-bits	8 x 16-bits
Consumo (modo ativo)	25uA (1 MHz)	300uA (1 MHz)	1,5mA (4 MHz)	260mA
Custo unitário (US\$)	0,89	1,42	2,66	3,46

---

# Eletrônica em SE

---

## Desafios de um projeto de HW

Não é um conhecimento nativo de um engenheiro de *software*;

A rotina de desenvolvimento em HW é bem diferente;

Testes podem ser destrutivos - a verificação prévia precisa ser exaustiva.

## Integração HW & SW

HW instável provocará problemas, independente da qualidade do SW;

Problemas no SW podem danificar o HW.

## Diversidade de componentes

Ao contrário do desenvolvimento de SW, não há um conjunto fixo de instruções e comandos;

Surgem novos sensores, atuadores e protocolos quase diariamente.

# Conceitos Básicos

## Grandezas:

Tensão, Corrente, Campos elétrico e magnético;

## Componentes:

Resistores, Capacitores e Indutores;

Diodos e transistores;

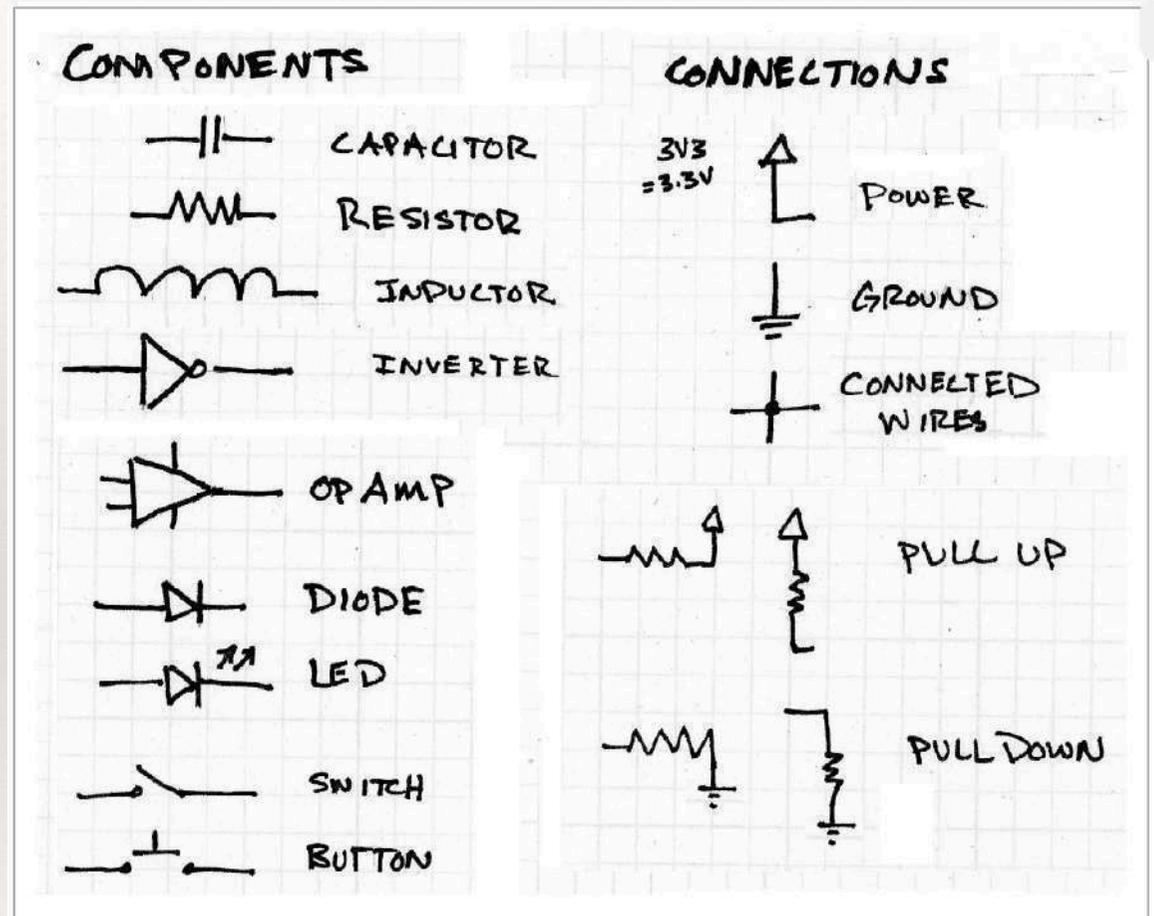
Amplificadores Operacionais;

Portas Lógicas;

## Circuitos:

Pull-Up e Pull-Down;

Amplificadores e Buffers;



---

# Tensão elétrica

---

É a diferença de potencial entre dois pontos, medida em Volts

Tipicamente um dos pontos é o "terra" (GND), com a tensão de zero volts;

Quando contínua, pode ser positiva ou negativa; Também pode ser alternada;

Provoca a circulação de corrente, que se move do ponto de tensão mais alto para o de tensão mais baixa.

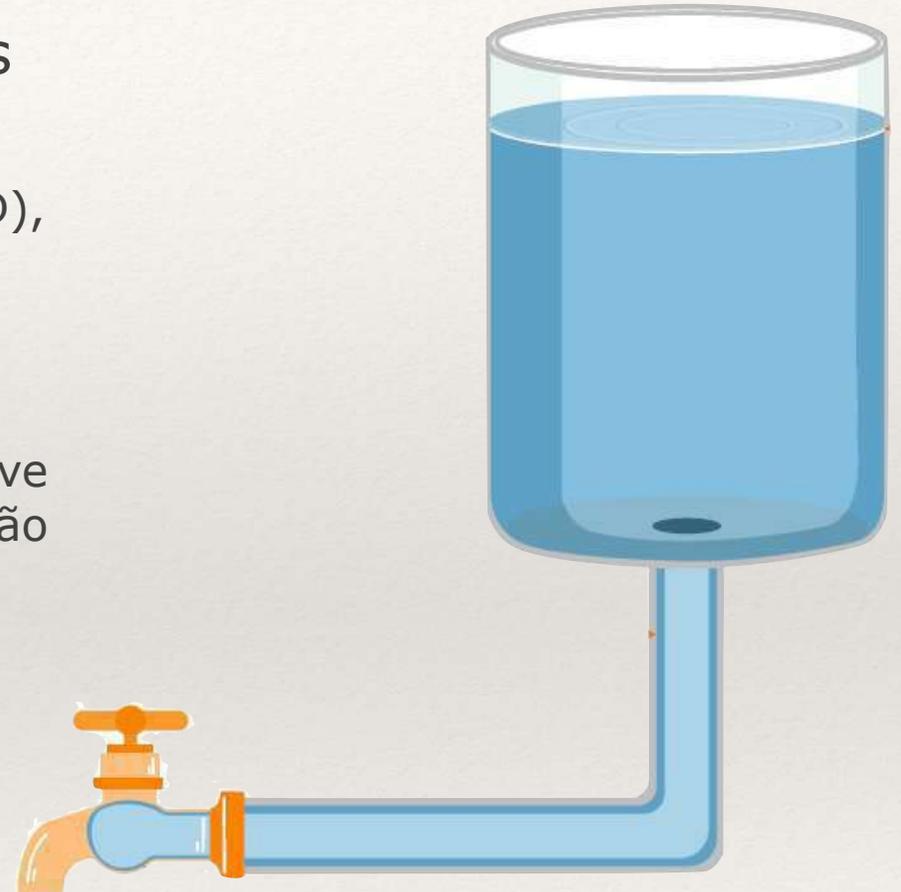
Pode ser causada por 3 efeitos

Campo elétrico estático;

Campo eletromagnético variável;

Corrente sob ação de um campo magnético.

Analogia mecânica: energia potencial.



# Corrente elétrica

Fluxo ordenado de cargas dentro de um condutor, medido em Ampères

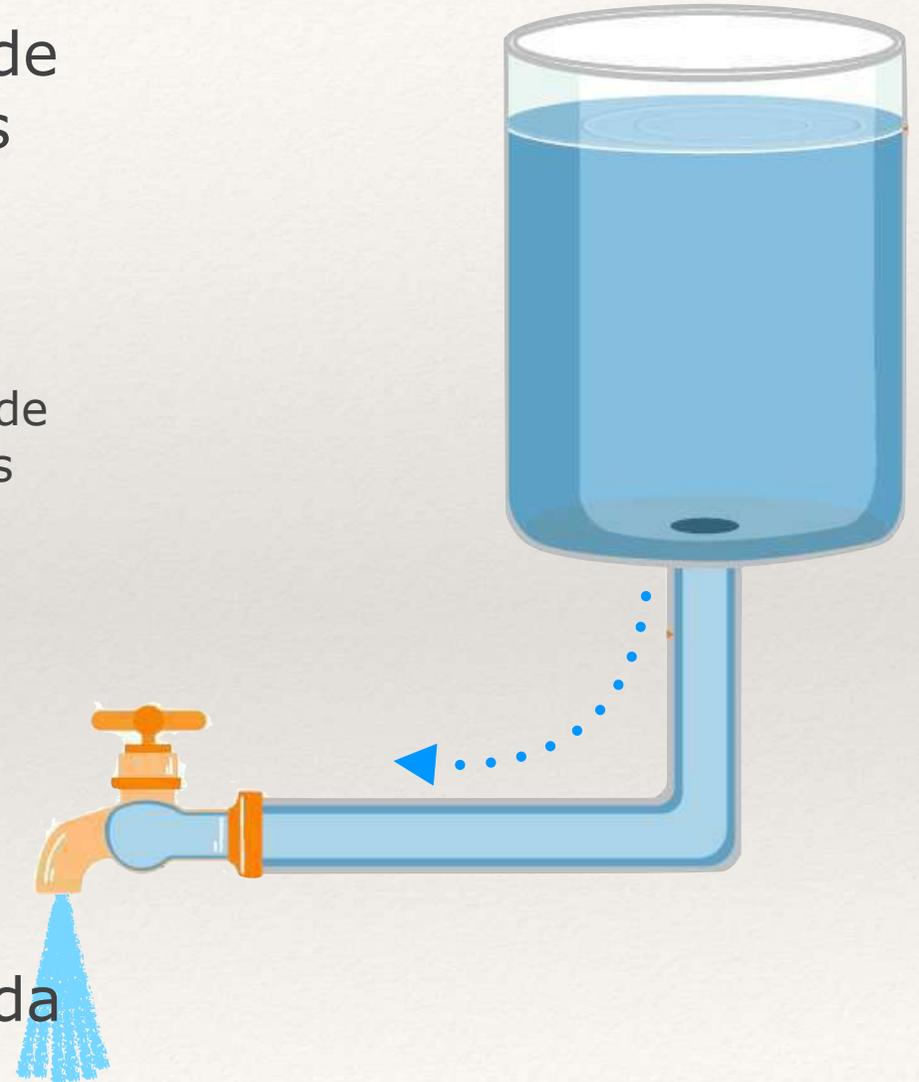
Para ser condutor, o material tipicamente tem elétrons livres em movimento desordenado;

Ocorre quando provocada pela existência de uma tensão elétrica entre as extremidades do condutor;

Quando contínua, o sentido do fluxo determina o seu sinal; pode também ser alternada.

Analogia mecânica: energia cinética;

Resistores se opõem à circulação da corrente elétrica.



---

# Campo Elétrico

---

É um campo de força criado pela eletricidade estática

Se inicia na carga positiva e termina na carga negativa;

É provocado pela ação de cargas elétricas (elétrons, prótons e íons);



Um campo elétrico variável provoca um campo magnético, e vice-versa;

Capacitores se opõem à alteração de um campo elétrico.

# Campo Magnético

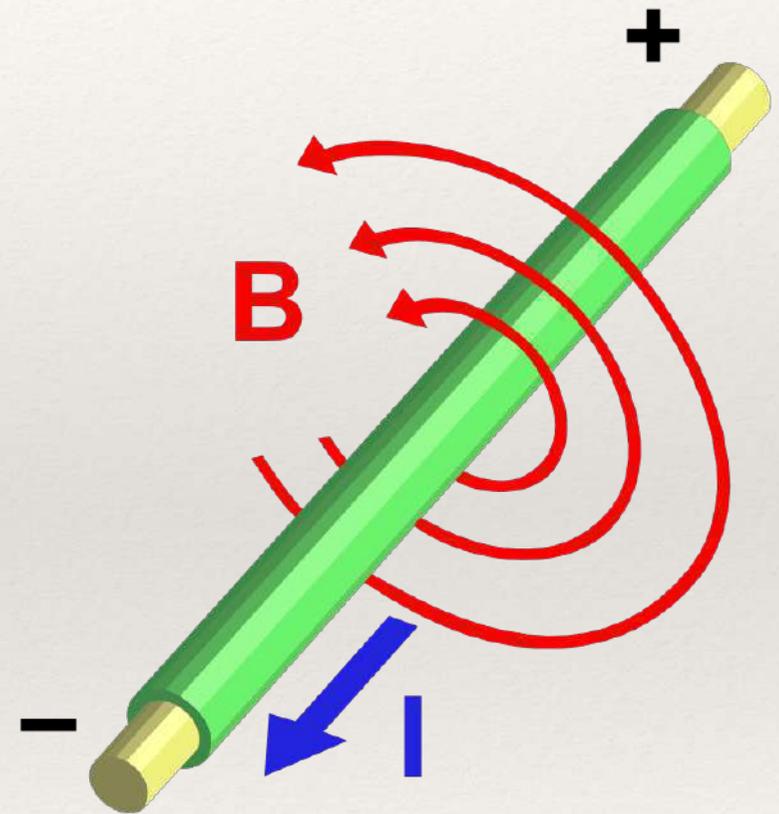
É um campo de força criado por materiais ferromagnéticos ou pela circulação de corrente elétrica

Formam círculos concêntricos perpendiculares à circulação da corrente;

A orientação do campo pode ser representada pela "regra da mão direita".

Um campo elétrico variável provoca um campo magnético, e vice-versa;

Indutores se opõem à alteração de um campo magnético.



# Resistor

Oferece resistência à passagem da corrente, e é medido em ohms ( $\Omega$ )

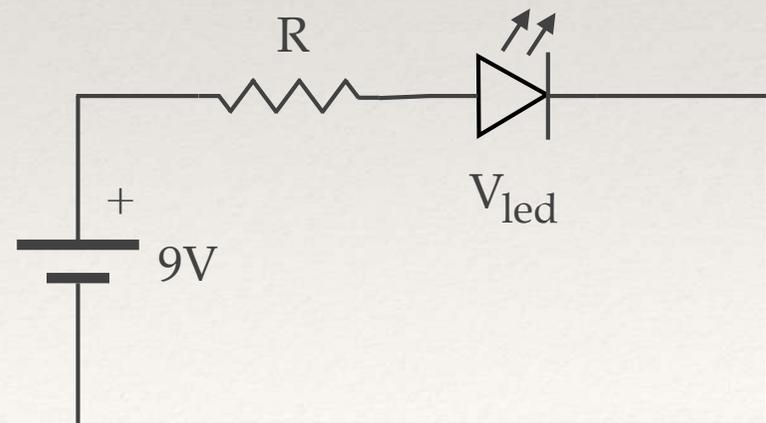
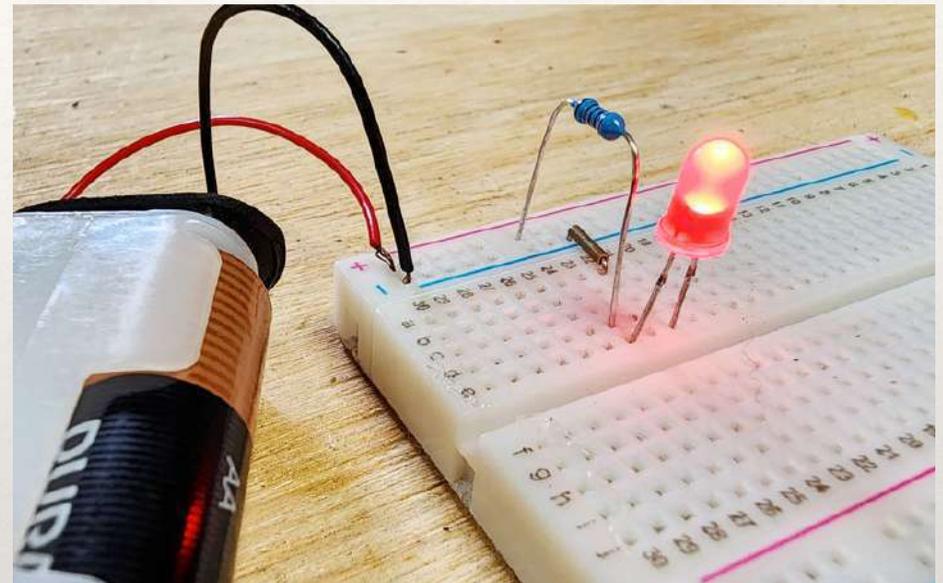
Remove energia do circuito transformando-a em potência térmica, que portanto precisa ser dissipada;

Reduz a corrente em circulação;

Provoca uma queda de tensão entre as suas extremidades.

Está relacionado à tensão e a corrente em um circuito com base na Lei de Ohm

Desenvolvida pelo físico alemão Georg Simon Ohm em 1827, baseada no trabalho de Fourier;



# Resistor

Oferece resistência à passagem da corrente, e é medido em ohms ( $\Omega$ )

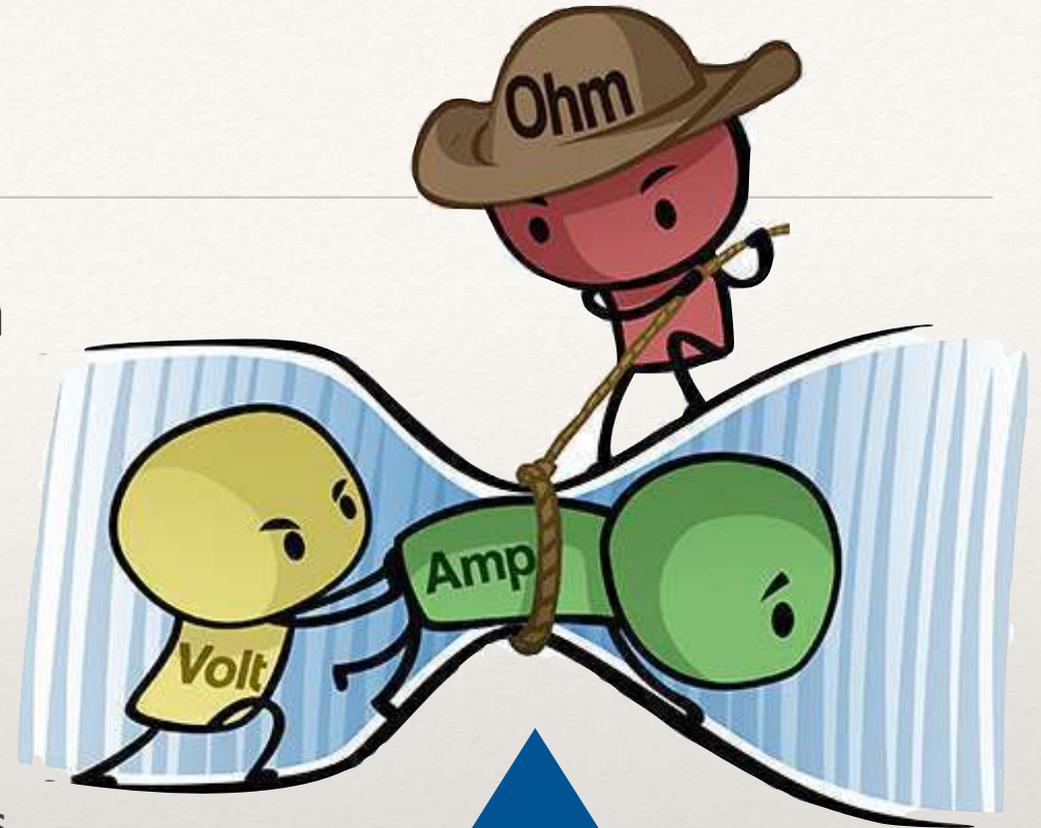
Remove energia do circuito transformando-a em potência térmica, que portanto precisa ser dissipada;

Reduz a corrente em circulação;

Provoca uma queda de tensão entre as suas extremidades.

Está relacionado à tensão e a corrente em um circuito com base na Lei de Ohm

Desenvolvida pelo físico alemão Georg Simon Ohm em 1827, baseada no trabalho de Fourier;



$$V = I \times R$$

# Tipos de Resistores



Carbon Composition



Carbon Film Resistors



Metal Film Resistors



Metal Oxide Film Resistors



Wire Wound Resistors



Thermistor Resistors



Fusible Resistors



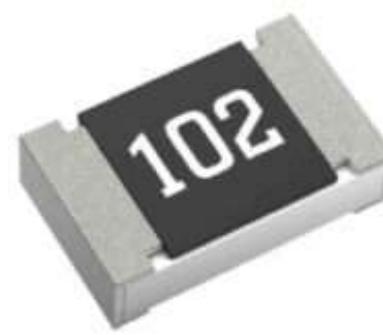
Carbon Potentiometer Resistors



Rheostat Resistors



Trimmer Resistors



Surface Mount Resistors



Wire Wound Potentiometer Resistors

engineeringlearn.com

engineeringlearn.com

# Resistores PTH - Potência

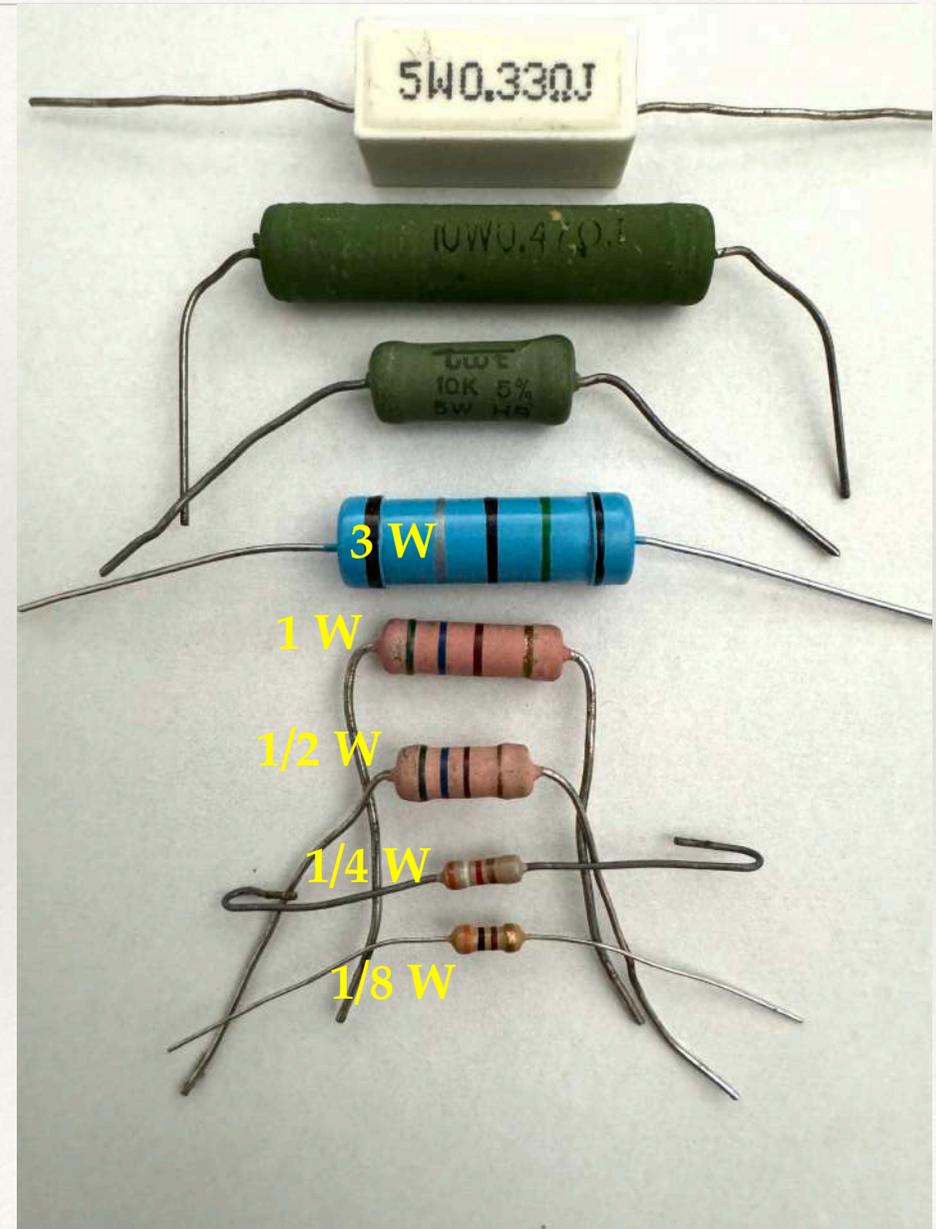
Tem relação com o tamanho

Resistores menores dissipam menos potência;

A temperatura (inclusive externa) afeta o valor da resistência !

Em alguns casos, pode estar gravada no corpo;

Projetos devem contemplar operação abaixo do limite.



# Resistores SMD - Potência



Invólucro	Tamanho em polegadas	Tamanho em milímetros	Potência
0201	0,024 "x 0,012"	0,6 milímetros x 0,3 milímetros	1/20W
0402	0,04 "x 0,02"	1,0 milímetro x 0,5 milímetros	1/16W
0603	0,063 "x 0,031"	1,6 milímetros x 0,8 milímetros	1/16W
0805	0,08 "x 0,05"	2,0 milímetro x 1,25 milímetros	1/10W
1206	0,126 "x 0,063"	3,2 milímetro x 1,6 milímetros	1/8W
1210	0,12 "x 0,10"	3,2 milímetros x 2,6 mm	1/4W
1812	0,18 "x 0,12"	4,6 mm x 3,0 milímetro	1/3W
2010	0,20 "x 0,10"	5,08 milímetros x 2,6 mm	1/2W
2.512	0,25 "x 0,12"	6,35 milímetros x 3,0 milímetros	1W

# Resistores - Resistência em $\Omega$

Cor	1ª Faixa	2ª Faixa	3ª Faixa	Multiplicador	Tolerância
Preto	0	0	0	x 1 $\Omega$	
Marrom	1	1	1	x 10 $\Omega$	+/- 1%
Vermelho	2	2	2	x 100 $\Omega$	+/- 2%
Laranja	3	3	3	x 1K $\Omega$	
Amarelo	4	4	4	x 10K $\Omega$	
Verde	5	5	5	x 100K $\Omega$	+/- .5%
Azul	6	6	6	x 1M $\Omega$	+/- .25%
Violeta	7	7	7	x 10M $\Omega$	+/- .1%
Cinza	8	8	8		+/- .05%
Branco	9	9	9		
Dourado				x .1 $\Omega$	+/- 5%
Prateado				x .01 $\Omega$	+/- 10%

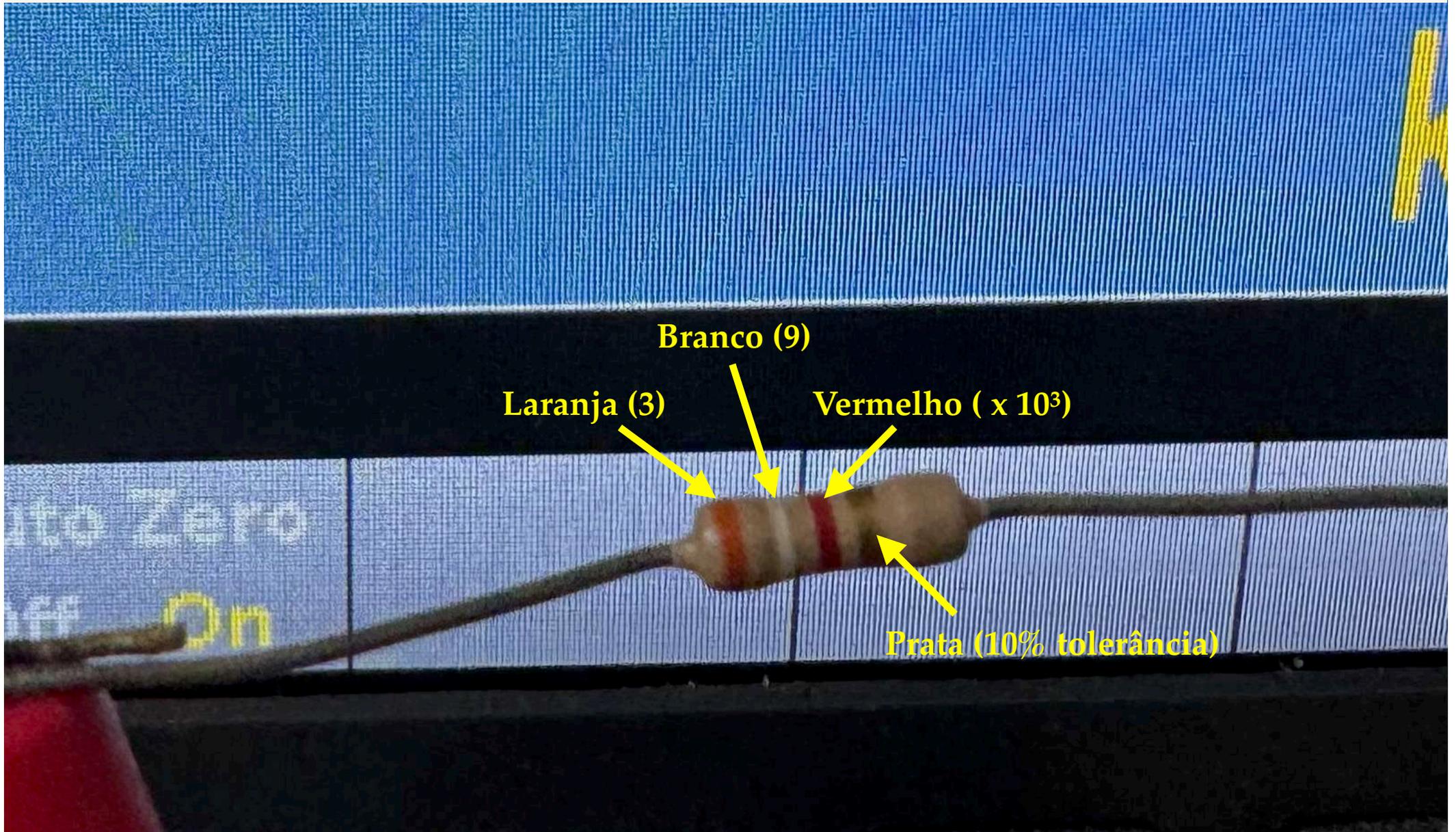
(código válido para maior parte dos resistores PTH)

3 faixas (20% tolerância), 4 faixas (comum), 5 faixas (precisão)

# Resistores - Resistência em $\Omega$



# Resistores - Resistência em $\Omega$



# Resistores - Resistência em $\Omega$

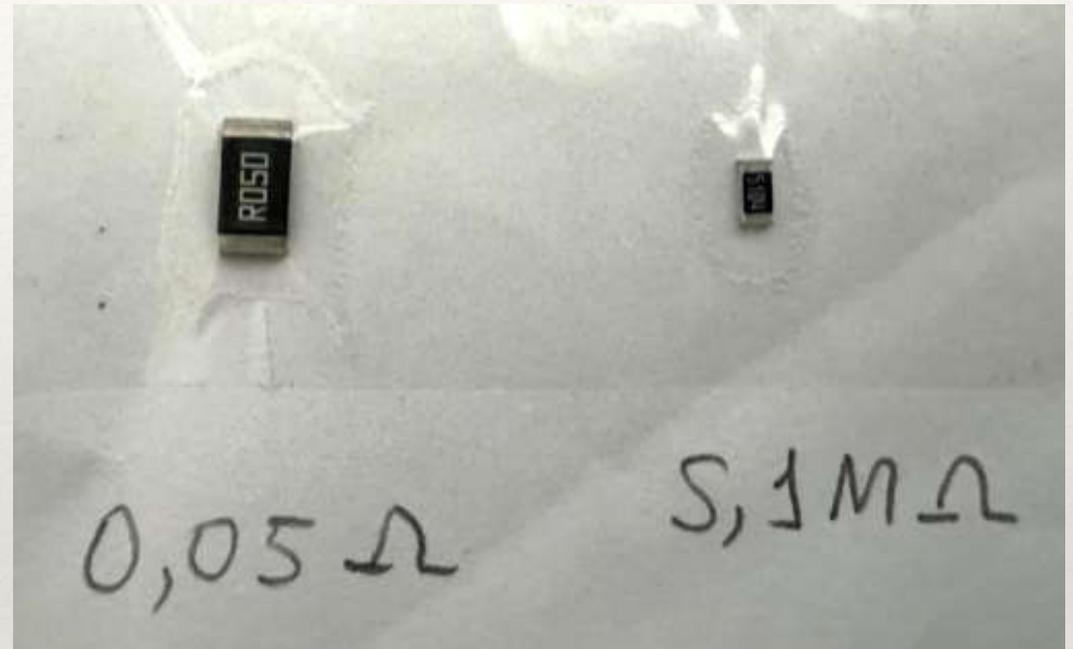
Em resistores SMD, o valor normalmente vem gravado no corpo

Tipicamente usam um código numérico de 3 dígitos;

Tal como as duas primeiras faixas de um resistor PTH, os dois primeiros números indicam os dígitos significativos, o terceiro é o multiplicador;

"104" por exemplo é igual a  $100\text{K}\Omega$ .

Se houverem quatro dígitos, vale a regra para resistores PTH com cinco faixas. Os três primeiros são dígitos significativos, e o último é o multiplicador.



---

# Resistores - Resistência em $\Omega$

---

Em resistores SMD, o valor normalmente vem gravado no corpo

Tipicamente usam um código numérico de 3 dígitos;

Tal como as duas primeiras faixas de um resistor PTH, os dois primeiros números indicam os dígitos significativos, o terceiro é o multiplicador;

"104" por exemplo é igual a  $100\text{K}\Omega$ .

Se houverem quatro dígitos, vale a regra para resistores PTH com cinco faixas. Os três primeiros são dígitos significativos, e o último é o multiplicador.



$0,05 \Omega - 1\text{W}$

Tamanho 2512 (6,35 x 3mm)

# Resistores - Resistência em $\Omega$

Em resistores SMD, o valor normalmente vem gravado no corpo

Tipicamente usam um código numérico de 3 dígitos;

Tal como as duas primeiras faixas de um resistor PTH, os dois primeiros números indicam os dígitos significativos, o terceiro é o multiplicador;

"104" por exemplo é igual a  $100\text{K}\Omega$ .

Se houverem quatro dígitos, vale a regra para resistores PTH com cinco faixas. Os três primeiros são dígitos significativos, e o último é o multiplicador.



$5,1\text{ M}\Omega - 1/4\text{W}$

Tamanho 1206 (3,2 x 1,6mm)

# Aula 04

# Resistores em Série

A resistência total é igual à soma das resistências individuais;

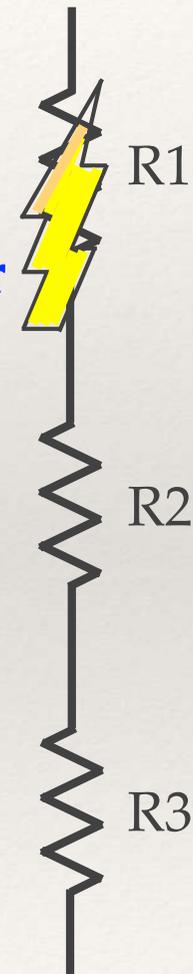
O arranjo permite:

Distribuir a potência dissipada;

Substituir resistências individuais;

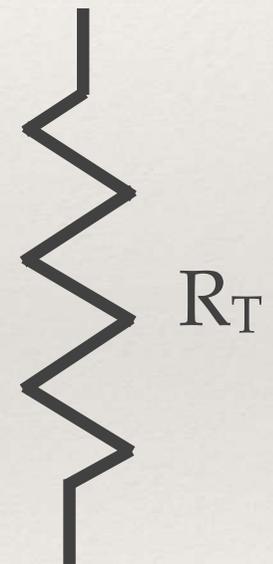
Aumentar a proteção contra descargas elétricas.

**Pular 1 resistor pode ser fácil**



$$R_T = R_1 + R_2 + R_3$$

≡



# Resistores em Série

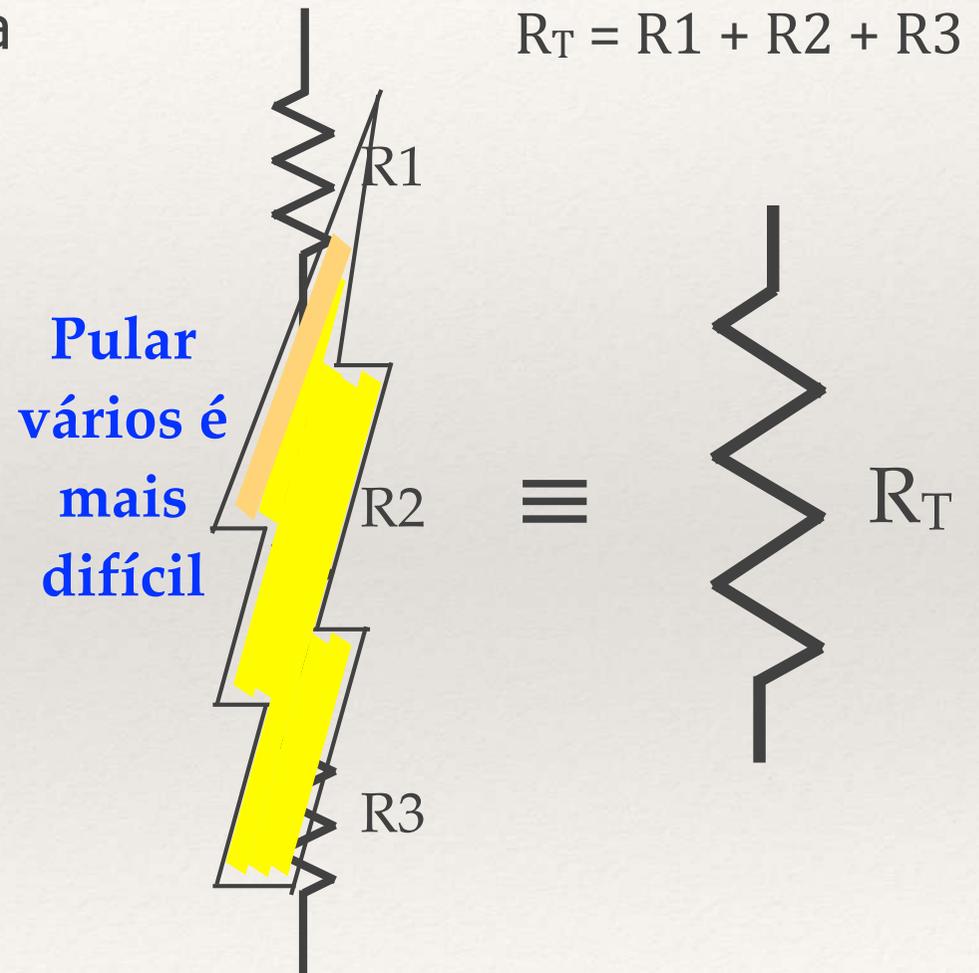
A resistência total é igual à soma das resistências individuais;

O arranjo permite:

- Distribuir a potência dissipada;

- Substituir resistências individuais;

- Aumentar a proteção contra descargas elétricas.



# Resistores em Série

A resistência total é igual à soma das resistências individuais;

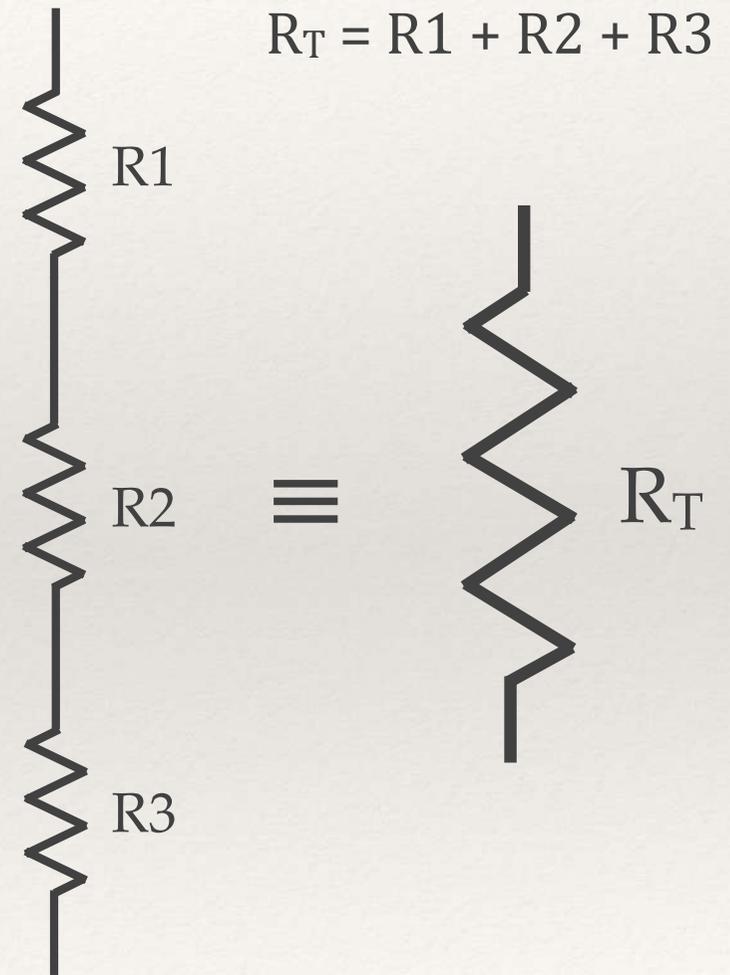
O arranjo permite:

- Distribuir a potência dissipada;

- Substituir resistências individuais;

- Aumentar a proteção contra descargas elétricas.

Existem, no entanto, aplicações mais "nobres", como o divisor de tensão.



---

# Resistores em Paralelo

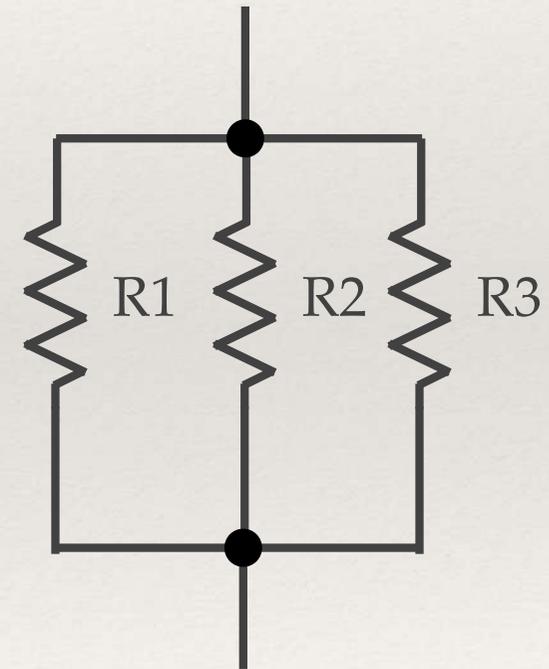
---

A resistência total é menor que a menor das resistências individuais;

$$\frac{1}{R_t} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$$

O arranjo permite:

- Distribuir a potência dissipada;
- Substituir resistências individuais.



# Resistores, valores comerciais

Norma IEC 60063:1963 estabeleceu as "séries E"

\* (IEC - *International Electrotechnical Commission*)

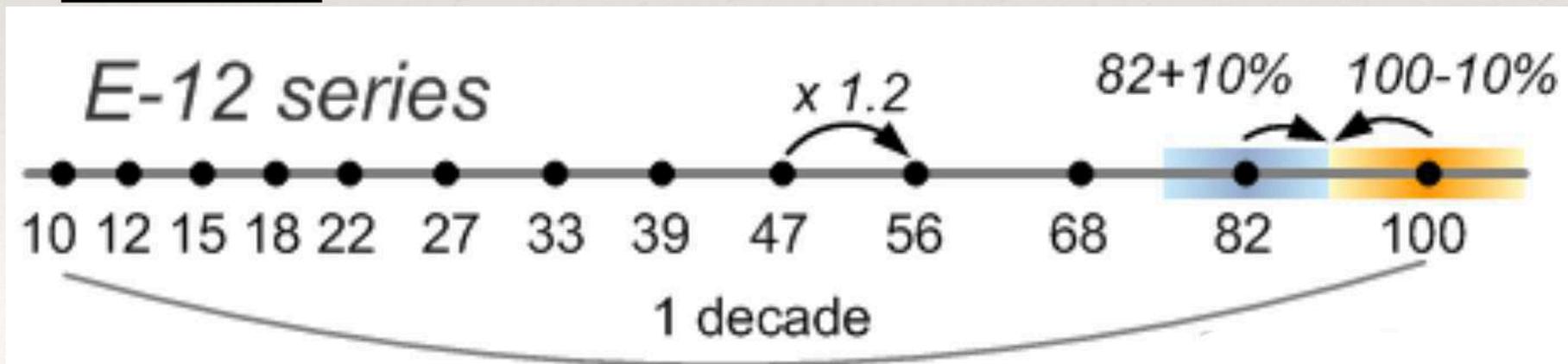
Cada década (0,1-1; 1-10 ...) está dividida em "n" espaços logarítmicos, a depender da tolerância:

20% de tolerância, 5 espaços (5 x 20% = 100%) e 6 valores (série E-6);

10% de tolerância envolve 12 valores (série E-12). Veja o cálculo:

$$10^{\frac{1}{12}} = 1,21$$

Cada valor é 21% maior que o anterior, com ajustes de aproximação e escala.



# Resistores, valores comerciais

Norma

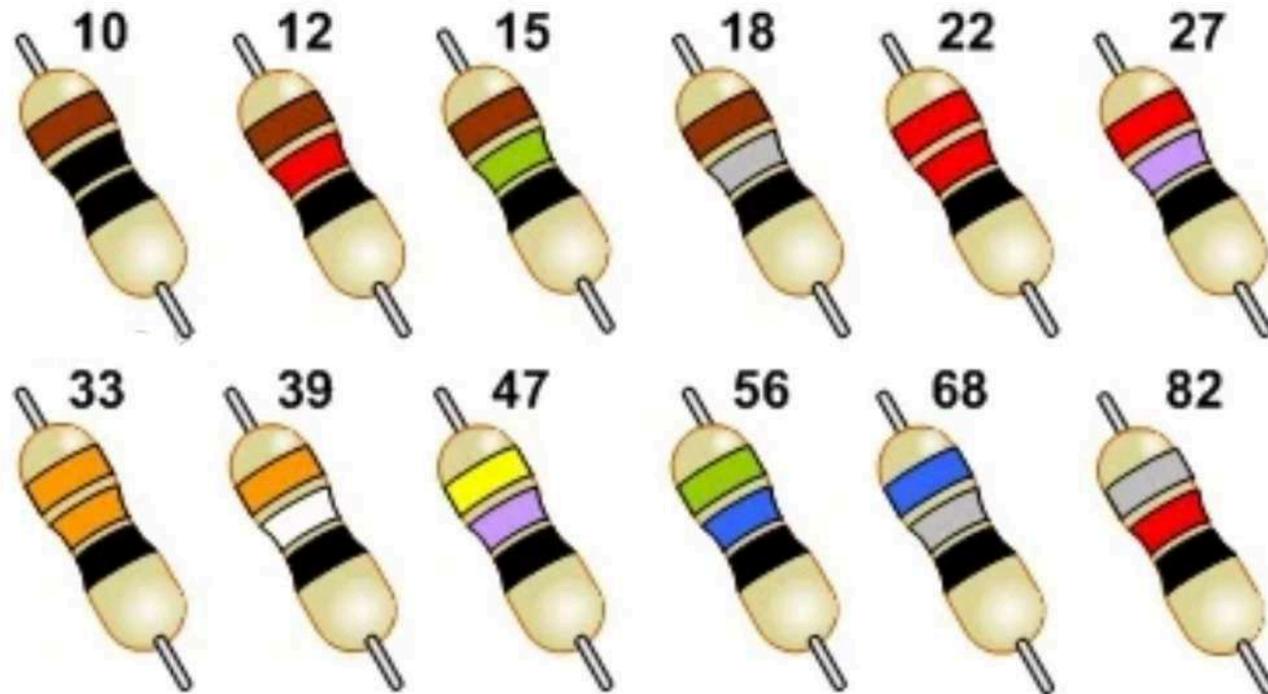
\* (IEC - I

Cada d  
logarítr

20% d

10% d

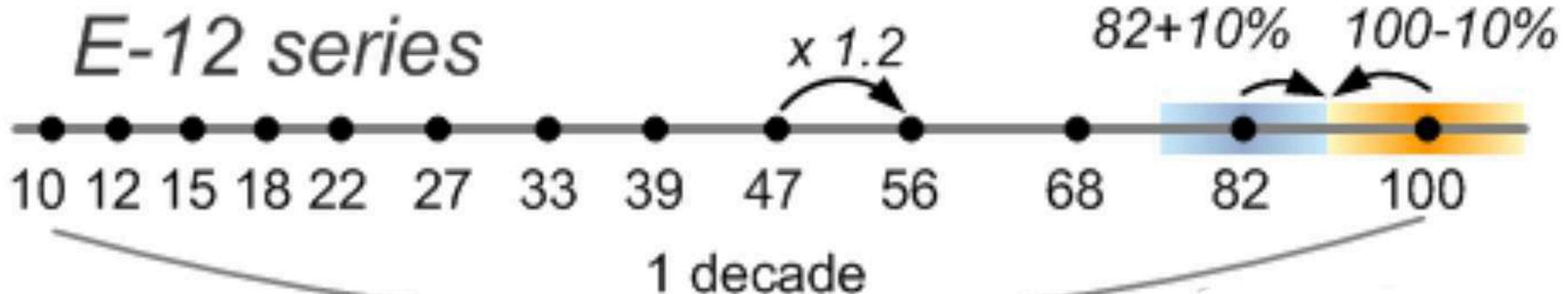
$$10^{\frac{1}{12}} =$$



paços

E-6);

o e escala.



# Resistores, valores comerciais

As séries valem para outros componentes:

Capacitores;

Indutores;

Diodos Zener.

A partir da série E-48 os componentes têm valores com 3 dígitos;

Outras séries:

E-96;

E-192.

Série E-6		
10	15	22
33	47	68

Série E-12 (10%)					
10	12	15	18	22	27
33	39	47	56	68	82

Série E-24 (5% e 1%)							
10	11	12	13	15	16	18	20
22	24	27	30	33	36	39	43
47	51	56	62	68	75	82	91

Série E-48 (2% e 1%)											
100	105	110	115	121	127	133	140	147	154	162	169
178	187	196	205	215	226	237	249	261	274	287	301
316	332	348	365	383	402	422	442	464	487	511	536
562	590	619	649	681	715	750	787	825	866	909	953

# Resistor - Acendendo um LED

## Resistância de R:

O cálculo é aplicação direta da lei de Ohm:

$$\text{Tensão: } 9 - V_{\text{led}} = 9 - 2 = 7 \text{ V}$$

Corrente: 10mA

$$R = 7 / 0,01 = 700 \Omega$$

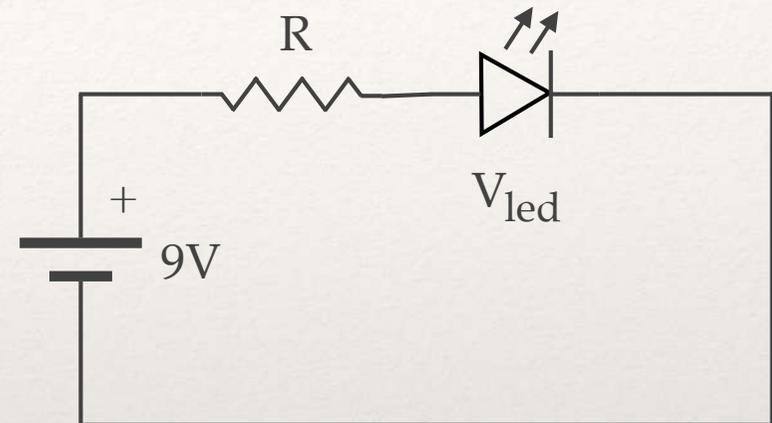
Valor comercial mais próximo é **680Ω**

## Potência de R:

$$i = 7 / 680 = 0,01\text{A (10mA)}$$

$$P = R \times i^2 = 680 \times 0,01^2 = 0,068 \text{ W}$$

Valor comercial é **1/8 W** (0,125 W)



# Resistor - 5V para 3,3V

## Divisor de tensão

Corrente baixa entre o GPIO de saída no Arduino e o terra;

5% da corrente máxima (20mA) = 1mA;

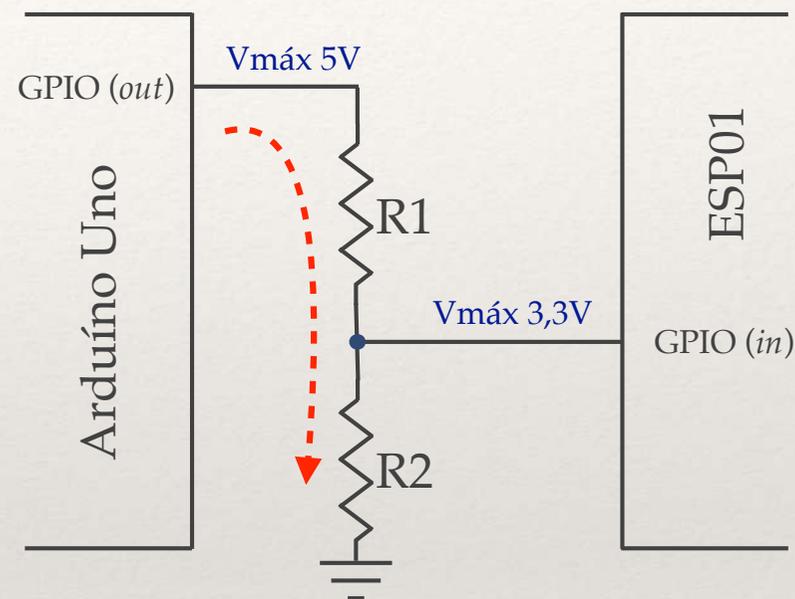
$$(R1 + R2) = 5V / 1mA = 5 K\Omega$$

$$R2 / (R1 + R2) = 3,3 / 5$$

$$R2 = 0,66 * 5K\Omega \cong \mathbf{3,3K\Omega} \text{ (valor comercial)}$$

$$R1 = 5K\Omega - 3,3K\Omega \cong 1,7K\Omega$$

$$R1 = \mathbf{1,8K\Omega} \text{ (valor comercial)}$$



# Resistor - 5V para 3,3V

## Divisor de tensão

Corrente baixa entre o GPIO de saída no Arduino e o terra;

5% da corrente máxima (20mA) = 1mA;

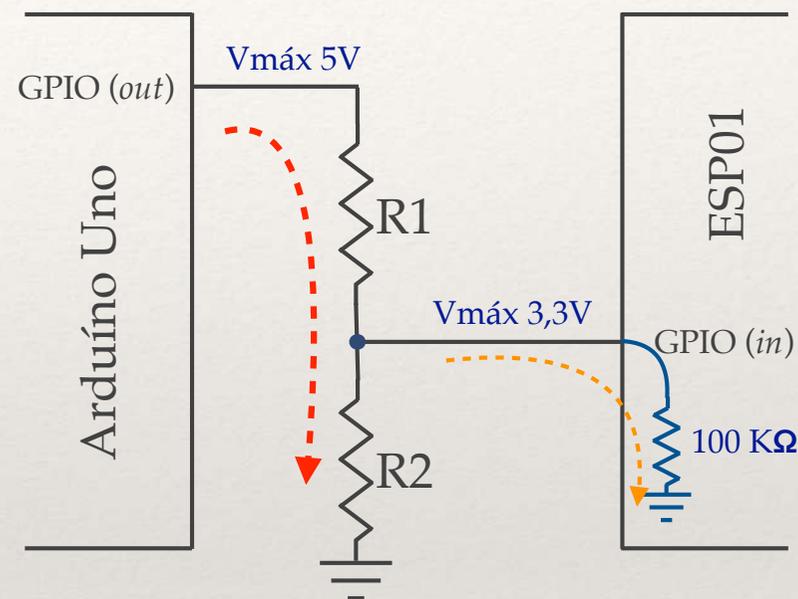
$$(R1 + R2) = 5V / 1mA = 5 K\Omega$$

$$R2 / (R1 + R2) = 3,3 / 5$$

$$R2 = 0,66 * 5K\Omega \cong \mathbf{3,3K\Omega}$$
 (valor comercial)

$$R1 = 5K\Omega - 3,3K\Omega \cong 1,7K\Omega$$

$$R1 = \mathbf{1,8K\Omega}$$
 (valor comercial)



## Corrente na carga

Entradas GPIO são de alta impedância (entre 100 KΩ e 1 MΩ);

$$R2 \text{ em paralelo com } 100 K\Omega = (1 / 3,3K\Omega + 1 / 100K\Omega)^{-1} = 3,2 K\Omega;$$

Variação de 3% para o menor valor de impedância.

# Resistor - *Pull-Up* / *Pull-Down*

## O problema das entradas desconectadas

É importante SEMPRE definir um valor padrão para uma entrada desconectada;

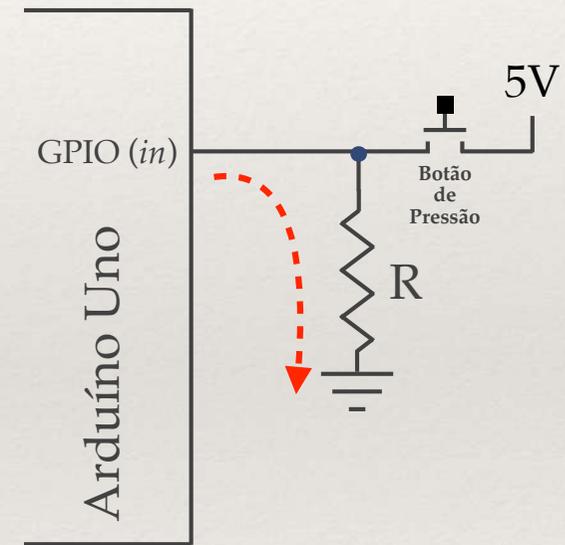
Ruídos elétricos podem determinar o valor de uma entrada desconectada;

5% da corrente máxima (20mA) = 1mA;

$R = 5V / 1mA = 4,7 K\Omega$

**Pull-Down:** pino funciona como fonte (*source*);

Pull-Up: pino funciona como dreno (*sink*)



# Resistor - *Pull-Up* / *Pull-Down*

## O problema das entradas desconectadas

É importante SEMPRE definir um valor padrão para uma entrada desconectada;

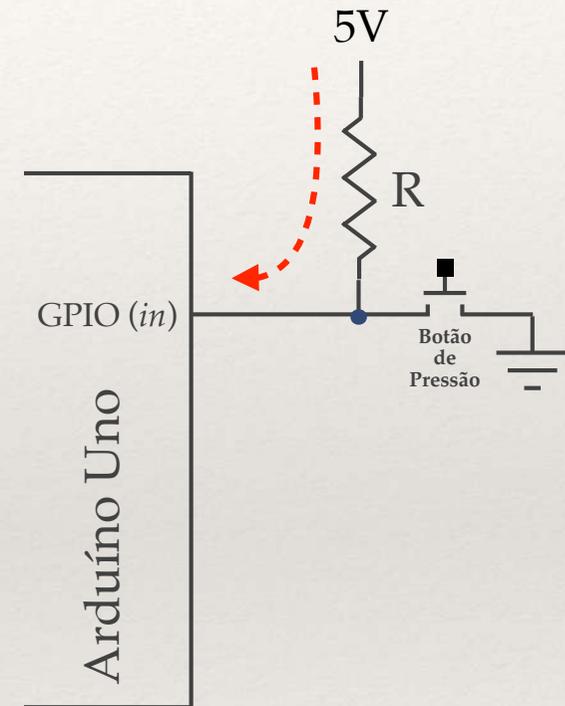
Ruídos elétricos podem determinar o valor de uma entrada desconectada;

5% da corrente máxima (20mA) = 1mA;

$$R = 5V / 1mA = 4,7 K\Omega$$

*Pull-Down*: pino funciona como fonte (*source*);

***Pull-Up***: pino funciona como dreno (*sink*)

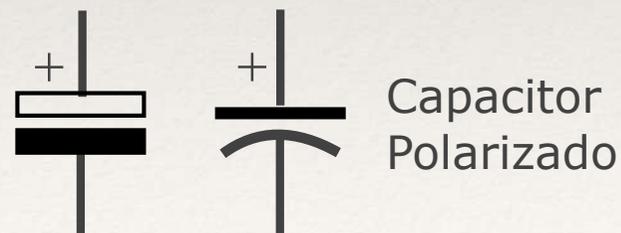
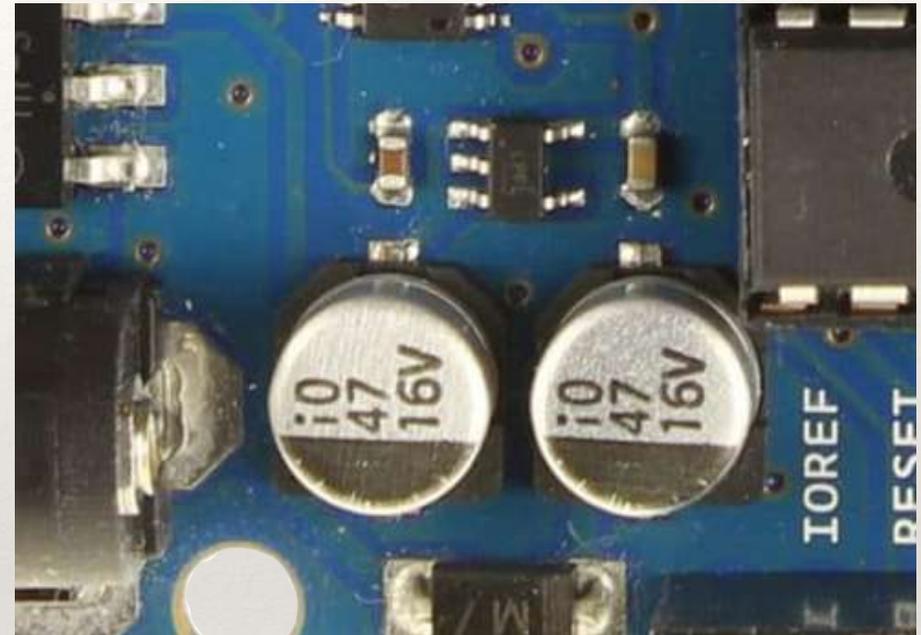


# Capacitor

Acumula tensão sob a forma de campo elétrico interno do componente, e é medido em faradays (F), tipicamente sub-múltiplos

Lembra uma bateria com carga e descarga muito rápida;

Diversas aplicações na eletrônica analógica, com algumas particularmente importantes em sistemas embarcados.



# Capacitor - Operação

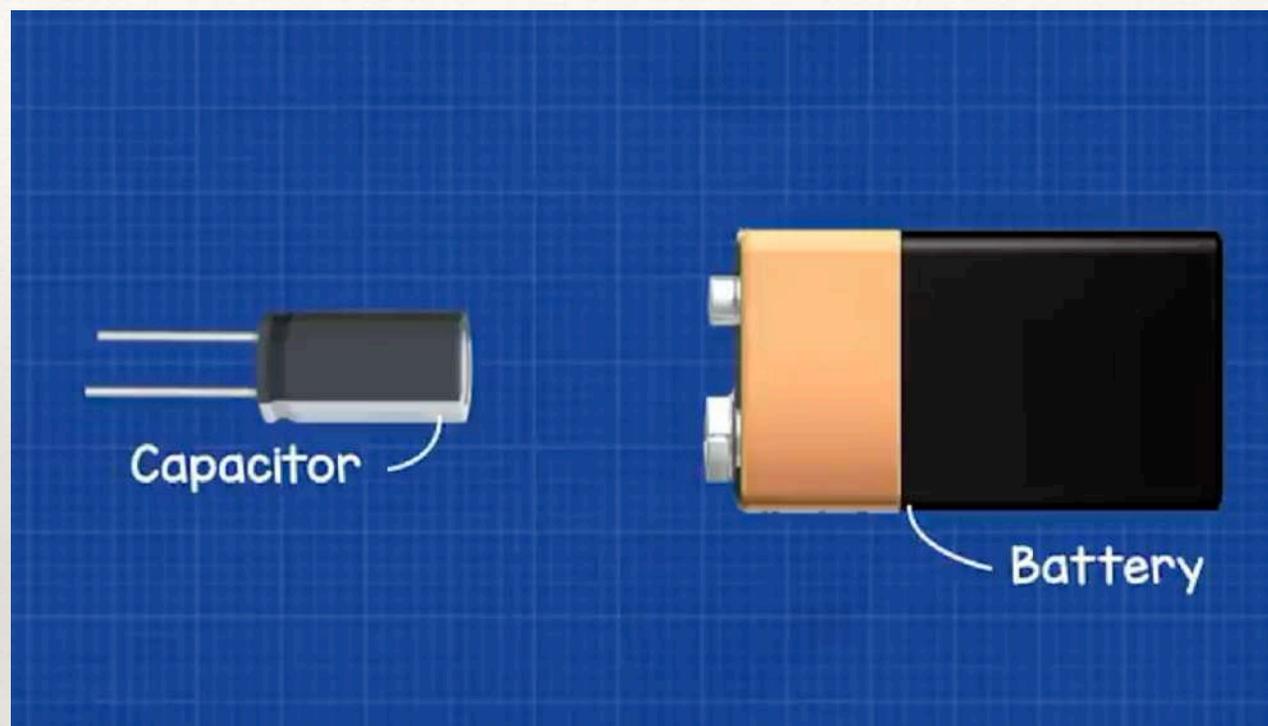
Quando conectado a uma fonte de tensão, acumula carga

Não há circulação de corrente entre os terminais, e sim apenas o acúmulo de carga;

A carga fica disponível até que circule corrente de descarga no sentido contrário;

Normalmente é especificado pela capacitância e tensão de operação. Algumas características específicas do tipo de capacitor também podem influenciar a aplicação;

A tensão de operação determina o tamanho.



# Tipos de Capacitor

## 1. Eletrolítico

Alta capacitância, polarizado, possui pequena indutância parasita

## 2. Poliéster

Autoregenerativo

## 3. Cerâmico

Um dos mais comuns

## 4. Tântalo

Substituem os eletrolíticos, polarizados, + vida útil, - espaço

## 5. Mica

Muito estável, usado em circuitos que exigem precisão

## 6. SMD

## 7. Variáveis



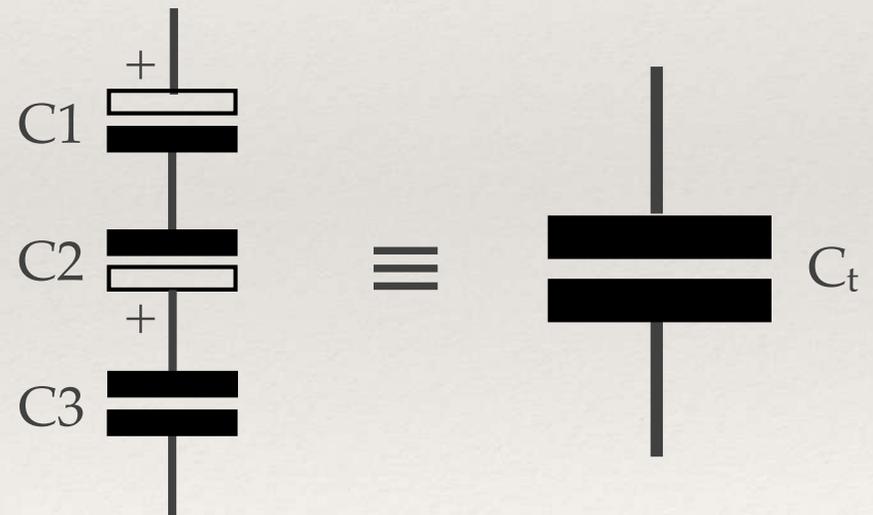
# Capacitores em Série

A capacitância total é menor que a menor das capacitâncias individuais;

$$\frac{1}{C_t} = \frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3}$$

O arranjo permite:

- Distribuir a tensão acumulada;
- Substituir capacitâncias individuais;
- Eliminar a polaridade (ver C1 e C2).



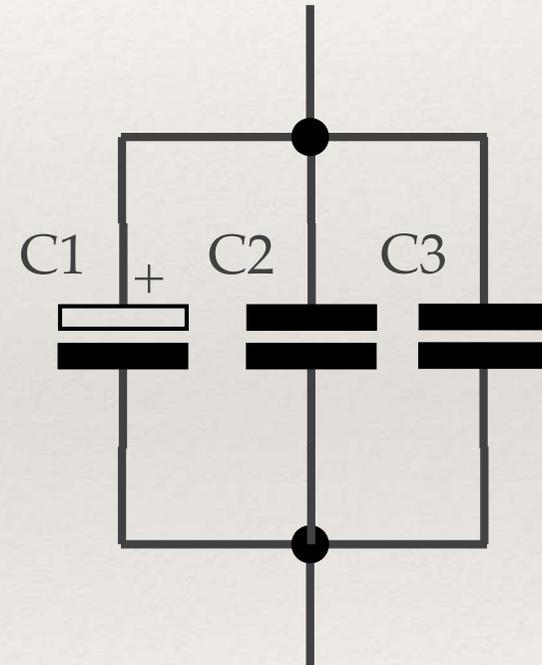
# Capacitores em Paralelo

A capacitância total é a soma das capacitâncias individuais;

O arranjo permite:

Substituir resistências individuais;

Agregar características de dois tipos diferentes de capacitor (na figura, C1 é eletrolítico e C2 cerâmico).



# Capacitores em Paralelo

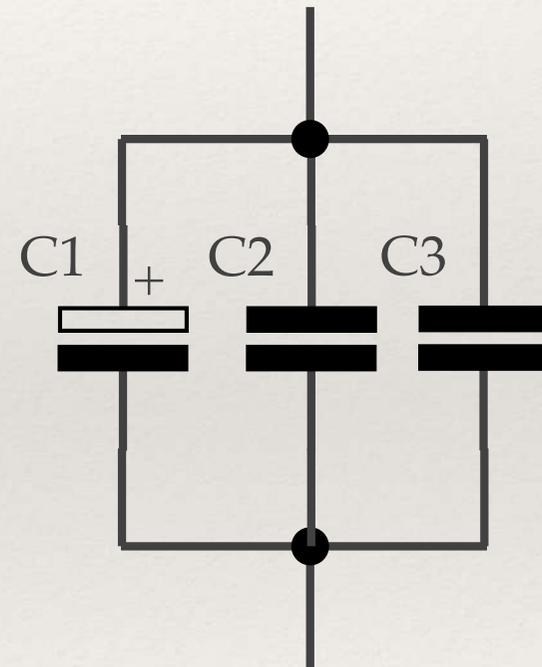
A capacitância total é a soma das capacitâncias individuais;

O arranjo permite:

Substituir resistências individuais;

Agregar características de dois tipos diferentes de capacitor (na figura, C1 é eletrolítico e C2 cerâmico).

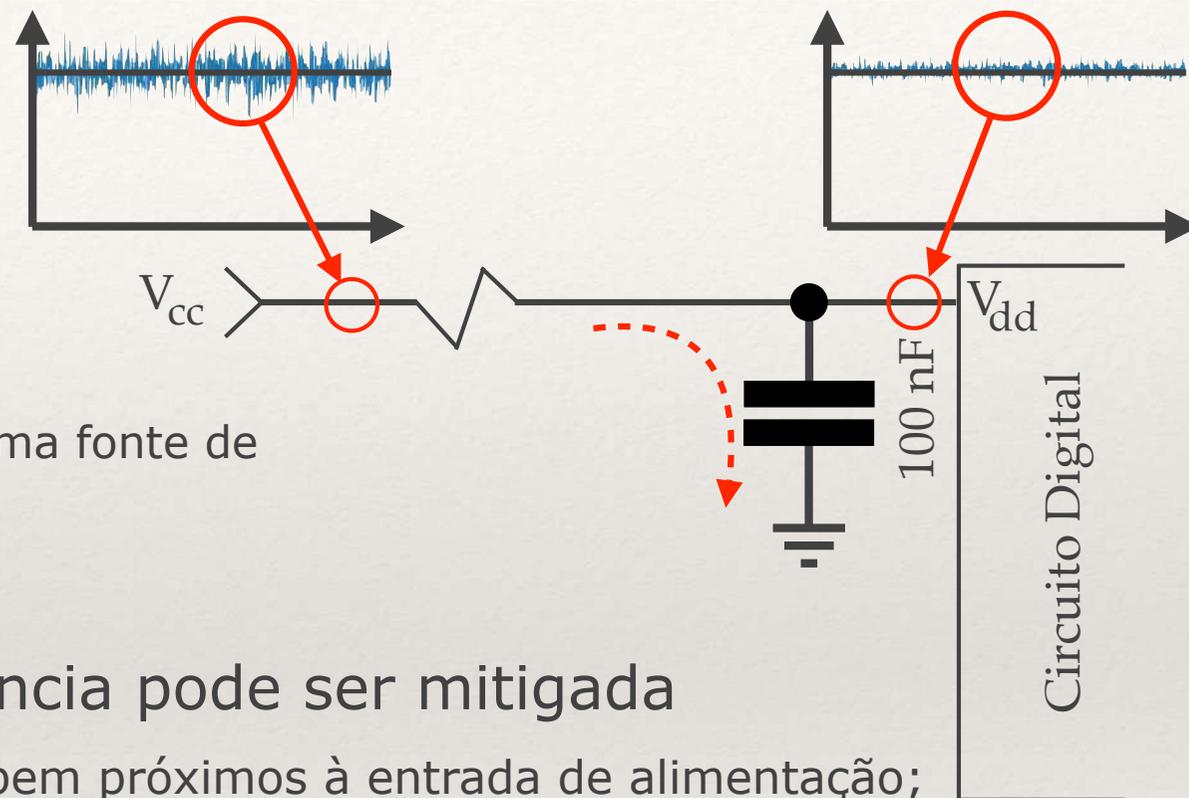
$$C_T = C_1 + C_2 + C_3$$



# Capacitor - Desacoplamento

É comum encontrarmos interferências na linha de alimentação, derivadas:

- do próprio circuito;
- de equipamentos ligados na mesma fonte de alimentação;
- de interferência eletromagnética.



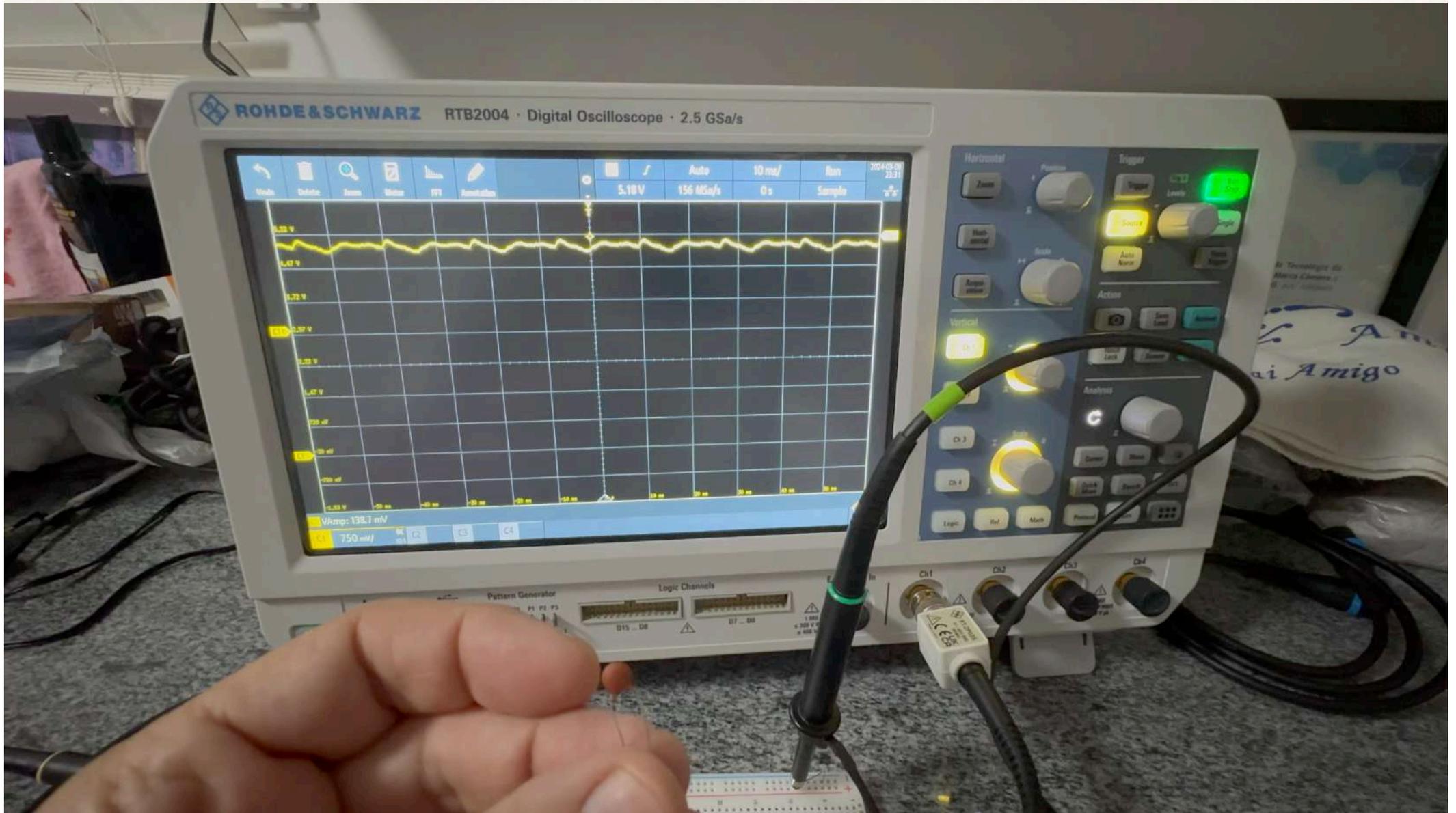
Interferência de alta frequência pode ser mitigada

Capacitores de 100nF instalados bem próximos à entrada de alimentação;  
*(cálculo foge do escopo da disciplina)*

Proteção advém da carga e descarga instantânea do capacitor;

Microcontroladores e circuitos integrados digitais são particularmente sensíveis a este problema.

# Capacitor - Desacoplamento



# Capacitor - Estabiliza tensão

Se houver instabilidade de baixa frequência na tensão de alimentação, usamos um capacitor maior

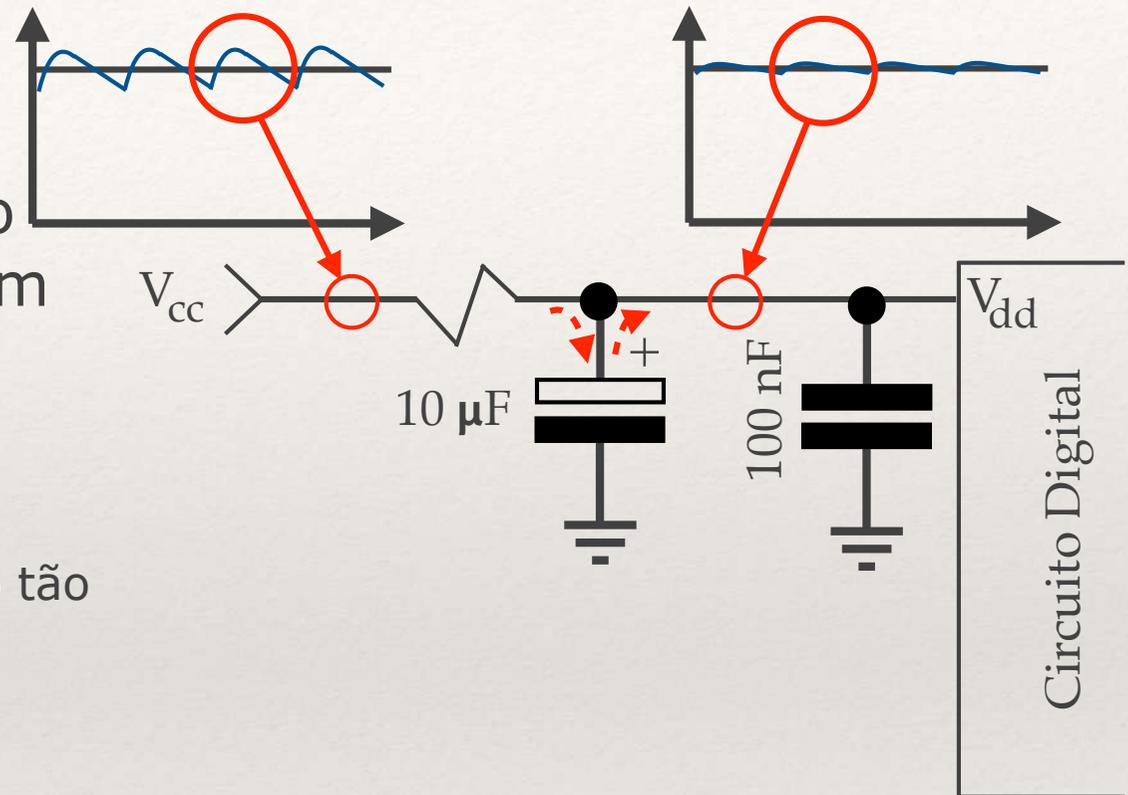
Isolado, ou em conjunto com anterior;

Capacitores eletrolíticos não são tão eficientes em altas frequências;

Além destas aplicações ...

Capacitores são essenciais em diversos circuitos analógicos;

Compõem filtros dos mais diversos tipos, fontes de alimentação, acopladores e desacopladores, sintonizadores, e dezenas de outras aplicações.



# Aula 06

# Semicondutores

Em um átomo, os elétrons podem ocupar determinadas órbitas ou níveis de energia

São sete: k, l, m, n, o, p e q

O nível mais externo é chamado de nível de valência;

Para conduzir uma corrente elétrica, os elétrons precisam saltar do nível de valência para o nível de condução.

A existência de níveis ou bandas de energia “proibidas” entre a condução e a valência determinam o tipo de material

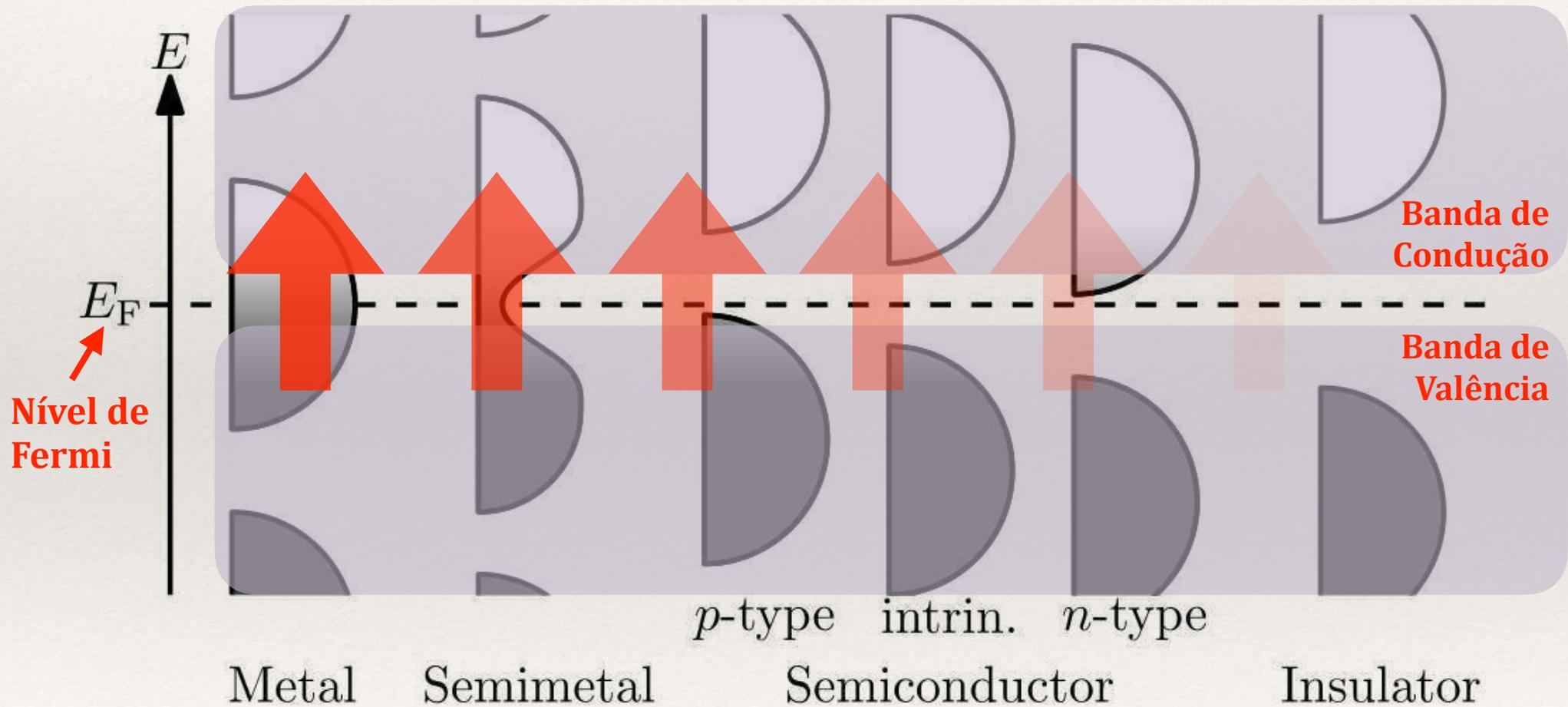
Não existe banda proibida: **CONDUTOR**

Existe uma banda proibida de grande dimensão: **ISOLANTE**

Existe uma banda proibida de pequena dimensão: **SEMICONDUTOR**



# Bandas de Valência e Condução



---

# Semicondutores

---

Para nosso estudo, interessa conhecer os semicondutores

Banda de valência próxima da banda de condução;

Temperatura, choques mecânicos, luz, radiação e, obviamente, tensão e corrente elétrica podem fazer os elétrons migrarem de banda. Com “elétrons livres”, o material funciona como um condutor;

O elemento semiconductor mais comum é o Silício, que possui 4 elétrons na camada de valência. É o segundo elemento mais comum na Terra, mas outros materiais podem ser utilizados, como o Germânio;

## Semicondutores tipo “N” e tipo “P”

São produzidos pela dopagem do Silício com “impurezas”;

Tipo “N”: dopagem com átomos pentavalentes como o Fósforo (sobra um elétron por átomo), tornando o material propenso a liberar elétrons;

Tipo “P”: dopagem com átomos trivalentes como o Alumínio (falta um elétron por átomo), tornando o material propenso a receber elétrons.

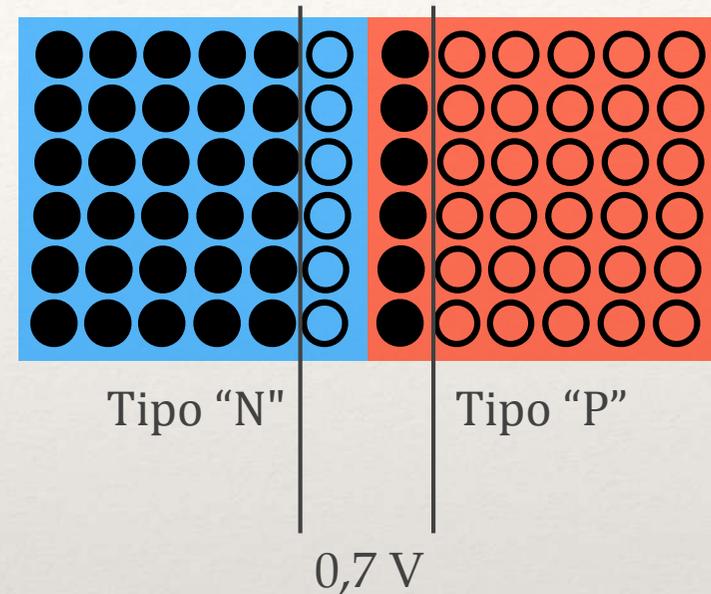
# Junção Semicondutora

Obtida pela junção de dois materiais dopados de forma diferente

Na junção, ocorre migração dos elétrons em excesso do material N para o material P, formando a camada de depleção;

Diferença de potencial ( $\approx 0,7V$  no silício) impede a continuidade de circulação de corrente.

\* No germânio, a tensão é de  $0,3V$ .

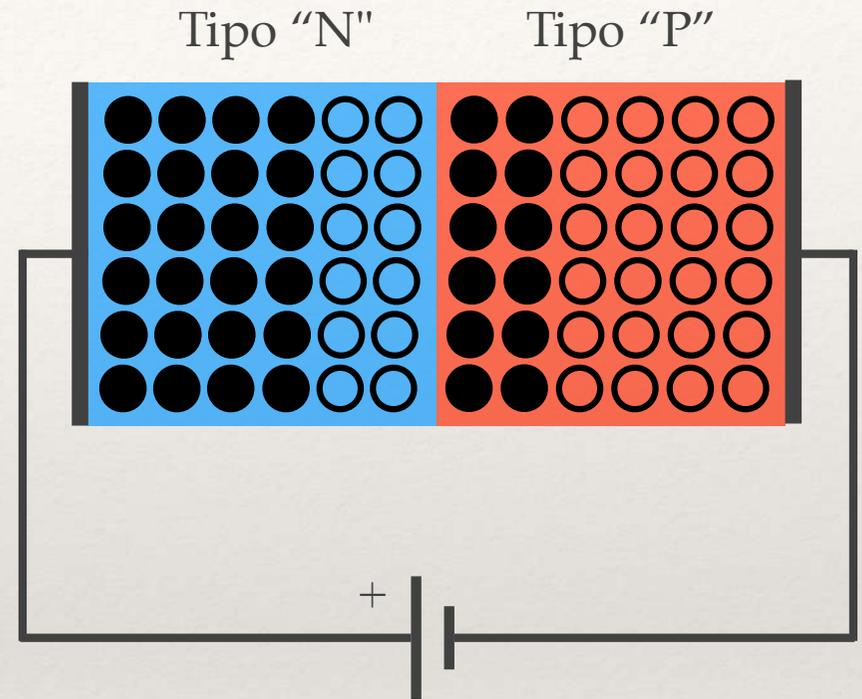


# Junção Semicondutora

E quando a junção é polarizada?

Com o material N ligado ao positivo, aumenta a camada de depleção, e não circula corrente;

Invertendo, desde que a tensão seja maior que a diferença de potencial da junção, haverá circulação de corrente.

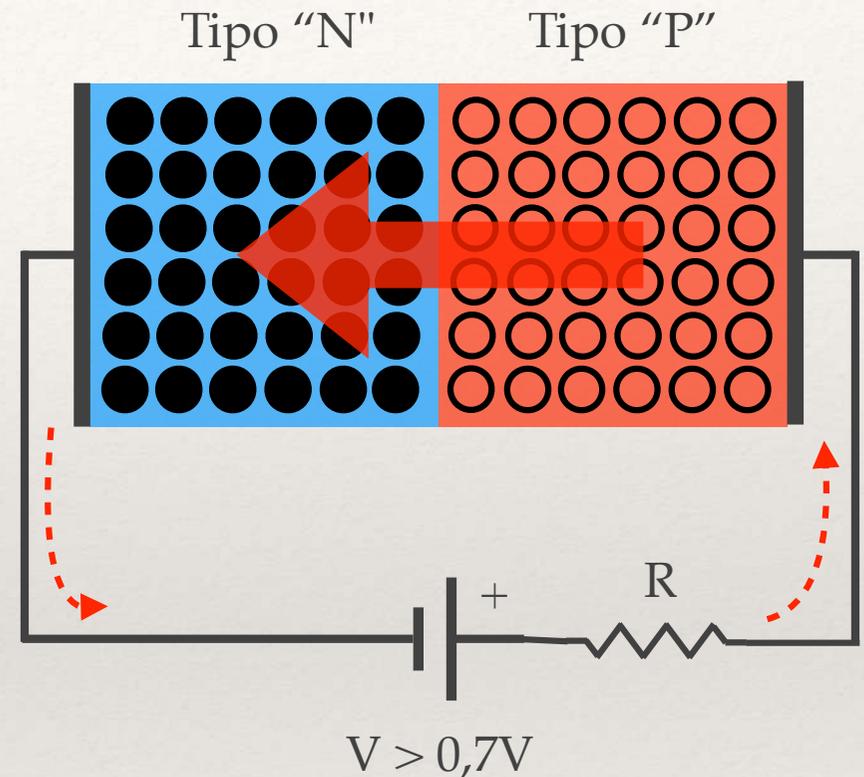


# Junção Semicondutora

E quando a junção é polarizada?

Com o material N ligado ao positivo, aumenta a camada de depleção, e não circula corrente;

Invertendo, desde que a tensão seja maior que a diferença de potencial da junção, haverá circulação de corrente.



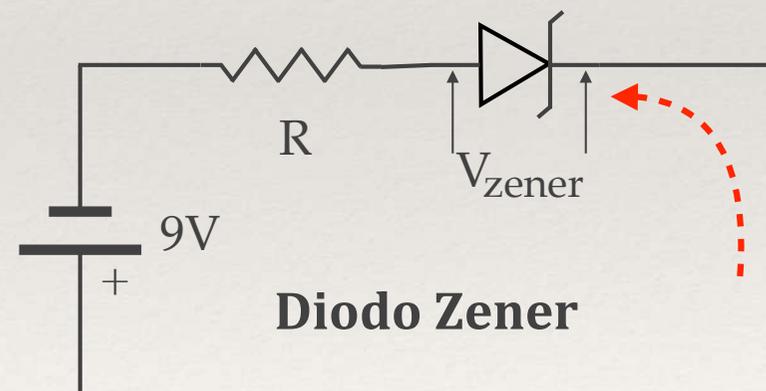
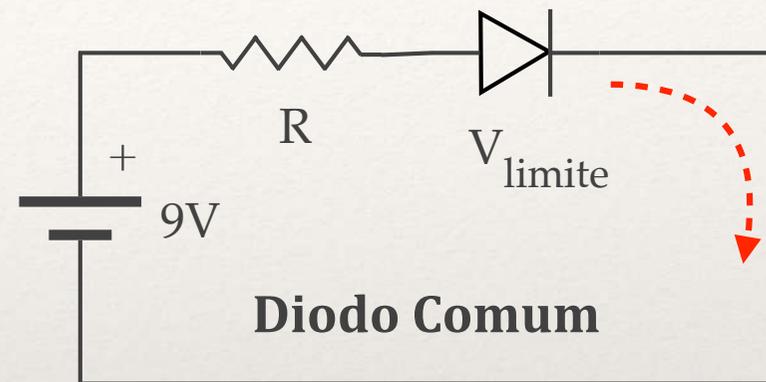
# Diodo: usando a junção P-N

Só permite a circulação de corrente em um sentido

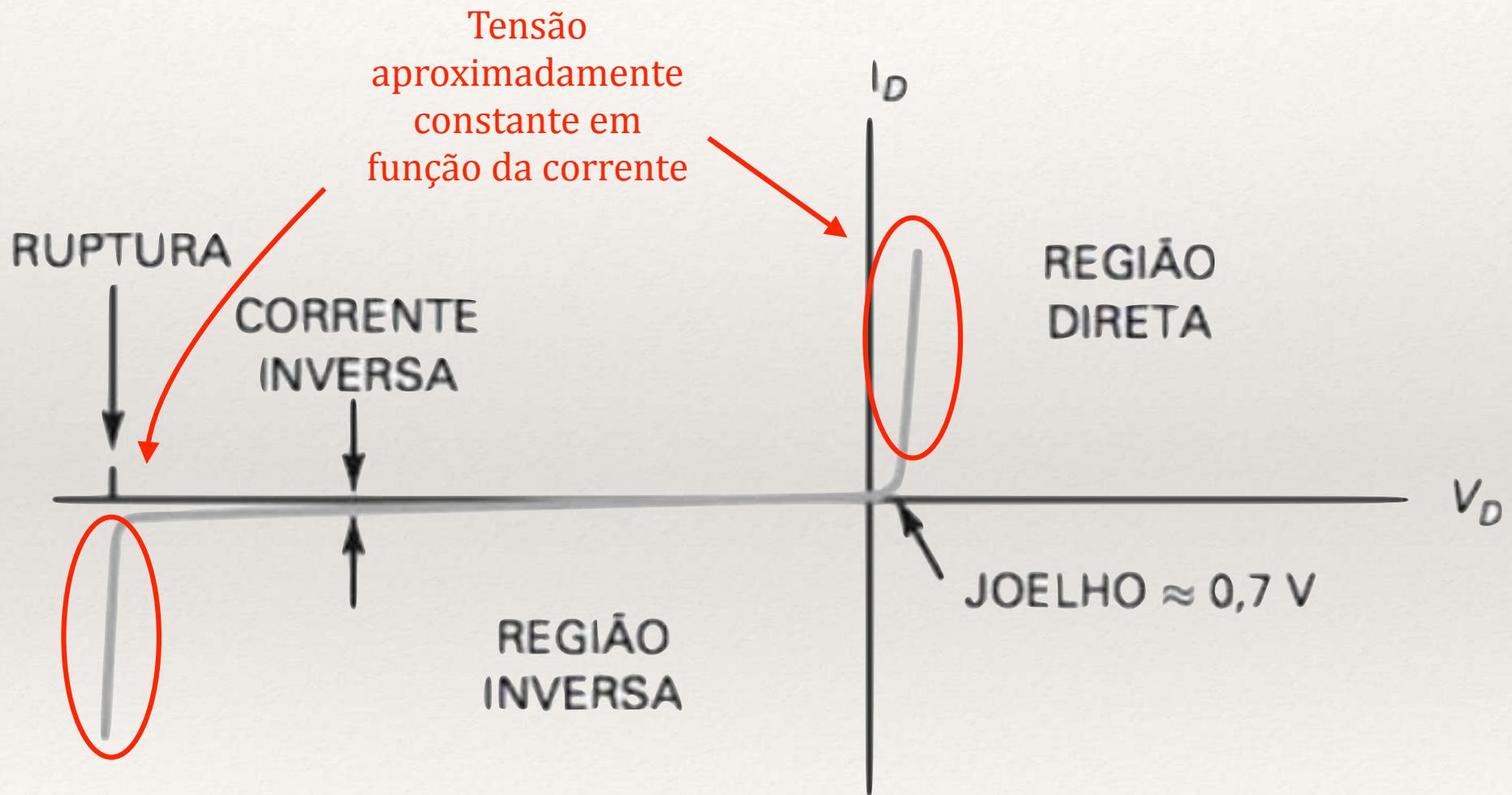
Exige tensão mínima limite ( $\approx 0,7V$ );

Corrente só circula no sentido contrário se a tensão aplicada ultrapassar a tensão de ruptura;

Um tipo específico de diodo aproveita este efeito (chamado de “efeito zener”) para outras aplicações.



# Diodo: curva característica



# Tipos de Diodo

## 1. Retificador

Fontes de alimentação, suporta altas correntes;

## 2. SMD

## 3. Alta potência

Para correntes elevadas;

## 4. Sinal

Utilizados em circuitos de alta frequência. Frequentemente são de germânio;

## 5. Duplo

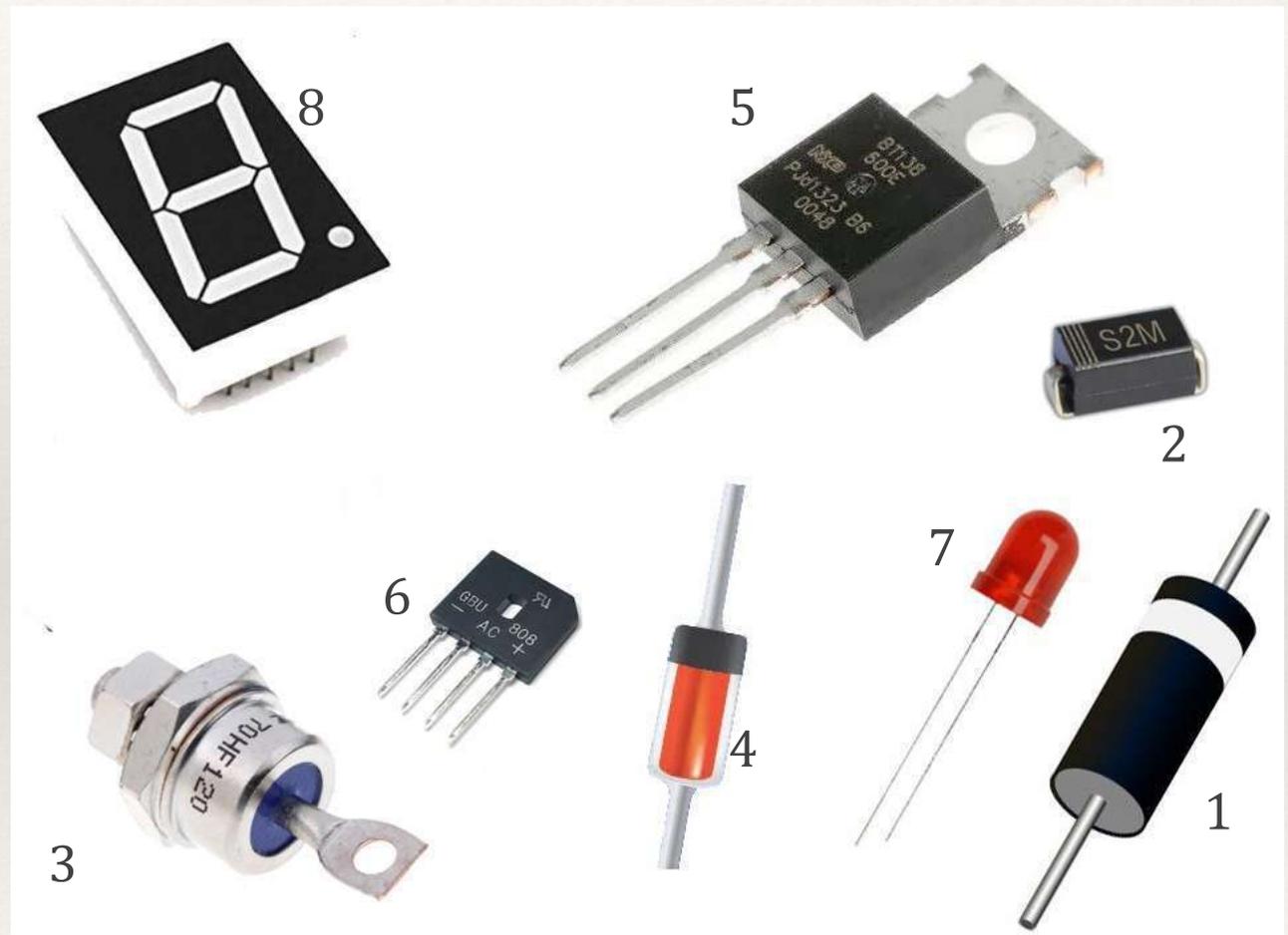
2 diodos em contra-posição;

## 6. Ponte retificadora

Arranjo de 4 diodos para fontes de alimentação

## 7. LED

## 8. Display de 7 segmentos



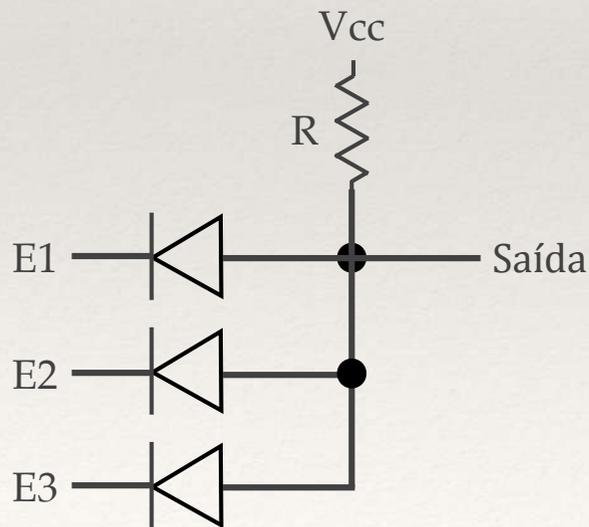
# Diodo: portas lógicas simples

## Porta AND (E)

Qualquer entrada (E1 a E3) no nível zero provoca o nível zero na saída;

Apenas se todas as entradas estiverem em nível um garantimos a saída no nível 1;

O resistor impede um curto-circuito.

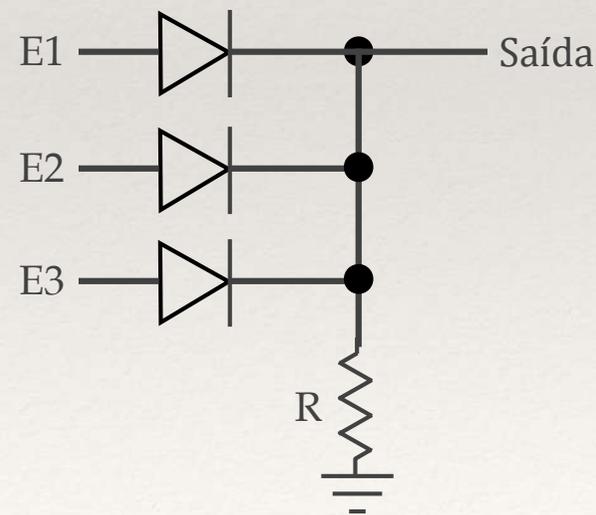


## Porta OR (OU)

Qualquer entrada (E1 a E3) no nível um provoca o nível um na saída;

Apenas se todas as entradas estiverem no nível zero garantimos a saída no nível zero;

O resistor impede um curto-circuito.

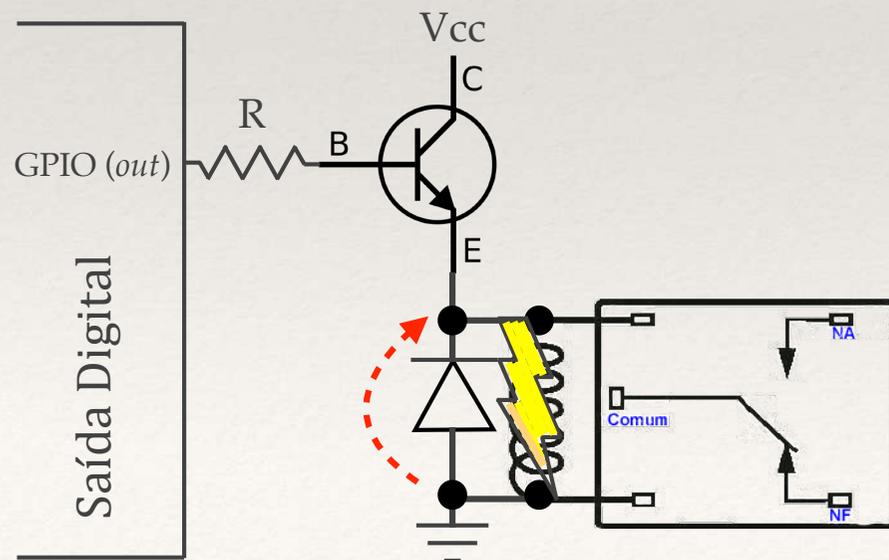


# Diodo: proteção de circuitos

Em circuitos digitais, são utilizados para proteção

Diodos comuns impedem polarização invertida;

Proteção funciona inclusive para pulsos reversos de alta tensão (relês).

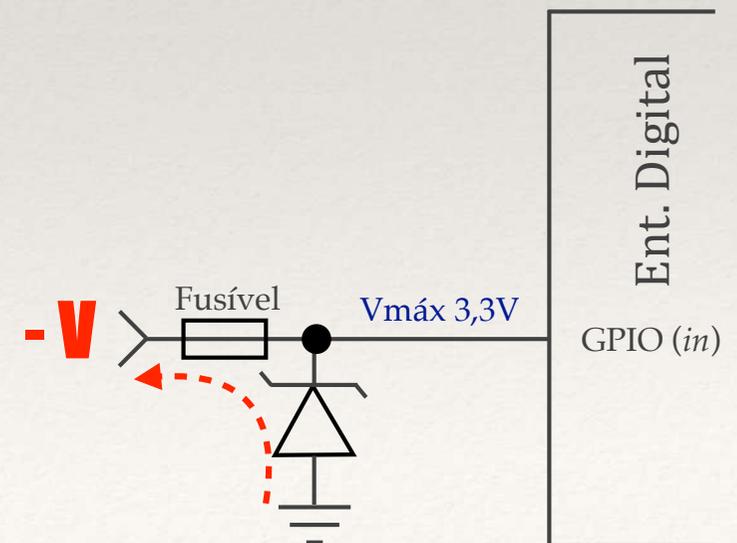


Diodos zener protegem entradas digitais sensíveis

Proteção contra inversão de polaridade;

Proteção contra sobretensões;

Fusível em série contra sobrecorrentes.

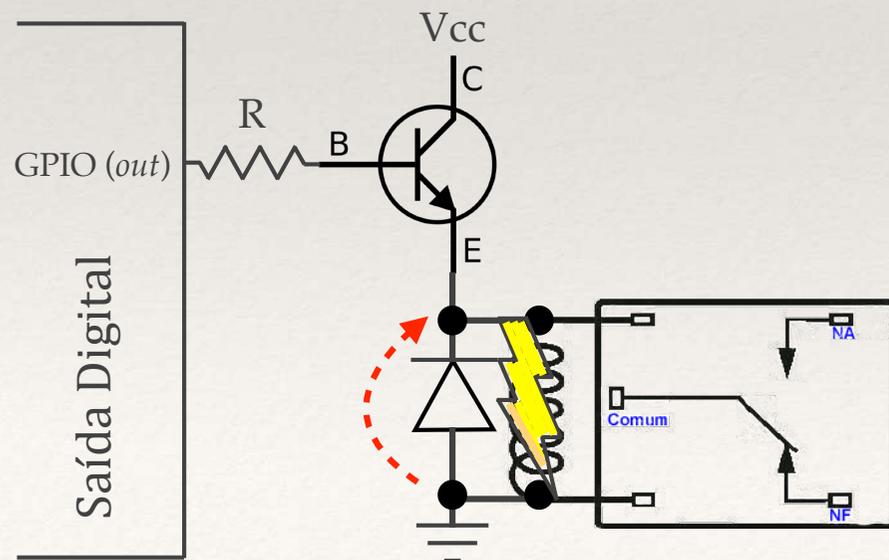


# Diodo: proteção de circuitos

Em circuitos digitais, são utilizados para proteção

Diodos comuns impedem polarização invertida;

Proteção funciona inclusive para pulsos reversos de alta tensão (relês).

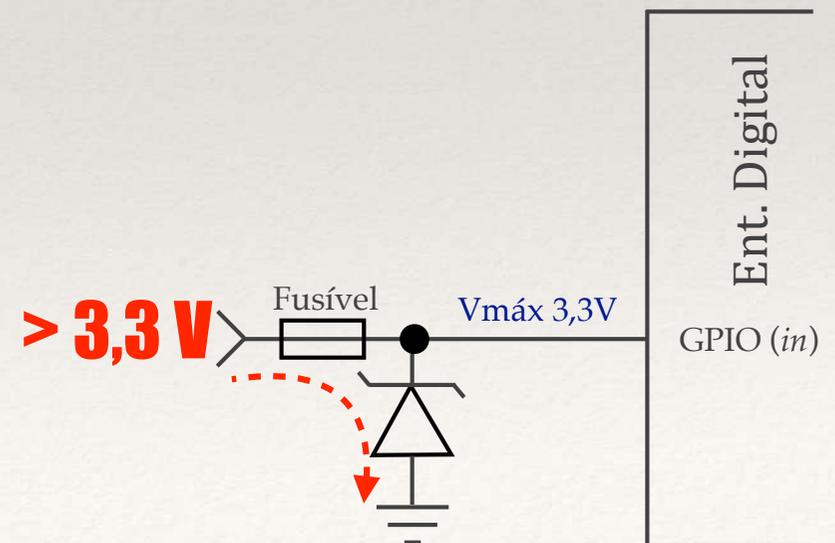


Diodos zener protegem entradas digitais sensíveis

Proteção contra inversão de polaridade;

Proteção contra sobretensões;

Fusível em série contra sobrecorrentes.



# O Transístor

São dois os principais tipos:

Bipolar de Junção: controlado pela corrente que circula pela base;

Efeito de Campo: controlado pela tensão presente no *gate*.

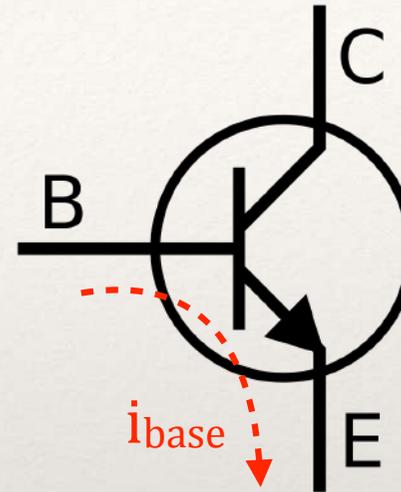
## Transistores Bipolares

Podem ser do tipo NPN ou PNP;

## Aplicações Típicas

Chaveadores;

Amplificadores.



Bipolar NPN



J-FET Canal N

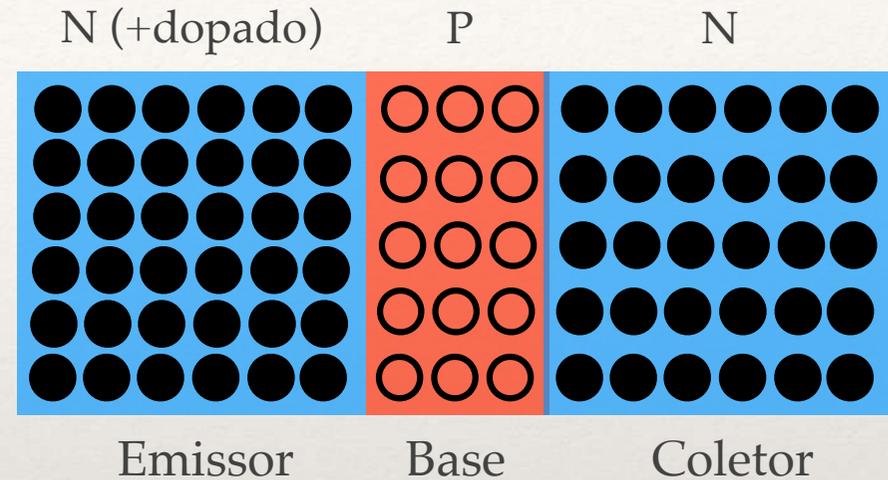
# Transistor: duas junções

Similar ao diodo, mas...

Temos duas camadas de depleção;

O emissor tem uma dopagem maior.

Com isso, a camada de depleção entre base e emissor libera elétrons em excesso;



# Transístor: duas junções

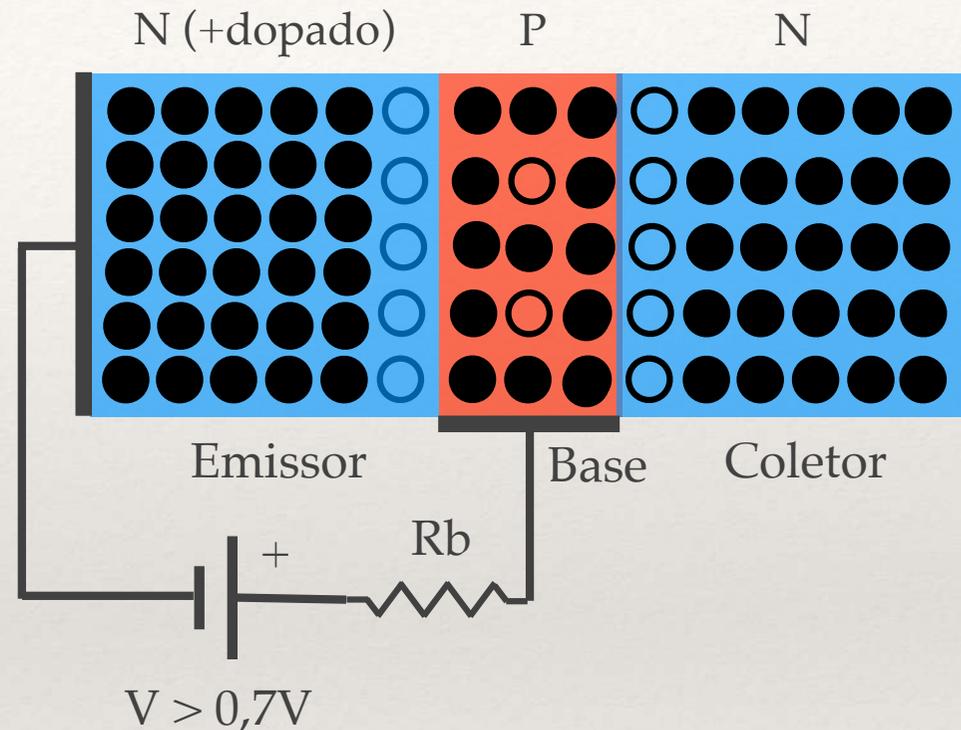
Similar ao diodo, mas...

Temos duas camadas de depleção;

O emissor tem uma dopagem maior.  
Com isso, a camada de depleção entre  
base e emissor libera elétrons em  
excesso;

Se fizermos circular uma  
corrente entre a base e o  
emissor:

A camada de depleção correspondente  
desaparecerá, e ainda ficarão elétrons  
livres devido à diferença na dopagem;



# Transistor: duas junções

Similar ao diodo, mas...

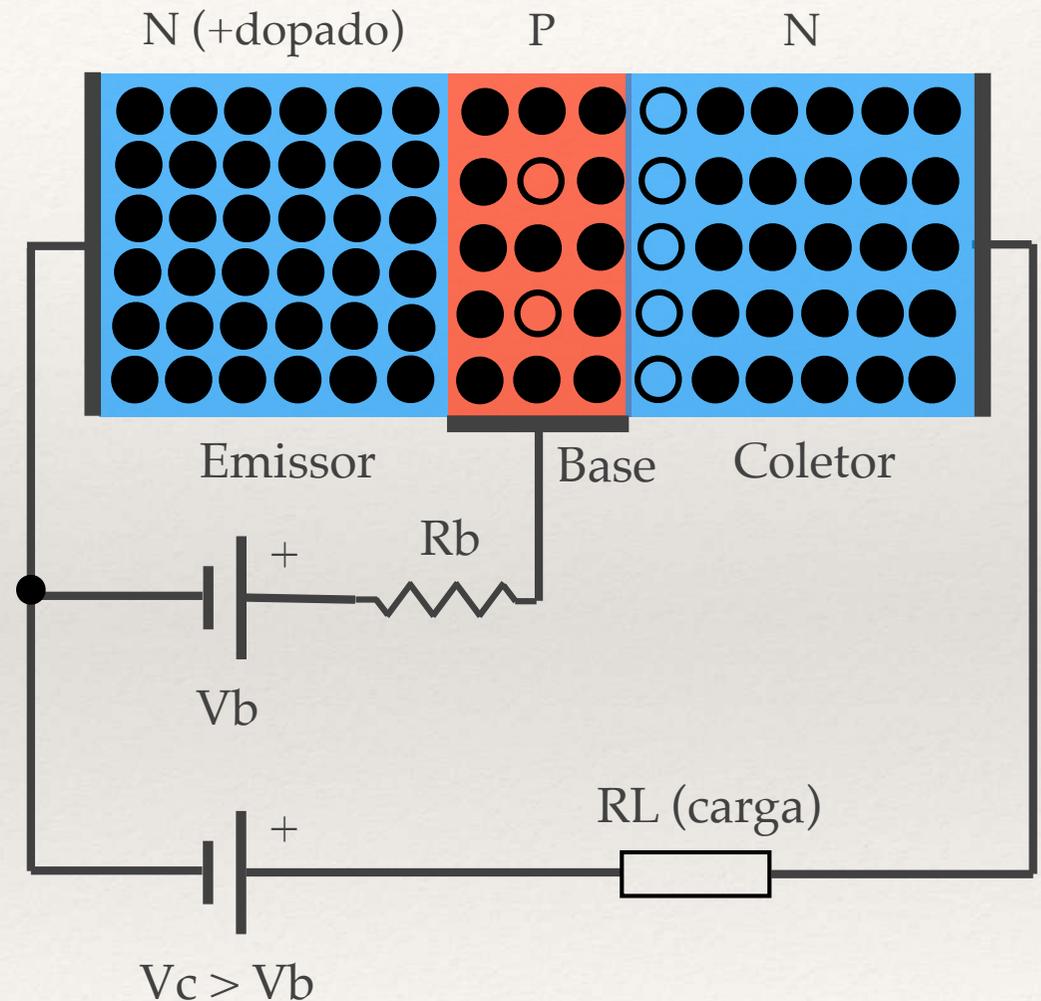
Temos duas camadas de depleção;

O emissor tem uma dopagem maior. Com isso, a camada de depleção entre base e emissor libera elétrons em excesso;

Se fizermos circular uma corrente entre a base e o emissor:

A camada de depleção correspondente desaparecerá, e ainda ficarão elétrons livres devido à diferença na dopagem;

Uma corrente entre coletor e base será controlada pela corrente de base.



# Transistor: duas junções

Similar ao diodo, mas...

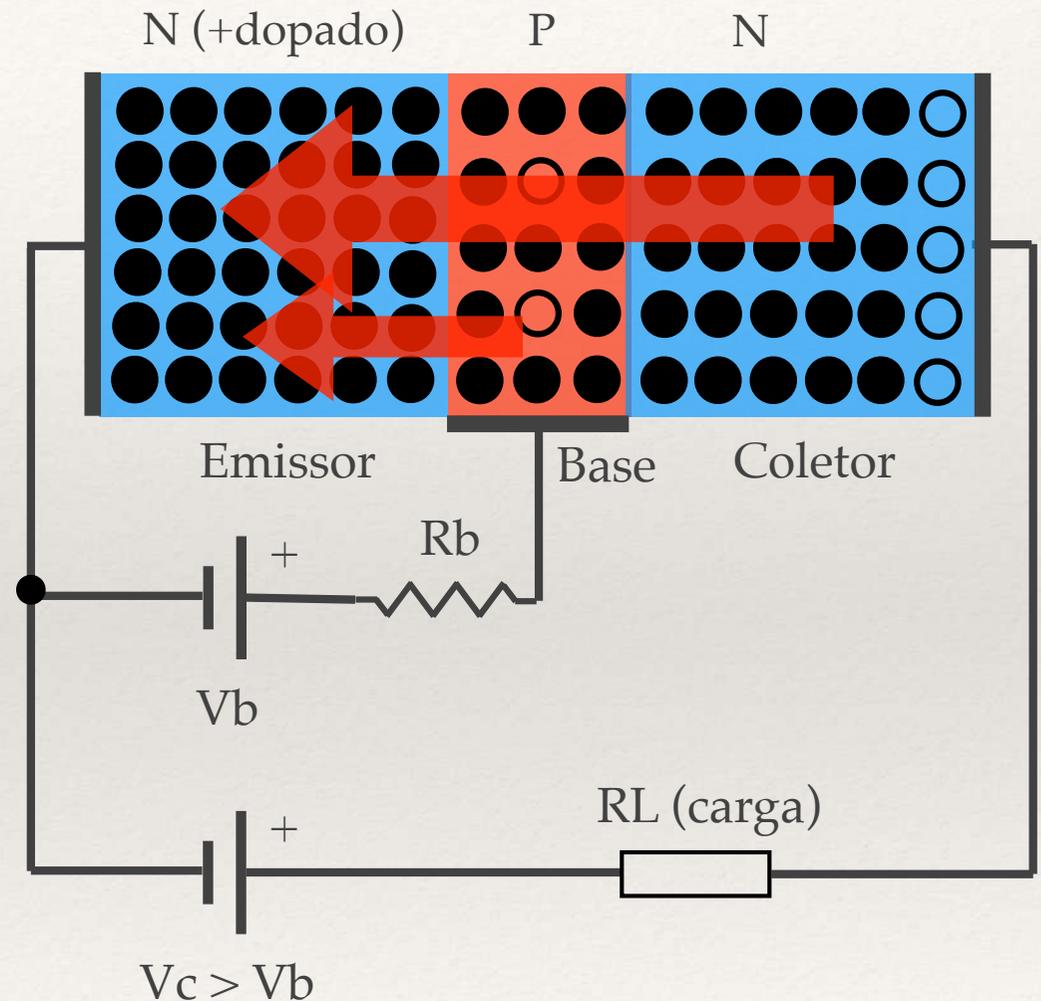
Temos duas camadas de depleção;

O emissor tem uma dopagem maior. Com isso, a camada de depleção entre base e emissor libera elétrons em excesso;

Se fizermos circular uma corrente entre a base e o emissor:

A camada de depleção correspondente desaparecerá, e ainda ficarão elétrons livres devido à diferença na dopagem;

Uma corrente entre coletor e base será controlada pela corrente de base.



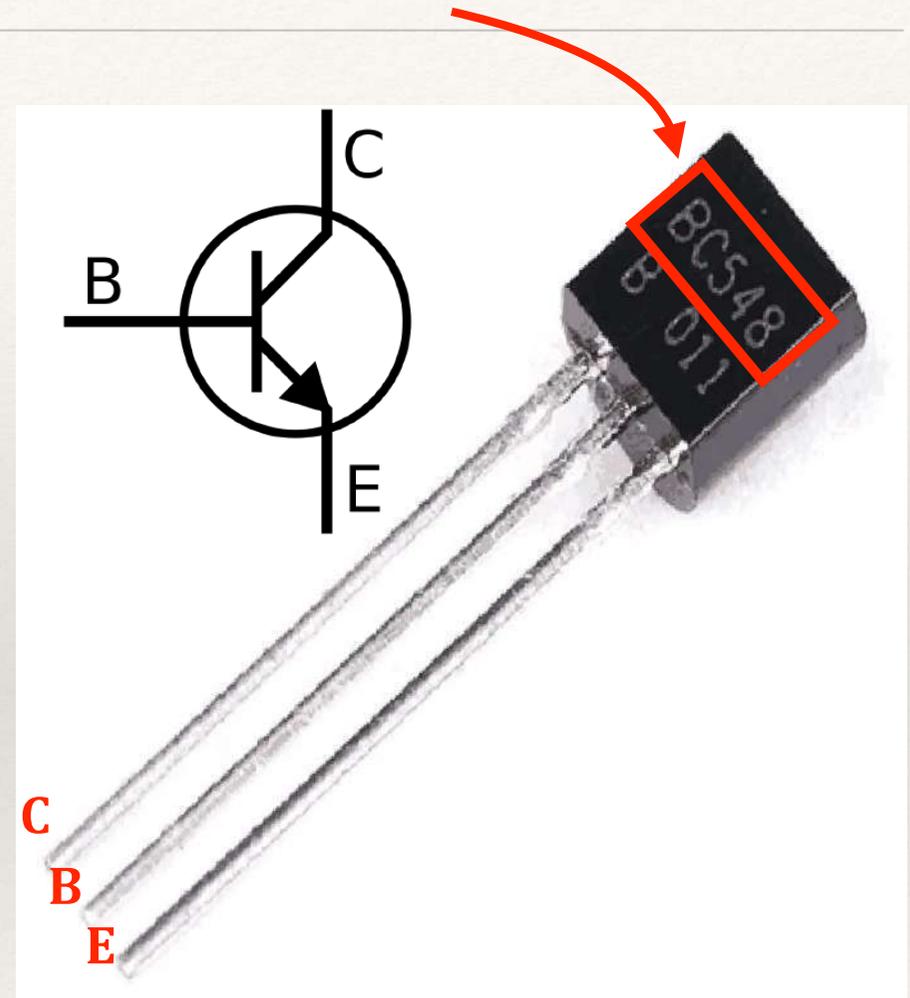
# Transístor

Especificações estão associadas ao código do componente

Código normalmente vem impresso no corpo do transistor, além de número do lote e outras especificações;

Na figura ao lado, temos o BC548, transístor NPN popular de baixa potência.

Na folha de dados ([datasheet](#)), encontramos, entre outras coisas, a "pinagem" do componente.



# Polarizando um Transistor

“Polarizar” um transistor é projetar as correntes que devem passar pelos seus terminais

$$i_c = \beta \times i_b \quad (\beta \text{ é o ganho do transistor})$$

$$i_e = i_c + i_b$$

$$V_{be} \approx 0,7 \text{ V (queda na junção)}$$

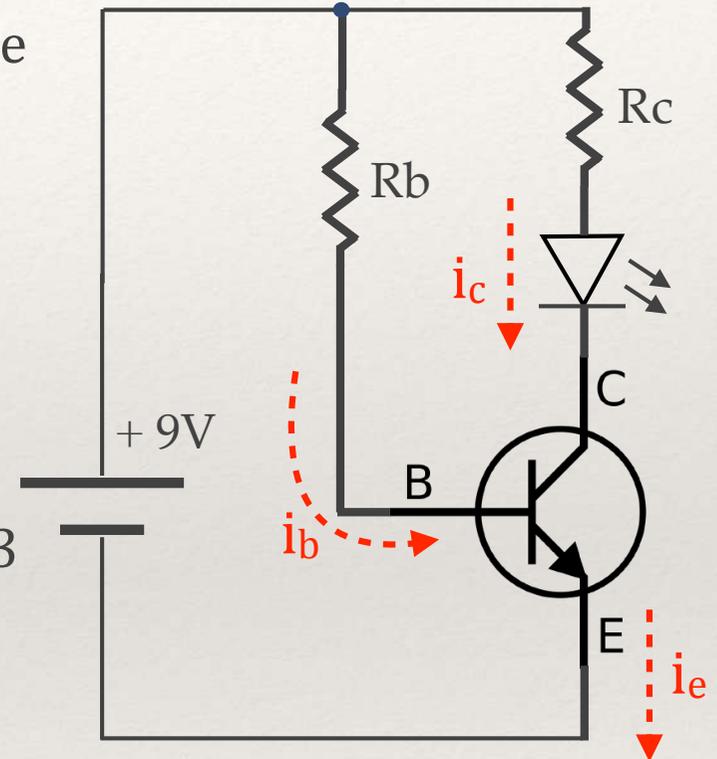
No PNP, o sentido de circulação da corrente se inverte.

A depender dos valores, o transistor pode ficar em 3 diferentes “zonas”:

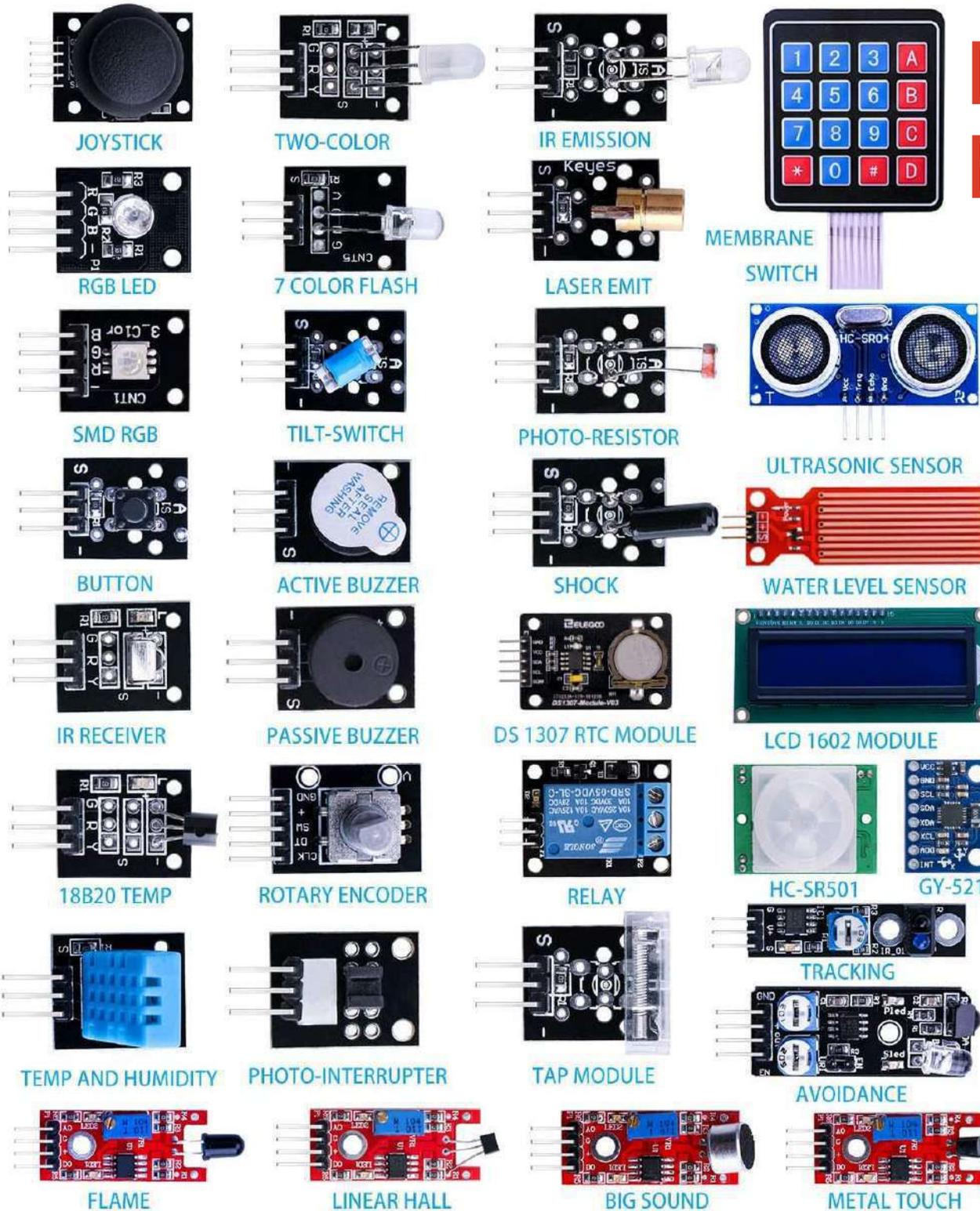
Corte, Ativa (amplificação) e Saturação;

Para operar como chave, nos interessa as zonas de corte e saturação;

Valor prático para  $i_b$  é 10% de  $i_c$  para operar na zona de saturação independente de  $\beta$ .



# Dispositivos de Entrada e Saída



---

# Dispositivos de Entrada

---

**Chaves e Botões;**

**Teclados** (ex. teclado de membrana);

**Sensores digitais** (chuva, presença e movimento, chave magnética, gases, sensor de digital etc);

**Sensores analógicos** (potenciômetro, umidade, pressão, iluminação, temperatura, ultrasom, tensão, corrente, campo magnético etc);

**Sensores de pulsos** (fluxo d'água, contadores etc);

**Dispositivos especiais** (giroscópio, acelerômetro, RTC).

# Aula 07

# Interface na Prática

A tensão elétrica dos sinais trocados com o mundo externo codificam a informação transferida

Portas digitais ESCREVEM E LEEM sinais codificados digitalmente

Tensão alta (HIGH) = 1;

Tensão baixa (LOW) = 0;

Portas analógicas LEEM sinais codificados analogicamente

O valor da tensão indica o valor analógico;

Teoricamente, existem "infinitos" valores (na prática não. Por quê? )

Sinais Elétricos  
(valores de tensão digitais)



Sinais Elétricos  
(valores de tensão analógicos)

---

# Dispositivos de Entrada

---

## Chaves e Botões

Método básico de entrada;

Chaves momentâneas, fixas e *encoders*;

Foto-isoladores;

*Pull Up* ou *Pull Down*

Efeito *Bouncing*

*Debouncing* por SW ou HW

---

# Dispositivos de Entrada

---

## Chaves e Botões

Método básico de entrada;

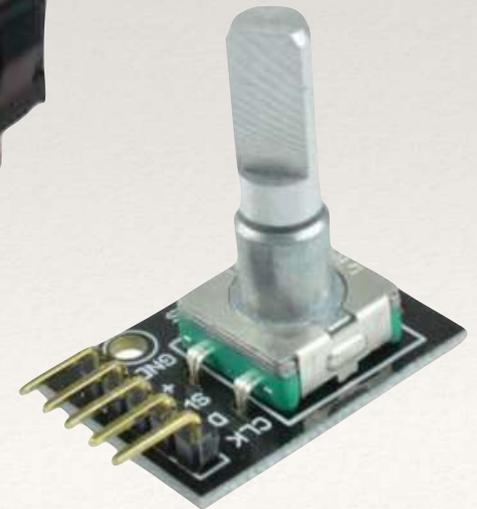
Chaves momentâneas, fixas e *encoders*;

Foto-isoladores;

*Pull Up* ou *Pull Down*

Efeito *Bouncing*

*Debouncing* por SW ou HW

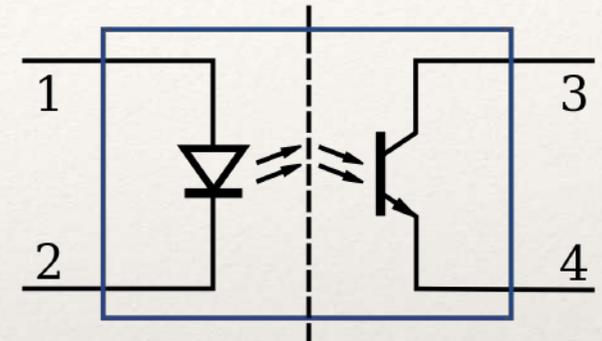


# Dispositivos de Entrada

## Chaves e Botões

Método básico de entrada;

Chaves momentâneas, fixas e encoders;



## Foto-isoladores

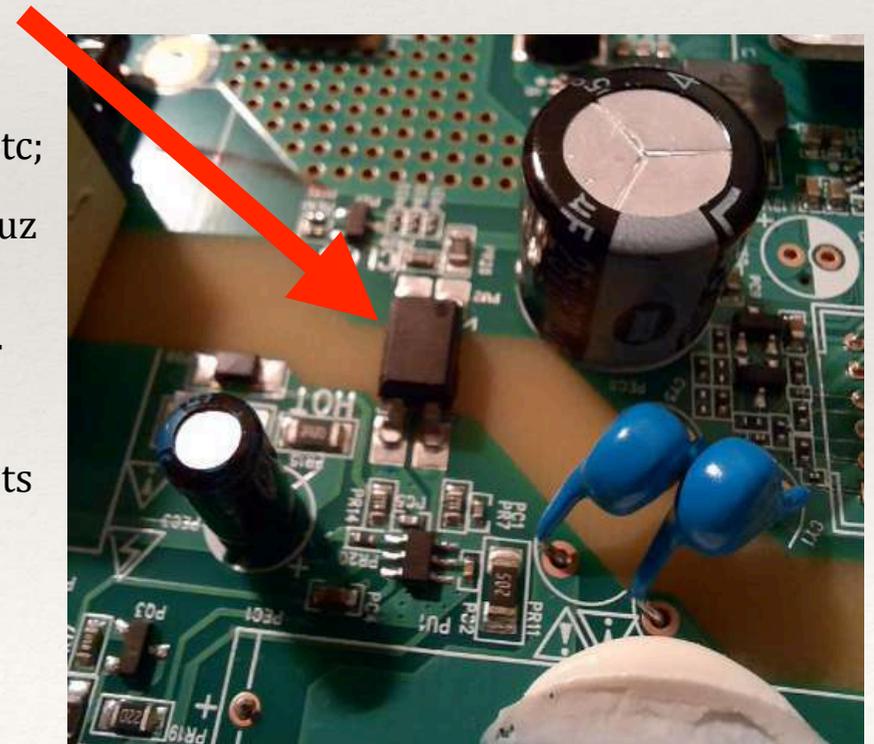
Existem sob a forma de foto-diodos, foto-transistores etc;

No foto-transistor do exemplo, quem excita a base é a luz do LED interno;

Embora tipicamente operem como chave, podem atuar como amplificadores;

Promovem isolamento típico de alguns milhares de volts (5.300V), com resistência de  $10^{11}\Omega$ .

Na placa do exemplo, notem o espaço de isolamento.



# Dispositivos de Entrada

## Chaves e Botões

Método básico de entrada

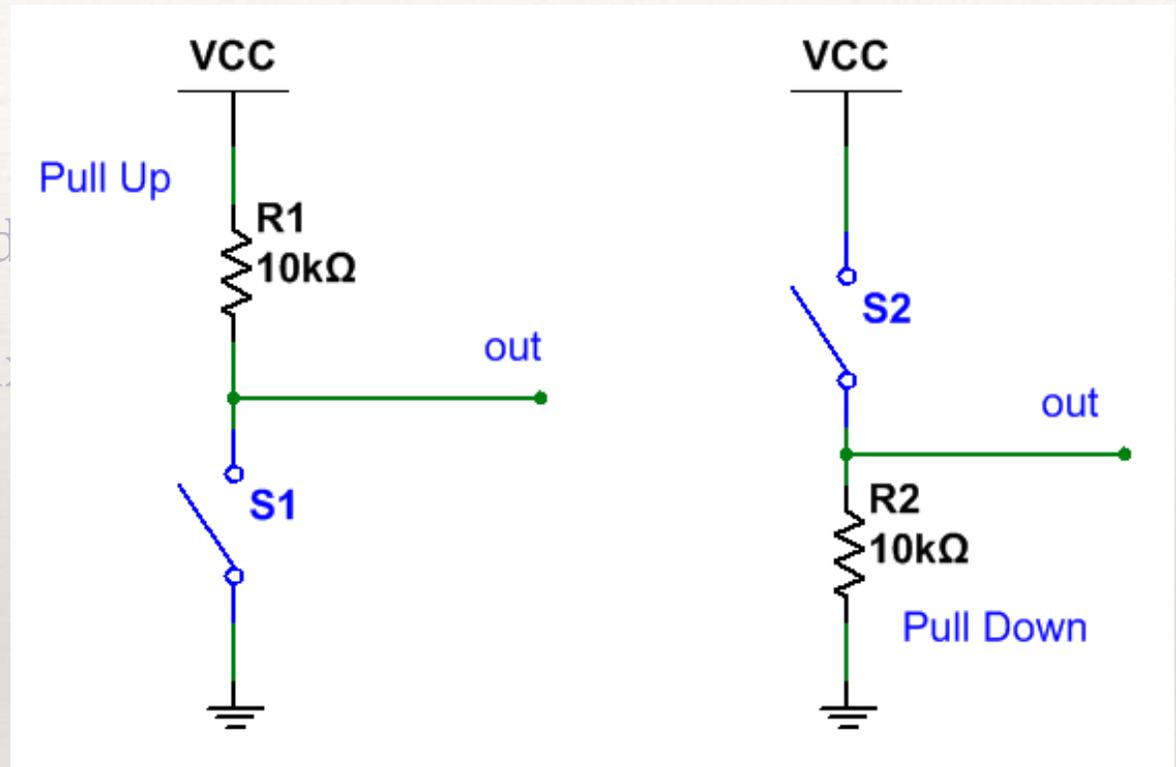
Chaves momentâneas, fixas

Foto-isoladores;

*Pull Up e Pull Down*

Efeito *Bouncing*

*Debouncing* por SW ou HW



# Dispositivos de Entrada

## Chaves e Botões

Método básico de entrada;

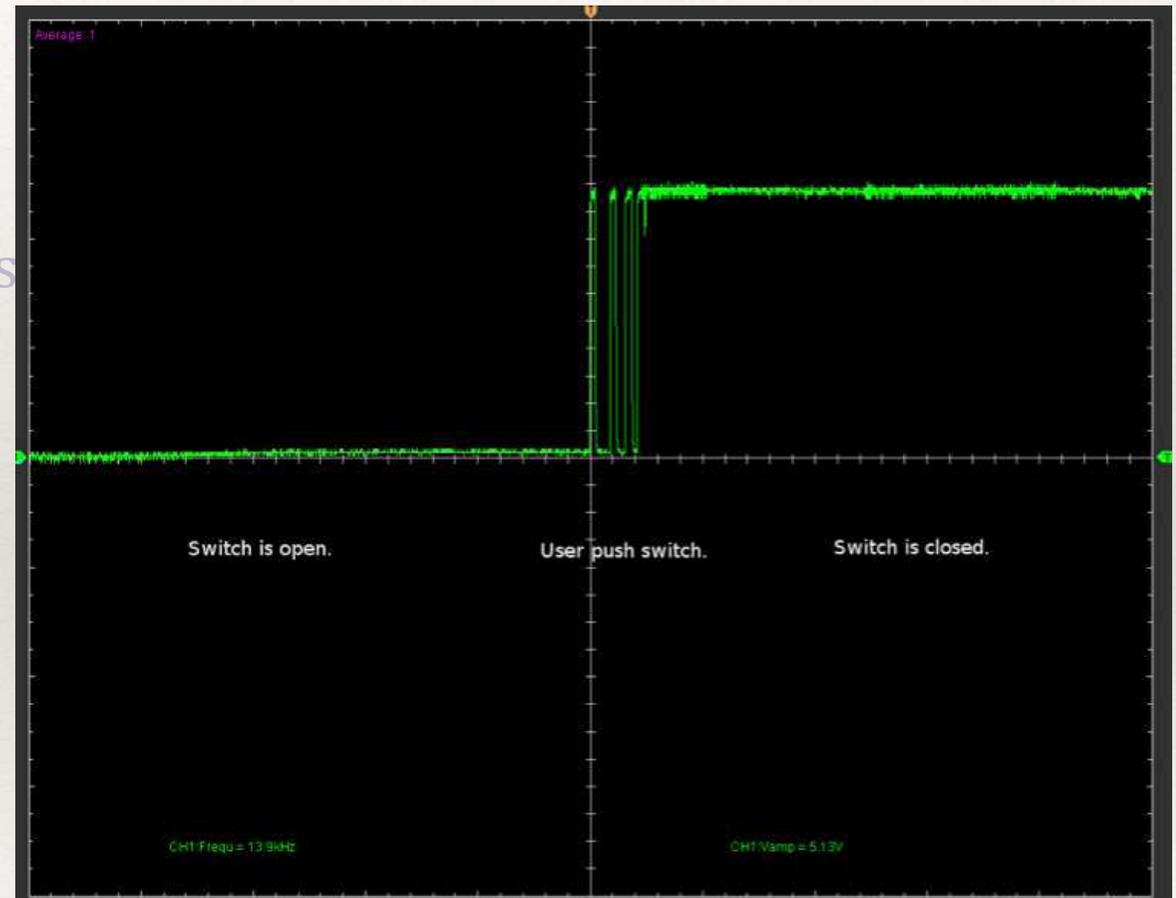
Chaves momentâneas, fixas

Foto-isoladores;

*Pull Up ou Pull Down*

Efeito *Bouncing*

*Debouncing: HW ou SW*



```
// Constantes para indicar número dos pinos utilizados
const int buttonPin = 2;
const int ledPin = 13;

// Variables will change:
int ledState = HIGH;      // status atual do pino de saída (LED)
int buttonState;         // status atual do pino de entrada
int lastButtonState = LOW; // status anterior do pino de entrada

unsigned long lastDebounceTime = 0; // ultimo momento em que o pino mudou de status
unsigned long debounceDelay = 50;  // tempo para o "debounce"

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);

  digitalWrite(ledPin, ledState); // Define status atual da saída
}
```

```
void loop() {
  // Lê o status atual do botão de entrada
  int reading = digitalRead(buttonPin);

  // Se o status do botão mudou, seja por ação do operador, seja por ruído, comece a contar o tempo
  if (reading != lastButtonState) lastDebounceTime = millis();

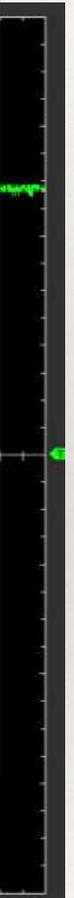
  // Se decorreu o tempo definido para o "debounce"
  if ((millis() - lastDebounceTime) > debounceDelay) {

    // Se o status mudou
    if (reading != buttonState) {
      buttonState = reading;

      // Altere o status do LED apenas se o status for HIGH
      if (buttonState == HIGH) {
        ledState = !ledState;
      }
    }
  }

  // Defina o status do LED
  digitalWrite(ledPin, ledState);

  // Salva a leitura para a próxima execução do loop
  lastButtonState = reading;
}
```



---

# Dispositivos de Entrada

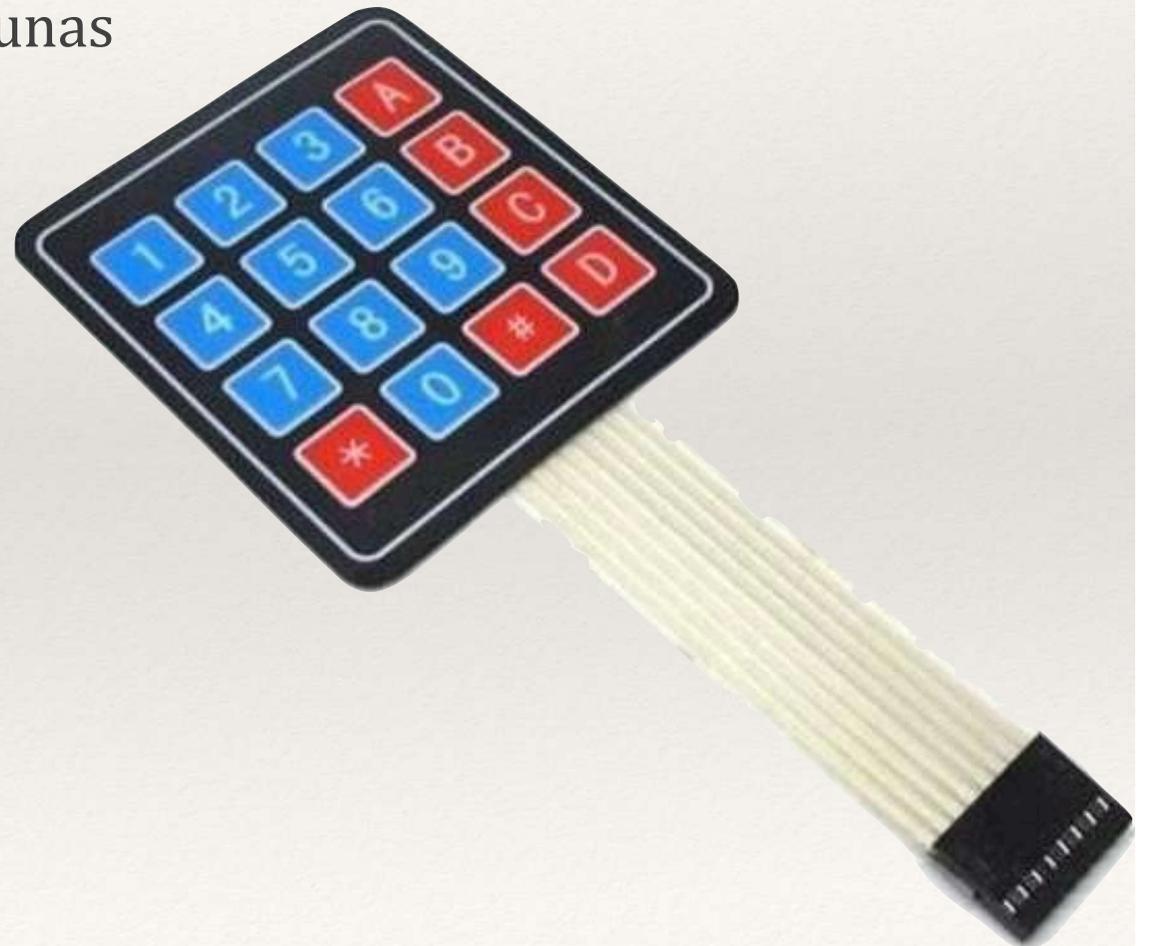
---

Teclados (ex. teclado de membrana)

Cruzamento de Linhas X Colunas

Varredura via SW

Analisar combinações?



---

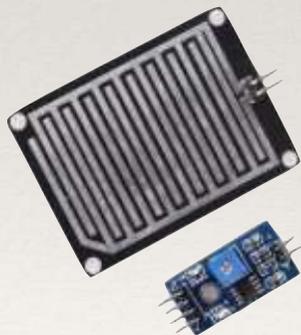
# Dispositivos de Entrada

---

**Sensores digitais** (chuva, presença e movimento, chave magnética, gases, leitura de digitais etc)

Funcionam como chaves (sem *bouncing*);

Alguns permitem ajuste da sensibilidade para ativação;





# Sensores "medem" tensão elétrica

Tensão elétrica é injetada em uma porta analógica do microcontrolador

Tensão elétrica precisa estar dentro dos limites do microcontrolador;

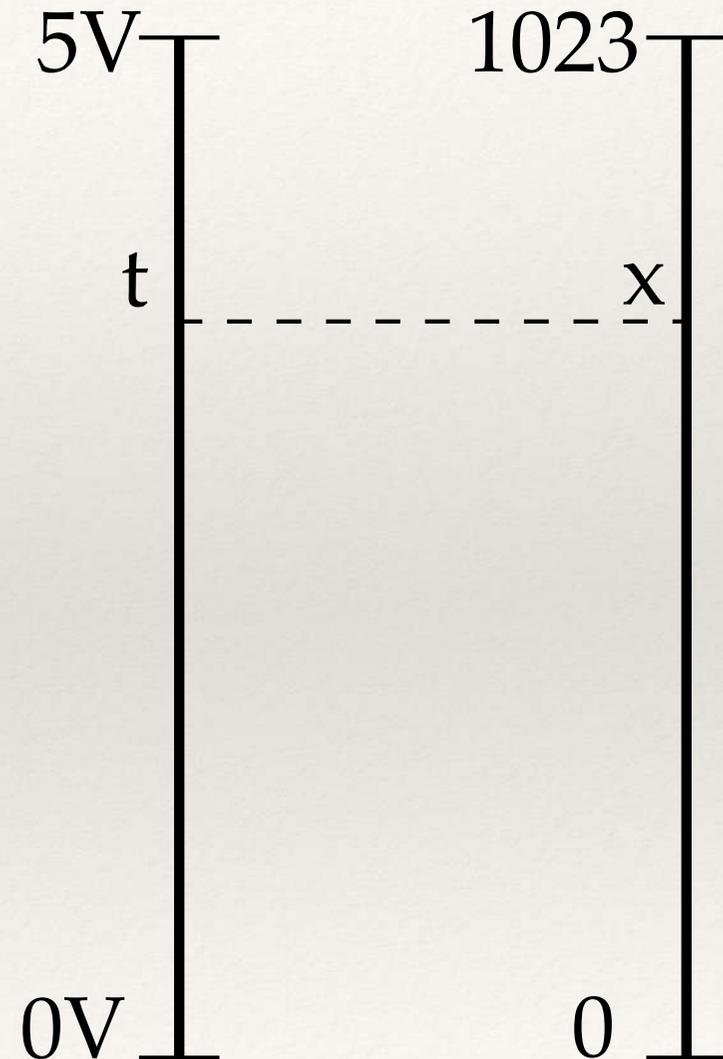
No Arduíno, pode estar entre 0 e 5V.

Na porta analógica, a tensão é convertida para um número inteiro pelo conversor analógico/digital interno;

Número é determinado pela quantidade de bits do conversor analógico digital

No Arduíno, o conversor é de 10bits

$$2^{10} = 1024$$



# Tensão é convertida para um número

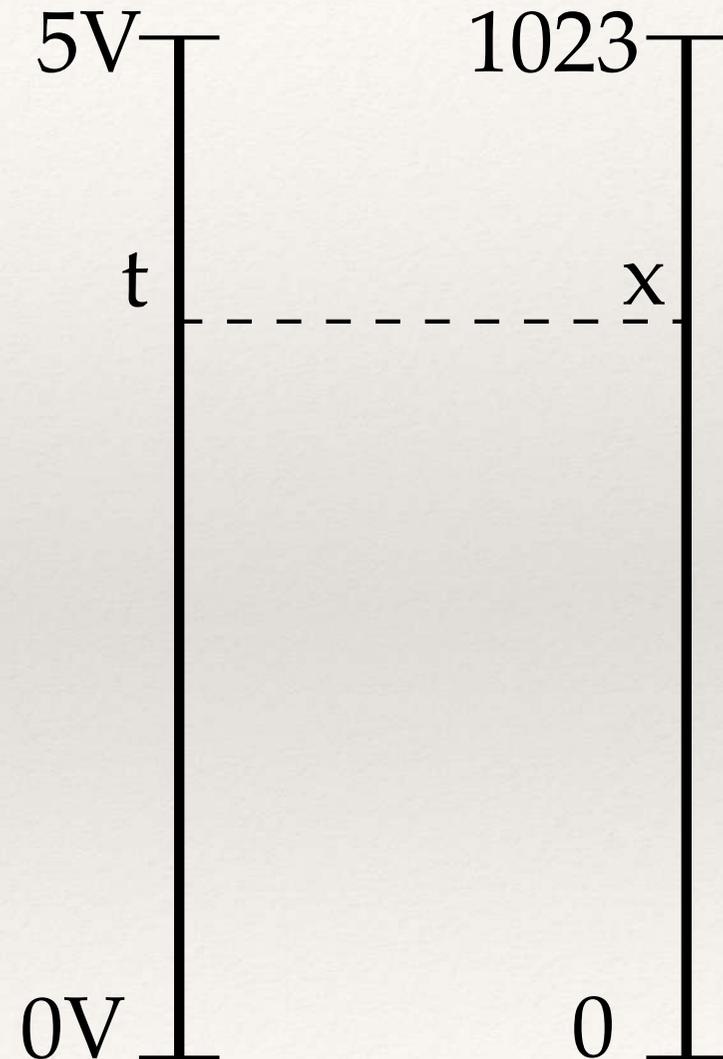
Medições são proporcionais graças à linearidade\* do sensor;

Conversão é básica, por regra de 3

$$\frac{t - 0}{5 - 0} = \frac{x - 0}{1023 - 0}$$

$$\frac{t}{5} = \frac{x}{1023}$$

$$t = \frac{5 \cdot x}{1023}$$



\* Veremos depois

# Medição analógica básica

## Medição de Tensões Disponíveis no próprio Arduino

Sem componentes externos;

Avaliar as medições com base no valor numérico medido pelo ADC;

Alimentação  
via USB

Entrada a ser medida →



# Medição analógica básica

Os resultados serão vistos no Monitor Serial  
Utilizaremos a entrada A0 do Arduíno

Salvamos o valor lido anteriormente

Leitura analógica da entrada

Mostra apenas se houver alteração

Número de amostragens < 10 por segundo

```
void setup() {  
  Serial.begin(115200);  
  pinMode(A0, INPUT);  
}  
  
int entrada, entradaAnterior;  
  
void loop() {  
  entrada = analogRead(A0);  
  
  if( entrada != entradaAnterior ) {  
    Serial.println( entrada );  
    entradaAnterior = entrada;  
  }  
  
  delay(100);  
}
```

# Medição analógica básica

Circuito básico, com apenas um Arduino e um cabo

Simplicidade, sem equip. externos.

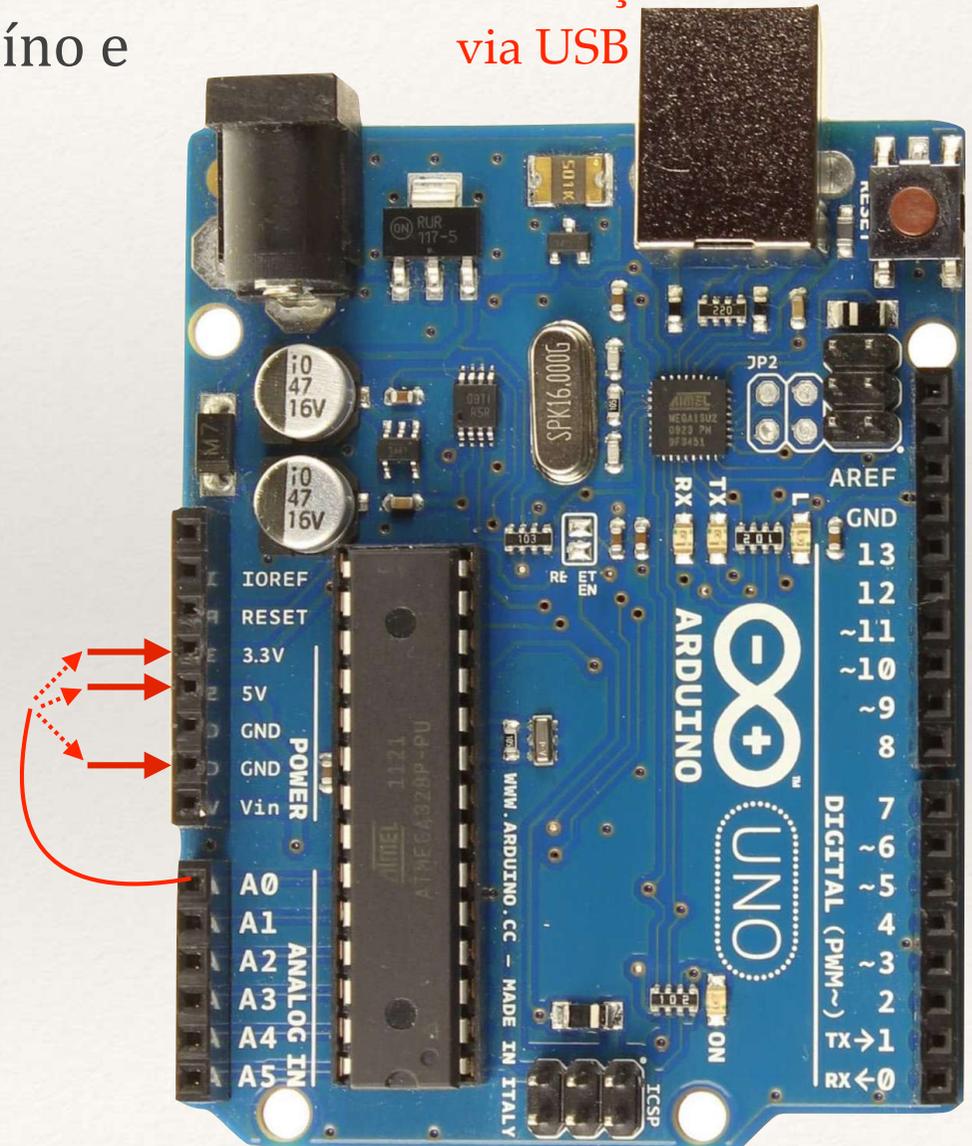
Escolha da tensão de entrada:

5V (Vcc)

3,3V (saída de alimentação)

0V (GND)

Alimentação  
via USB



# Medição analógica básica

Circuito básico, com apenas um Arduino e um cabo

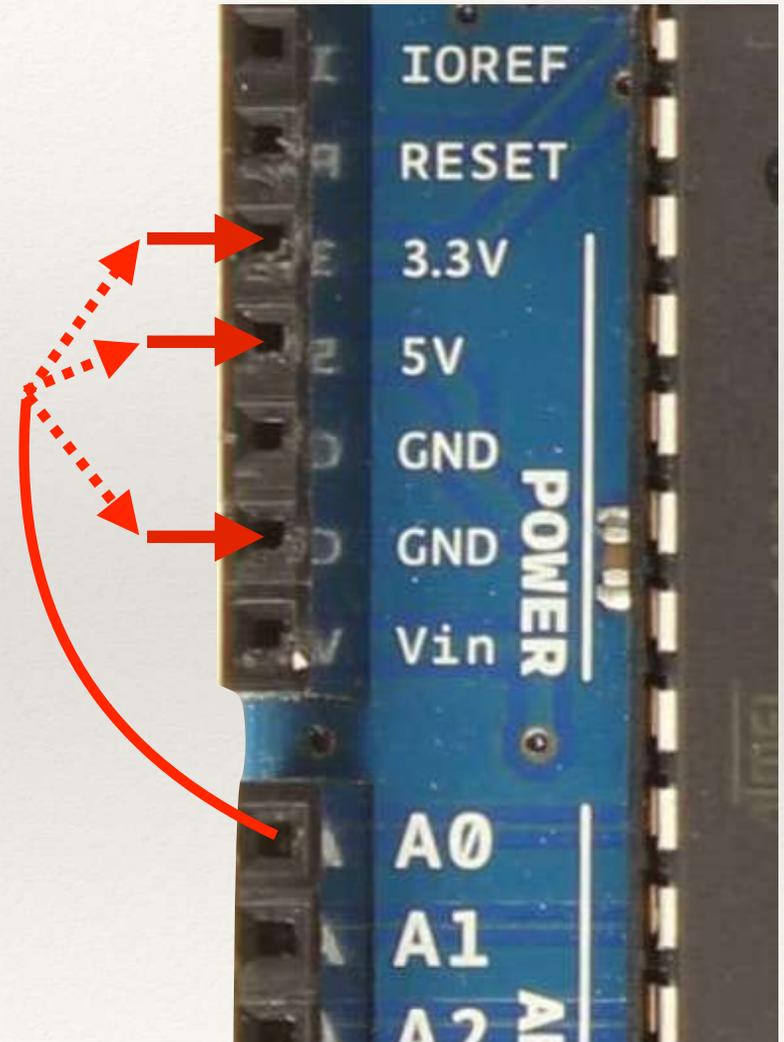
Simplicidade, sem equip. externos.

Escolha da tensão de entrada:

5V (Vcc)

3,3V (saída de alimentação)

0V (GND)



# Medição analógica básica

```
402
398
405
395
404
396
406
389
386
390
388
395
393
127
179
343
```

Auto-rolagem  Show timestamp

## O que percebemos?

Entradas desconectadas são “dominadas” pelo ruído;

Há alguma instabilidade quando medimos a tensão de 3,3V.

## Especificação básica

Conversor A/D de 10 bits (0~1023).

## Cuidado com a conexão !

Interligar portas erradas pode danificar o Arduíno, e até a porta USB do seu computador !

---

# Instabilidade

---

A instabilidade é um sintoma da falta de precisão;

Mas o que é “precisão” ?

A precisão, exatidão e a resolução, além da linearidade, são parâmetros importantes, e serão posteriormente;

A imprecisão costuma afetar mais os valores intermediários de uma escala (no caso do experimento, seus efeitos apareceram mais na medição dos 3,3V).

# Medição analógica típica

Medição de Tensão Externa 0~5V;

Sem ajuste de escala;

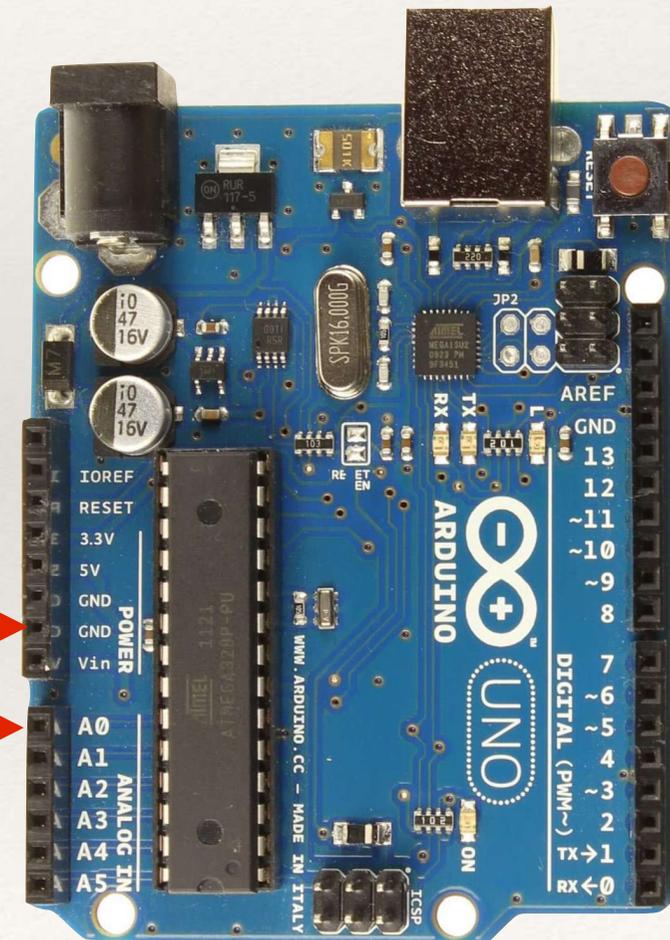
Referência: Vcc (5V);

Medição numérica e de tensão;

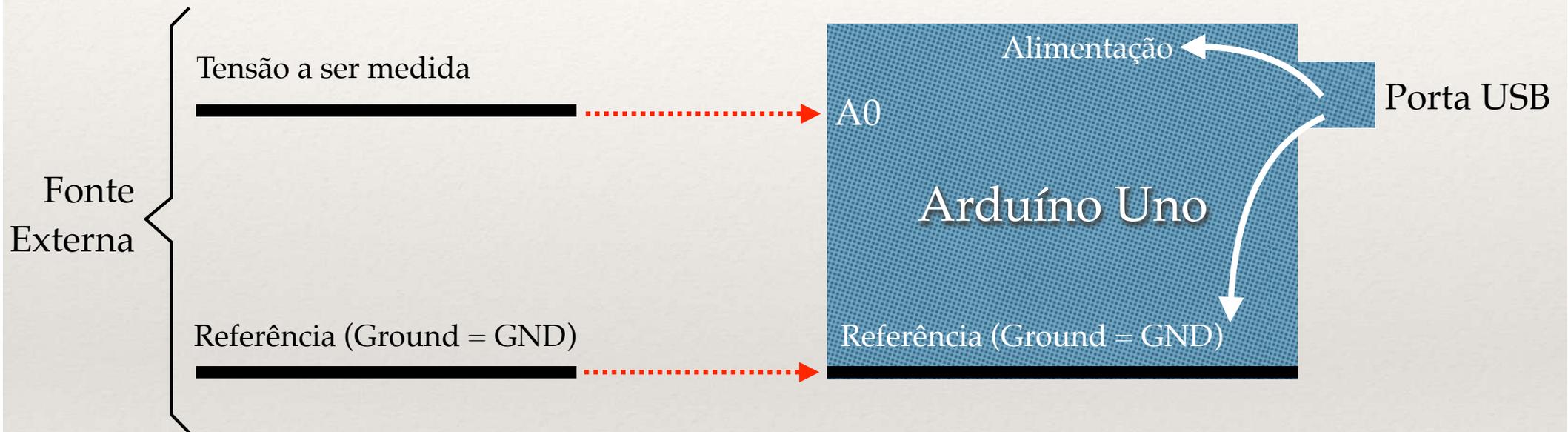
Apenas o Arduíno, sem componentes externos.

Alimentação  
via USB

GND de referência →  
Entrada a ser medida →



# Medição analógica típica



# Medição analógica típica

O valor da tensão não é inteiro

Cálculo da tensão

(usamos a casa decimal para garantir o processamento das constantes como "float")

O valor da tensão terá 2 casas decimais

```
void setup() {  
  
  Serial.begin(115200);  
  pinMode(A0, INPUT);  
  
}  
  
int entrada, entAnt;  
float tensao;  
  
void loop() {  
  
  entrada = analogRead(A0);  
  tensao = 5.0 * entrada / 1023.0;  
  
  if( entrada != entAnt ) {  
    Serial.print( entrada );  
    Serial.print( ": " );  
    Serial.print( tensao, 2);  
    Serial.println("V");  
    entAnt = entrada;  
  }  
  
  delay(100);  
  
}
```

# Medição analógica típica

O que percebemos?

A exatidão é razoável;

As tensões foram representadas com 2 dígitos decimais;

Percebe-se alguma instabilidade nos valores medidos, exceto para 0 e 5V.

Cuidado importante !

Não há qualquer proteção da entrada, logo tensões negativas, ou superiores a 5V podem danificar o Arduíno !

---

# Casas Decimais

---

O número de casas decimais não é só uma “escolha”; ele depende diretamente da resolução;

Outro parâmetro também importante é o “número de contagens”;

Vamos ver estes parâmetros ?

---

# Precisão e Exatidão

---

A informação é instável e imprecisa

Nota-se também um erro de medição, mesmo que façamos uma média;

Vamos entender cada um dos parâmetros?



Pouco preciso,  
pouco exato

---

# Precisão e Exatidão

---



Muito preciso,  
pouco exato



Pouco preciso,  
pouco exato

Precisão não tem relação  
com a obtenção do valor  
exato

Se o valor medido for fixo, medições  
precisas retornarão valores sempre  
muito próximos;

Ou seja, para valores fixos, a medição  
será estável, mesmo que errada.

---

# Precisão e Exatidão

---

## Exatidão não tem relação com precisão (ou estabilidade)

Muitas vezes medições exatas perdem precisão devido à presença de erros aleatórios (ruído branco, ou aleatório);

Nestes casos, a média dos valores medidos é muito próxima do valor real.



Pouco preciso,  
pouco exato



Pouco preciso,  
muito exato

---

# Precisão e Exatidão

---

## Exatidão não tem relação com precisão (ou estabilidade)

Muitas vezes medições exatas perdem precisão devido à presença de erros aleatórios (ruído branco, ou aleatório);

Nestes casos, a média dos valores medidos é muito próxima do valor real.



Pouco preciso,  
pouco exato



Pouco preciso,  
muito exato

---

# Precisão e Exatidão

---

## Exatidão não tem relação com precisão (ou estabilidade)

Muitas vezes medições exatas perdem precisão devido à presença de erros aleatórios (ruído branco, ou aleatório);

Nestes casos, a média dos valores medidos é muito próxima do valor real.



Pouco preciso,  
pouco exato



Muito preciso  
muito exato

# Precisão

Se a causa for um ruído com características aleatórias

Cálculo da média pode resolver;

Solução Típica: média simples;

Número de amostragens depende do desvio padrão

A1 A2 A3 A4 A5 A6 A7 A8 A9 A10...

M1

M2

Médias móveis normalmente são melhores devido à performance

A1 A2 A3 A4 A5 A6 A7 A8 A9 A10

M7

# Exatidão

Normalmente a inexatidão é um problema mais grave

A informação correta não está disponível;

Se houver precisão, no entanto, a comparação com outra medição mais exata pode determinar um padrão para correção do erro;

A exatidão é um percentual

Ex.:  $\pm (2\% + 2)$  significa 2% de exatidão + 2. Se estiver lendo 10,0, o valor pode estar entre 9,78 e 10,22.

# Linearidade

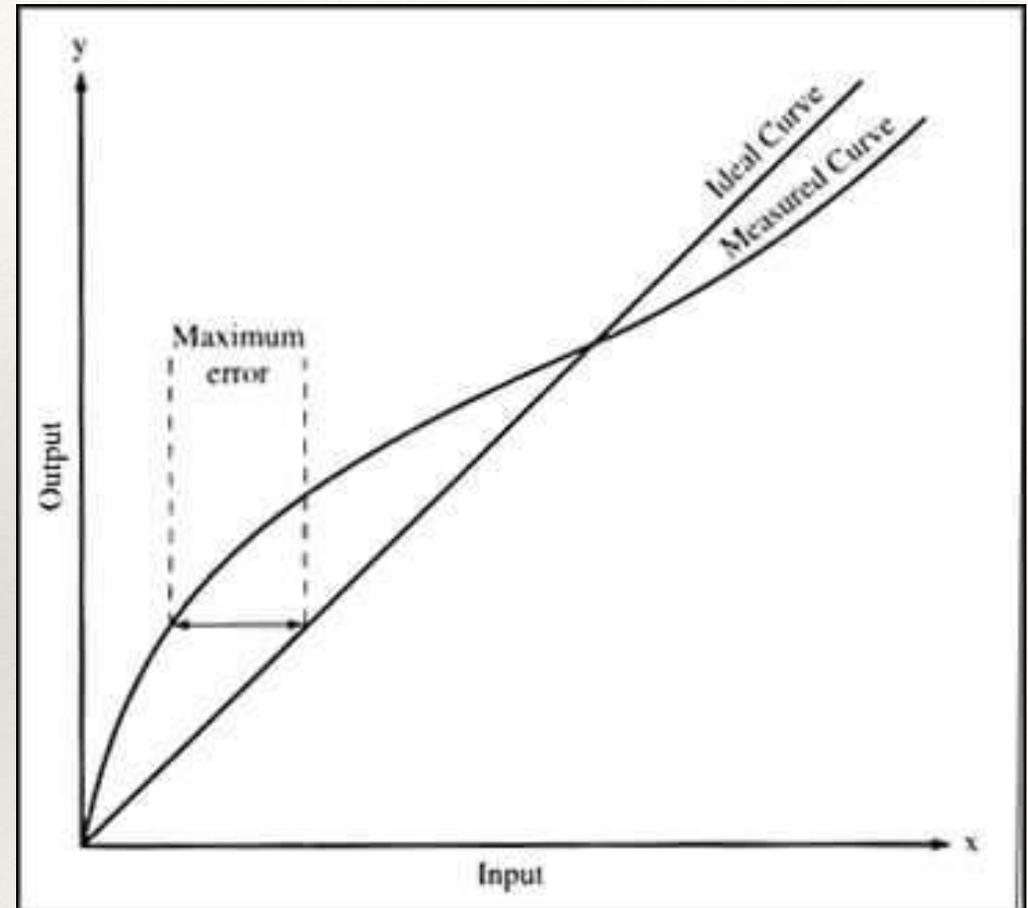
O comportamento da leitura nem sempre corresponde ao comportamento real do valor medido

A não-linearidade provoca erros de medição em parte dos pontos avaliados;

É comum encontrar a não-linearidade nos extremos da faixa de valores mensuráveis;

Em algumas situações, pode valer a pena limitar a faixa de valores medidos, e com isso a resolução;

Em casos específicos, pode-se corrigir a não-linearidade matematicamente.



---

# Precisão, Exatidão e Linearidade

---

Nenhum destes parâmetros tem relação obrigatória com medições digitais !

Qualquer tipo de medida, inclusive com instrumentos analógicos ou manuais, é afetada pela precisão, exatidão e linearidade;

No entanto, medições digitais em sistemas automatizados também precisam ser avaliadas nestes aspectos.

## Fatores Universais

Referência

Ruídos

Escala

## Fatores em Medições Digitais

Resolução

Taxa de Amostragem

---

# Referência

---

Qualquer medição baseia-se em uma referência;

Microcontroladores e sensores podem usar tipicamente uma das seguintes fontes de referência:

- Tensão de Alimentação;

- Tensão de Referência Interna;

- Tensão de Referência Externa.

Referências possuem exatidão e precisão pré-definidas

- Quanto melhor a referência, melhor a medição (e maior o custo);

- Fatores externos, como a temperatura, muitas vezes afetam as referências.

---

# Fontes de Referência

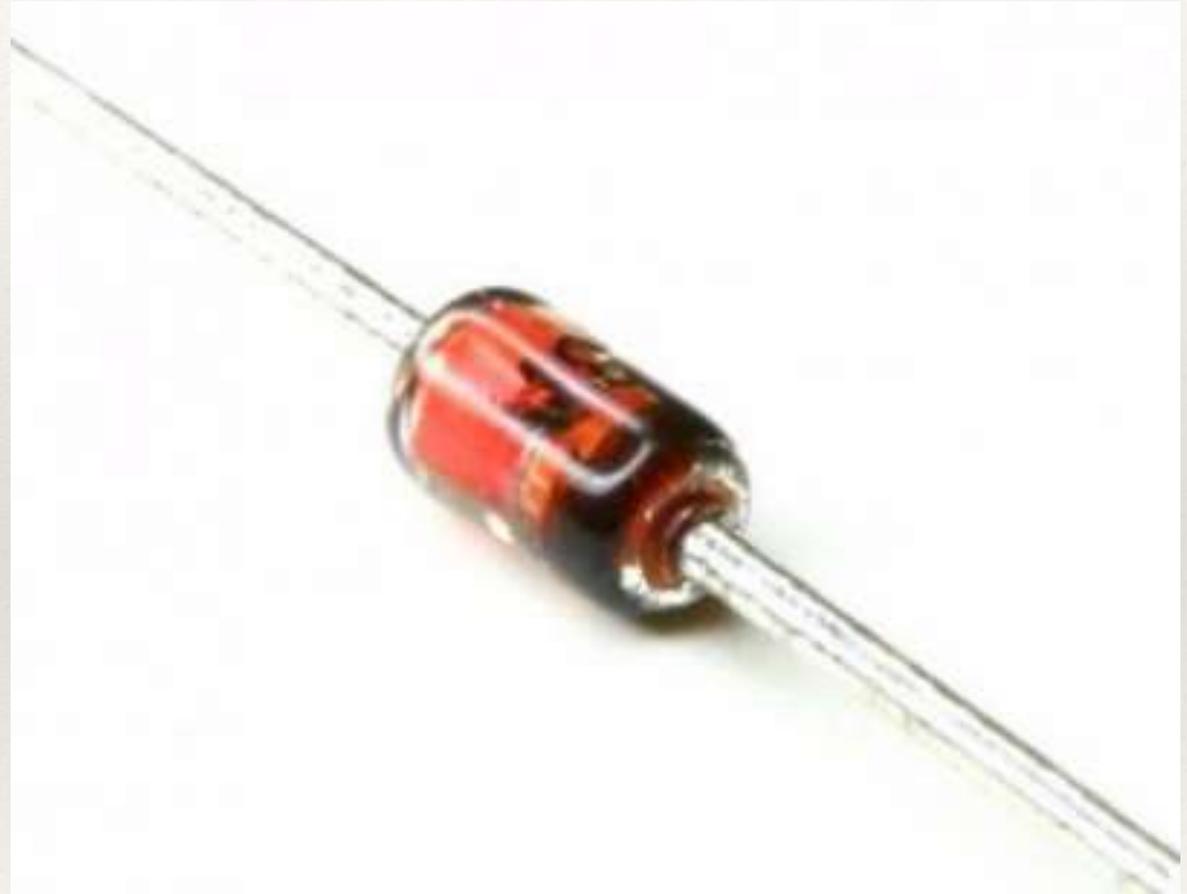
---

## Diodo Zener

Tensões diversas;

Tomar cuidado com a  
linearidade;

Afetado pela temperatura  
(700ppm/°C)



---

# Fontes de Referência

---

Referências Programáveis de Precisão

Ex.: TL431 (Texas)

Tensão ajustável de 2,5 a 36V;

0,5% de Exatidão;

Afetado pela temperatura  
(600ppm/°C).



---

# Fontes de Referência

---

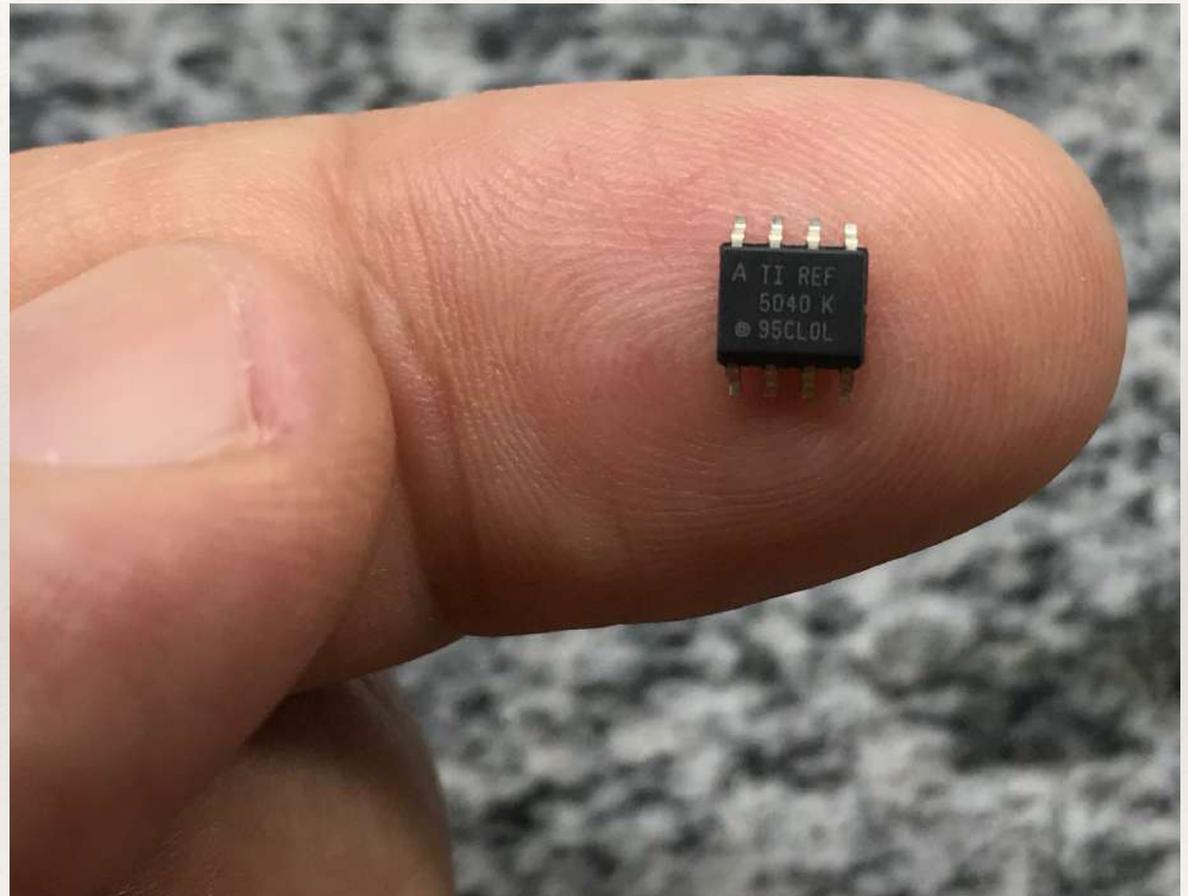
Chips especializados

Ex.: 5040AIDG4 (Texas)

4,096 Volts

0,05% de exatidão;

3 ppm/°C

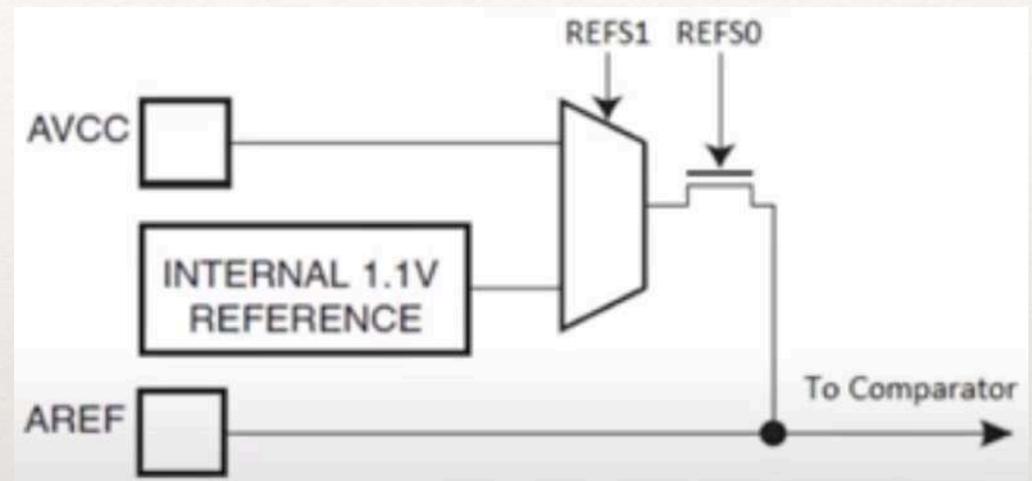


# Ex: Referência no Arduino

Por padrão, o Arduino utiliza a tensão de alimentação (5V) como referência;

Outras referências exigem *software* e até *hardware* específico;

Após mudança, as primeiras medições podem ser imprecisas.



**analogReference( DEFAULT )** : utiliza Vcc, e varia com ela.

**analogReference( INTERNAL )** : utiliza referência interna de 1,1V

Medir valor efetivo (medir pino AREF durante operação normal).

**analogReference( EXTERNAL )** : utiliza tensão de entrada no pino AREF

0V < Referência <= 5V (a qualidade da referência determinará a exatidão das medidas);

Não usar analogRead() antes de mudar a referência, pois isso pode danificar o Arduino.

# Tratando o ruído

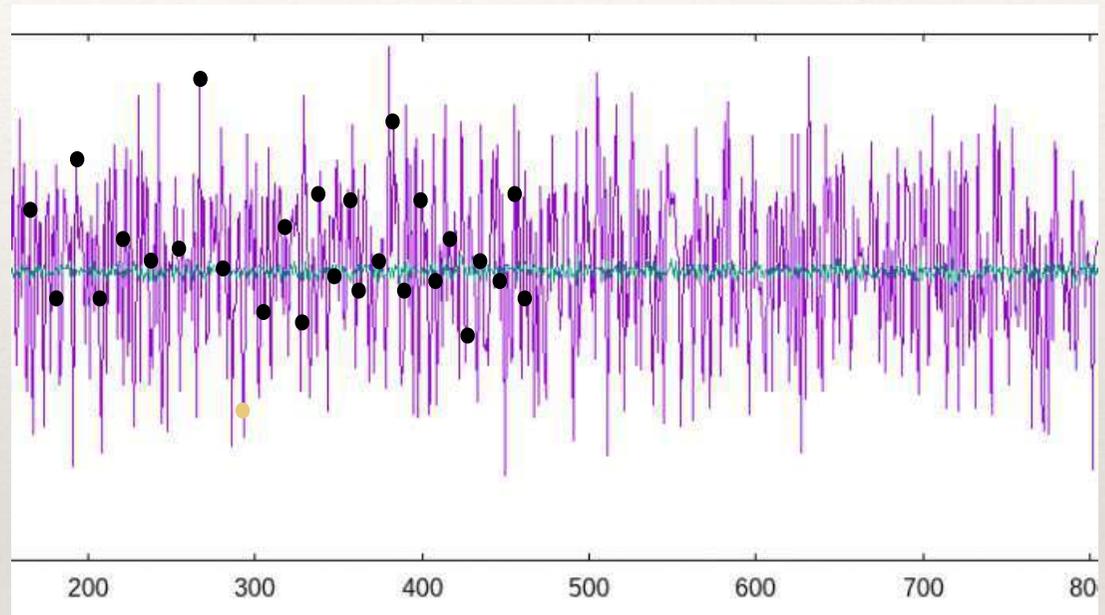
A melhor opção é buscar a redução do ruído no sensor, circuito ou meio ambiente

Qualidade do sensor e componentes;

Proximidade de Fontes de ruído;

Conexão e cabos;

Alimentação elétrica;



Ruídos tipicamente afetam a precisão

Quando aleatórios, ruídos podem ser reduzidos por algoritmos de média;

Se o desvio padrão for elevado, o número de amostragens necessárias pode ser grande.

# Escalas

## Adequam a leitura à faixa de valores das entradas

Devem ser dimensionadas para manter a resolução máxima;

Tipicamente usam Divisores de Tensão;

Avaliar impedâncias;

Cuidado com a qualidade dos componentes.

## Outros "detalhes"

Unificar o "zero volts";

Medições diferenciais;

Algumas escalas não começam em zero;

Medição de valores negativos;

Proteção de entradas.



# Escalas

## Adequam a leitura à faixa de valores das entradas

Devem ser dimensionadas para manter a resolução máxima;

Tipicamente usam Divisores de Tensão;

Avaliar impedâncias;

Cuidado com a qualidade dos componentes.

## Outros "detalhes"

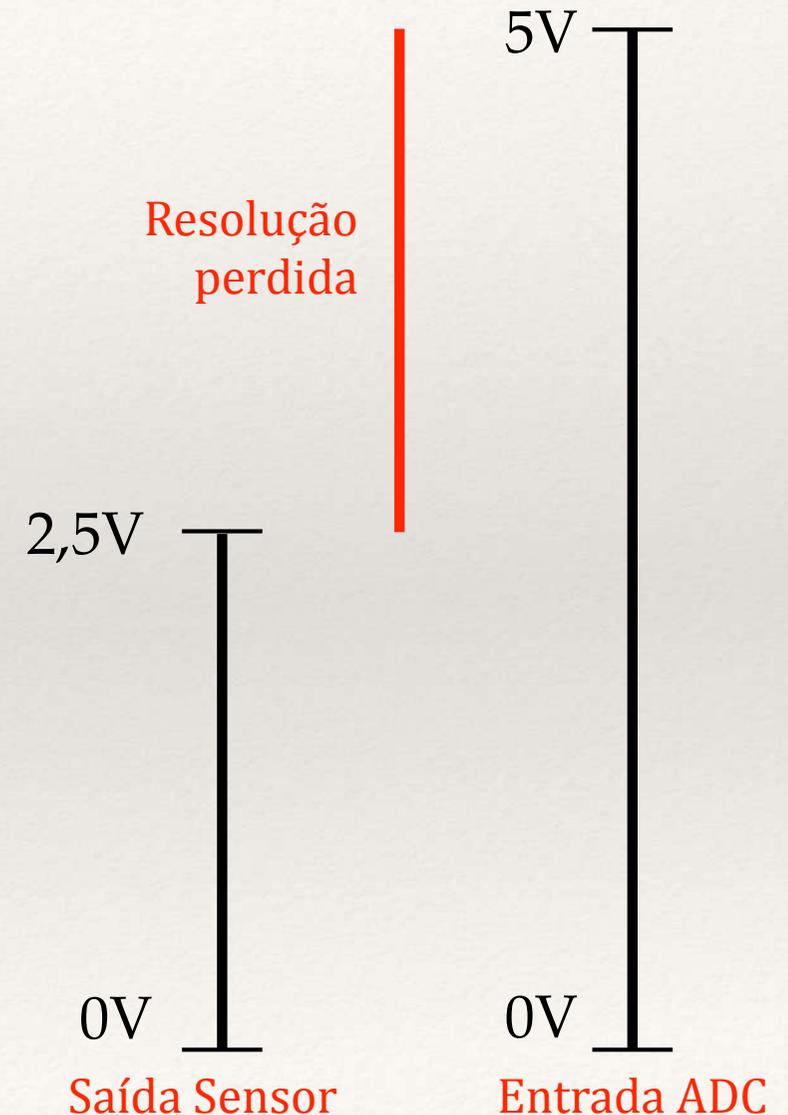
Unificar o "zero volts";

Medições diferenciais;

Algumas escalas não começam em zero;

Medição de valores negativos;

Proteção de entradas.



# Escalas

## Adequam a leitura à faixa de valores das entradas

Devem ser dimensionadas para manter a resolução máxima;

Tipicamente usam Divisores de Tensão;

Avaliar impedâncias;

Cuidado com a qualidade dos componentes.

## Outros "detalhes"

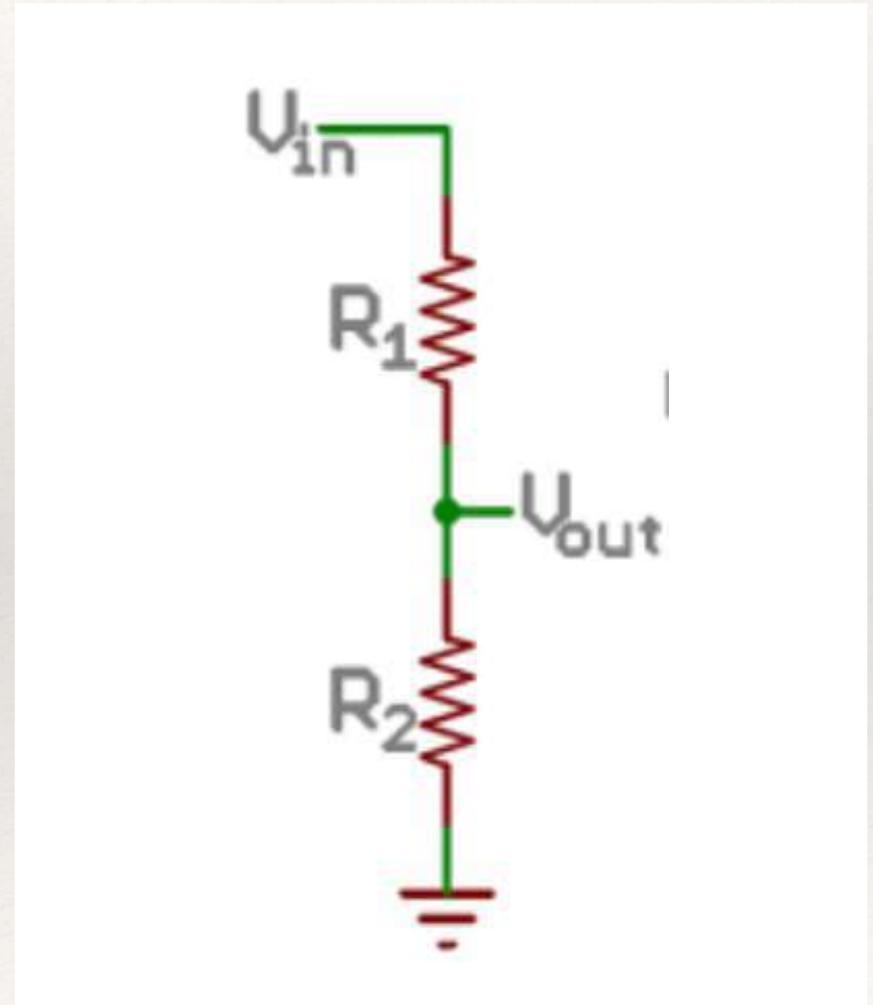
Unificar o "zero volts";

Medições diferenciais;

Algumas escalas não começam em zero;

Medição de valores negativos;

Proteção de entradas.



# Escalas

## Adequam a leitura à faixa de valores das entradas

Devem ser dimensionadas para manter a resolução máxima;

Tipicamente usam Divisores de Tensão;

**Avaliar impedâncias;**

Cuidado com a qualidade dos componentes.

## Outros "detalhes"

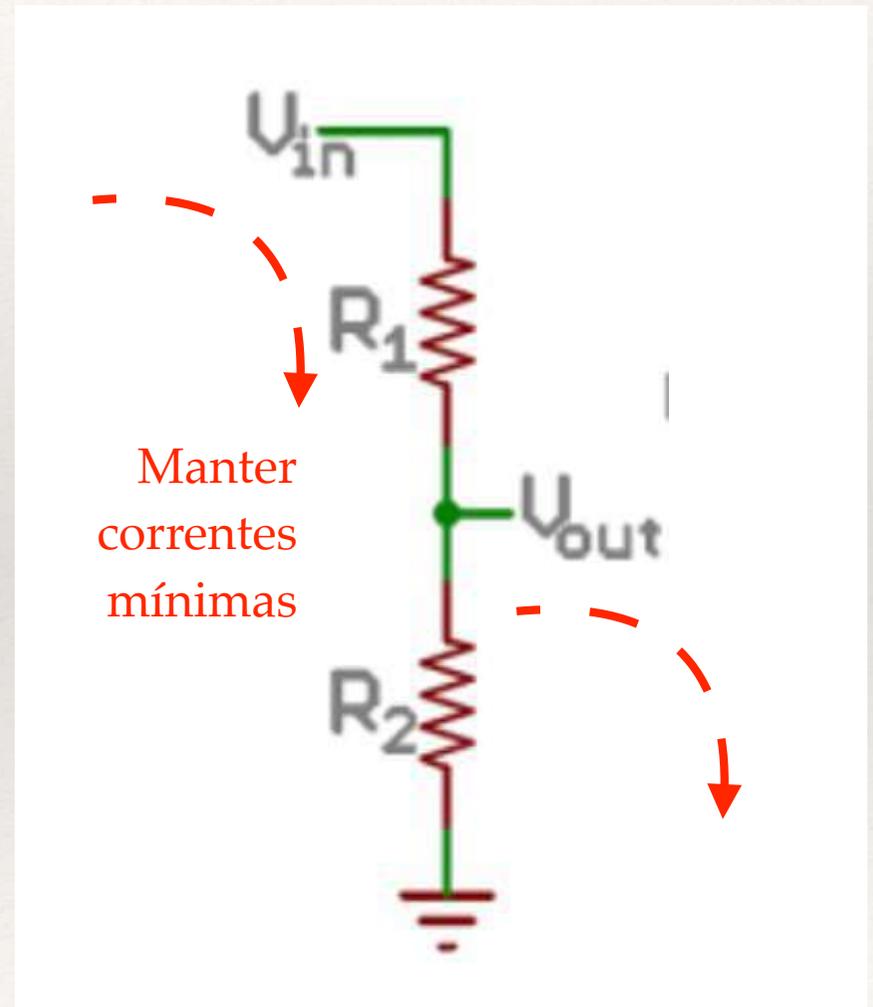
Unificar o "zero volts";

Medições diferenciais;

Algumas escalas não começam em zero;

Medição de valores negativos;

Proteção de entradas.



# Escalas

Adequam a leitura à faixa de valores das entradas

Devem ser dimensionadas para manter a resolução máxima;

Tipicamente usam Divisores de Tensão;

Avaliar impedâncias;

Cuidado com a qualidade dos componentes.

Outros "detalhes"

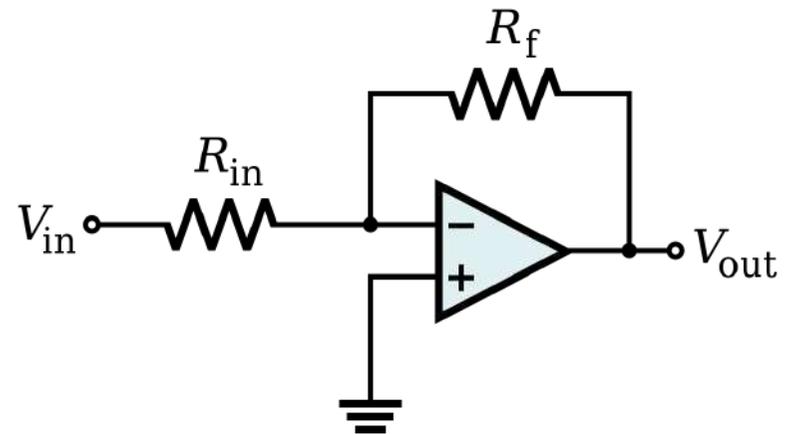
Unificar o "zero volts";

Medições diferenciais;

Algumas escalas não começam em zero;

Medição de valores negativos;

Proteção de entradas.



---

# Escalas

---

Adequam a leitura à faixa de valores das entradas

Devem ser dimensionadas para manter a resolução máxima;

Tipicamente usam Divisores de Tensão;

Avaliar impedâncias;

Cuidado com a qualidade dos componentes.

Outros "detalhes"

Unificar o "zero volts";

Medições diferenciais;

Algumas escalas não começam em zero;

Medição de valores negativos;

Proteção de entradas.

**CHALLENGE ACCEPTED**

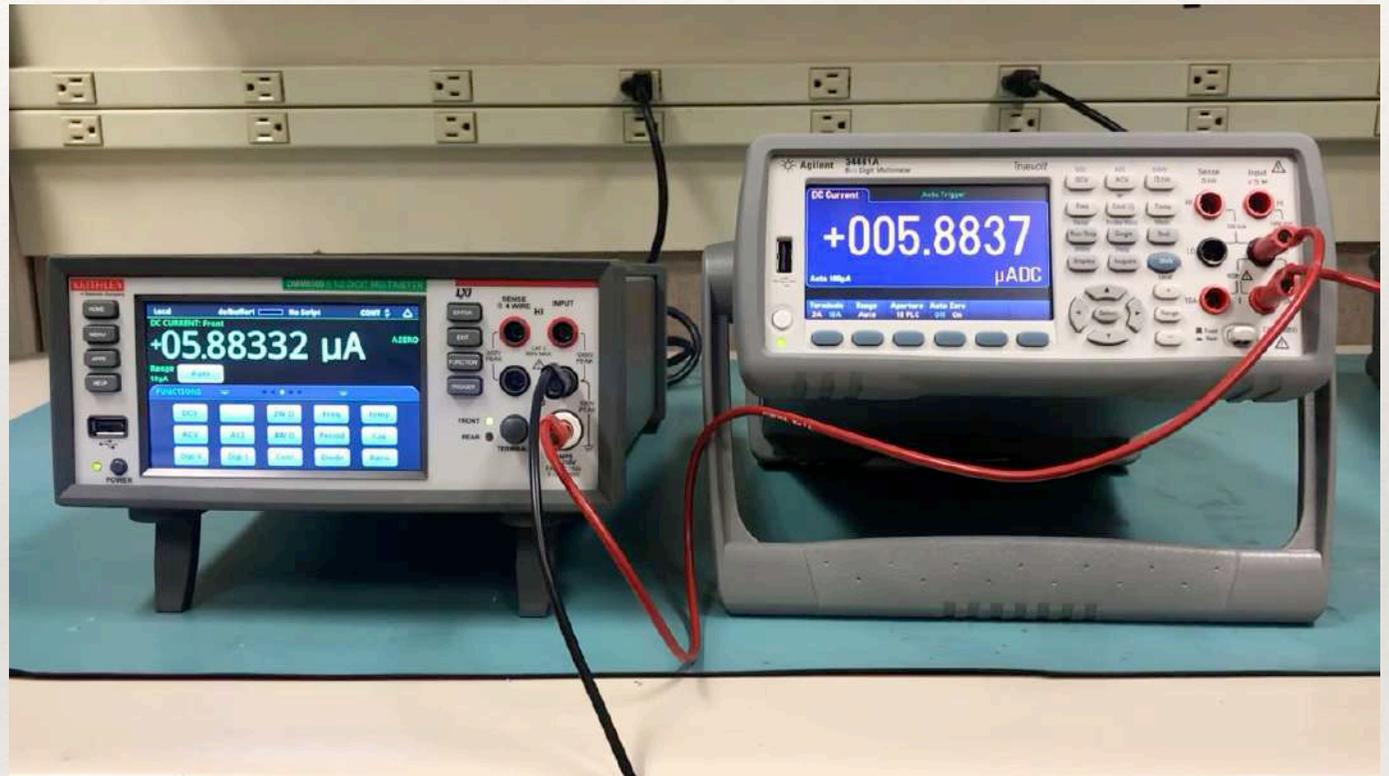


# Resolução

Parâmetro mais usado em medições digitais;

Medida em bits, é a capacidade de identificar valores diferentes, mesmo que próximos;

Outra forma de avaliar é a “contagem”: quantos diferentes valores podem ser representados ?



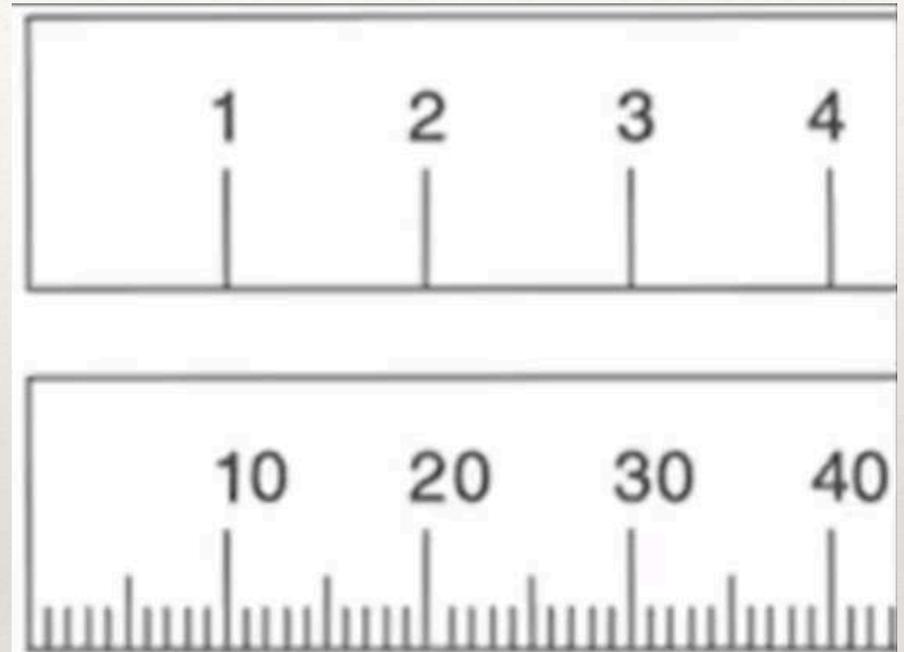
Quem determina a resolução é o ADC (Conversor Analógico / Digital);

Vídeo mostra a “disputa” de dois equipamentos de alta resolução. Notem que, mesmo possuindo a mesma quantidade de dígitos, há diferenças.

# Resolução

Existe resolução em medidas analógicas ?

Em medições digitais, a resolução é definida em bits, já que o sinal de entrada precisa ser convertido para um número digital e binário;

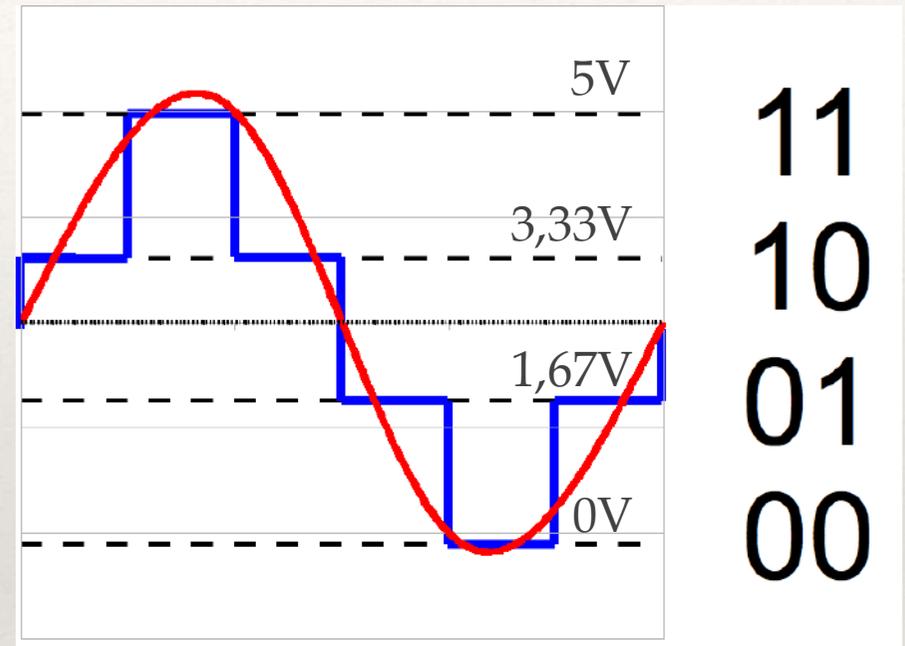


# Resolução

Existe resolução em medidas analógicas ?

Em medições digitais, a resolução é definida em bits, já que o sinal de entrada precisa ser convertido para um número digital e binário;

Resolução de 2 bits: 4 valores diferentes;

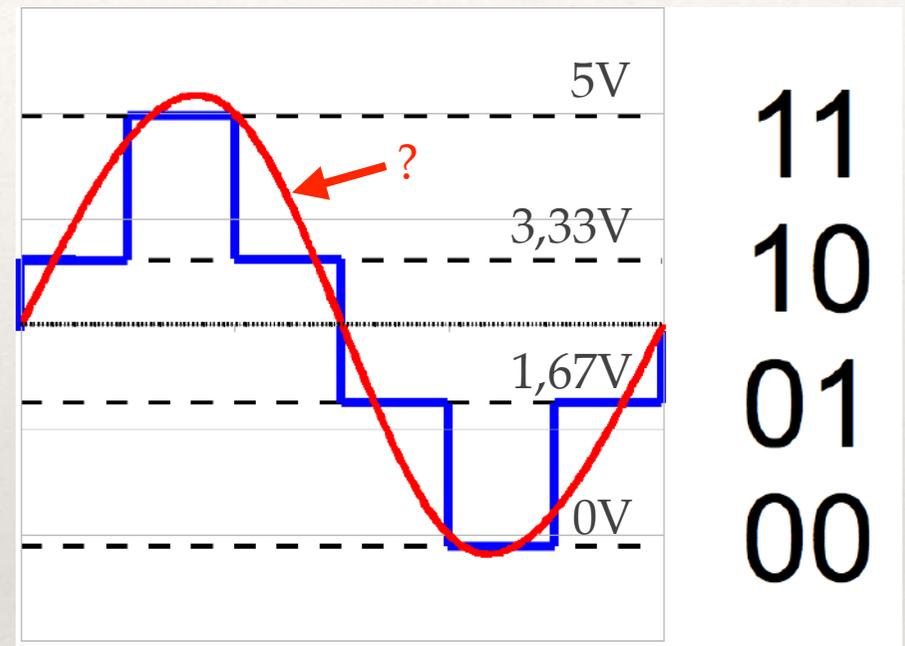


# Resolução

Existe resolução em medidas analógicas ?

Em medições digitais, a resolução é definida em bits, já que o sinal de entrada precisa ser convertido para um número digital e binário;

Resolução de 2 bits: 4 valores diferentes;



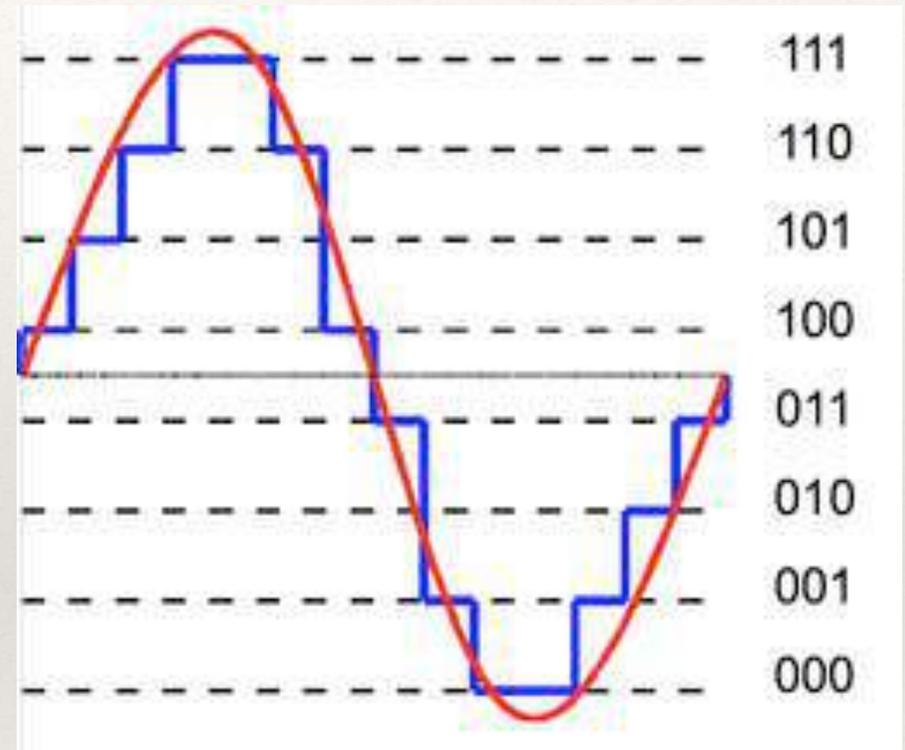
# Resolução

Existe resolução em medidas analógicas ?

Em medições digitais, a resolução é definida em bits, já que o sinal de entrada precisa ser convertido para um número digital e binário;

Resolução de 2 bits: 4 valores diferentes;

Resolução de 3 bits: 8 valores diferentes;



# Resolução

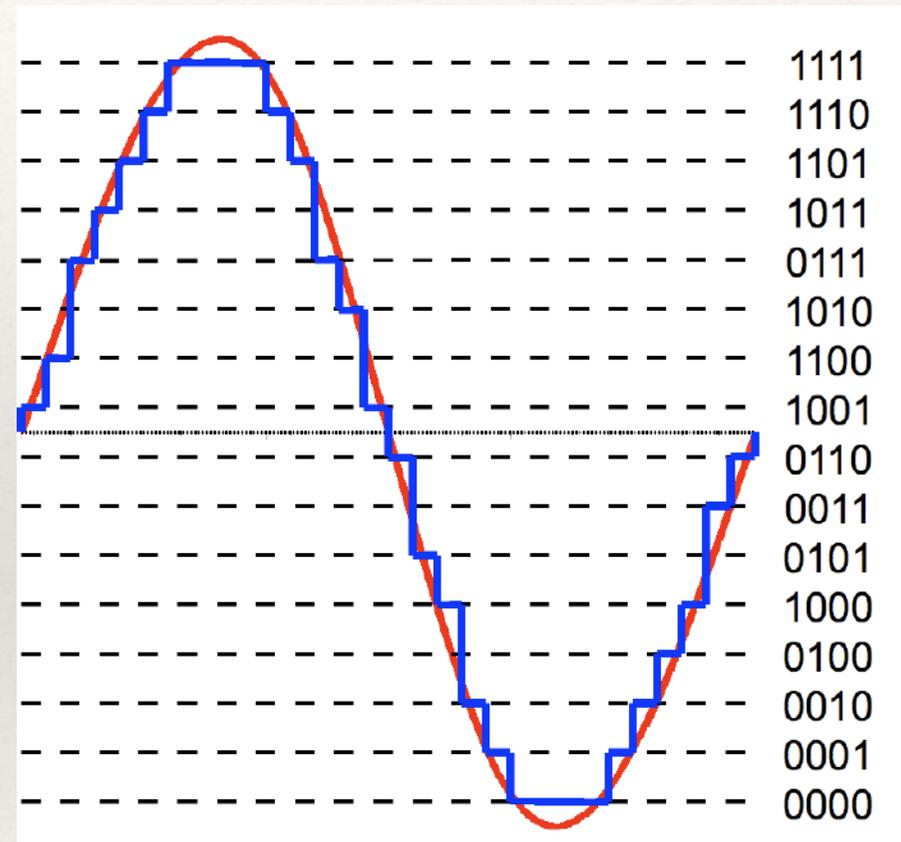
Existe resolução em medidas analógicas ?

Em medições digitais, a resolução é definida em bits, já que o sinal de entrada precisa ser convertido para um número digital e binário;

Resolução de 2 bits: 4 valores diferentes;

Resolução de 3 bits: 8 valores diferentes;

Resolução de 4 bits: 16 valores diferentes;



# Resolução

Existe resolução em medidas analógicas ?

Em medições digitais, a resolução é definida em bits, já que o sinal de entrada precisa ser convertido para um número digital e binário;

Resolução de 2 bits: 4 valores diferentes;

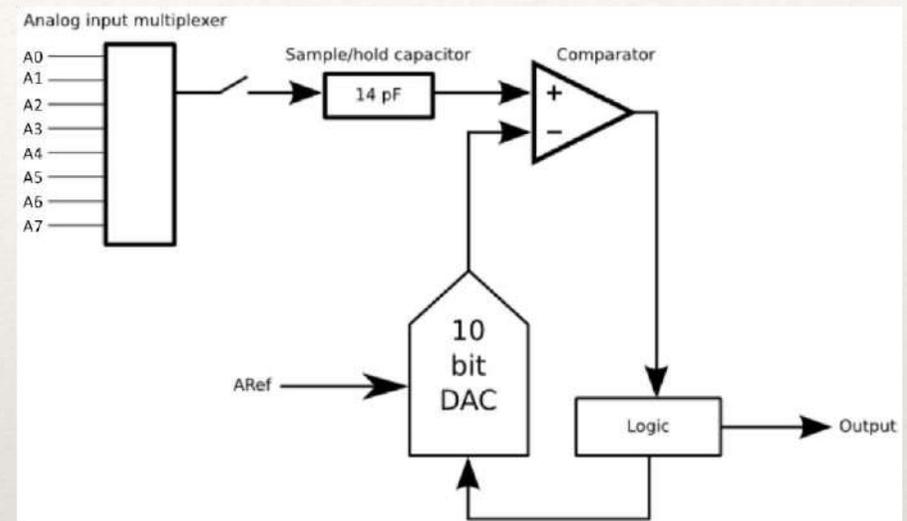
Resolução de 3 bits: 8 valores diferentes;

Resolução de 4 bits: 16 valores diferentes;

Arduíno: 10 bits de resolução no ADC

São 4,89mV entre dois valores (  $5V / 1023$  ). Isso implica em 3 dígitos (0,00 a 5,00V), pois o terceiro dígito (mV) não pode ser usado. Como são 500 valores, temos "500 contagens";

Com a referência de 1,1V, teríamos 1mV. Isso implicaria em 4 dígitos, ou 1100 "contagens".



# Resolução - como calcular

E se, dados os limites de uma medição, eu precisar calcular a resolução?

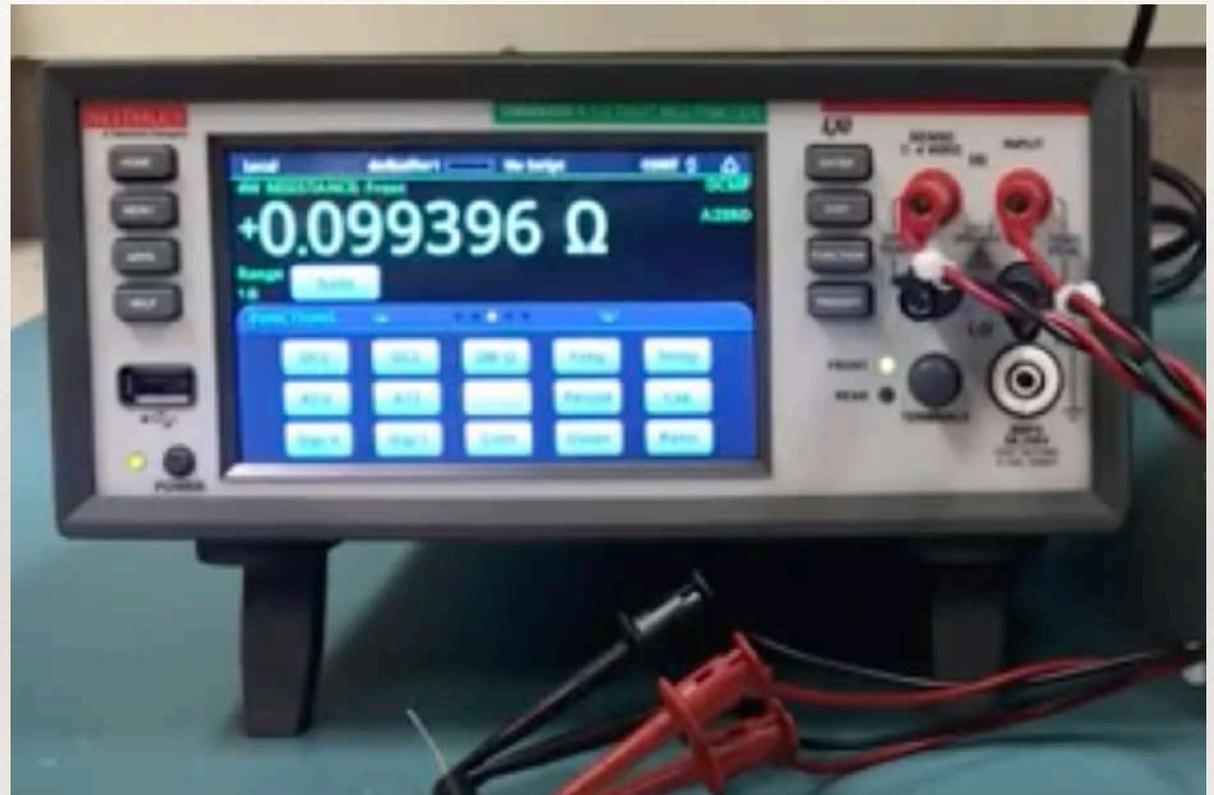
Qual a resolução necessária para termos "x" contagens?

No vídeo, os equipamentos possuem 6 1/2 dígitos, logo:

De 0 a 1.999.999: 2 milhões de valores;

$2^{21} = 2.097.152 \rightarrow$  são 21 bits;

Você não vai conseguir fazer com um Arduino.



---

# Resolução - como calcular

---

Outro exemplo: termômetro digital;

Medir temperaturas até 42,0°C

De 0,0 até 42,0: 421 valores;

$2^9 = 512 \rightarrow$  são 9 bits;

Este dá para fazer com o Arduíno!



# Aula 08

# E/S Analógicas X Digitais

Grandezas Digitais e Analógicas precisam ser processadas constantemente;

Para manipular informações analógicas, há dois desafios:

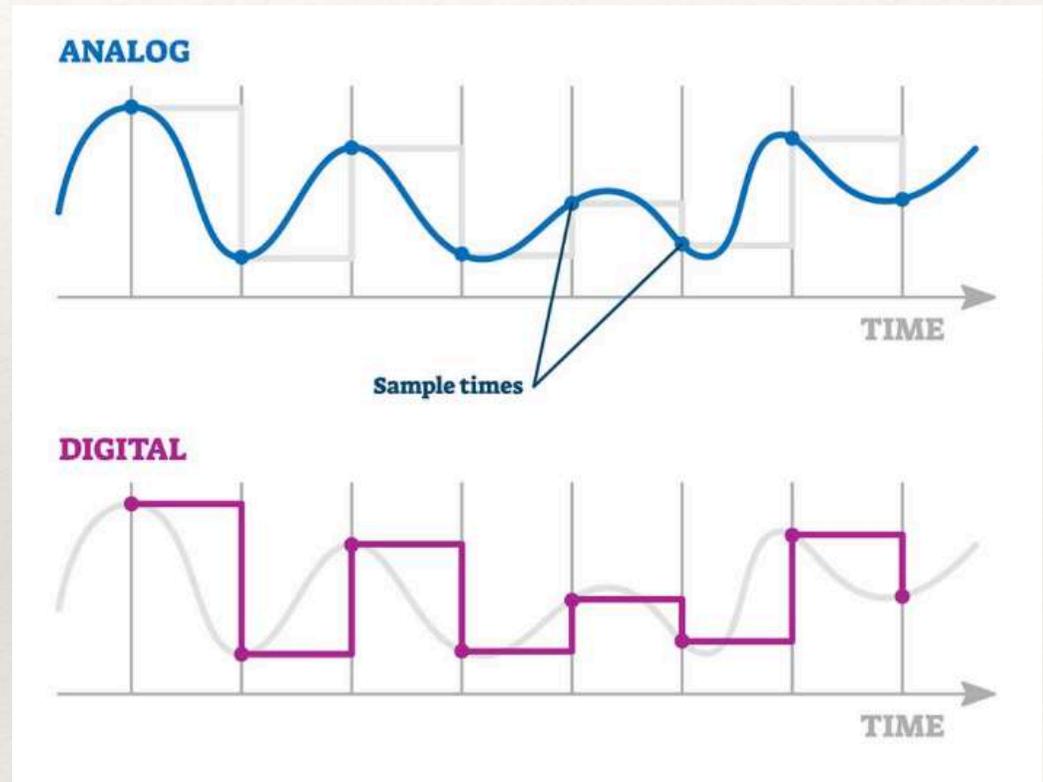
Leitura:

ADCs (*Analog to Digital Converters*);

Escrita:

DACs (*Digital to Analog Converters*);

PWM (*Pulse Width Modulation*).



# ADC: como funciona?

## Arquitetura Paralela

Um divisor resistivo divide a tensão de referência máxima em intervalos;

As saídas do divisor resistivo são comparadas com o valor da entrada analógica através de "n" comparadores, gerando uma tabela verdade com "n" saídas;

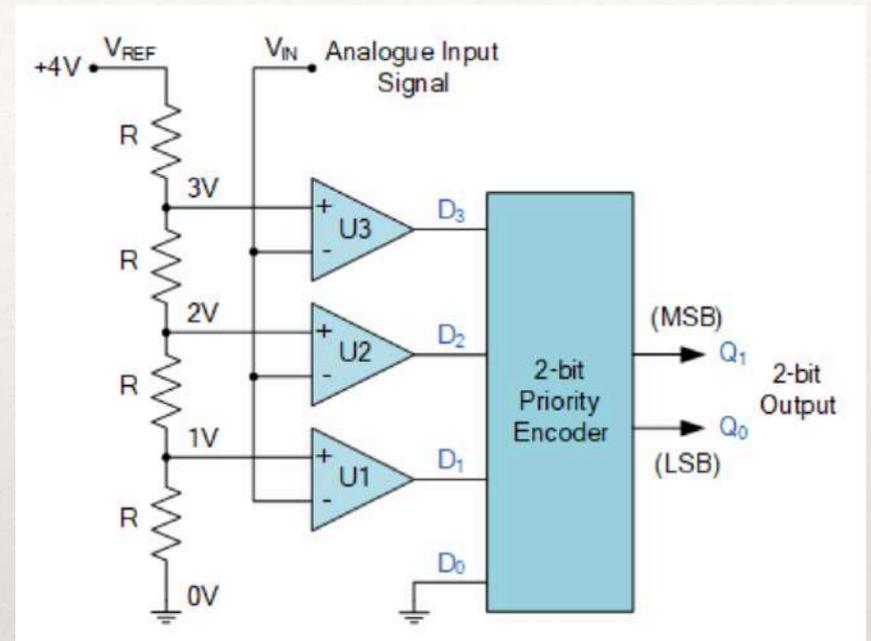
As "n" saídas são convertidas para o número binário correspondente.

A arquitetura é rápida (tempo real), mas sua complexidade é muito grande para resoluções maiores;

Outras arquiteturas são mais adequadas para resoluções mais altas

Pipeline, SAR, Sigma-Delta

Estas arquiteturas geram retardo, e são muito dependentes da linearidade, ruído, exatidão e precisão.



Analogue Input Voltage ( $V_{IN}$ )	Comparator Outputs				Digital Outputs	
	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	Q <sub>1</sub>	Q <sub>0</sub>
0 to 1V	0	0	0	0	0	0
1 to 2V	0	0	1	X	0	1
2 to 3V	0	1	X	X	1	0
3 to 4V	1	X	X	X	1	1

# ADC: como funciona?

## Arquitetura Pa

Um divisor res em intervalos;

As saídas do di da entrada ana gerando uma t

As "n" saídas s correspondent

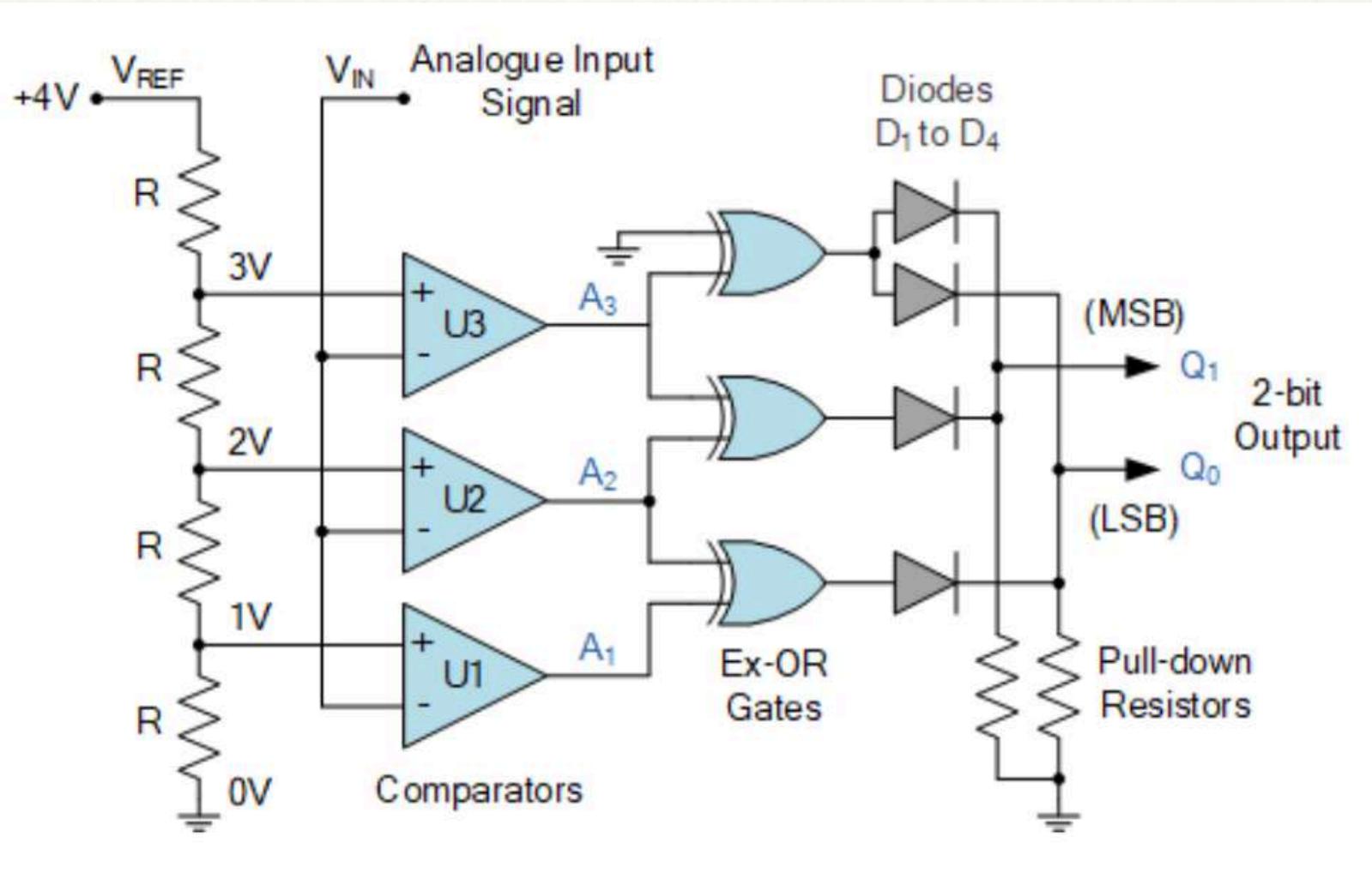
A arquitetura é complexidade maiores;

Outras arquitetura resoluções ma

Pipeline, SAR, S

Estas arquitetura

dependentes da linearidade, ruído, exatidão e precisão.



---

# ADC: como funciona?

---

## Arquitetura Paralela

Um divisor resistivo divide a tensão de referência máxima em intervalos;

As saídas do divisor resistivo são comparadas com o valor da entrada analógica através de "n" comparadores, gerando uma tabela verdade com "n" saídas;

As "n" saídas são convertidas para o número binário correspondente.

A arquitetura é rápida (tempo real), mas sua complexidade é muito grande para resoluções maiores;

Outras arquiteturas são mais adequadas para resoluções mais altas

Pipeline, SAR, Sigma-Delta

Estas arquiteturas geram retardo, e são muito dependentes da linearidade, ruído, exatidão e precisão.

# Amostragem

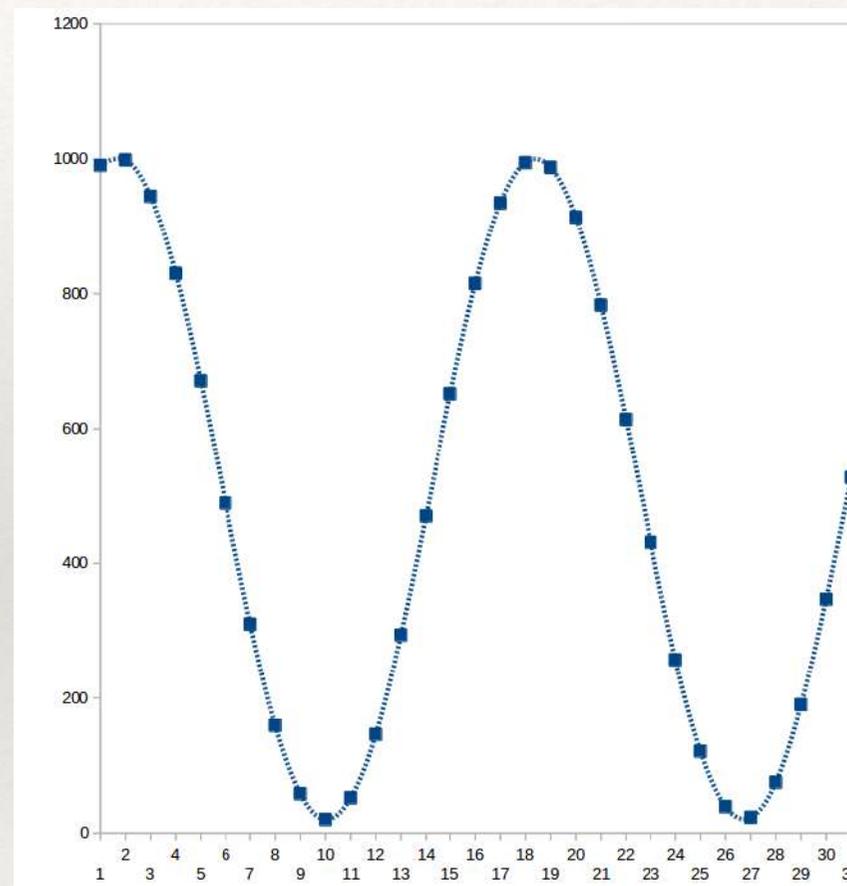
As medições são realizadas em um determinado momento, onde o valor medido é registrado;

A quantidade de medições em um determinado período é chamada de taxa de amostragem

É necessária uma frequência de amostragem mínima para conseguirmos efetivamente representar um determinado sinal, e suas variações com o tempo;

O teorema de Nyquist diz que, para uma amostragem infinita, uma frequência de amostragem pelo menos 2 vezes superior à maior componente de frequência do sinal é capaz de representar adequadamente o sinal original;

Sinais digitais, devido às suas harmônicas, podem ter frequências de amostragem ideais muito elevadas.



A frequência de amostragem está relacionada à performance do ADC.

# Amostragem

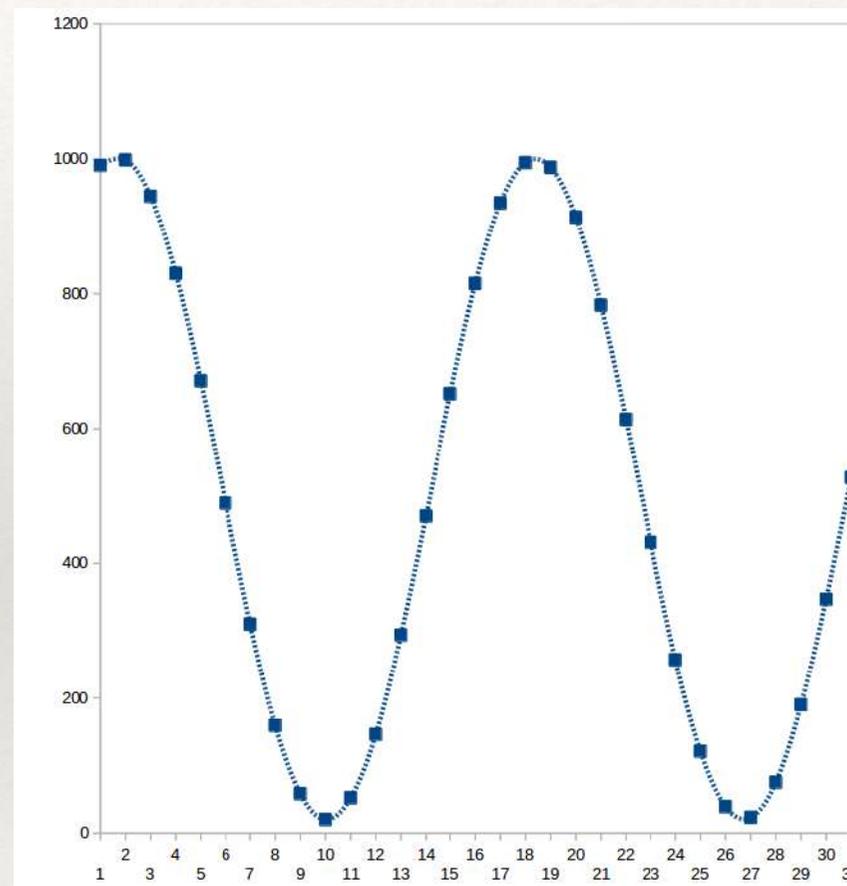
A taxa de amostragem está associada à aplicação

Se você quer ler um valor fixo, ou com variação lenta, a taxa pode ser menor;

Se você deseja analisar o comportamento de um sinal que varia rapidamente, a taxa precisa ser maior;

Um voltímetro tem taxa típica de 4 SA/s (sem considerar cálculo de média);

Um osciloscópio tem taxa típica de 1 GSA/s (você não vai conseguir fazer isto com um Arduino).



# O ADC do Arduino

## Hardware do ADC

Uma amostragem é, na verdade, uma comparação;

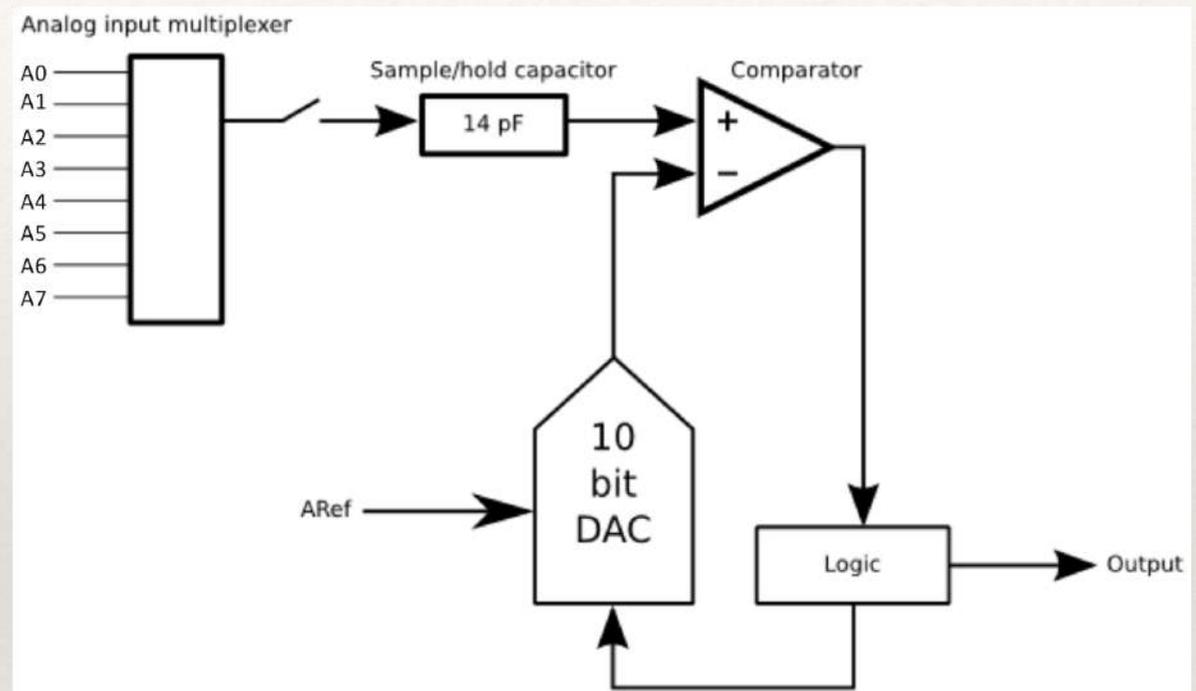
Capacitor para "filtro de ruído analógico";

O filtro é mais eficiente para ruídos de frequência mais alta.

Multiplexação de 8 diferentes entradas analógicas

Poderíamos ter mais?

Tudo é uma questão de performance e aplicação.

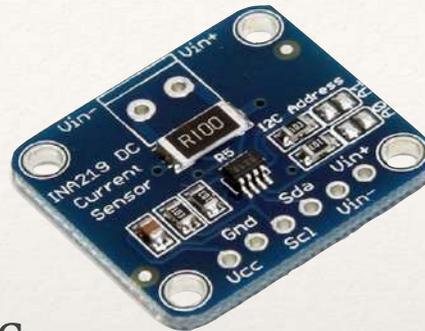


# ADC externo

INA219 - 12 bits com interface i2C

Até 128 SA/s;

Leitor de Tensão e Corrente.



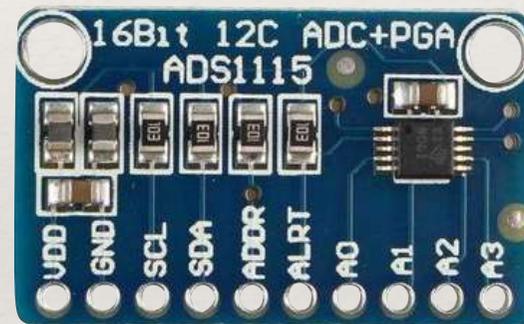
ADS1115 - 16 bits com interface i2C

Até 860 SA/s;

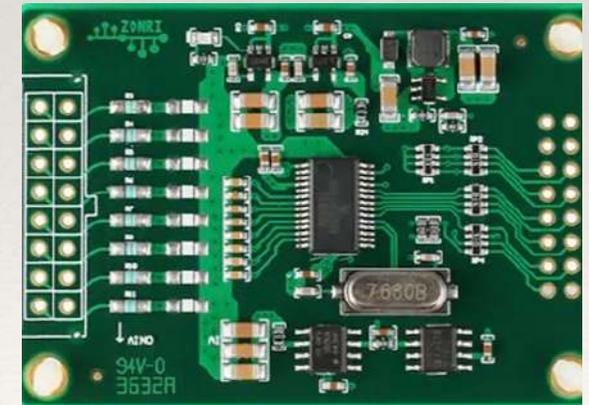
4 canais de entrada ou ...

2 canais diferenciais;

Amplificador de ganho programável.



ADS1252 - 24 bits com interface SPI



# DAC: como funciona?

Um DAC converte uma entrada digital em uma saída de "n" níveis, simulando um valor analógico

Quanto mais bits, "mais analógico";

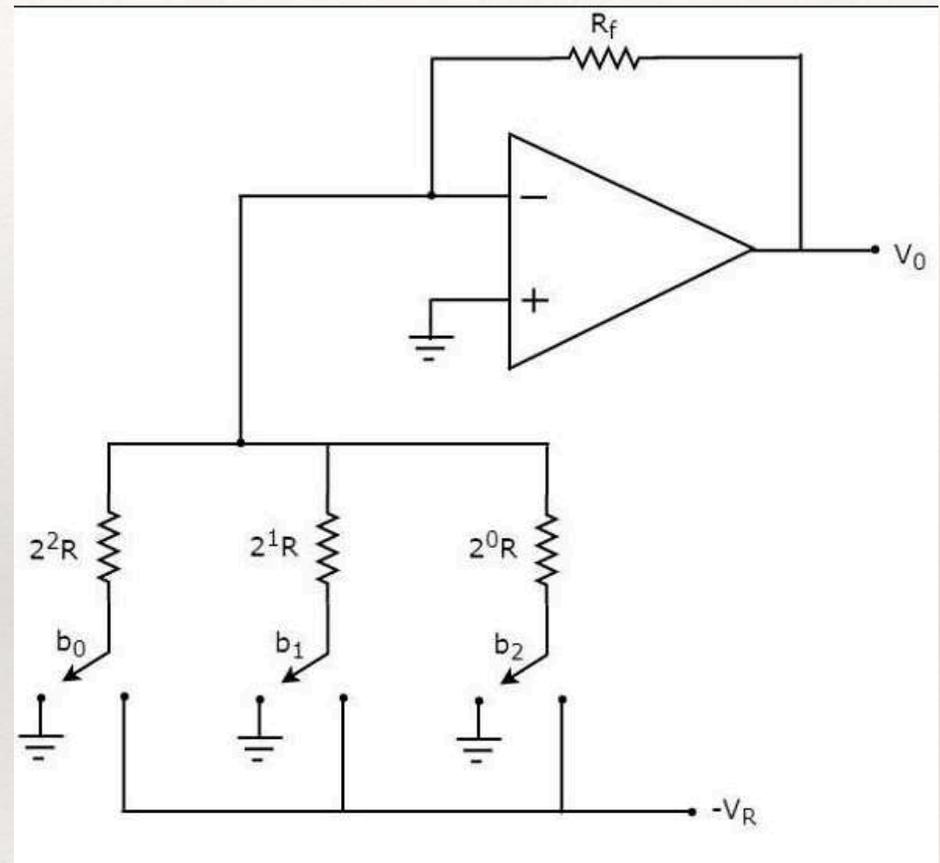
Propriedades: resolução, monotonicidade, linearidade, exatidão e tempo de acomodação.

Existem basicamente 2 tipos de DAC

DAC por Resistores Ponderados;

DAC por Rede R-2R em escada.

O Arduíno Uno não possui DAC integrado



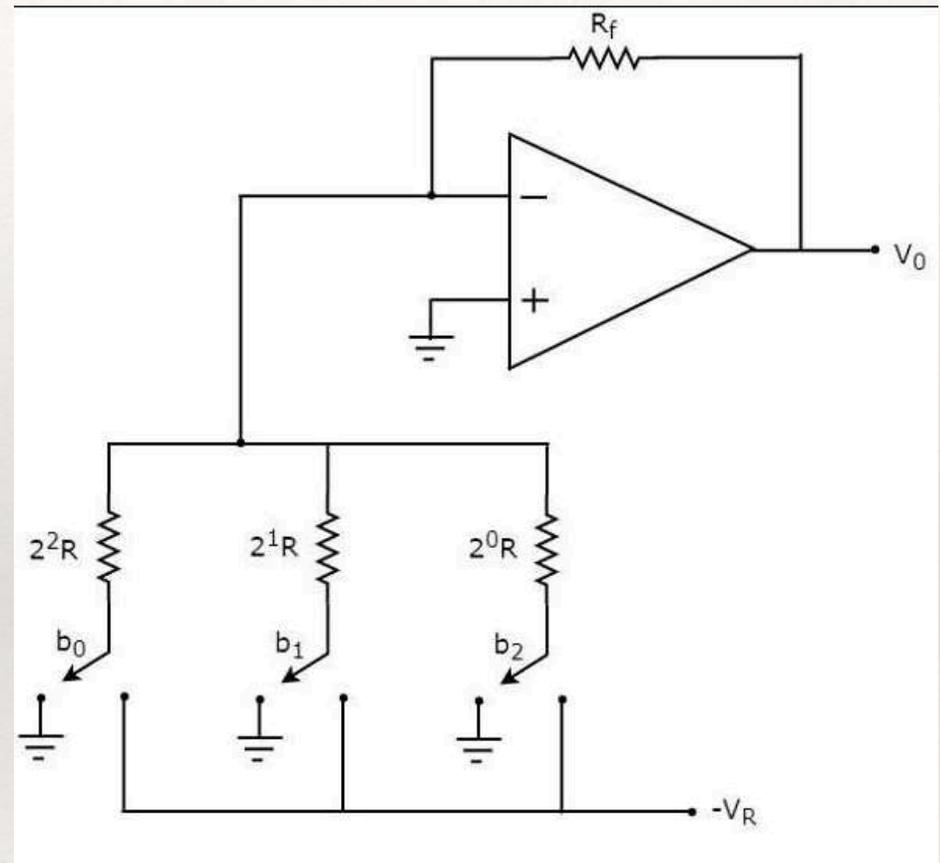
# DAC: como funciona?

## DAC por Resistores Ponderados

Sequência de resistores acompanham as potências de 2 representadas por cada dígito do número;

Complexidade associada à precisão dos resistores, especialmente para muitos dígitos;

Na figura temos um exemplo com entrada de 3 bits.



# DAC: como funciona?

## DAC por Rede R-2R em escada

A vantagem clara está nos dois únicos valores para os resistores, facilitando a construção;

Na figura temos um exemplo com entrada de 3 bits.

## Parâmetros

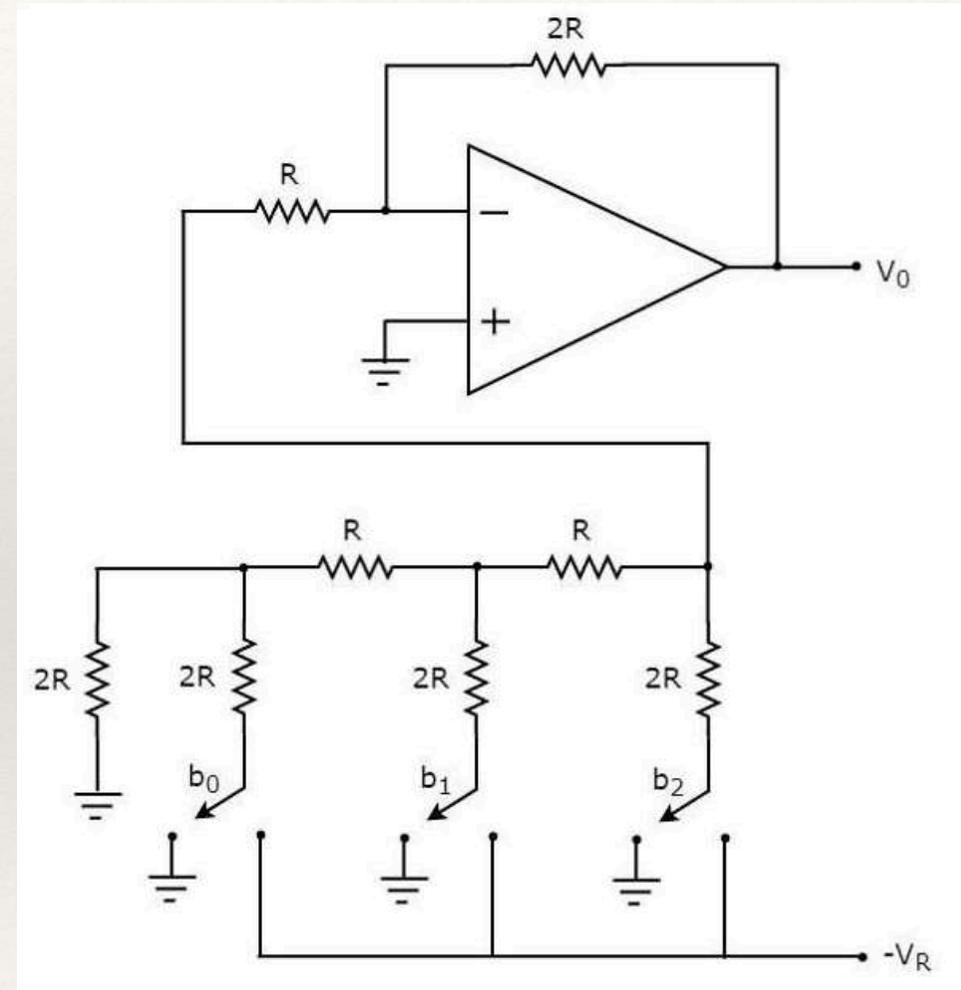
**Resolução:** número de bits

**Monotonicidade:** crescimento constante

**Linearidade**

**Exatidão**

**Tempo de acomodação:** tempo necessário para a saída representar uma mudança entre o menor e maior valor binário na entrada.



# PWM: como funciona?

*Pulse Width Modulation*: Modulação por Largura de Pulso;

Opção mais comum para simular uma saída analógica

Microcontrolador controla o percentual de tempo da saída no nível 1;

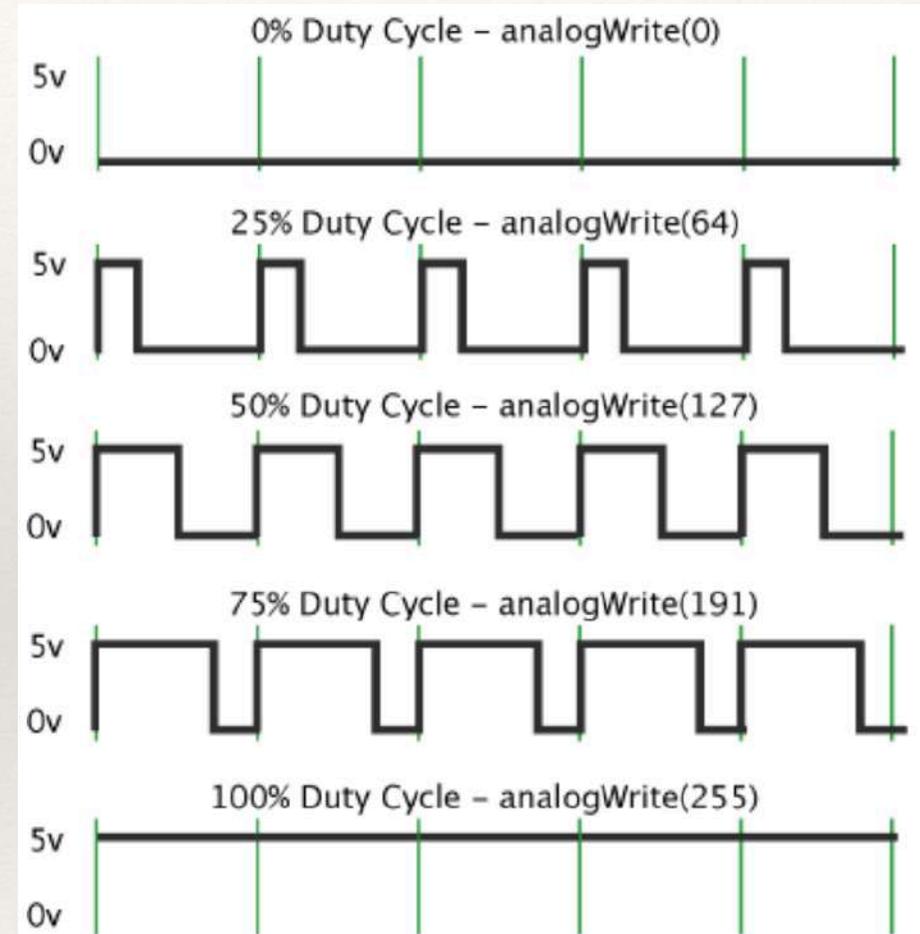
O percentual é chamado de “ciclo de trabalho”, ou *Duty Cycle*;

O efeito “analógico” vem da tensão média na saída.

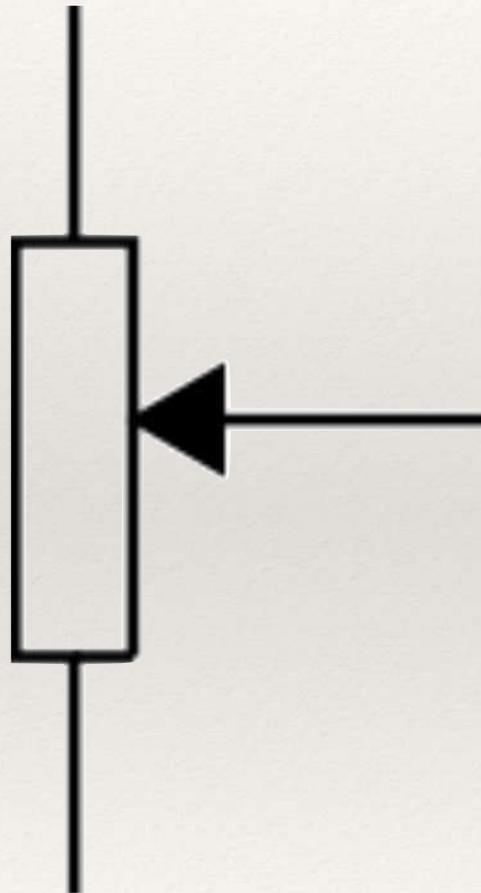
Arduíno Uno

Frequência típica 490Hz, mas pode subir até 4MHz;

Resolução para ajuste do Duty Cycle: 8 bits



# Entrada: Divisor de Tensão



Com a circulação de corrente pelo potenciômetro, a tensão no cursor varia de acordo com a posição do mesmo;

A posição do cursor determina, então, a tensão obtida;

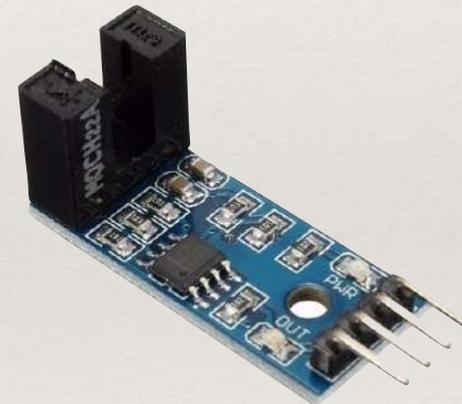
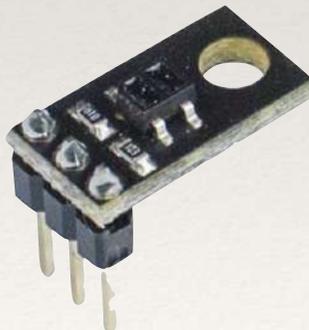
O potenciômetro funciona como um sensor de posição angular, e assim está sendo usado no projeto.

---

# Dispositivos de Entrada

---

Sensores de pulsos (fluxo d'água, contadores etc)

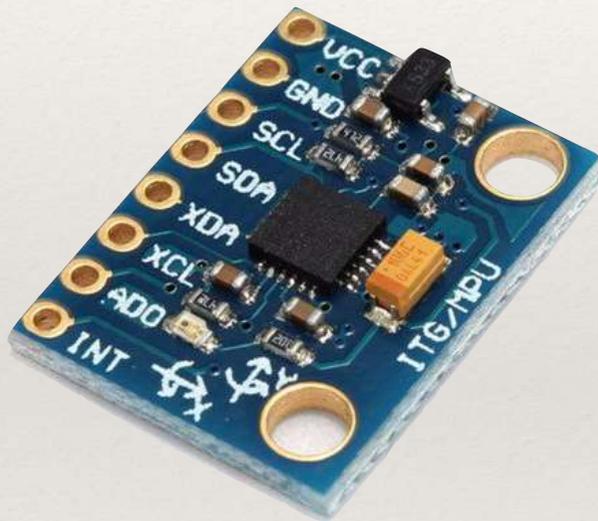


---

# Dispositivos de Entrada

---

Dispositivos especiais (giroscópio, acelerômetro, RTC)



---

# Dispositivos de Saída

---

## LEDs

Saída analógica típica para testes;

Admite modulação via PWM;

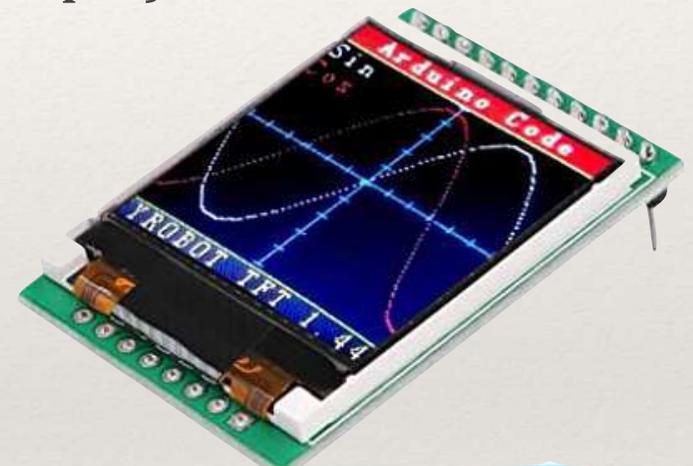
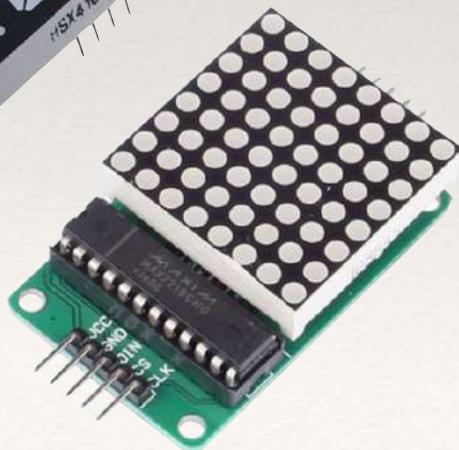
Vermelho, Verde, Amarelo, Azul, Branco e RGB (3 saídas).

# Dispositivos de Saída

Displays (7 segm., matriciais, LCD, TFT, OLED)

Interface convencional ou de rede (i2C, por exemplo);

Cátodo ou Anodo comum;



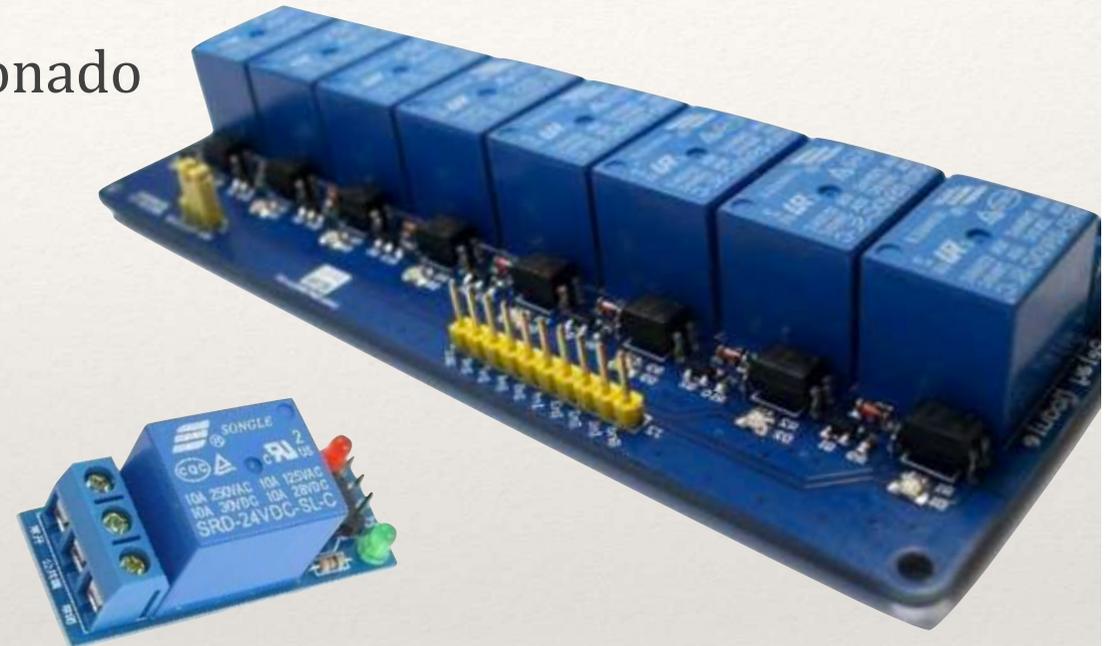
# Dispositivos Saída (Relês)

Nem todo dispositivo pode ser acionado por um transístor

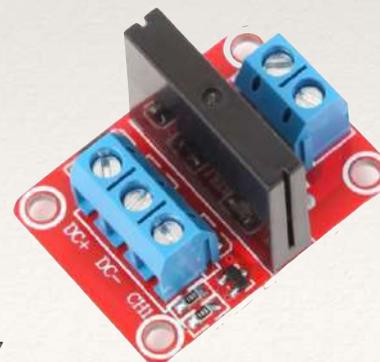
- Cargas AC ou de alta potência;
- Necessidade de isolamento;

Relês são dispositivos mecânicos

- ↑ Promovem grande isolamento;
- ↑ Suportam cargas DC ou AC;
- ↓ São lentos;
- ↓ Consomem potência significativa;
- ↓ Vida útil é baixa  
Suportam até 300.000 acionamentos (típ)



“Relês” de estado sólido



- ↑ São rápidos;
- ↑ Consomem pouca potência;
- ↑ Vida útil elevada;  
Suportam até 10.000.000 de acionamentos (típ)

Tem isolamento inferior aos relês

Suportam cargas AC

---

# Dispositivos de Saída

---

Servos, motores convencionais e de passo



# Aula 09

# Memória no ATmega328P

O ATmega328P não segue a arquitetura padrão Von Neumann

A Memória que armazena o programa é diferente da memória que contém as variáveis;

São 3 tipos de memória diferentes;

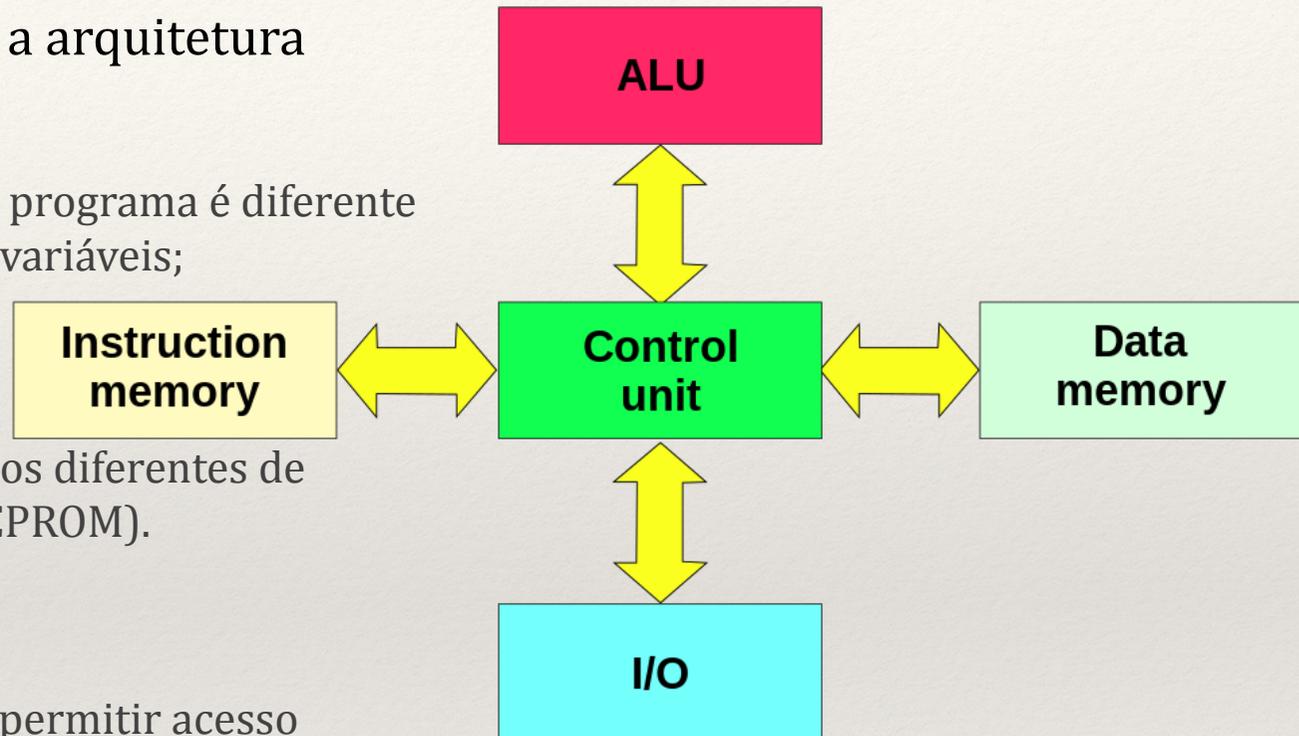
O ATmega328P possui 3 tipos diferentes de memória (Flash, SRAM e EEPROM).

"Arquitetura Harvard"

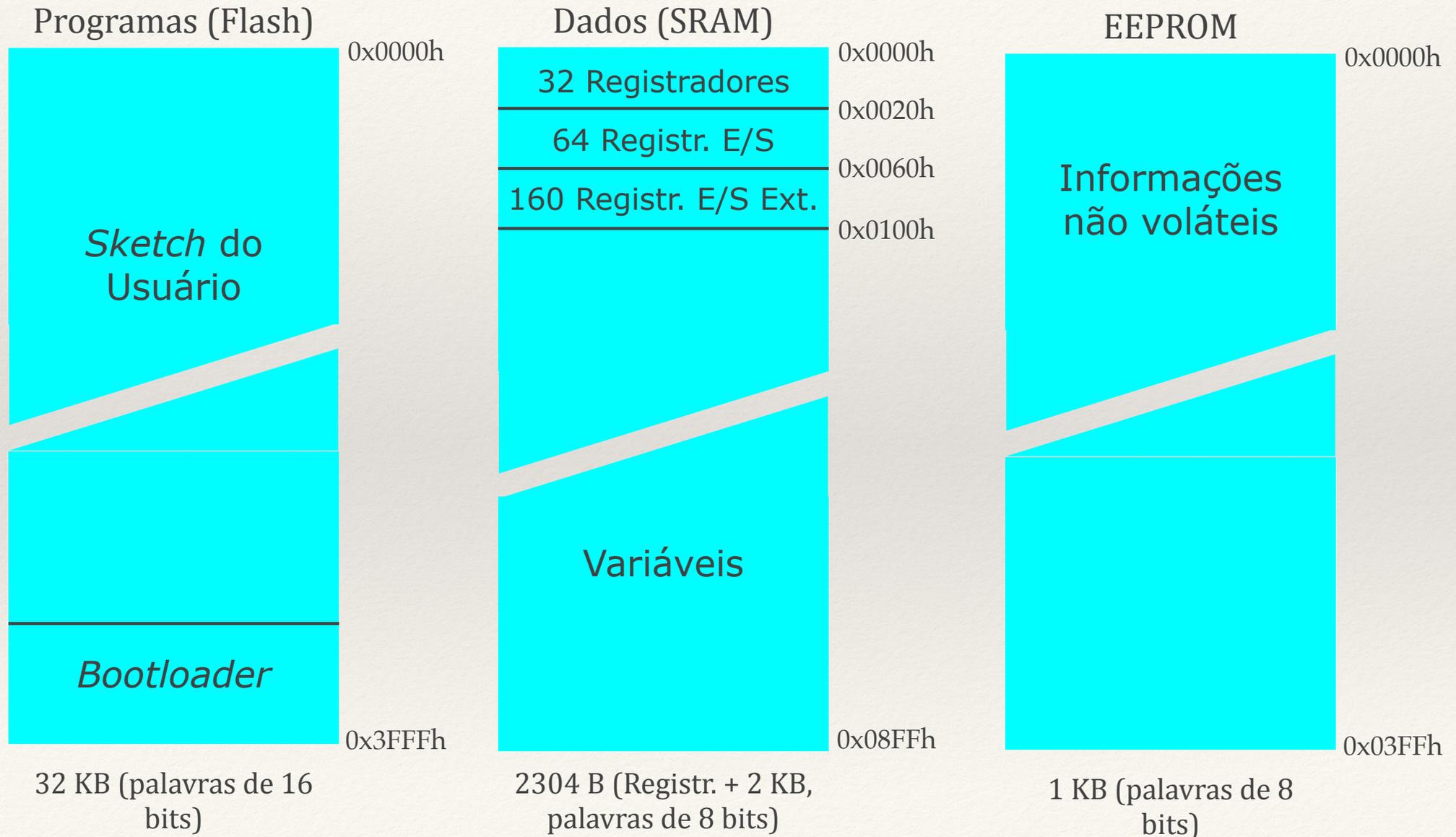
Ganho de performance por permitir acesso simultâneo às instruções e aos dados;

Tamanho que contém uma instrução (16 bits) pode ser diferente da palavra de dados (8 bits);

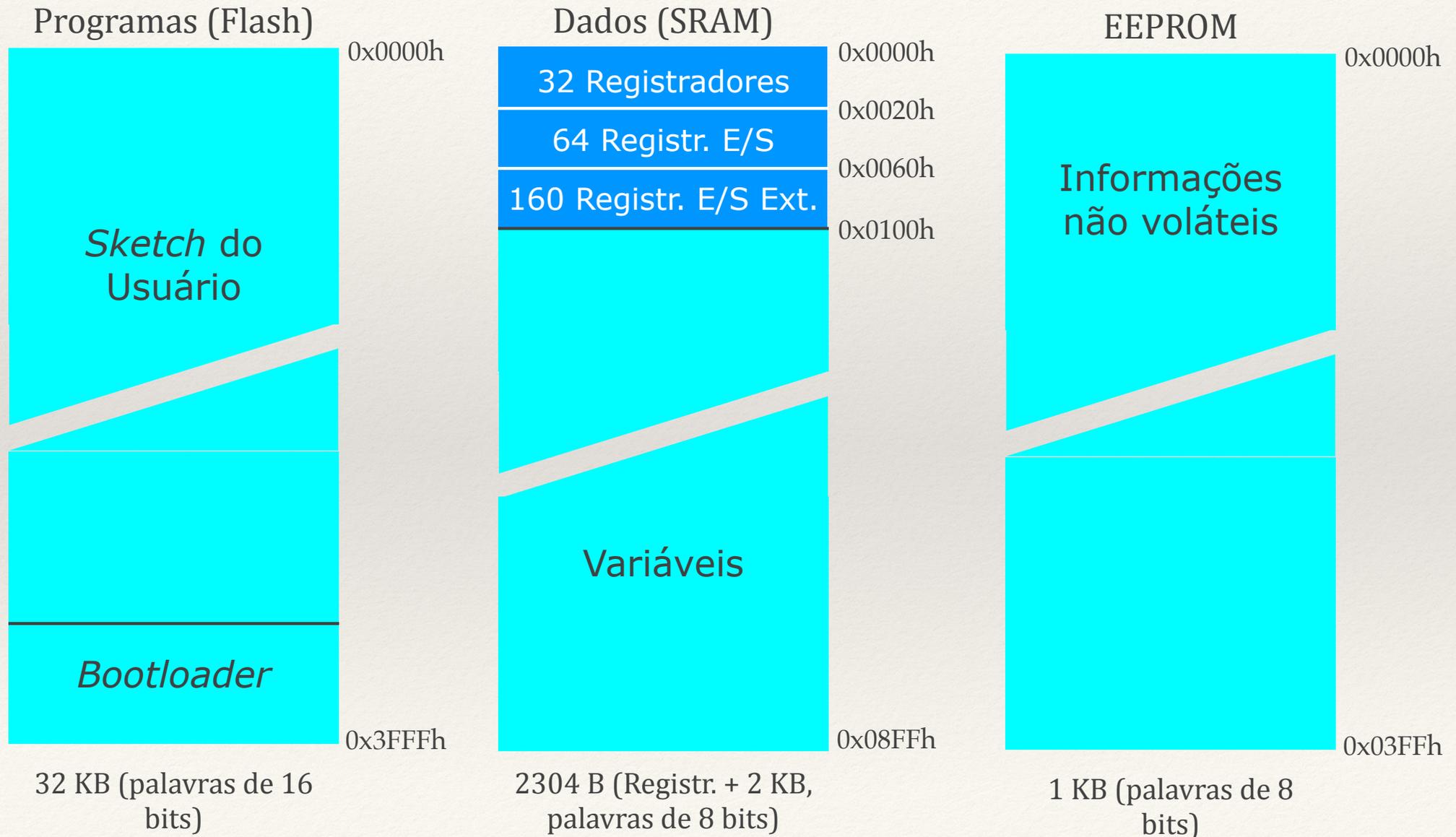
Processadores modernos misturam Von Neumann com Harvard, através do uso do cache.



# Memória no ATmega328P



# Memória no ATmega328P



# ATMega328P em blocos

Os registradores controlam toda a operação do ATMega328P

Muitos são acessados indiretamente quando usamos o conjunto de instruções padrão do Arduino;

Alguns recursos exigem acesso direto

Interrupções de Relógio;

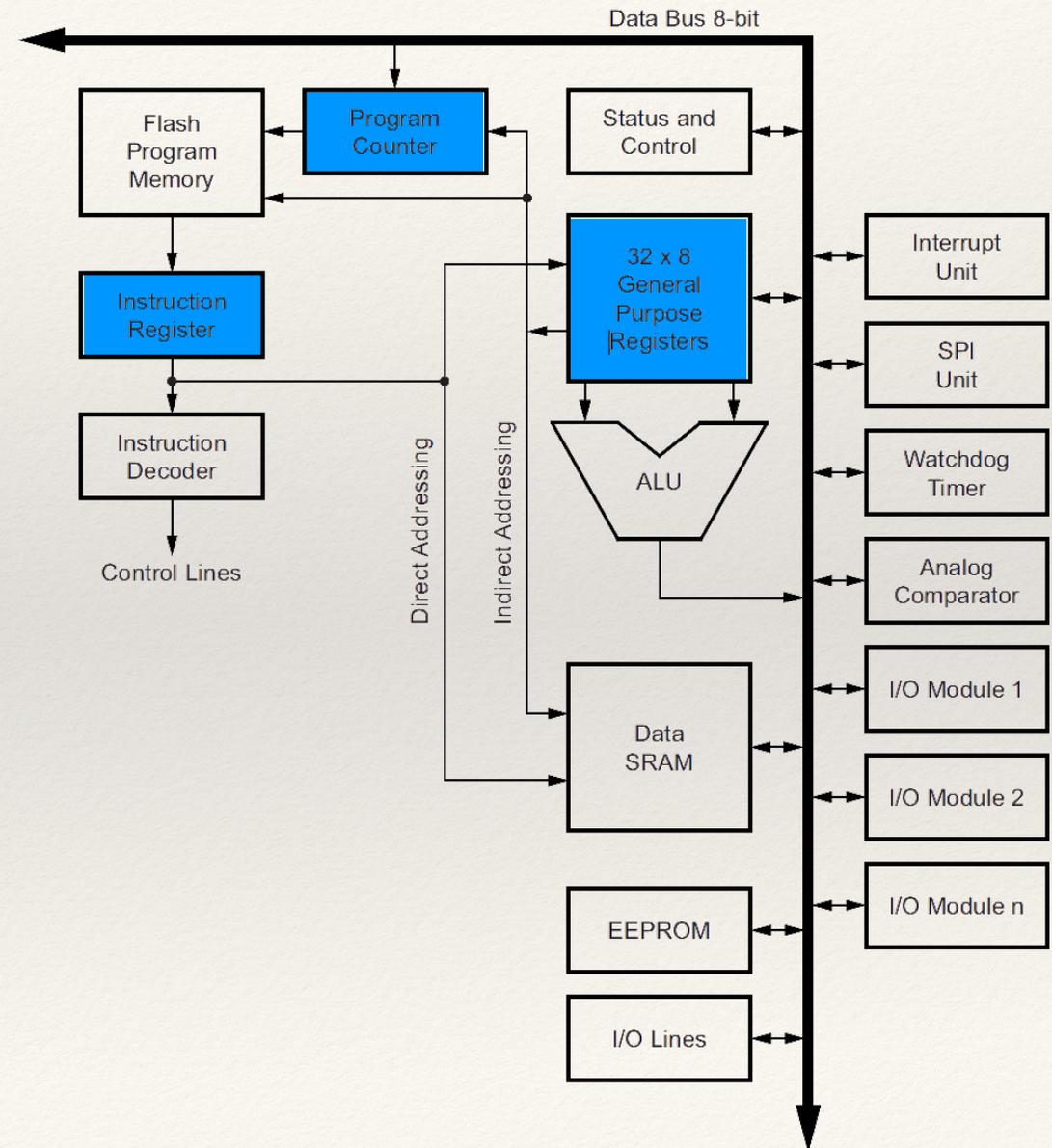
Alteração de frequência PWM;

Manipulação "criativa" de portas.

Acesso via registradores

Códigos menores, e mais rápidos;

Manipulação e consulta binária.



---

# Acesso Binário

---

Bits dos registradores agregam funções na mesma palavra

Alterações e consultas bit-a-bit;

Operações lógicas binárias: E, OU, NÃO, deslocamento.

Registradores de 8 ou 16 bits

Registradores de 16 bits tipicamente são tratados byte a byte.

Ativando bits (*set bits*)

Diretamente pela carga da palavra;

Carga de bit com operação OU (*OR*).

Desativando bits (*reset bits*)

Desativação de bit com operação E (*AND*).

# Acesso Binário - exemplo

## Projeto básico com Arduíno Uno

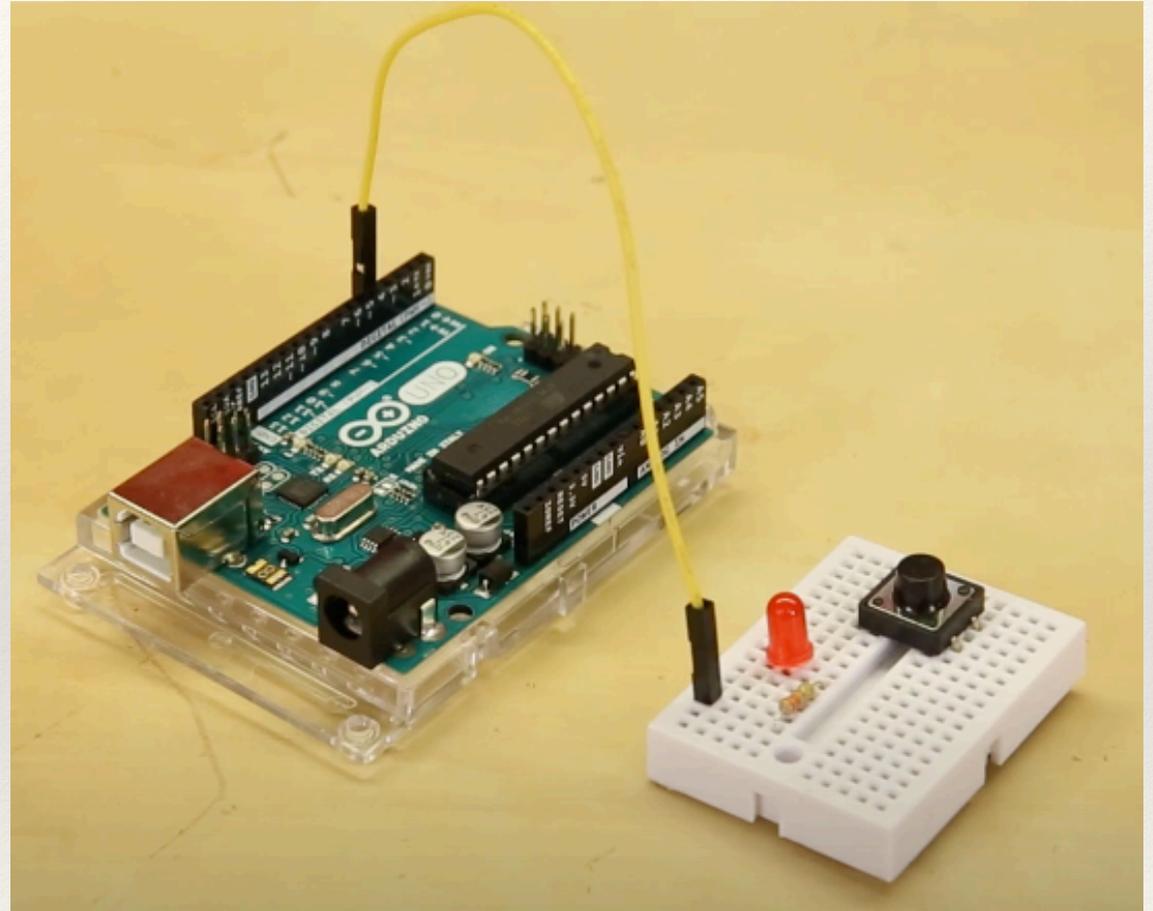
Ler botão;

Se apertado, acende LED; caso contrário, apaga.

## Desenvolvimento do teste

Teste inicial com código em C padrão do Arduíno;

Migração de parte do código, desta vez manipulando diretamente os registradores.



# Acesso Binário - exemplo

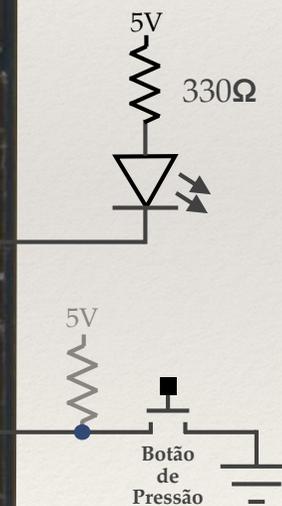


Componentes alimentados pelo Arduino Uno

LED (digital 5);

Botão de Leitura (digital 2);

Pull-Up interno.



# Projeto Básico

*Sketch* define entrada e saída

Pino 2 utilizado para leitura do botão,  
com uso de PullUp interno;

Pino 5 utilizado para ativar o LED.

## Função básica Arduino

```
pinMode ( ) ;
```

Opera indiretamente com registradores.



```
TesteRegistradores $
#define pinoBotao 2
#define pinoLed 5

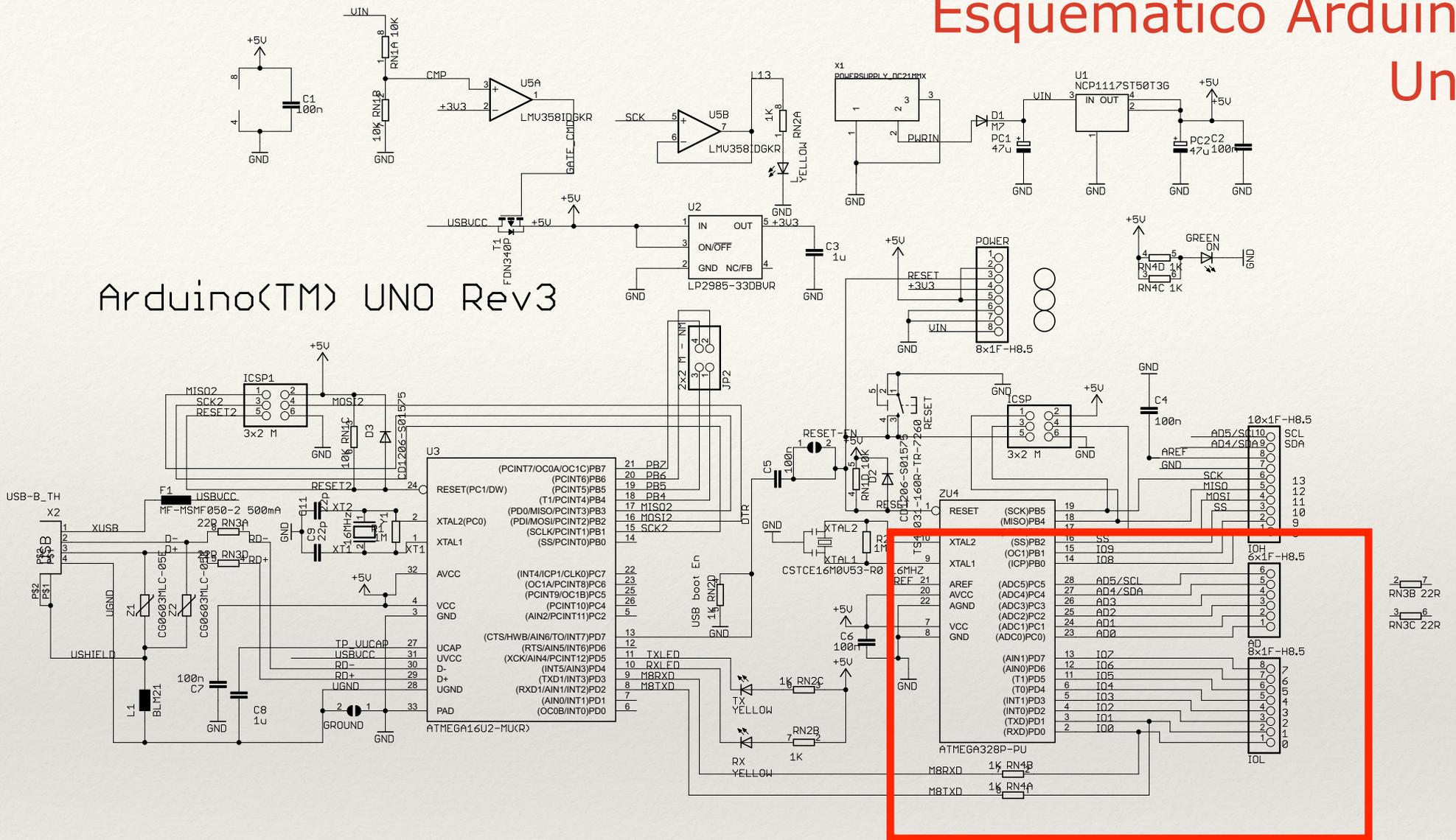
void setup() {
  pinMode( pinoBotao, INPUT_PULLUP );
  pinMode( pinoLed, OUTPUT );
}

void loop() {
  int botao = digitalRead( pinoBotao );

  if( botao == LOW ) {
    digitalWrite( pinoLed, LOW );
  } else {
    digitalWrite( pinoLed, HIGH );
  }
}
```

# Esquemático Arduino Uno

## Arduino(TM) UNO Rev3

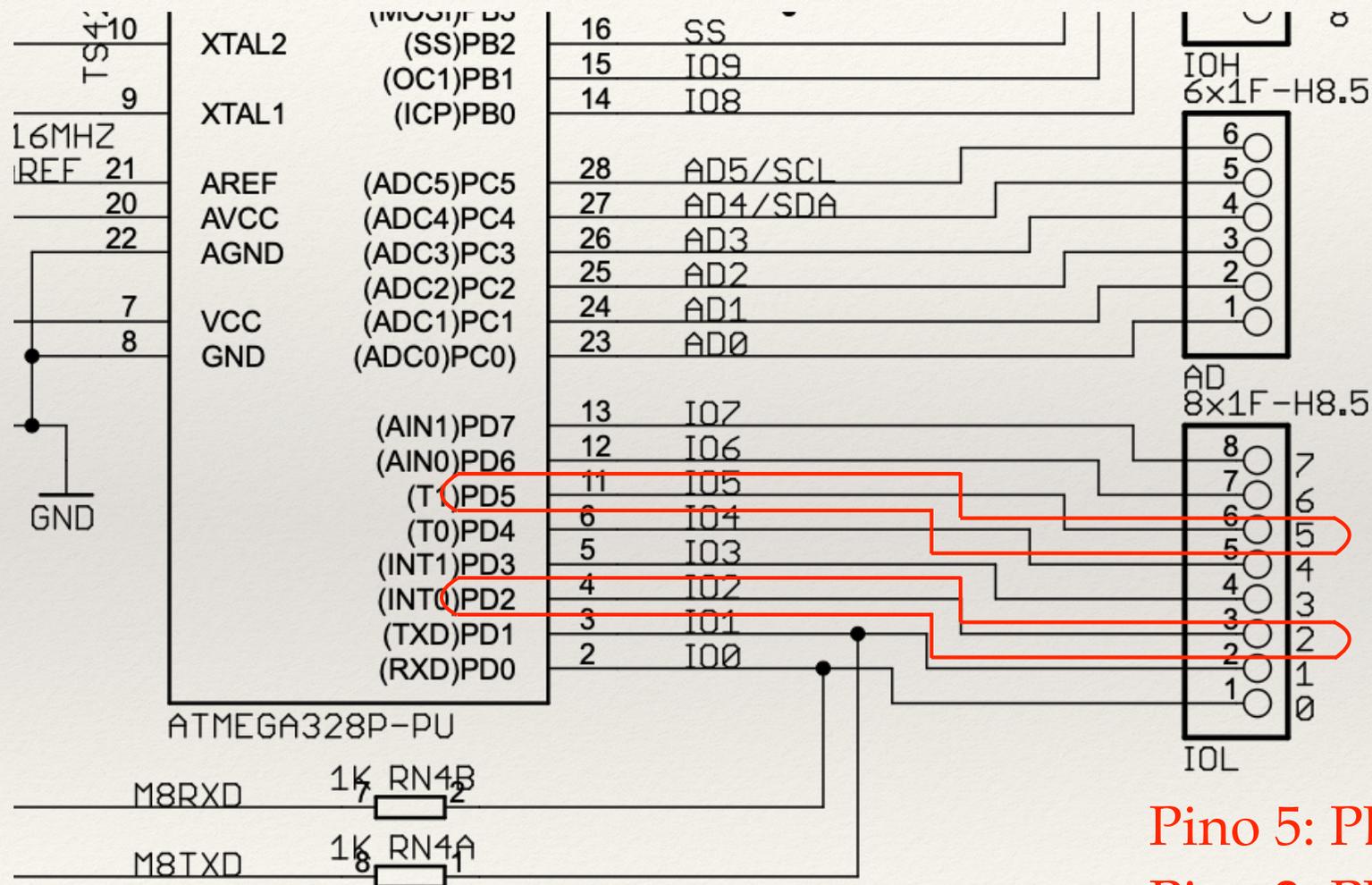


Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

ARDUINO is a registered trademark.  
Use of the ARDUINO name must be compliant with <http://www.arduino.cc/en/Main/Policy>

Diagrama esquemático do Arduino Uno v3

# Registradores pinos 2, 5 ?

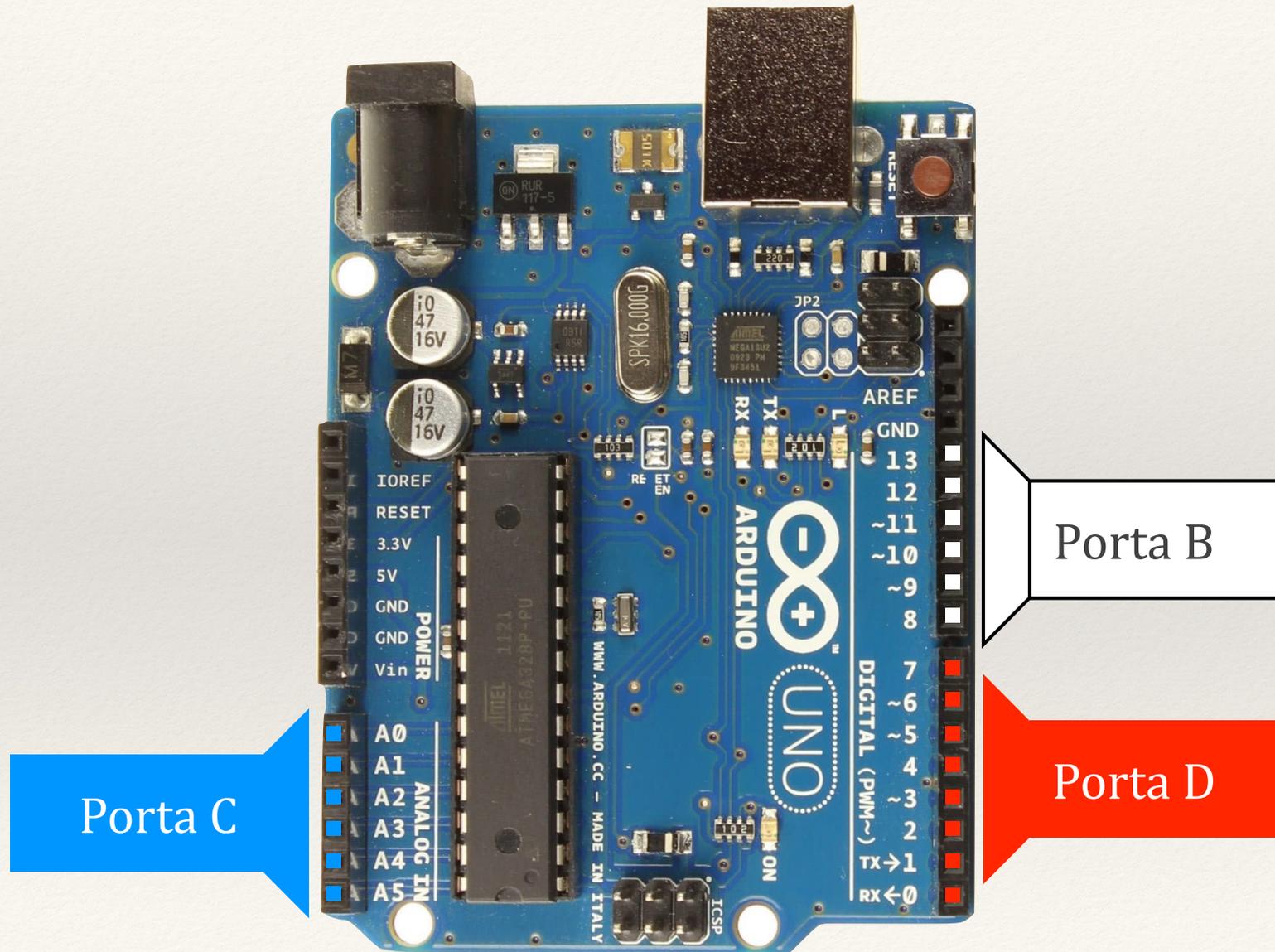


Pino 5: PD5

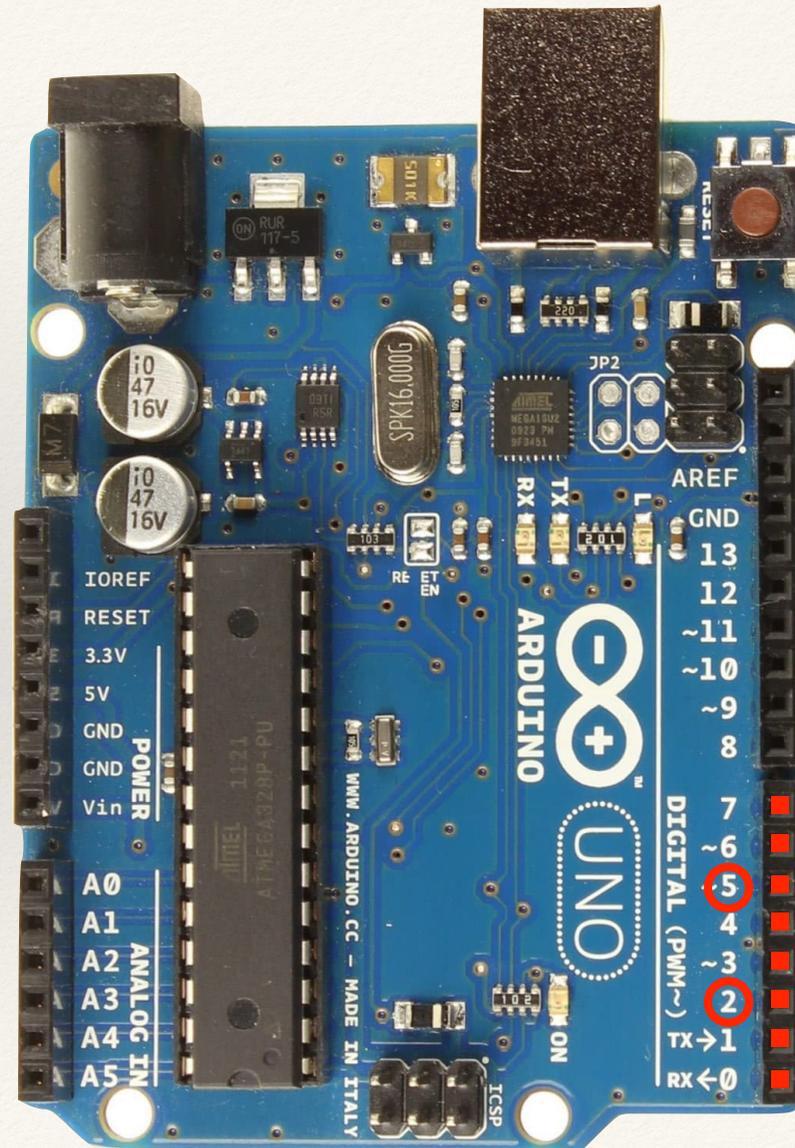
Pino 2: PD2

(ambos na porta D)

# Arduíno: portas B, C e D



# Arduíno: portas B, C e D



Porta D

# Registadores - Porta D

## PORTD – The Port D Data Register

Bit	7	6	5	4	3	2	1	0	
0x0B (0x2B)	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

## DDRD – The Port D Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x0A (0x2A)	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

## PIND – The Port D Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x09 (0x29)	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A								

# Registadores - Porta D

## PORTD – The Port D Data Register

Bit	7	6	5	4	3	2	1	0	
0x0B (0x2B)	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

## DDRD – The Port D Data Direction Register

0x0A								DDD0	DDRD
Read/Write								R/W	
Initial Value								0	

O registrador DDR determina quais são os pinos que serão utilizados como entrada (valor 0) ou saída (valor 1)

## PIND – The Port D Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x09 (0x29)	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A								

# Registadores - Porta D

## PORTD – The Port D Data Register

Bit  
0x0B (0x0B)  
Read/Write  
Initial Value

O registrador PORT determina, no caso de SAÍDA, qual o valor a ser apresentado; no caso de entrada, habilita o resistor interno de *PullUp* (1), ou desabilita (0)

## DDRD – The Port D Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x0A (0x2A)	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

## PIND – The Port D Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x09 (0x29)	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A								

# Registadores - Porta D

## PORTD – The Port D Data Register

Bit	7	6	5	4	3	2	1	0	
0x0B (0x2B)	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

## DDRD – The Port D Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x0A (0x2A)	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

## PIND – The Port D Input Pins Address

Bit	1	0	
0x0C (0x2C)	PIND1	PIND0	PIND
Read/Write	R	R	
Initial Value	N/A	N/A	

O registrador PIN, que deve ser tratado apenas para leitura, possui o status atual de cada pino

# Acesso Binário - exemplo

*Sketch* define pino 5 como saída e pino 2 como entrada

1º passo: carregar 1 no bit 5 do registrador DDRD;

2º passo: carregar 1 no bit 2 do registrador PORTD;

Atribuindo o valor da palavra

Em binário:

```
DDRD = B00100000;
```

```
PORTD = B00000100;
```

Ou, em decimal:

```
DDRD = 32;
```

```
PORTD = 2;
```



```
TesteRegistradores $
#define pinoBotao 2
#define pinoLed 5

void setup() {
  pinMode( pinoBotao, INPUT_PULLUP );
  pinMode( pinoLed, OUTPUT );
}

void loop() {
  int botao = digitalRead( pinoBotao );

  if( botao == LOW ) {
    digitalWrite( pinoLed, LOW );
  } else {
    digitalWrite( pinoLed, HIGH );
  }
}
```

# Acesso Binário - exemplo

## Atribuindo o valor ao BIT !

Operação OU com conteúdo anterior:

```
DDRD  |= B00100000;
```

```
PORTD |= B00000100;
```

Embora funcione, este procedimento força a criação de uma constante, e de uma variável provisória (lento?);

Existe um procedimento mais comum para este tipo de operação, que utiliza o registro de deslocamento.



```
TesteRegistradores §
#define pinoBotao 2
#define pinoLed 5

void setup() {
  pinMode( pinoBotao, INPUT_PULLUP );
  pinMode( pinoLed, OUTPUT );
}

void loop() {
  int botao = digitalRead( pinoBotao );

  if( botao == LOW ) {
    digitalWrite( pinoLed, LOW );
  } else {
    digitalWrite( pinoLed, HIGH );
  }
}
```

# Acesso Binário - exemplo

Ativando o bit 5 em DDRD:

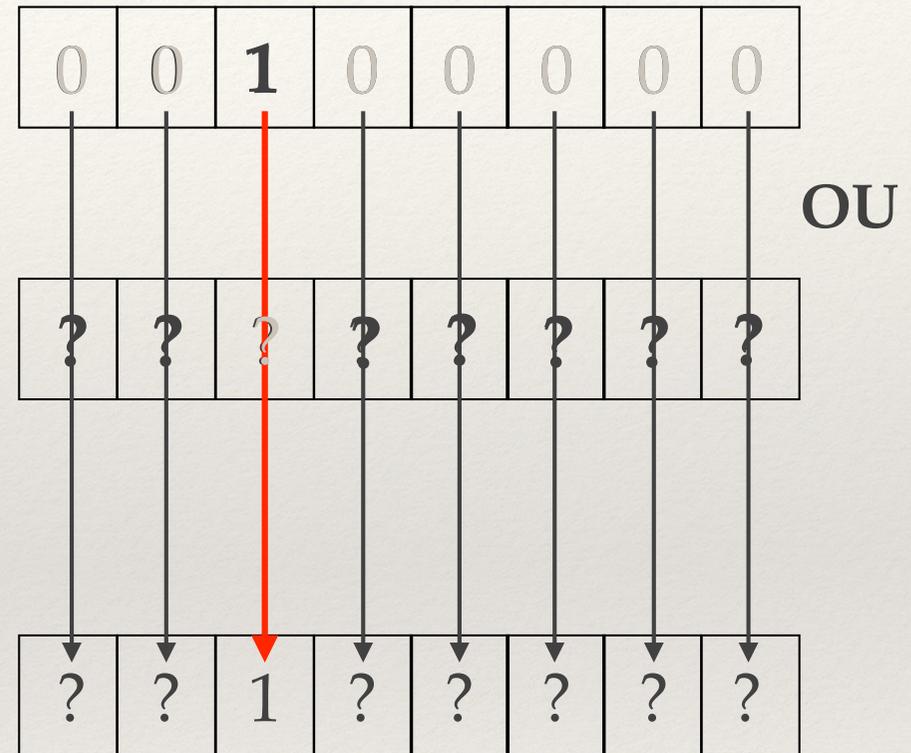
Criamos uma palavra com o bit desejado ativo:

```
(1<<pinoLed)
```

Fazemos uma operação OU (|) com o valor atual de DDRD:

```
DDRD = ( 1 << pinoLed ) | DDRD;
```

Todos os bits de DDRD são mantidos, exceto o bit 5, que assume o valor 1.



# Acesso Binário - exemplo

## Desativando o bit 5 em DDRD:

Criamos uma palavra com o bit desejado ativo:

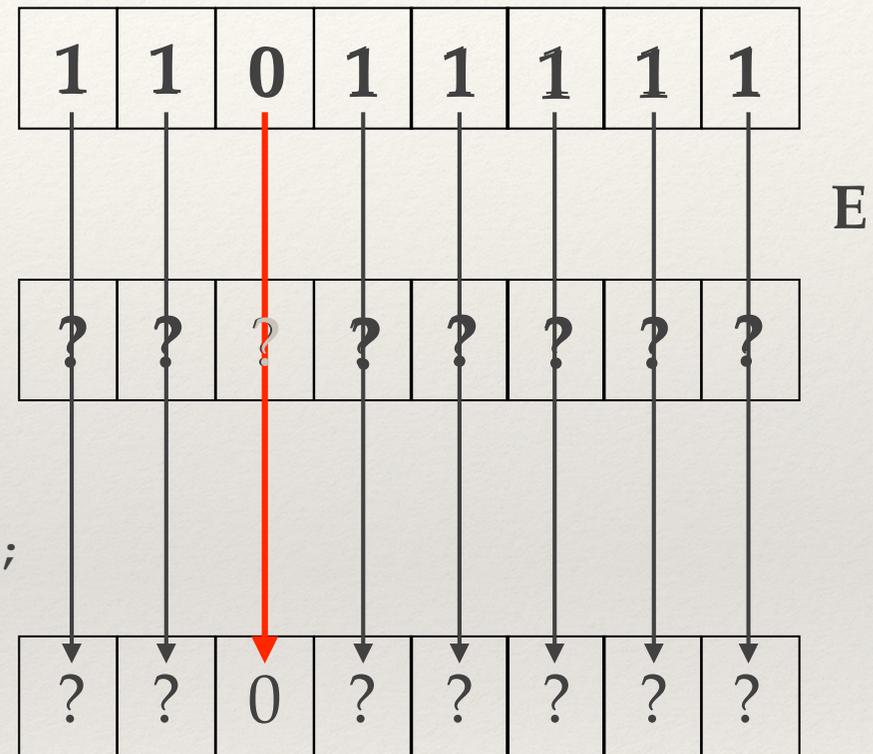
```
(1<<pinoLed)
```

Invertemos a palavra bit-a-bit ( ~ );

Fazemos uma operação E (& ) com o valor atual de DDRD:

```
DDRD = ~( 1 << pinoLed ) & DDRD;
```

Todos os bits de DDRD são mantidos, exceto o bit 5, que assume o valor 0.



# Acesso Binário - exemplo

Para ativar o bit 2 de PORTD em DDRD, o processo é similar

Criamos uma palavra com o bit desejado ativo:

```
(1<<pinoBotao)
```

Fazemos uma operação OU (|) com o valor atual de PORTD:

```
PORTD =( 1 << pinoBotao ) |  
PORTD;
```



```
TesteRegistradores  
#define pinoBotao 2  
#define pinoLed 5  
  
void setup() {  
  DDRD = ( 1 << pinoLed ) | DDRD;  
  PORTD = ( 1 << pinoBotao ) | PORTD;  
}  
  
void loop() {  
  int botao = ( ( 1 << pinoBotao ) & PIND ) >> 2;  
  
  if( botao == LOW ) {  
    digitalWrite( pinoLed, LOW );  
  } else {  
    digitalWrite( pinoLed, HIGH );  
  }  
}
```

# Acesso Binário - exemplo

Para leitura do Botão, utilizamos o registro PIND

Criamos uma palavra colocando 1 no bit a ser lido:

```
( 1 << pinoBotao )
```

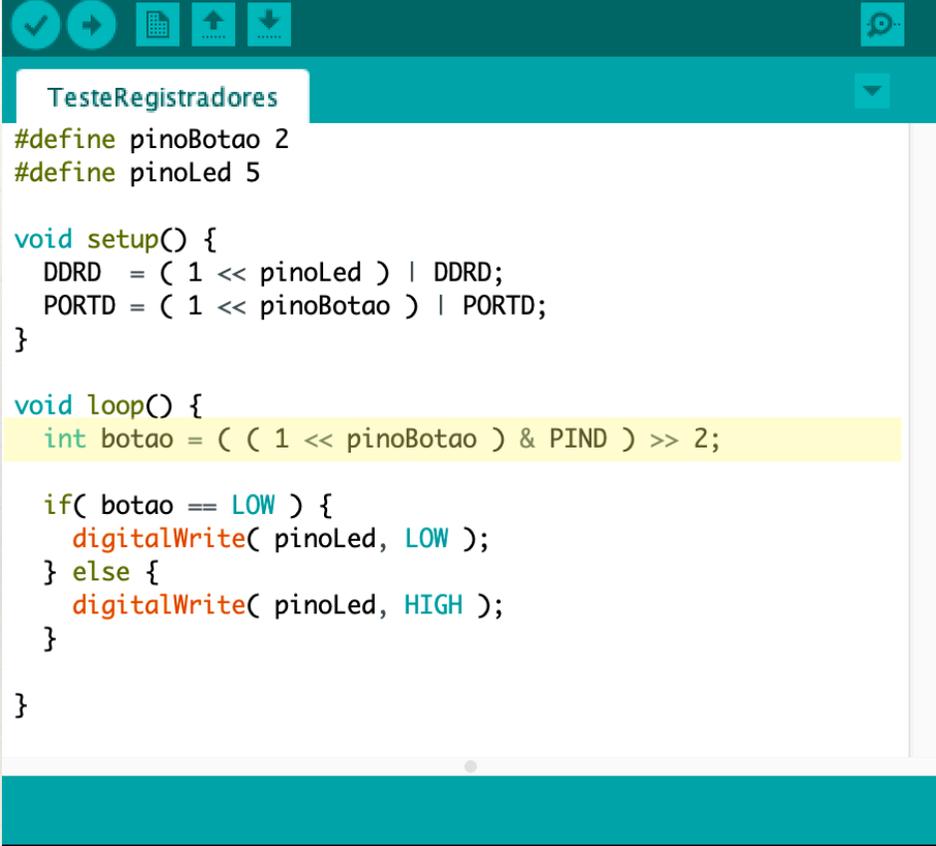
Fazemos uma operação E ( & ) com o valor atual de PIND:

```
( 1 << pinoBotao ) & PIND;
```

Revertemos o deslocamento para colocar o bit testado na posição zero:

```
botao = ( ( 1 << pinoBotao )  
& PIND ) >> 2;
```

O valor será 1 (verdadeiro) ou 0 (falso), a depender do valor lido na posição desejada do PIND.



```
TesteRegistradores  
#define pinoBotao 2  
#define pinoLed 5  
  
void setup() {  
  DDRD = ( 1 << pinoLed ) | DDRD;  
  PORTD = ( 1 << pinoBotao ) | PORTD;  
}  
  
void loop() {  
  int botao = ( ( 1 << pinoBotao ) & PIND ) >> 2;  
  
  if( botao == LOW ) {  
    digitalWrite( pinoLed, LOW );  
  } else {  
    digitalWrite( pinoLed, HIGH );  
  }  
}
```

# Acesso Binário - exemplo

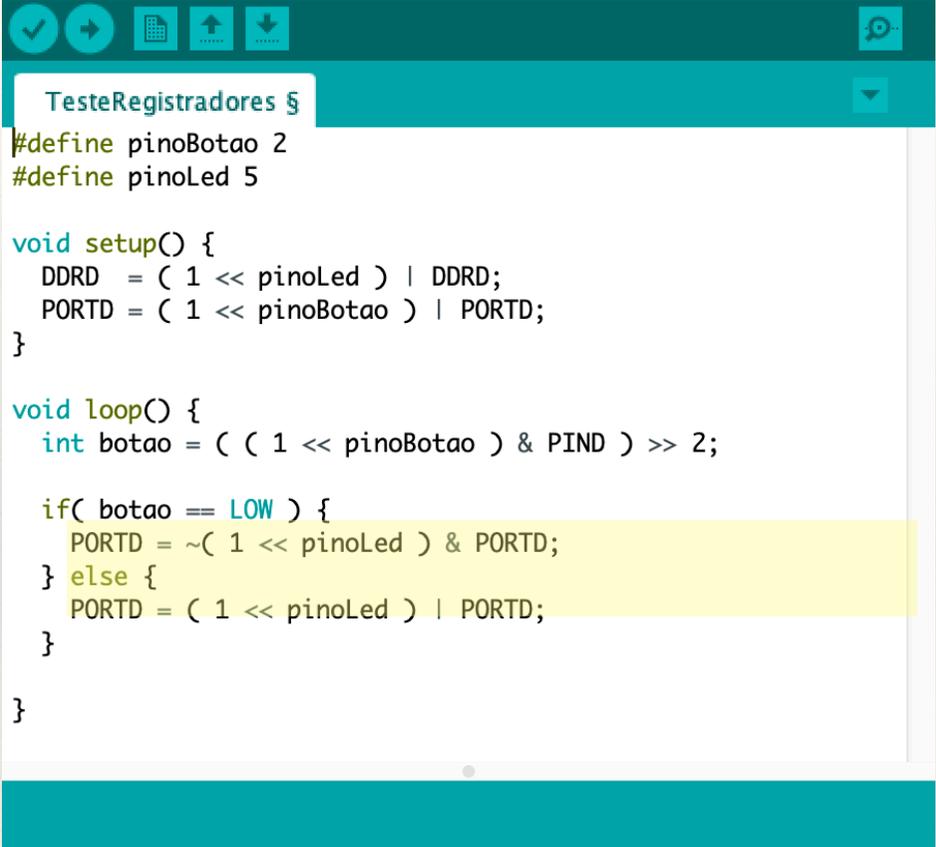
Para acender ou apagar o LED, os processos são bem similares

Criamos uma palavra com o bit desejado ativo:

```
( 1 << pinoLed )
```

Fazemos uma operação OU ( | ) com o valor atual de PORTD:

```
if( botão == LOW ) {  
    PORTD = ~( 1 << pinoLed )  
& PORTD;  
} else {  
    PORTD =( 1 << pinoLed ) |  
PORTD;  
}
```



```
TesteRegistradores §  
#define pinoBotao 2  
#define pinoLed 5  
  
void setup() {  
    DDRD = ( 1 << pinoLed ) | DDRD;  
    PORTD = ( 1 << pinoBotao ) | PORTD;  
}  
  
void loop() {  
    int botao = ( ( 1 << pinoBotao ) & PIND ) >> 2;  
  
    if( botao == LOW ) {  
        PORTD = ~( 1 << pinoLed ) & PORTD;  
    } else {  
        PORTD = ( 1 << pinoLed ) | PORTD;  
    }  
}
```

# Aula 10