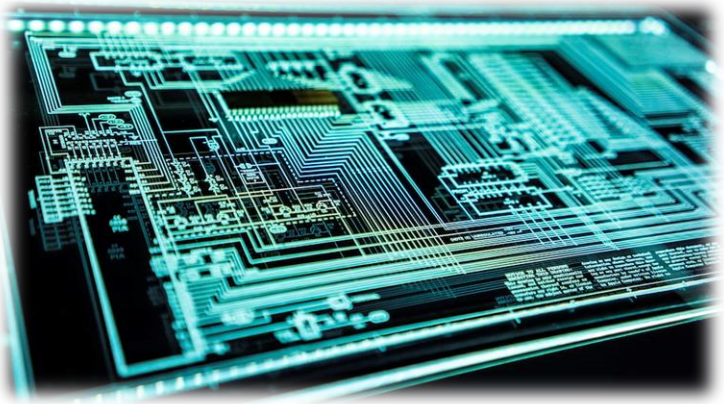
UNIVERSIDADE DE SÃO PAULO
Escola de Engenharia de São Carlos
Departamento de Engenharia Elétrica e de Computação

SISTEMAS EMBARCADOS



Características, Mercado, Projeto, e Soluções

Prof. Pedro de Oliveira C. Junior

pedro.oliveiracjr@usp.br

1

CARACTERÍSTICAS

Conceito, definições, requisitos, organização e programação

2

MERCADO

Principais destaques e tendências

3

PROJETO

Aspectos envolvidos em produtos

4

SOLUÇÕES

Qual sistema usar? Sistemas embarcados no curso de Engenharia de Computação

1 – Características dos sistemas embarcados

Algumas definições

SISTEMAS EMBARCADOS / EMBEDDED SYSTEMS

- Sistemas embarcados (S.E.), e/ou "embutidos" – "incorporados" – "integrados" – inserido/introduzido em um elemento - no contexto da eletrônica e da computação.
- É um sistema microprocessado no qual o computador é completamente encapsulado ou dedicado ao dispositivo ou sistema que ele controla: um equipamento eletrodoméstico, por ex.

> "***Is a system whose principal function is not computational, but which is controlled by a computer embedded within it.***" ([digital file] - Dhiraj Rane) (percepção).

Sist. Eletrônicos/computacionais embarcados: É definido pela **IEEE 610.12-1990** (IEEE *Standard Glossary of Software Engineering Terminology*) como "um sistema computacional que faz parte de um sistema maior e implementa alguns dos requerimentos deste sistema. << **Produtos que incorporam um computador**!>>

Aplicações




Qualquer produto/dispositivo que use um microprocessador, mas não seja um PC de uso geral


SAÚDE:

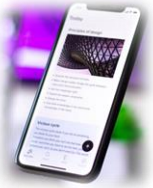
Monitoramento EEG e ECG.

PERIFÉRICOS DE COMPUTADOR

Impressoras, scanners ...

AUTOMAÇÃO E SEGURANÇA RESIDENCIAL

Condicionadores de ar, sprinklers, alarmes de incêndio



SISTEMAS DE REDE DE COMPUTADORES

Roteadores e switches de rede.

ELETRÔNICOS DE CONSUMO E LEITORES

Telefones celulares, tablets, eletrodomésticos; smart watches, Leitores de código de barras, smart card.

APLICAÇÕES ESPACIAIS E COMUNICAÇÃO

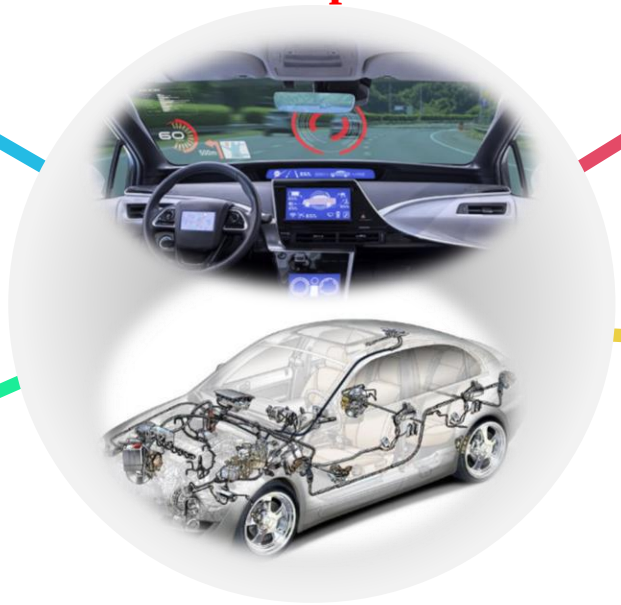
Telecomunicação, satélites, sondas robóticas espaciais, aeronaves

Aplicações automotivas

Um automóvel moderno pode ter cerca de 100 microprocessadores:

Controle da mistura ar/combustível, sincronização do motor: proporciona menos emissões, melhor eficiência de combustível

Multimídia e comunicação sem fio;



Veículos autônomos;

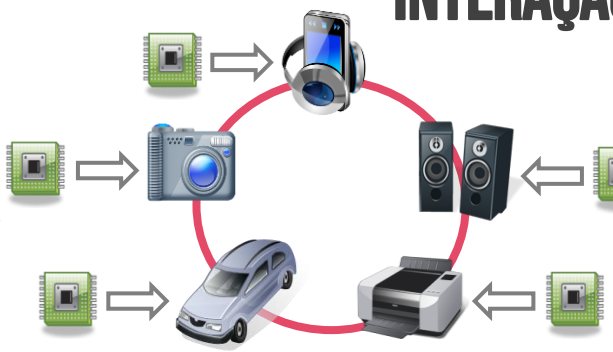
Robótica na linha de montagem;

- ✓ **Microcontrolador para verificar o cinto de segurança**
- ✓ **Microcontroladores para sinalizar dispositivos no painel**
- ✓ **Controle Automático de Estabilidade (ASC+T): Controla o motor para melhorar a estabilidade.**

Características

DEDICADOS À TAREFAS ESPECÍFICAS E COM APLICAÇÃO BEM DEFINIDA!

- ✓ Confiabilidade e segurança - tolerantes a falhas, disponibilidade, confidencialidade
- ✓ Baixo custo, peso e tamanho reduzidos
- ✓ Restrição de memória
- ✓ Restrição de software;
- ✓ Sob-medida (fábrica)
- ✓ Eficiencia energética;
- ✓ Hostilidade ambiental



INTERAÇÃO POR SENSORES E ATUADORES

Software: correção lógica, tempo real, tolerante a falhas, atualização segura, concorrência de elementos físicos, específico.

Combinação de hardware e software

Interfaces de entrada e saída

Realiza uma função específica para a qual foi programado

sob medida (desempenho inferior a PCs).

É reconfigurável via software* (não p/usuário final)

Interage continuamente com o ambiente a sua volta

Usualmente um sistema embarcado executa somente um programa repetidamente (enquanto um PC desktop pode executar uma variedade de programas)

Função específica e arquitetura

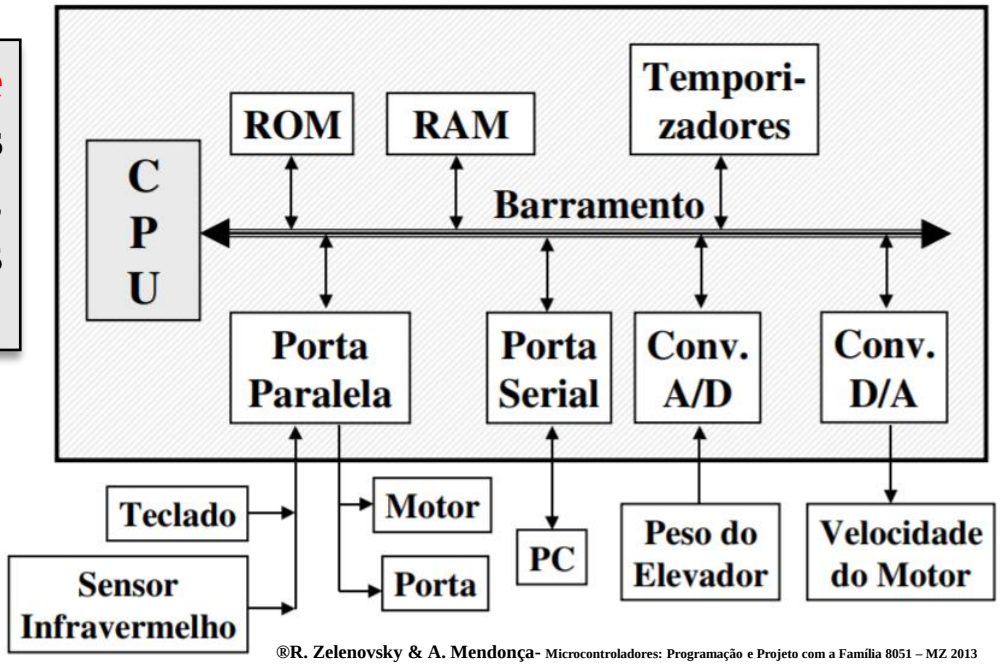
HARDWARE E SYSTEM CORE

- O coração do S.E., composto por chip semicondutor (processador/CPU e memória de alta velocidade).
- Periféricos, E/S, interfaces, memória e barramento

Exemplos: uma lavadora de roupas que exerce apenas sua funcionalidade; um ar condicionado, projetado para controlar a temperatura etc.

O hardware embarcado também requer interface com usuário, interfaces de entrada/saída, visualização e memória.

Microcontroladores, processadores digitais de sinais (DSP), dispositivos lógicos programáveis complexos (FPGA), tecnologia de GPU, circuitos integrados específicos (ASIC) podem ser usados.



®R. Zelenovsky & A. Mendonça- Microcontroladores: Programação e Projeto com a Família 8051 – MZ 2013

SOFTWARE

- Sistema operacional, RTOS, Compiladores, Ferramentas de Debug, Firmware da aplicação

Complexidade (tarefas simples?)

EXEMPLO: FORNO DE MICRO-ONDAS:

- Ao pressionar uma tecla, como PIPOCA, um sistema deve ajustar a potência correta, selecionar e medir o tempo em que o forno deve ficar acionado e emitir um sinal quando a tarefa for concluída.

PARA EXECUTAR A SIMPLES OPERAÇÃO DO FORNO:

- **O "cérebro" do forno deve: receber sinais de sensores (da porta, para saber se foi realmente fechada), acionar o equipamento de potência, calcular o tempo da operação, acionar o motor de rotação do prato, permitir que o usuário interrompa a operação a qualquer tempo, atualizar o display, medir o tempo que se passou desde o início da operação, etc.**

Classificação de Sistemas Embarcados

SISTEMAS EMBARCADOS EM PEQUENA ESCALA:

MCU de 8 – 16 bits, editor e IDE - aplicação não crítica em termos de desempenho.

SE SOFISTICADOS:

Enorme complexidade, 32-64 bits, reconfigurável, maior capacidade de memória, comunicação sem fio, maior velocidade, computadores de placa única (SBC); Programação em alto nível; Sit. Operacional, processamento de imagens, inteligência artificial etc.

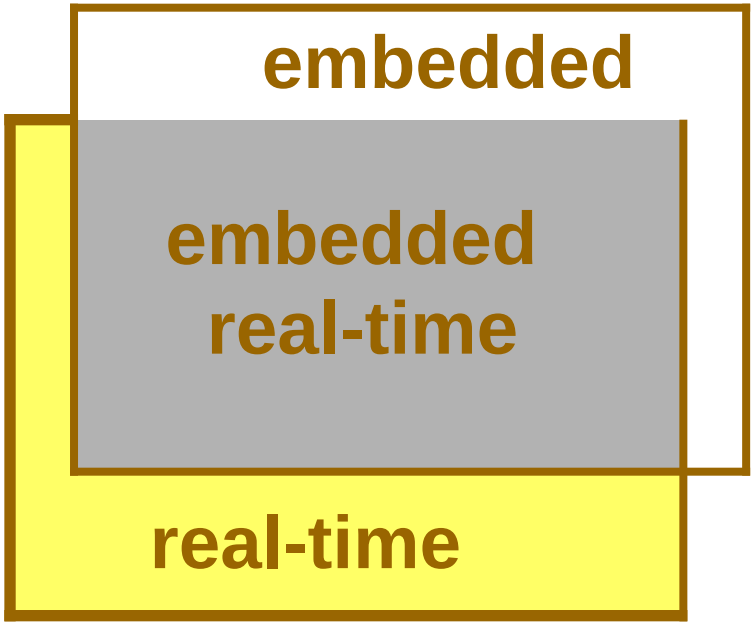
SISTEMAS EMBARCADOS EM ESCALA MÉDIA:

16 -32 bits, RISC, DSP, ASICs, maior complexidade, com ferramentas de programação em alto nível, depuração etc.- Geralmente contêm sistema operacional de tempo real
Exemplos: Máquinas industriais, robótica.

Sistemas de tempo real

Este é um tipo de SE com restrições de tempo, ou seja, um sistema que responde a eventos externos ou estímulos de entrada em tempo hábil (dentro do tempo finito e especificado).

O requisito de tempo é no sentido da dispobilidade da resposta quando for necessária e dentro de uma deadline



A correção depende não apenas do resultado lógico, mas também a hora em que foi entregue: a falta de resposta é tão ruim quanto a resposta errada!

Sistemas de tempo real - *hard vs. soft*

➢ SISTEMAS DE TEMPO REAL COM "TEMPO CRÍTICO" – *HARD REAL TIME*

- As tarefas de tempo crítico são intolerantes a atrasos, devem ser realizadas dentro de um intervalo preciso de tempo ou haverá falha.
- O sistema de acionamento dos airbags de um carro, por exemplo.
- Nele, alguns milissegundos podem fazer a diferença entre salvar ou não a vida de condutor(a).

➢ SISTEMAS DE TEMPO "QUASE" REAL – *SOFT REAL TIME*

- São mais tolerantes.
- Se a tarefa responsável pelo fechamento da válvula de água de uma máquina de lavar atrasar um ou dois segundos, a roupa será lavada com um pouco mais de água além do necessário, perdendo um pouco da eficiência almejada.

Sistemas de tempo real

➢ OUTRO EXEMPLO: CÂMERA DIGITAL - ALGUMAS RESTRIÇÕES RÍGIDAS

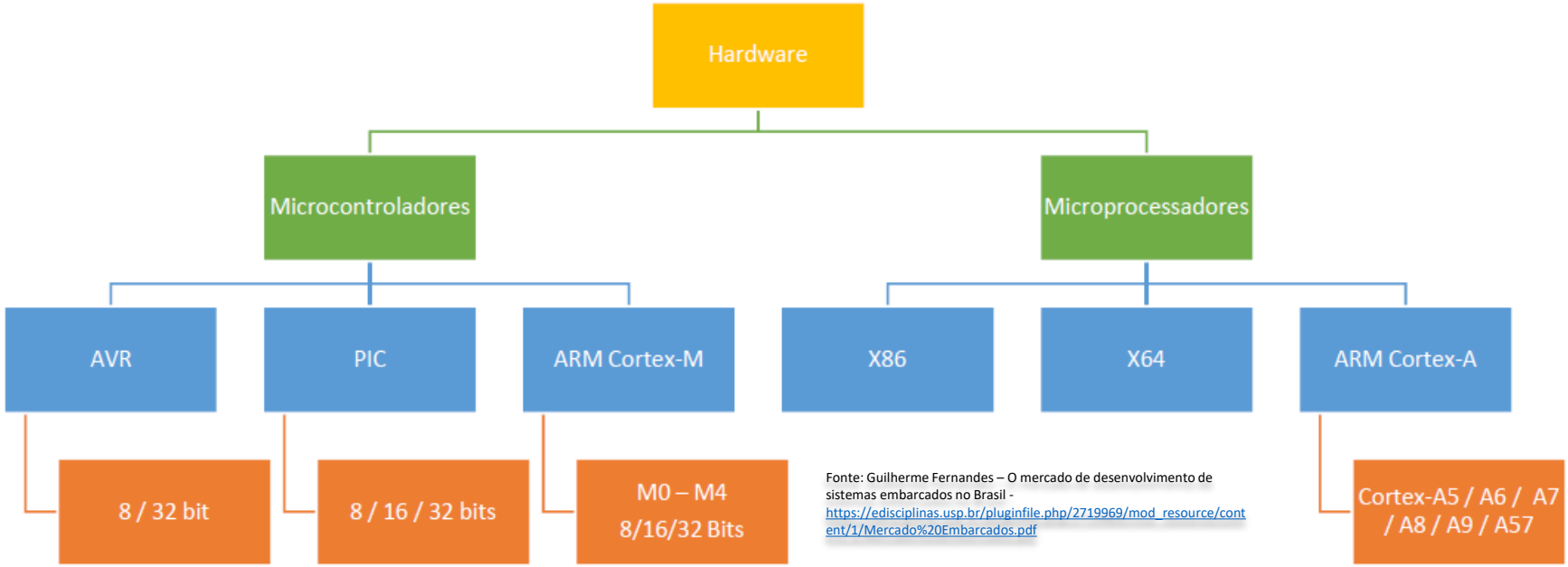
- ✓ **O sistema deve ter um baixo custo para permitir que uma câmara seja acessível a maioria das pessoas, deve ser pequeno para caber numa câmara convencional, deve ser rápido para processar várias imagens em milissegundos e deve consumir pouca potência para uma maior duração da bateria.**

- ✓ **Reflita: este exemplo, particularmente, possui um alto grau de reatividade de tempo real quanto a resposta ao clique do botão ?**

Sistemas de tempo real

- PENSE EM QUAIS SERIAM OUTROS EXEMPLOS DE APLICAÇÕES EM QUE O TEMPO DE RESPOSTA É CRÍTICO ?

Microprocessador ou microcontrolador?



Fonte: Guilherme Fernandes – O mercado de desenvolvimento de sistemas embarcados no Brasil - https://edisciplinas.usp.br/pluginfile.php/2719969/mod_resource/content/1/Mercado%20Embarcados.pdf

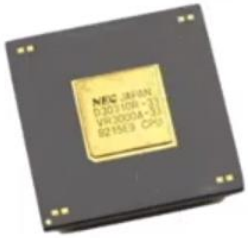
Exemplos de microprocessadores de propósito geral:

Intel Core i9

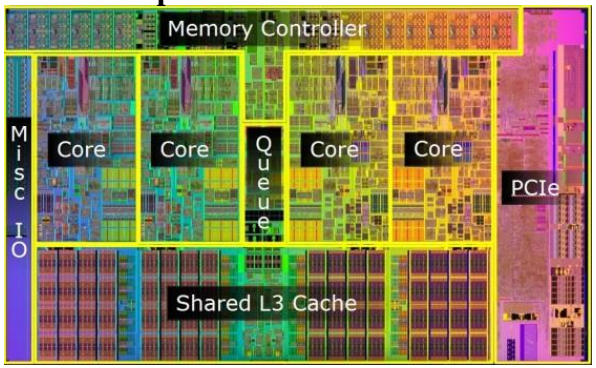
AMD Ryzen

ARM Cortex-A14

MIPS - R3000



Microprocessador Intel core i7

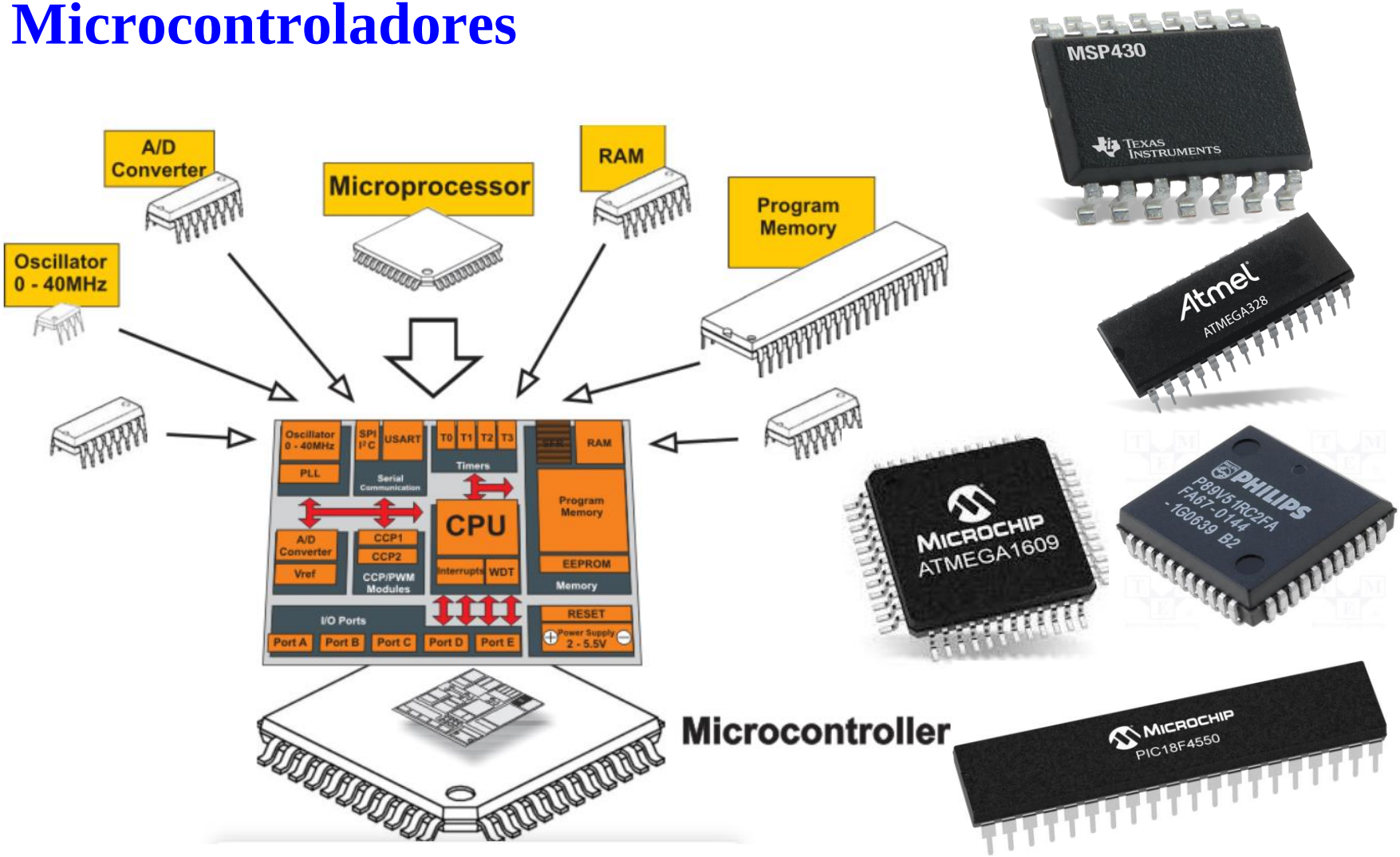


https://www.cs.uaf.edu/2009/fall/cs441/proj1/russell/index.html

Microcontrolador *vs.* Microprocessador

MCU		Microprocessador
Baixo	Custo	Alto
Menor	Capacidade	Maior
Pequeno	Tamanho	Grande
Muitos	Periféricos	Poucos
Versátil	Foco	Velocidade
Limitado	Escalabilidade	Grande
Específico	Aplicação	Geral

Microcontroladores

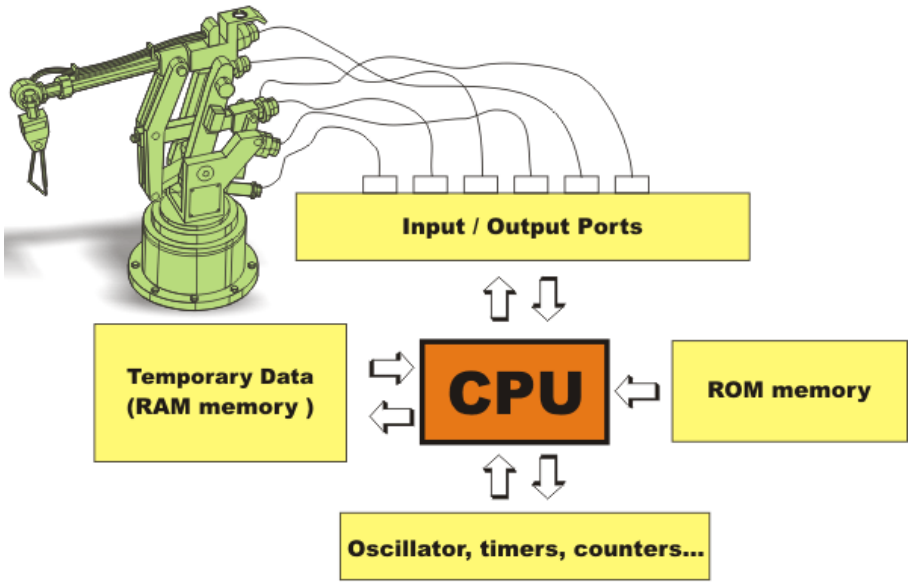


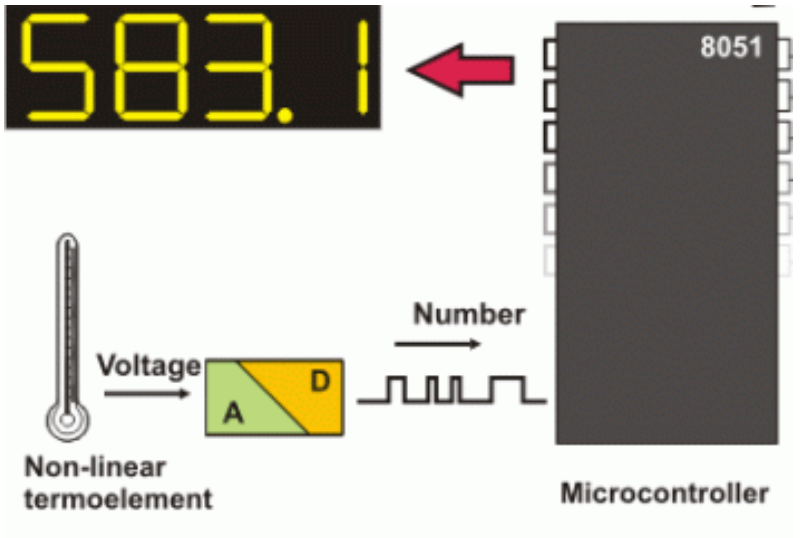
Fonte https://www.mikroe.com

Microcontroladores: Microchip, Atmel, Texas, Philips

Microcontroladores – Sistemas embarcados de pequena e média escala

- É um circuito integrado que contém **núcleo de um processador, memória e periféricos programáveis de entrada e saída** para controlar totalmente ou algumas funções de um dispositivo eletrônico.
- Apresentam menor desempenho e foco em maior economia e baixo custo em relação ao **microprocessadores** e são destinados **a aplicações embarcadas**.



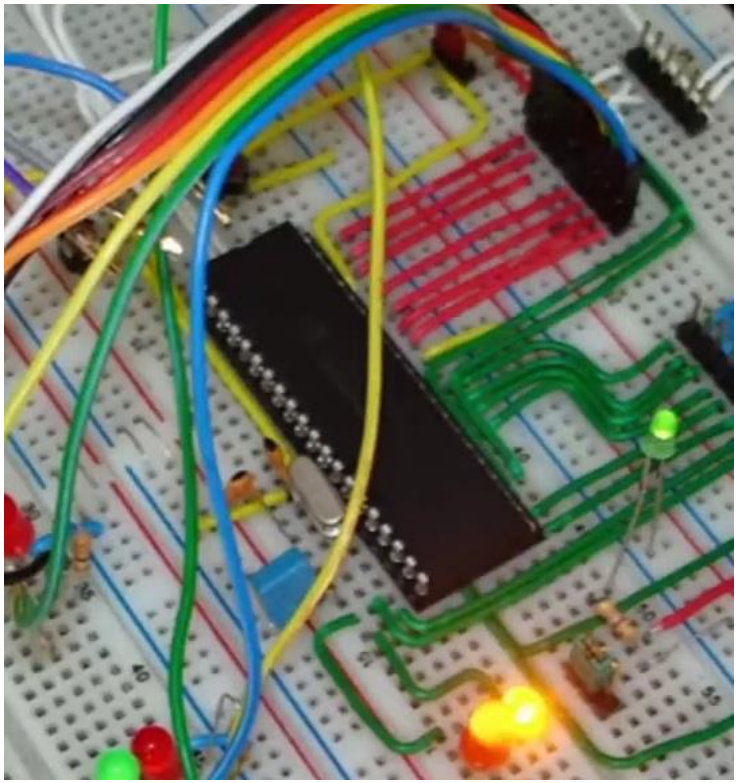


Fonte https://www.mikroe.com

Microcontroladores – Funcionamento

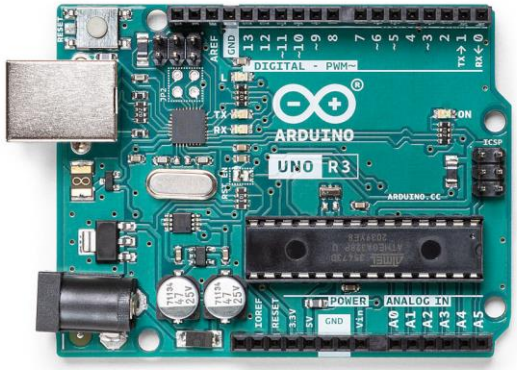
- **Microcontrolador;**
- **CI conversor de nível serial;**
- **Cristal oscilador;**
- **Capacitores;**
- **Regulador de tensão;**
- **LEDs indicadores;**
- **Botão com resitor**

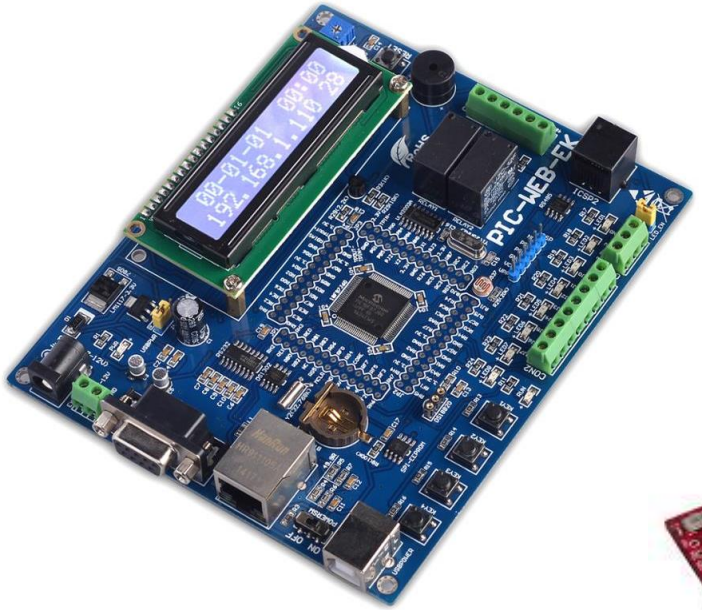
Solução para protótipos: plataforma de desenvolvimento com MCU e outros periféricos e circuitos necessários já embarcados

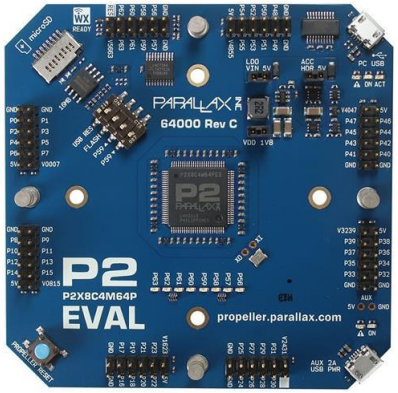


Microcontroladores – kits de desenvolvimento

Além dos circuitos mínimos para o funcionamento do microcontrolador, permitem simular vários projetos





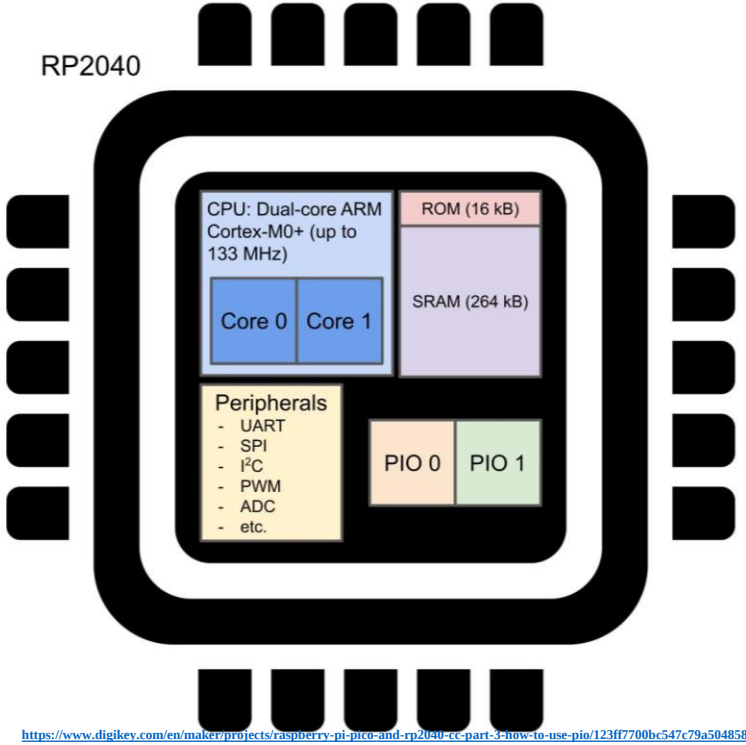


Fonte https://www.mikroe.com

Kits de desenvolvimento: PIC, Arduino, Parallax, Texas

Microcontroladores de 32 bits

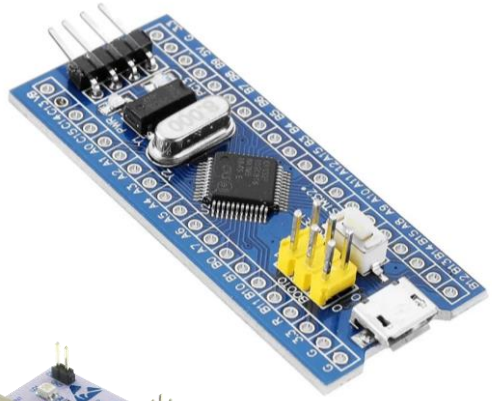
- **Microntroladores com CPU ARM Cortex-M:** família de 32 bits RISC licenciadas pela ARM: **Cortex M0, M1, M3 e M4**. Processadores de alto desempenho, suporta recursos para RTOS, 2 núcleos etc. Exemplos: **STM32** (ARM Cortex-M3); **RP2040** (ARM Cortex M0).

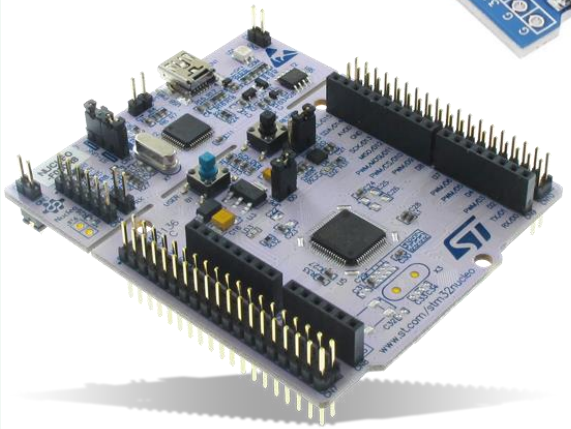


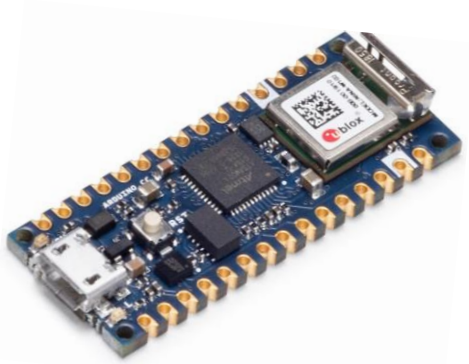
https://www.digikey.com/en/maker/projects/raspberry-pi-pico-and-rp2040-cc-part-3-how-to-use-pio/123ff7700bc547c79a504858c1bd8110

Microcontroladores modernos -kits de desenvolvimento

- **Blue Pill/Nucleo ST; Raspberry Pi Pico, Arduino nano 33 IoT, NXP Freedom/Kinetis**



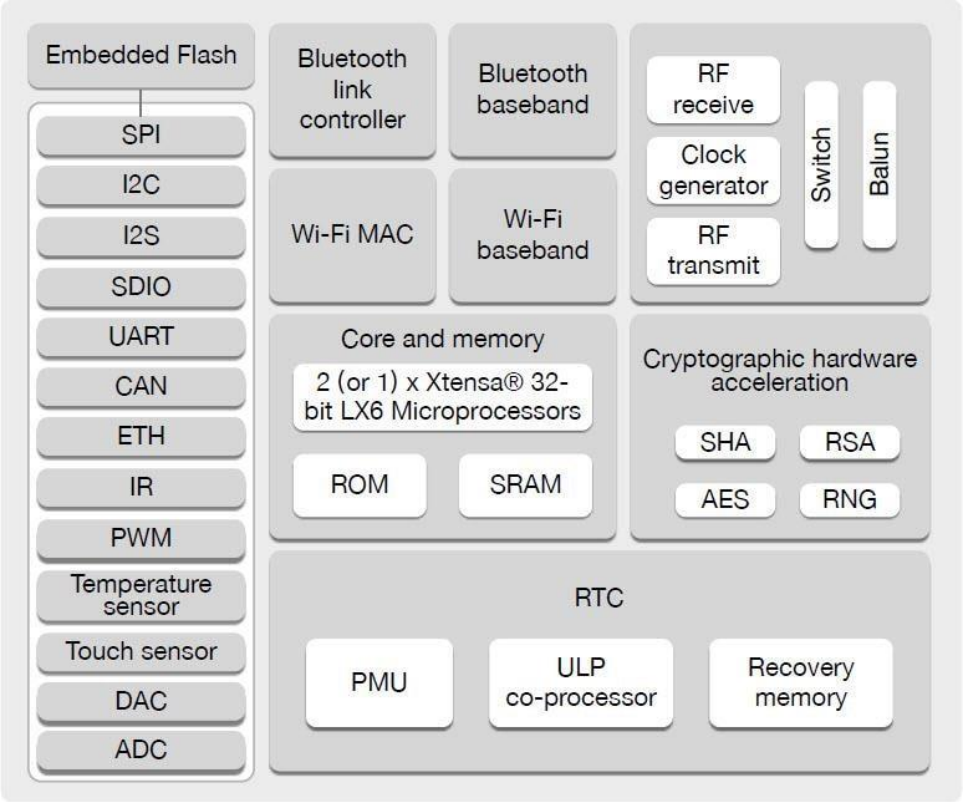


Fonte:Raspberry Pi Foundation - https://www.raspberrypi.com/documentation/microcontrollers/rp2040.html

Microcontroladores em SoC

- **Micontroladores em SoC (system on chip):** ESP32 32 bits RISC da empresa Espressif. Comporta em único chip além das caraterísticas de um MCU de 32 bits de 2 núcleos, Wi-Fi, Bluetooth, criptografia de dados, radiofrequência etc.

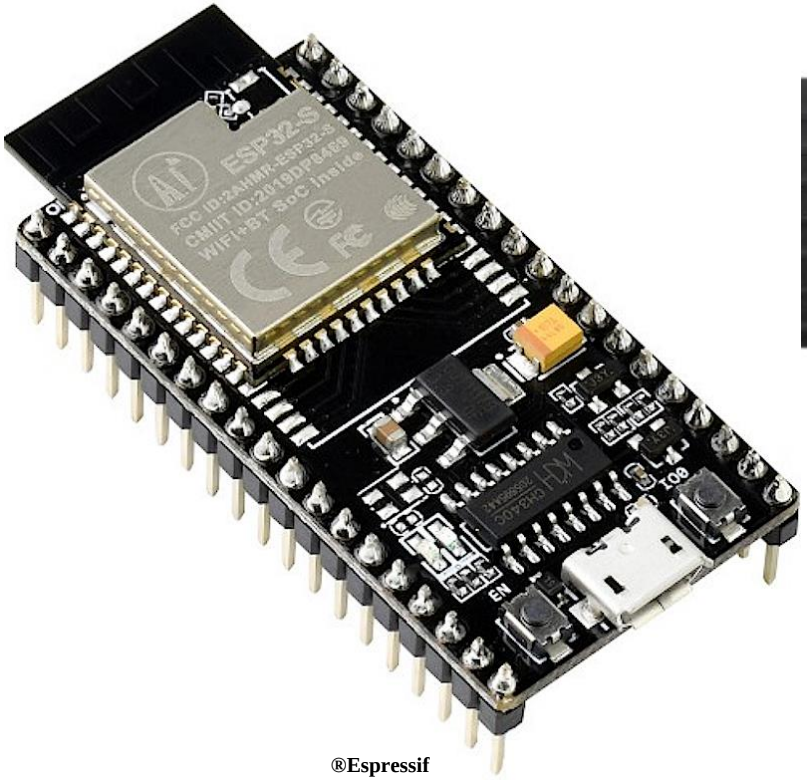
®Expressif



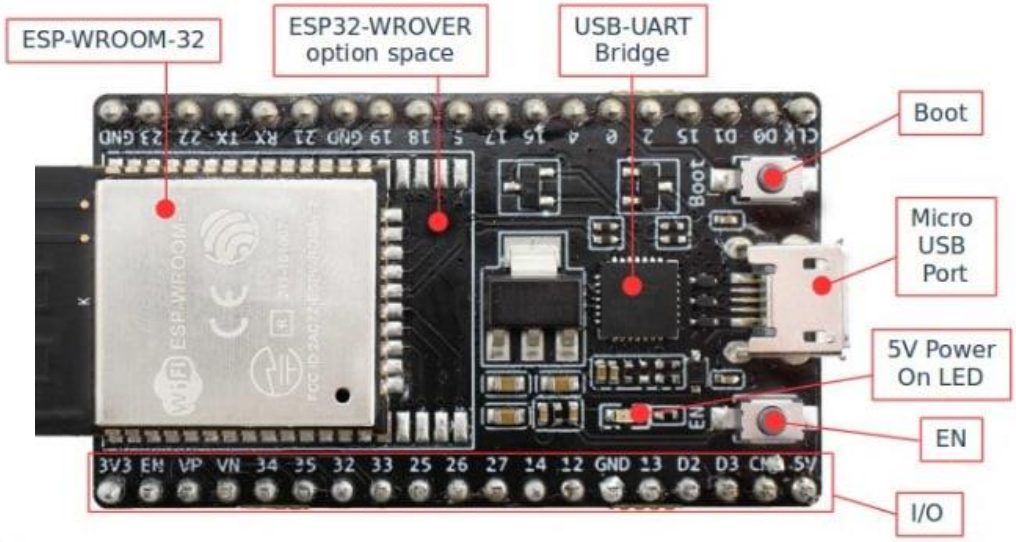
https://br.mouser.com/images/marketingid/2017/microsites/108753931/Espressif_ESP32_Diagram.jpg

Microcontroladores em SoC -kits de desenvolvimento

Módulo ESP32: plataforma com o SoC de 32 bits Dual-Core Wi-Fi e bluetooth– Espressif



®Espressif



Fabricantes e famílias

- ✓ **Alguns dos principais:** Microchip, NXP, Samsung, ST, Texas Instruments, Intel, Panasonic, Parallax, Raspberry Pi Foundation, Rockwell, Sony, Toshiba, Phillips, Siemens, Atmel.

MICROCHIP ATMEL

NXP freescale semiconductor

intel ALTERA

ESPRESSIF

TEXAS INSTRUMENTS

8-bit Microcontrollers
- AVR, PIC
- HCS12
- 8051 family etc.

Internal bus is 8 bit. ALU performs operations on 8 bit (1 Byte)

6 Bit Microcontrollers
- Extended 8051XA
- Intel 8096,
- MC68HC12 etc.

Internal bus is 16 bit. ALU performs operations on 16 bit (2 Byte or word), Greater precision than 8 bit

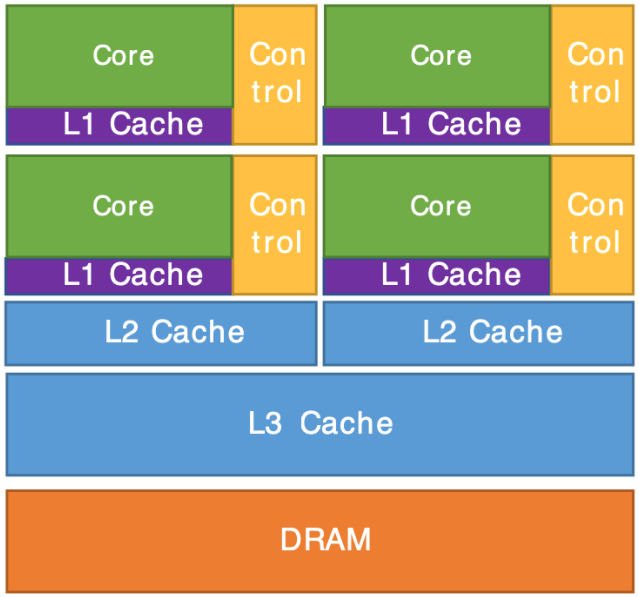
32-bit microcontrollers
- ARM
- PIC32
- Intel 80960, Intel/Atmel 251 family

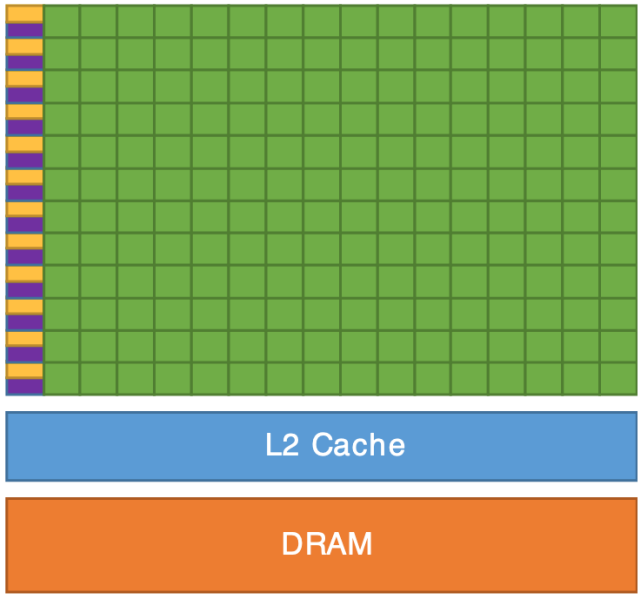
Internal bus is 32 bit. ALU performs operations on 32 bit (4 Byte or word), Greater precision than 16 bit

Sistemas embarcados sofisticados

- Processadores de alto desempenho, multi núcleos (**4, 8, 16**), processamento gráfico, (**GPU**) aplicações em telas gráficas (**ARM Cortex A**), **32/64 bits**, e que rodam **sistema operacional**.

NXP BROADCOM Toradex TEXAS INSTRUMENTS

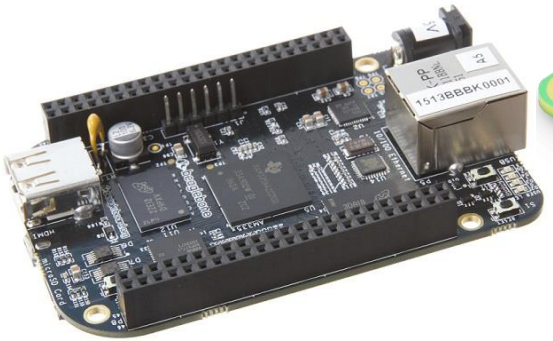


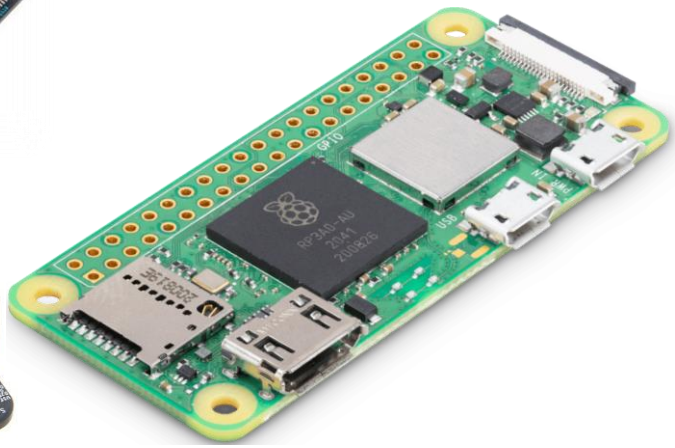


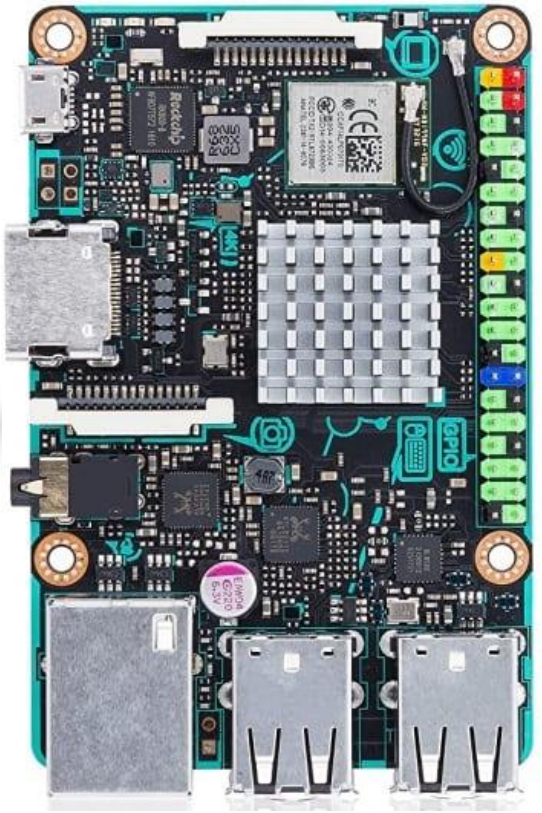
https://cvw.cac.cornell.edu/GPUarch/gpu_characteristics

SBCs – single board computers

- Combinação de processador, memória, entrada e saída e outros recursos (Wi-Fi, Bluetooth, etc.) em uma única placa (bastando conectar teclado, mouse, e monitor para se obter um PC, por ex.).







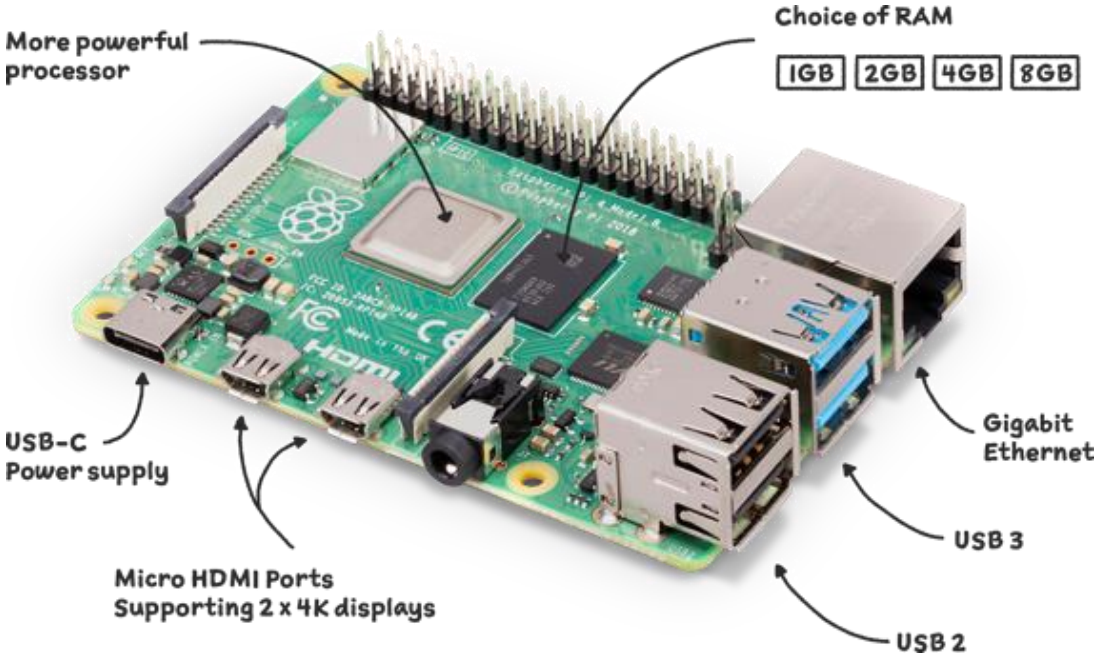
Raspberry Pi

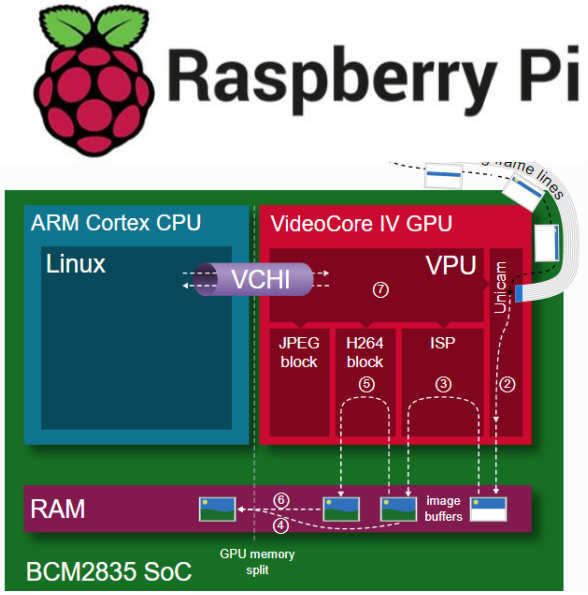
A proposta é a redução de custo, solução integrada, tamanho reduzido, flexibilidade e facilidade.

É uma plataforma embarcada pronta para uso!

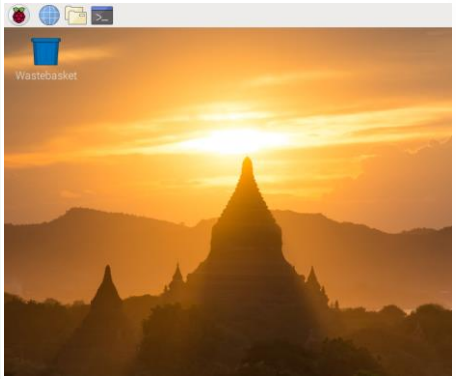
Envolve SoC (por vezes com microprocessador ARM)

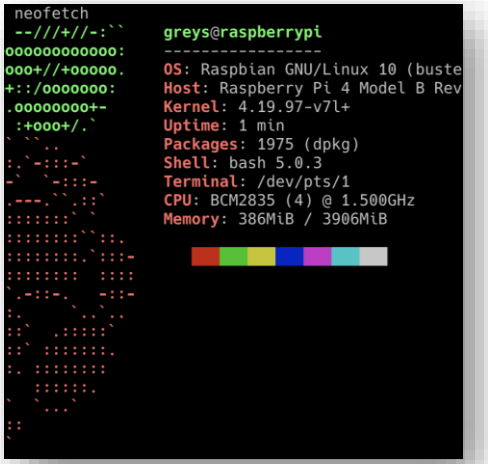
Exemplo: O projeto Raspberry Pi lançou em 2012 o primeiro SBC de tamanho reduzido visando promover computação!

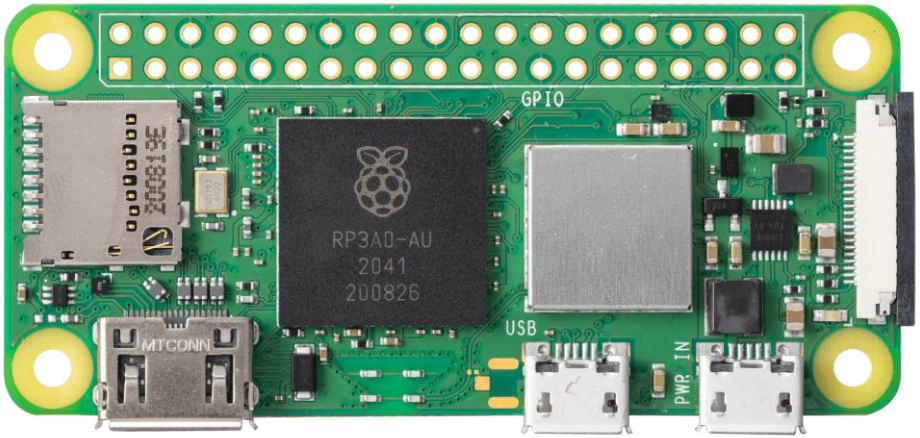


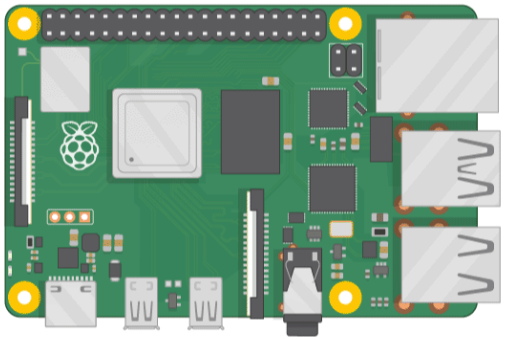


Raspberry Pi







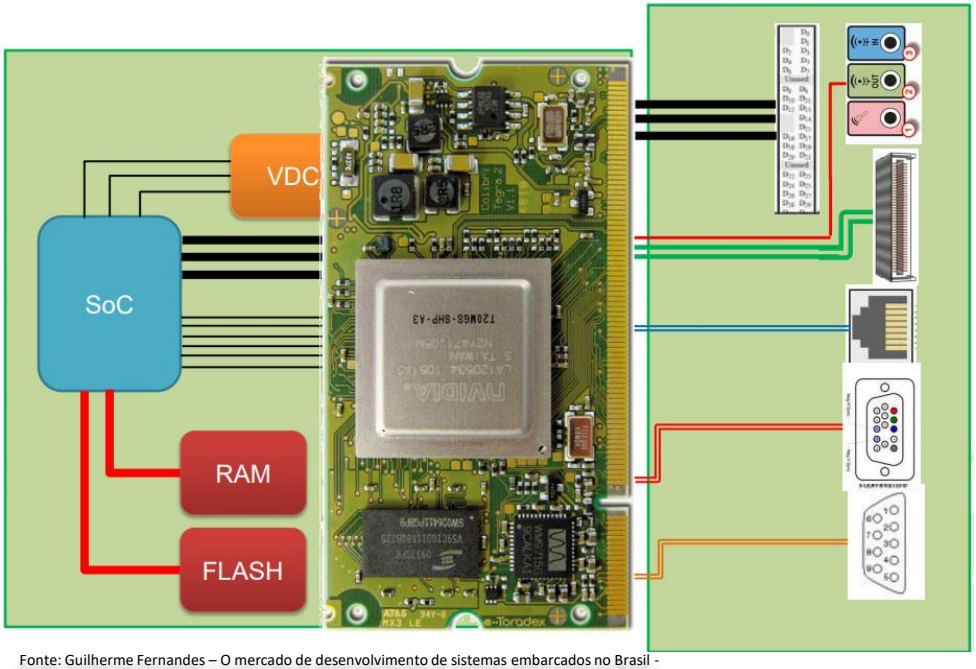


CoM – *Computer on Module*

Situa-se entre uma SBC e um Microcontrolador!

Solução computacional complexa (core)

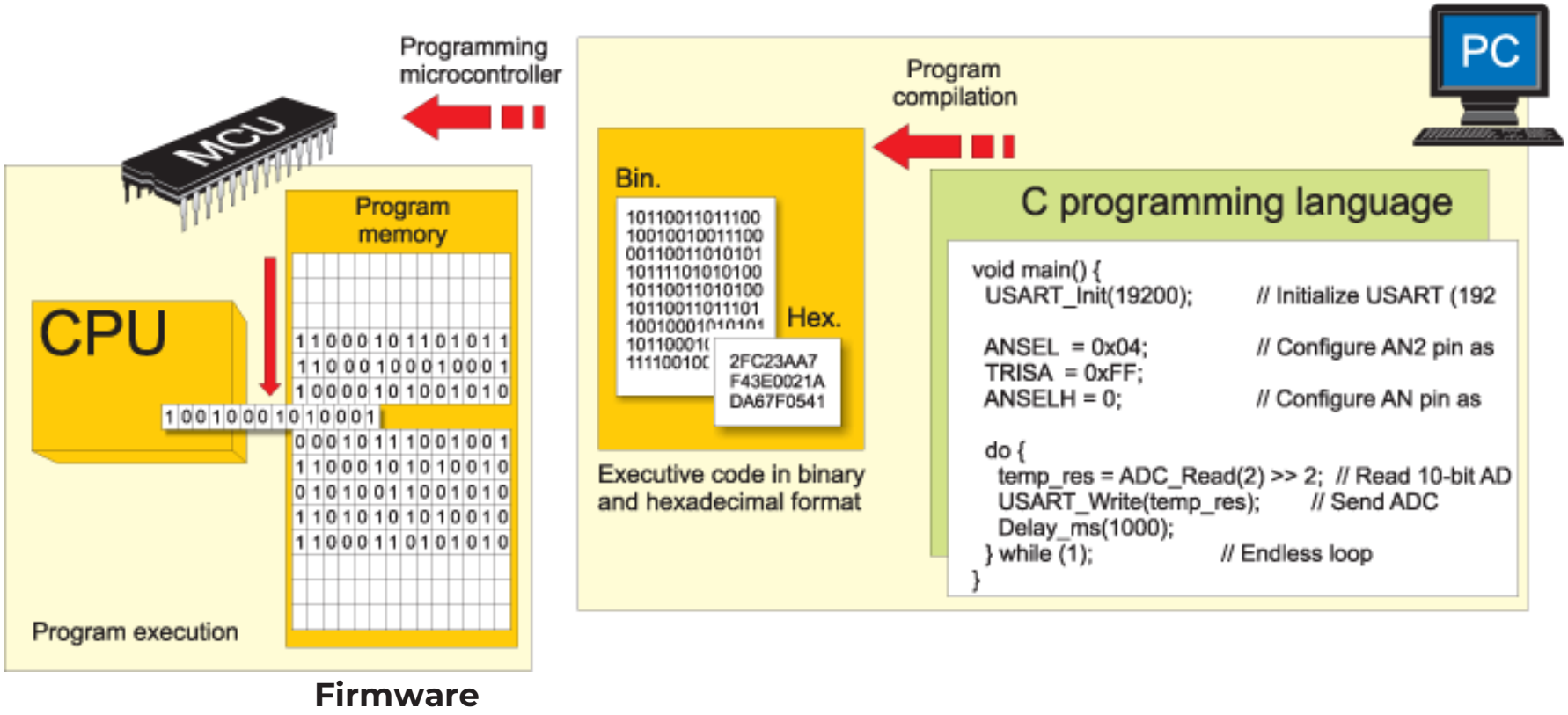
Usada com a placa base específica para um produto (conectores, interfaces)



Fonte: Guilherme Fernandes – O mercado de desenvolvimento de sistemas embarcados no Brasil - https://edisciplinas.usp.br/pluginfile.php/2719969/mod_resource/content/1/Mercado%20Embarcados.pdf

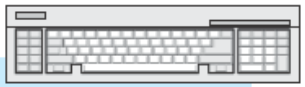
Programação do microcontrolador

- ✓ **Etapas:** I escrita do programa em linguagem de alto nível; II compilação e geração do arquivo hex; III gravação do código bin na ROM do MCU



M. Verle -PIC Microcontrollers Programming in C – Mikroe https://www.mikroe.com/ebooks/pic-microcontrollers-programming-in-c

Programação "*bare-metal*"



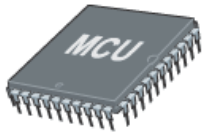
```
void main() {

   TRISB = 0;              // All port B pins are configured as
                           // outputs
   PORTB = 0b01010101;     // Logic state on port B pins
}
```

Program written in C

```
:100000000428FF3FFF3FFF3F03138316860155304F
:10001000831286000A28FF3FFF3FFF3FFF3FFF3F5D
:04400E00F22FFFFF8F
:00000001FF
```

Executable Code of the program (HEX code)




```
MOV RØ, #Ø4h
```

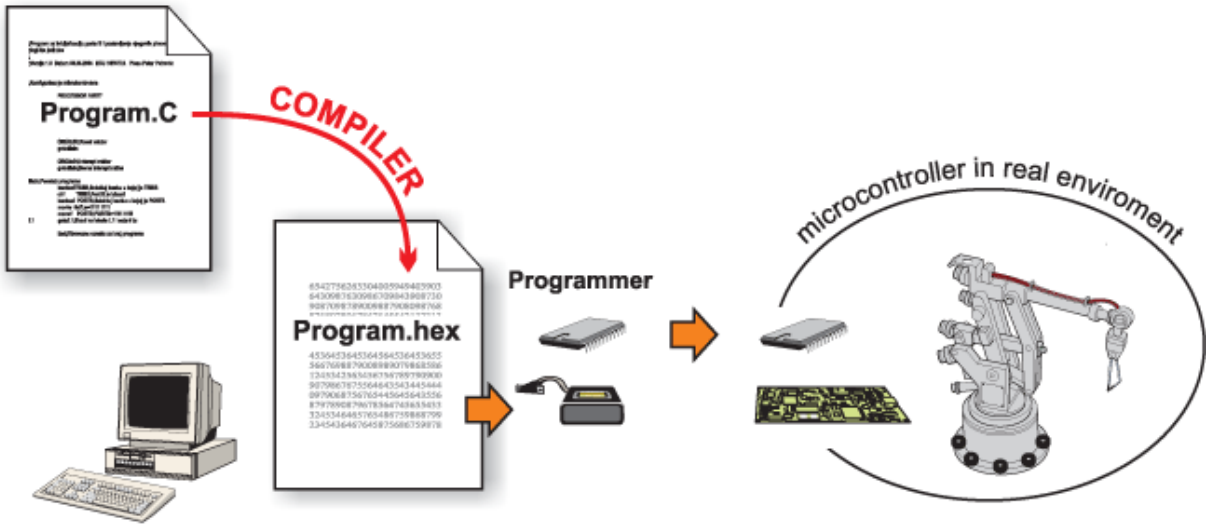
Code Memory				
	0	1	2	3
00	78	04	00	00
10	00	00	00	00

0111 1000 – 0000 0100

Primeiro computador
ALTAIR 8800 1975

Linguagens de programação

- ✓ **Alto nível:** programas mais transparentes e sem conhecer o HW (hardware); MicroPython, C++...
- ✓ **Baixo nível:** opera, acessa e manipula registradores, endereços de memória e dispositivos I/O (exige conhecimento do HW) – Assembly / linguagem de máquina;
- ✓ **Médio nível:** características dos dois níveis acima para agregar flexibilidade. – Linguagem C



M. Verle -PIC Microcontrollers Programming in C – Mikroe https://www.mikroe.com/ebooks/pic-microcontrollers-programming-in-c

Sistema operacional

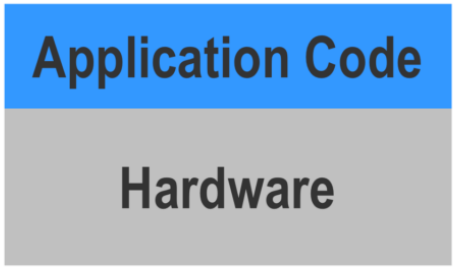
➢ POR QUE PRECISA DE S.O.?

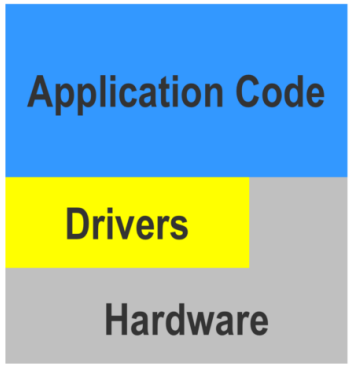
- **O Sistema Operacional fornece uma interface de usuário para interação com o hardware.**
- **Sistemas avançados como set-top boxes, que podem ser reproduzidos em 1000 canais, podem gravar vídeos de alta definição e suportar mídia de dispositivos externos como USB, definitivamente requerem S.O. e sistema embarcado sofisticado (microprocessador, CPU multicore, GPU etc).**

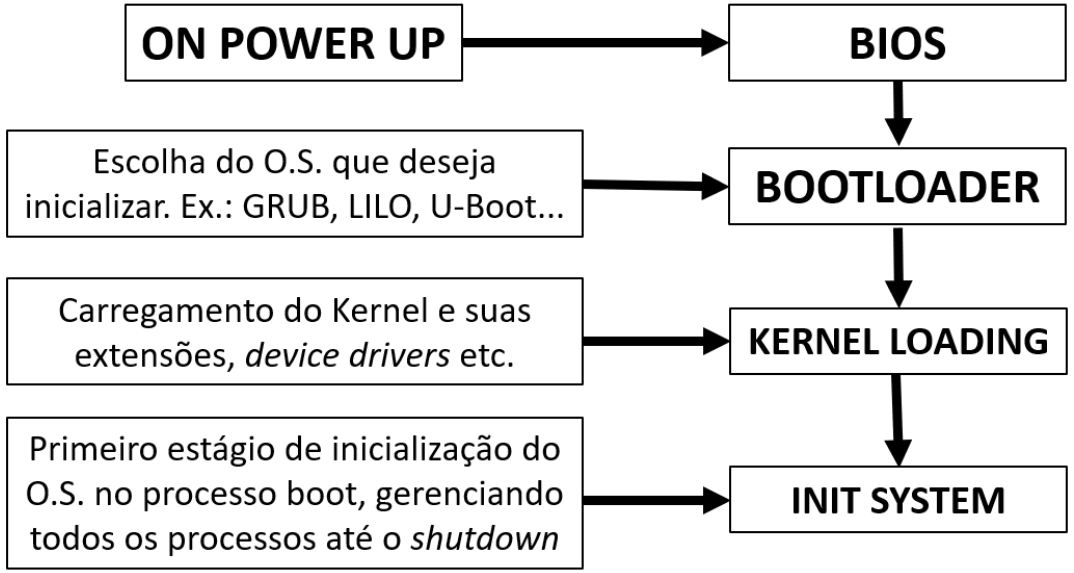
Linux

Sistema embarcado com S.O.

➢ FIRMWARE VS. S.O.







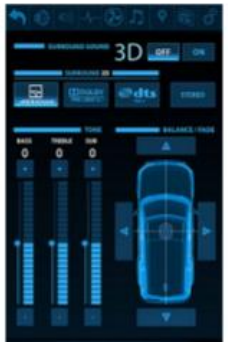
BOOT	RASPBERRY (rootfs)		
FAT32	EXT4		
• bootcode.bin • start.elf • kernel7.img • fixup.dat • config.txt • cmdline.txt • *.dtb	• /bin • /boot • /dev • /etc • /home • /lib	• /media • /mnt • /opt • /proc • /root • /run	• /sbin • /srv • /sys • /tmp • /usr • /var

Linux embarcado








Linux embarcado

- SO (kernel)
- Detectar e preparar o hardware
- Gerenciar a memória
- Fornecer as interfaces gráficas para o usuário
- Proporcionar o acesso e autenticação de usuário
- Oferecer os utilitários administrativos
- Virtualização
- Computação em tempo real

GNOME™

THE LINUX FOUNDATION

Linux sob a licença GNU GPL

GPL Free Software
Free as in Freedom

https://www.gnu.org/

fedora

CentOS

SUSE

debian

Red Hat

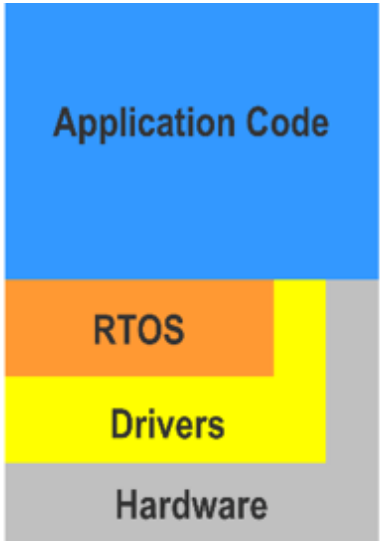
Sistema operacional de tempo real

➢ SISTEMA OPERACIONAL EM MICROCONTROLADORES

- Os sistemas embarcados possuem grandes limitações, se comparados com computadores que utilizamos no cotidiano.
- Portanto, **não podemos utilizar o mesmo SO que comumente empregamos em computadores convencionais**.

- **Em aplicações embarcadas, a solução é quase sempre um Sistema Operacional de Tempo Real (RTOS – *Real Time Operating System*)**
- **RTOS difere de um SO convencional no quesito tempo.**

freeRTOS

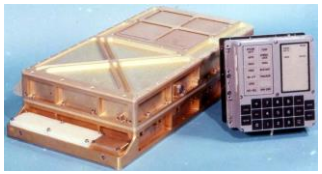


2 – Mercado de sistemas embarcados

Primeiro grande sistema embarcado vs. Sistema computacional moderno

Apollo Guidance Computer:

- ✓ Operava em tempo real, era considerado o item mais arriscado do projeto (uso de CIs monolíticos elevou tal risco)
- ✓ 4K bytes de ROM e 256 bytes de RAM
- ✓ Clock de ~2 MHZ
- ✓ 11 instruções e 16 bits. Aprox. 5 mil CIs, **RTL**



Apple MAC Studio M1 Ultra

- ✓ Chip M1 Ultra da Apple com CPU de 20 núcleos, GPU de 48 núcleos e Neural Engine de 32 núcleos
- ✓ Memória unificada de 128 GB
- ✓ SSD de 8 TB

Mercado de Sistemas Embarcados

RELATÓRIO DE PESQUISA – EMPRESA EMBARCADOS

- Relatório de pesquisa de responsabilidade da empresa Embarcados sobre o mercado brasileiro de Sistemas Embarcados e IoT em 2021.

Descreve particularidades do mercado brasileiro, o processo de desenvolvimento, tecnologias e fabricantes utilizados em projetos de SE e de profissionais que atuam na área.

Fonte: https://embarcados.com.br/relatorio-da-pesquisa-sobre-o-mercado-brasileiro-de-sistemas-embarcados-e-iot-2021/

Mercado Brasileiro de Sistemas Embarcados e IoT

RELATÓRIO DE PESQUISA – MERCADO BRASILEIRO 2021 – EMPRESA EMBARCADOS

Ferramentas para Sistema Embarcados e IoT

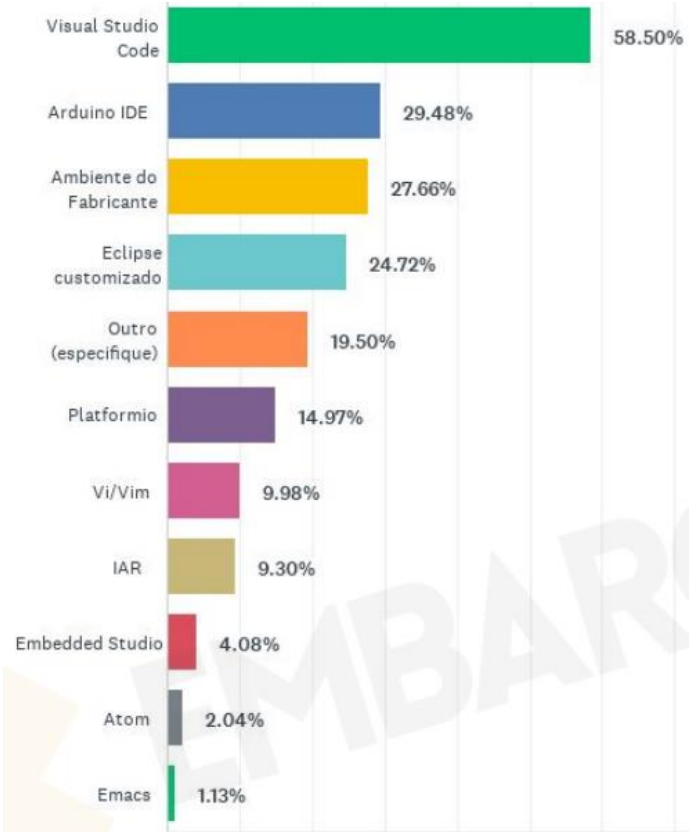
- IoT (48.70%) e Sistemas Industriais (37.17%) são as principais áreas de aplicação dos projetos desenvolvidos atualmente.
- Comunicação sem fio (75.22%), alimentação por bateria (62.61%) e Capacidade de Tempo Real (54.57%) são os recursos mais utilizados em projetos de sistemas embarcados.
- A principal comunicação sem fio utilizada atualmente é Wi-Fi (56.74%), seguido por Bluetooth LE/Smart (36.96%) e 3G/4G (34.78%).
- Kits de desenvolvimento fornecidos pelos fabricantes do processador ou módulo (50.11%) são as ferramentas de prototipação rápida preferida pelos profissionais, seguido de kits com ESP8266/ESP32 (46.49%), Arduino (46.03%) e Raspberry Pi (39.00%). No entanto, a maioria não incorpora kits ou placas de prototipação em seus produtos.

Fonte: https://embarcados.com.br/relatorio-da-pesquisa-sobre-o-mercado-brasileiro-de-sistemas-embarcados-e-iot-2021/

Mercado Brasileiro de Sistemas Embarcados e IoT

RELATÓRIO DE PESQUISA – MERCADO BRASILEIRO 2021 – EMPRESA EMBARCADOS

- IDEs e linguagens de programação mais usadas



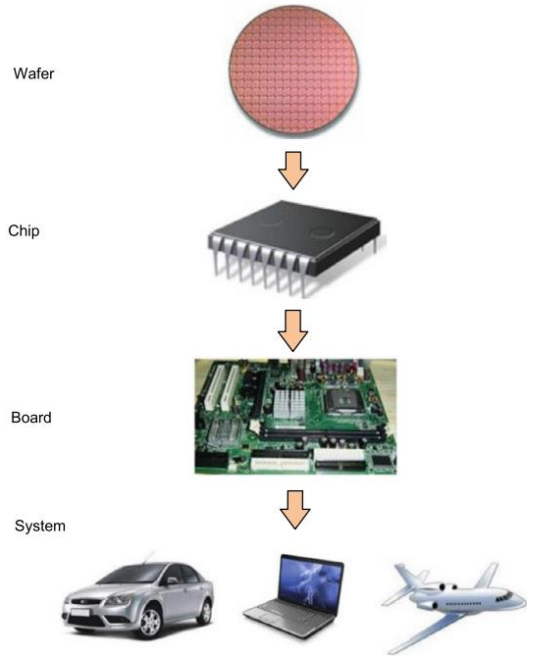
OPÇÕES DE RESPOSTA	RESPOSTAS
C	74.72%
C++	45.79%
C/C++(Arduino)	30.07%
Python	28.47%
JavaScript	9.79%
C#	7.74%
Java	7.52%
.NET	5.69%
MicroPython	5.69%
Linguagem Assembly	5.47%
MATLAB	5.01%
Outra (especifique)	3.64%
VHDL	3.42%
LabVIEW	2.96%
Verilog	2.05%
CircuitPython	1.59%
Go	1.59%
Rust	1.14%
SystemVerilog	0.68%
Ada	0.23%

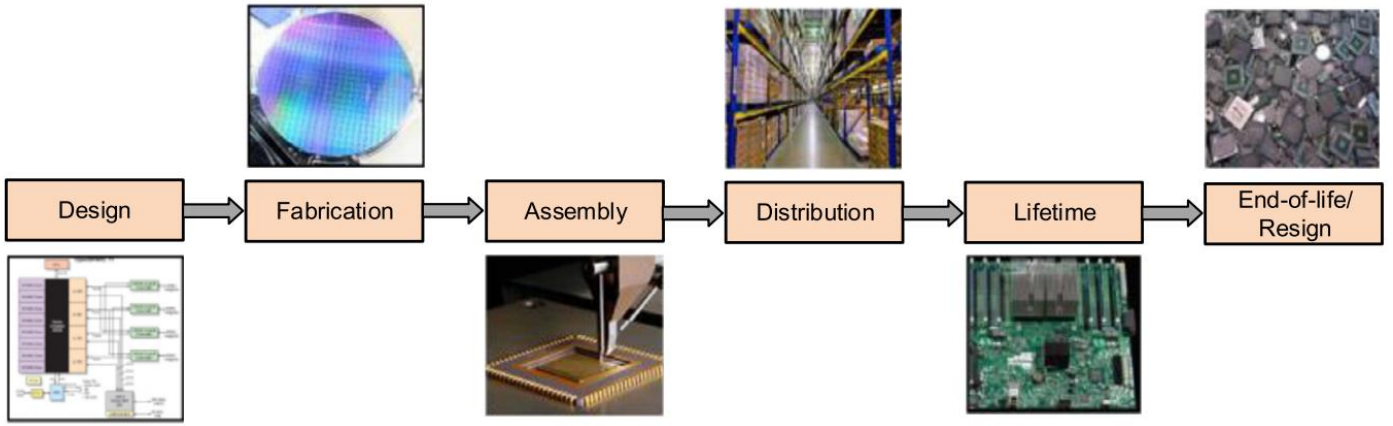
Fonte: https://embarcados.com.br/relatorio-da-pesquisa-sobre-o-mercado-brasileiro-de-sistemas-embarcados-e-iot-2021/

3 – Projeto de sistemas embarcados para produtos

Projeto de sistemas embarcados

- **QUAIS PERGUNTAS DEVEM SER FEITAS?**
  - ✓ **Tipo de hardware e software necessário?**
  - ✓ **Arquitetura, velocidade, requisito de tempo, software otimizado**
  - ✓ **Tamanho dos dados?**
  - ✓ **Quantidade de Memória necessária?**
  - ✓ **Como reduzir consumo de energia?**





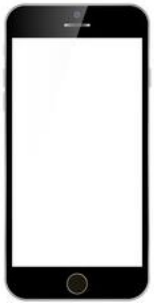
Fonte: **Hardware Security.** https://doi.org/10.1016/B978-0-12-812477-2.00008-3 - **Elsevier2019**

Projeto de sistemas embarcados

➢ ASPECTOS ENVOLVIDOS NO PROJETO

- ✓ **Requisitos:** Confiabilidade; Preço acessível; Segurança; Escalabilidade; Vida útil
- ✓ **Ciclo de vida:** Requisitos; Fabricação; Distribuição; Logística; Suporte
- ✓ **Sistemas embarcados envolvem conceitos multidisciplinares:** Hardware eletrônico; circuito, layout PCB, prototipagem, Software, programação e algoritmo de controle, Estrutura mecânica, prototipagem mecânica,; Interação humana etc.

- ✓ Ciclo de vida de um smartphone é de aprox. 2 anos, porém, milhares de unidades são vendidas, o que viabiliza a produção do zero do sistema embarcado, totalmente customizado.
- ✓ Ciclo de vida de uma tela gráfica de interface homem-máquina é de apróx. 10 anos, e a quantidade vendida é bem menor em relação a um smartphone, o que justifica talvez usar partes do sistema embarcado (Ex. plataforma embarcada, computador em módulo) e produzir a placa base (conectores) específicos .

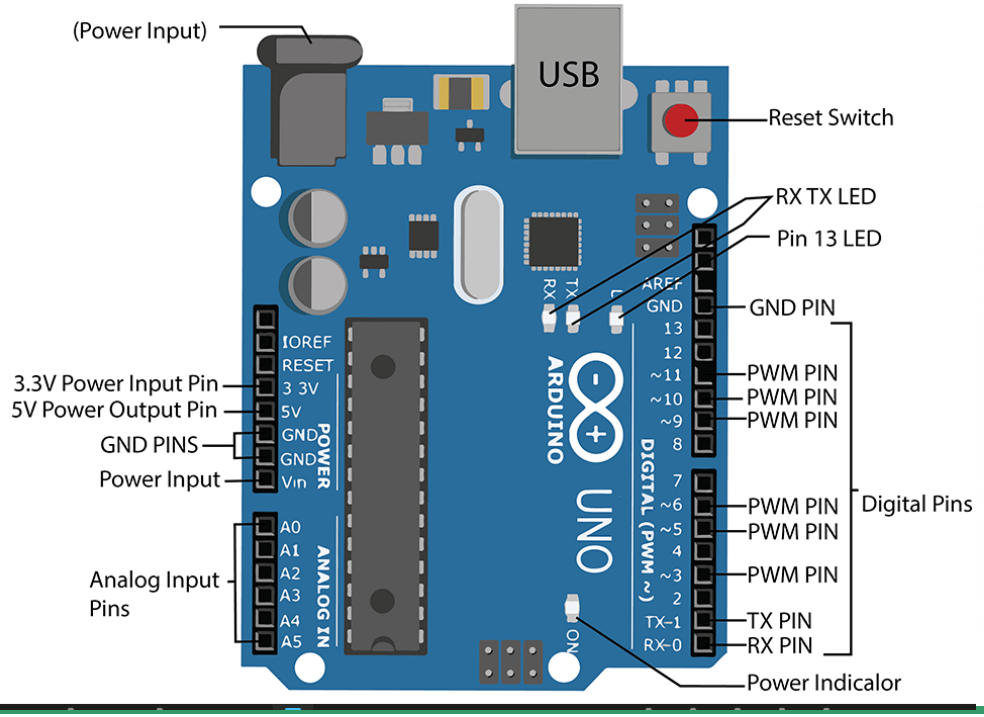




4 – Soluções e sistemas embarcados no curso

Qual plataforma usar para projetos da disciplina?

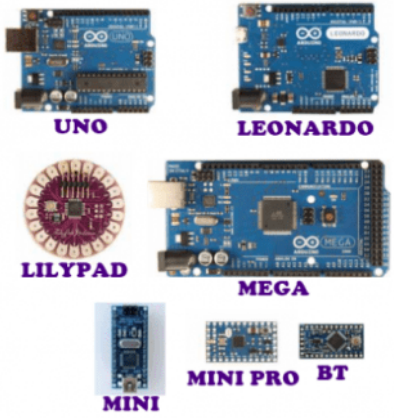
- **Plataforma Arduino: baixo custo, periféricos, IDE, fácil programação. Maior comunidade.**

Arduino: Plataforma de prototipagem eletrônica que advém de uma comunidade (https://www.arduino.cc) que visa permitir o desenvolvimento de controle de sistemas interativos, de baixo custo e acessível a diferentes públicos. IDE amigável



Arduino IDE

- Arduino Uno
- Arduino Leonardo
- Arduino LilyPad
- Arduino Mega
- Arduino Nano
- Arduino Mini
- Arduino Mini Pro
- Arduino BT



Qual plataforma usar para projetos da disciplina?

- **Exemplos**

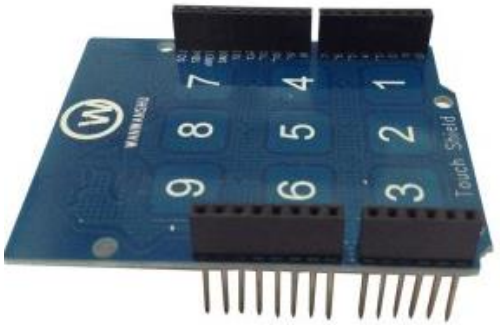
Ethernet

Relé

Protoboard

Motor

LCD



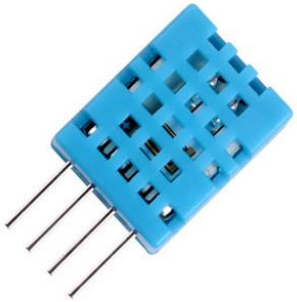
Teclado capacitivo

Qual plataforma usar para projetos da disciplina?

- **Exemplos**

Detector de fumaça

Joystick



Sensor de temperatura

Sensor de movimento

TAG RF

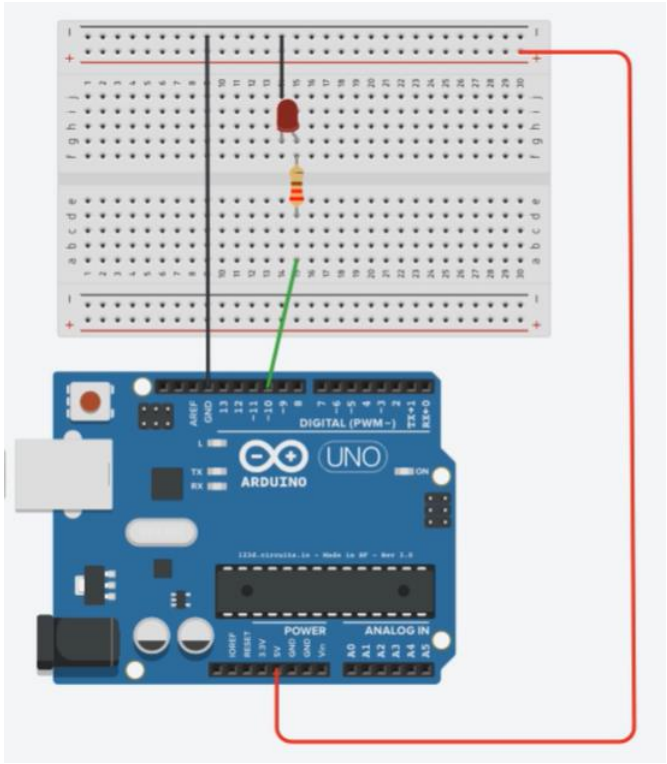
Ultrassonico

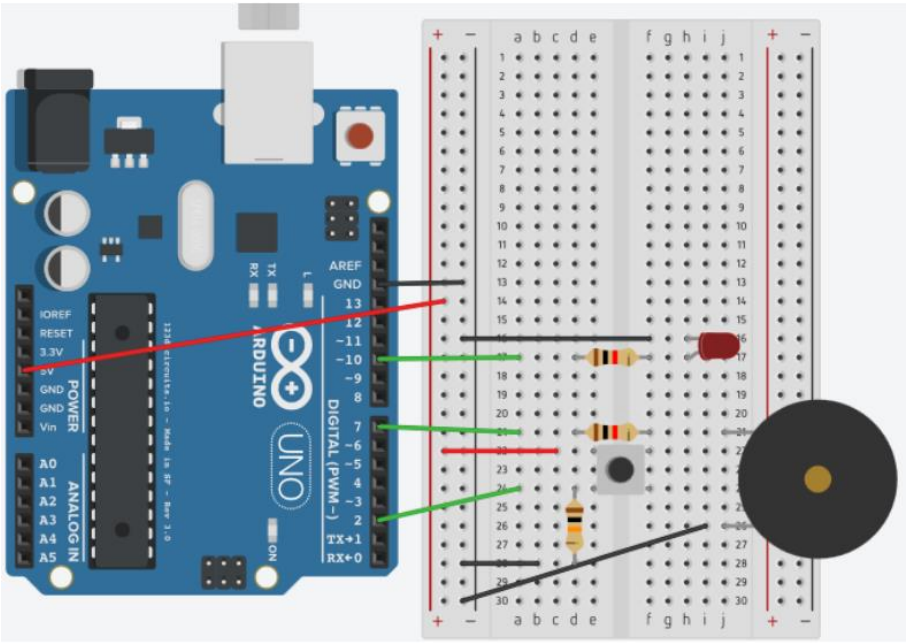
Umidade

Qual plataforma usar para projetos da disciplina?

- **Alguns exemplos de projetos com Arduino**

Piscar LED, projeto com botão, LED e Buzzer

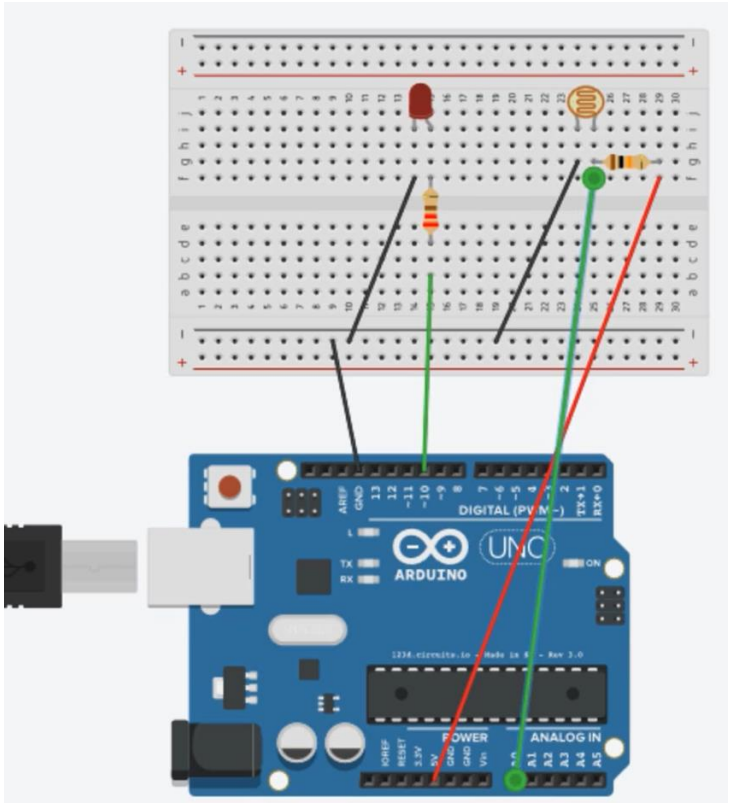


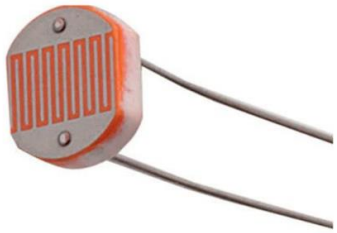


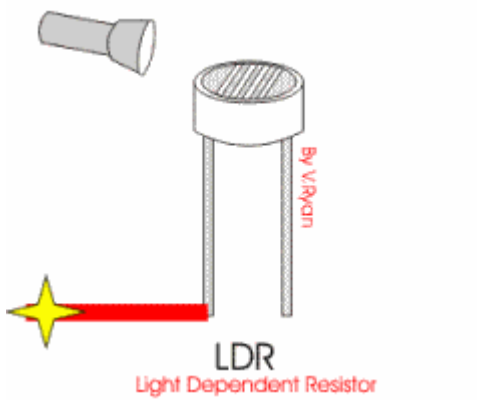
Qual plataforma usar para projetos da disciplina?

- Alguns exemplos de projetos com Arduino

Projeto com sensor de luminosidade LDR



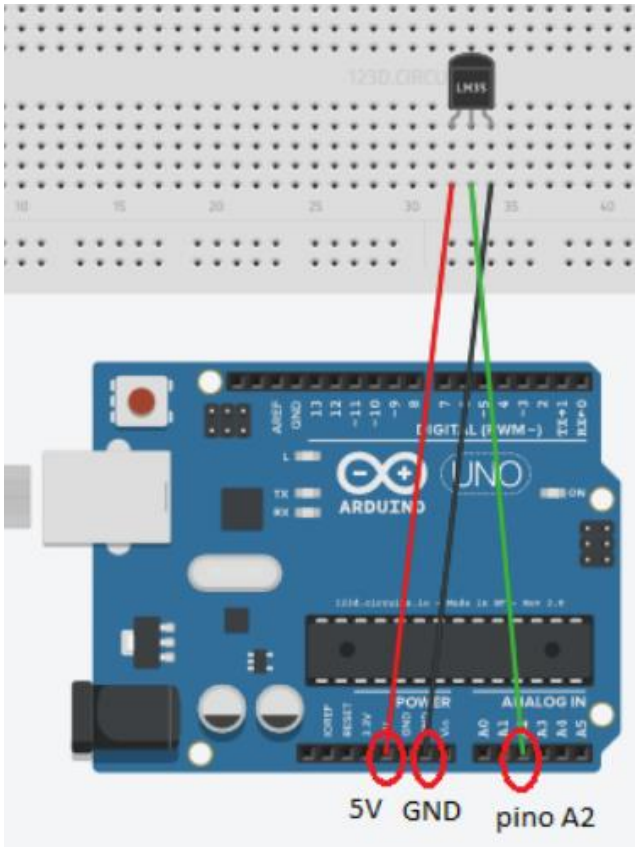


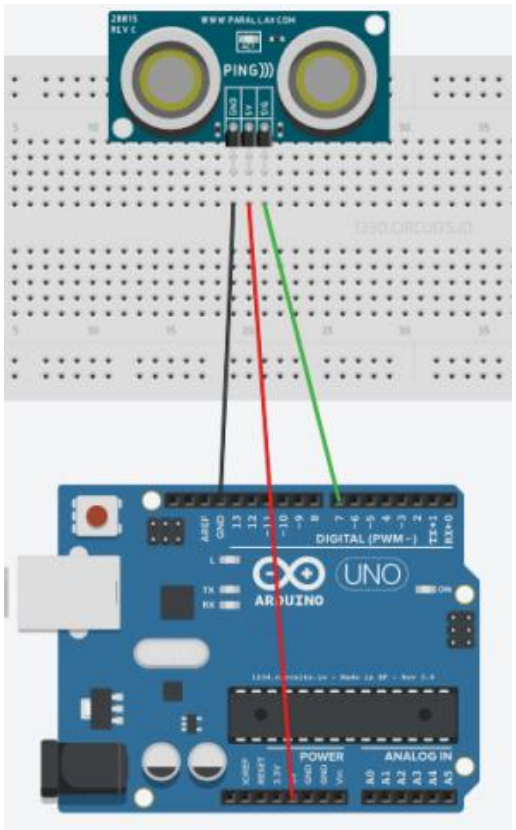


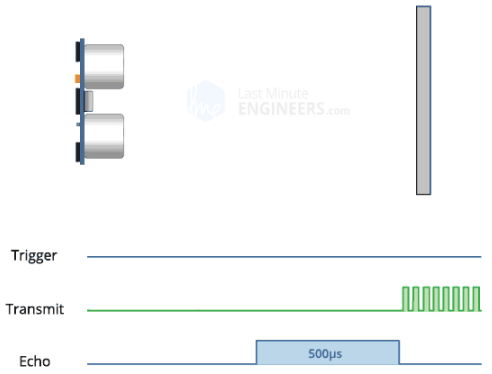
Qual plataforma usar para projetos da disciplina?

- **Alguns exemplos de projetos com Arduino**

Projeto com sensor de temperatura LM35 e sensor ultrassonico

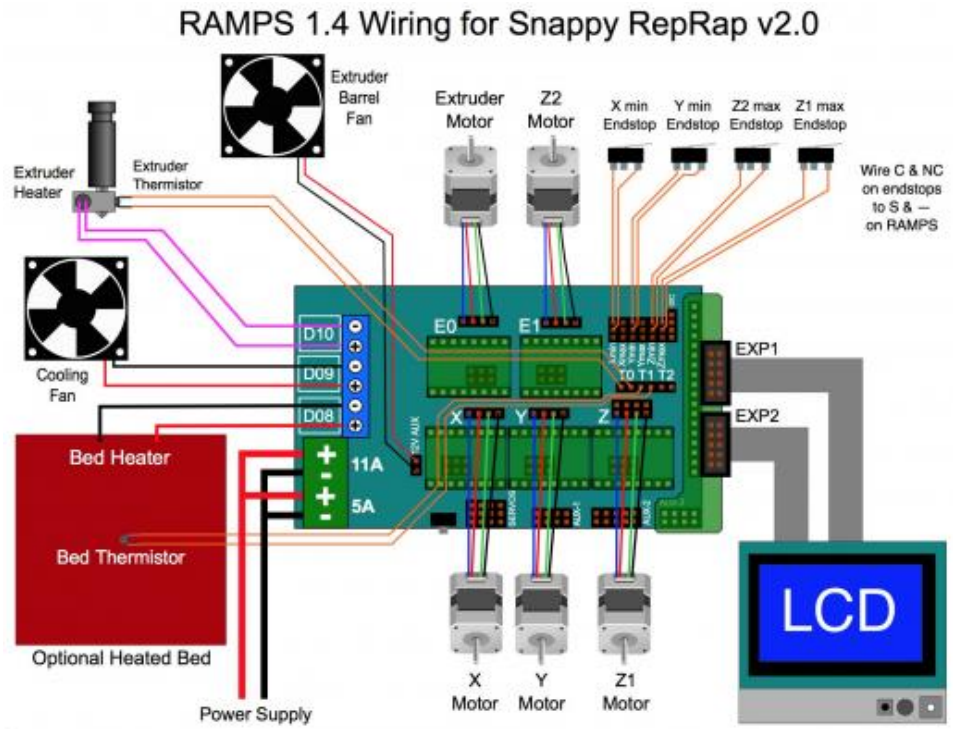


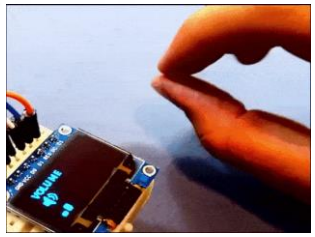


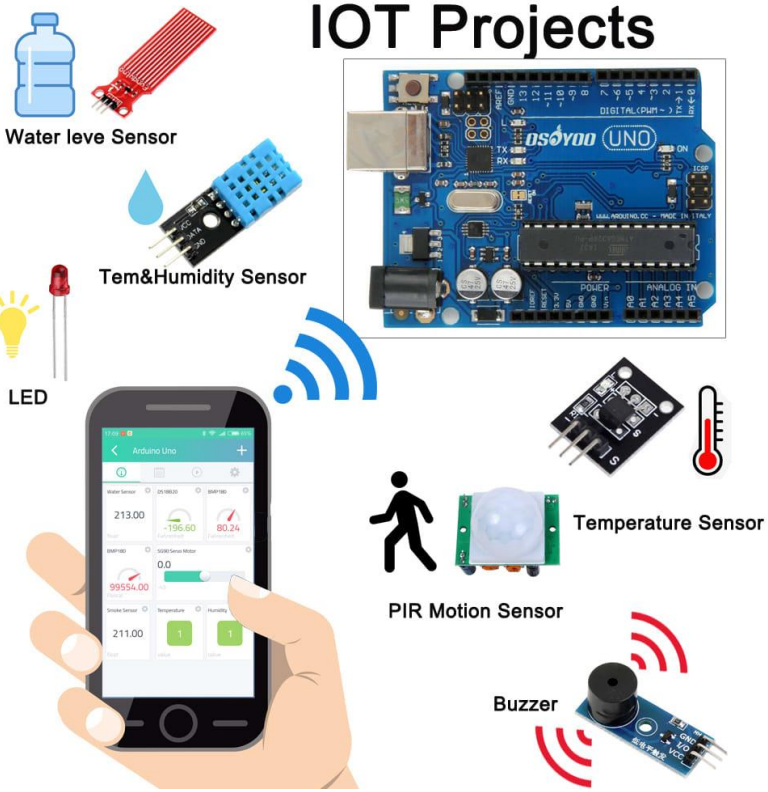


Qual plataforma usar para projetos da disciplina?

- **E uma infinidade de outras possiblidades**



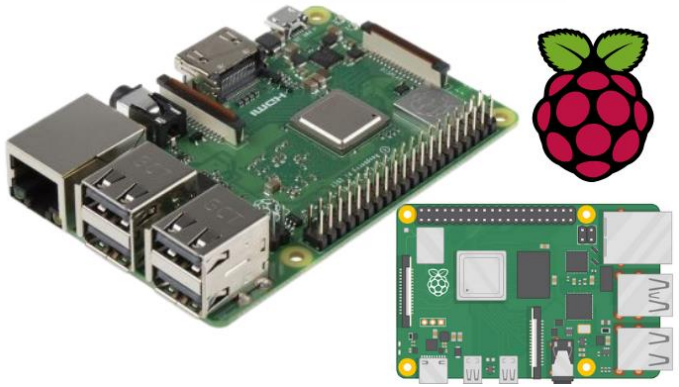




Qual plataforma usar para projetos da disciplina?

- **Raspberry Pi: Sistema operacional, processamento de imagem (câmera), programação em alto nível (bibliotecas Python)(comunidade)**

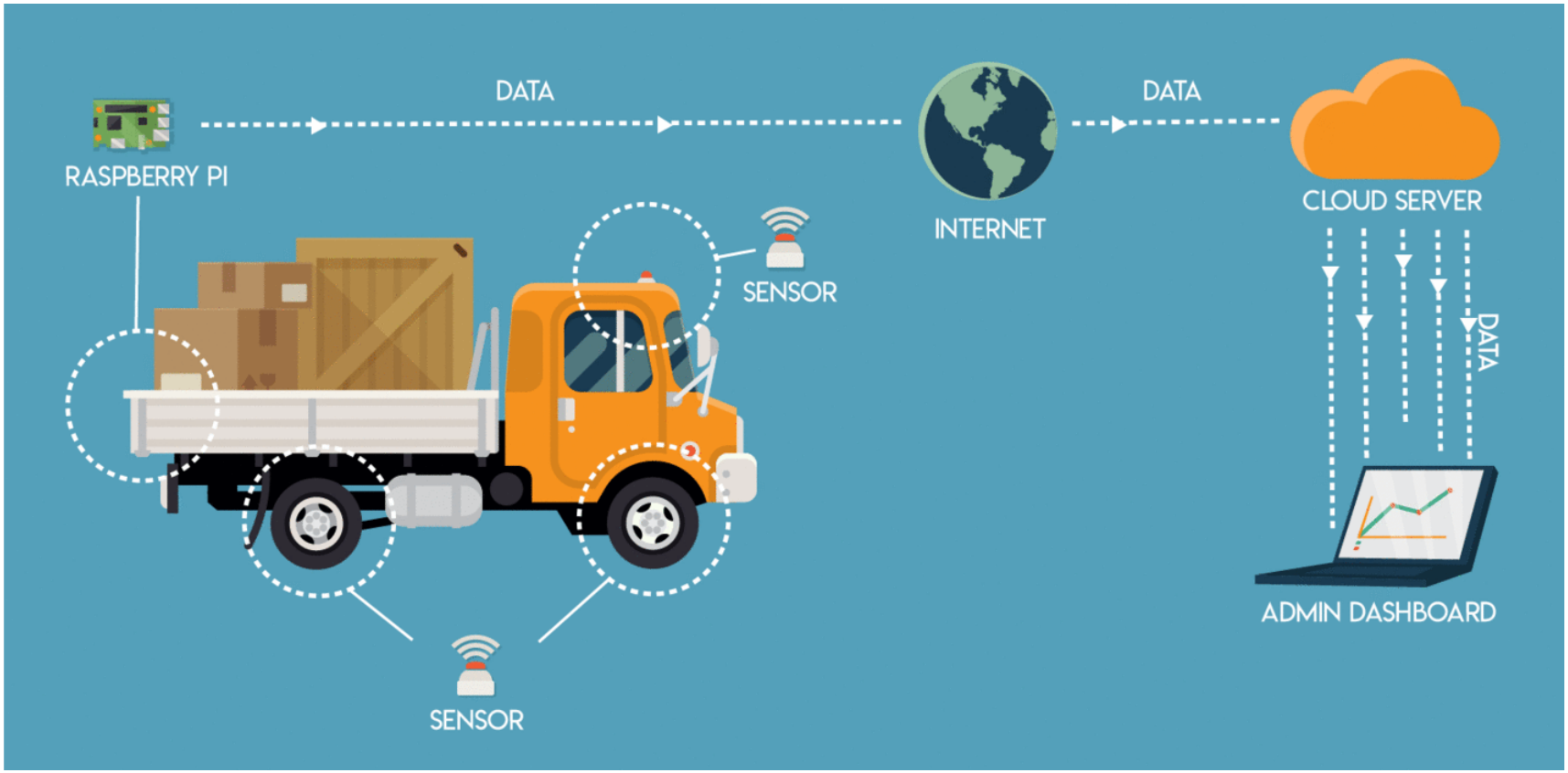
- Ponto de acesso wireless
- Sistema de monitoramento com câmera e reconhecimento de voz;
- Media Center com reprodução de músicas, vídeos e fotos;
- Estação meteorológica com envio dos dados pela internet;
- Console de videogame “retrô”;
- Servidor de Arquivos / File Server;
- Web Server



Qual plataforma usar para projetos da disciplina?

- Quando usar a **Raspberry Pi**

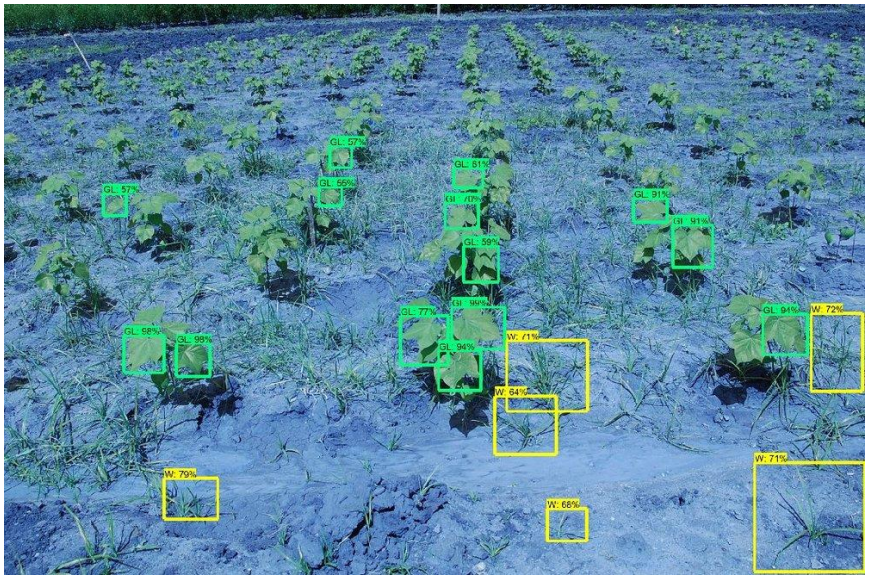
Gateway de dados para monitoramento de veículos de carga

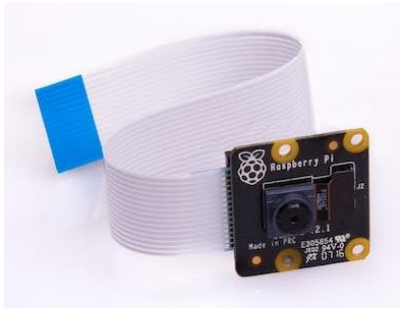


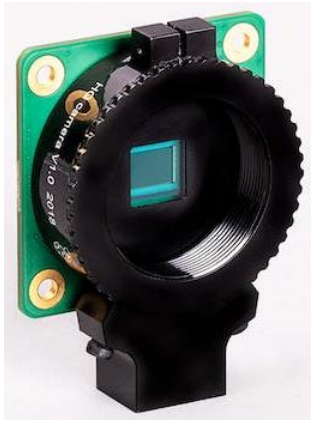
Qual plataforma usar para projetos da disciplina?

- Quando usar a **Raspberry Pi**

Visão computacional/reconhecimento facial

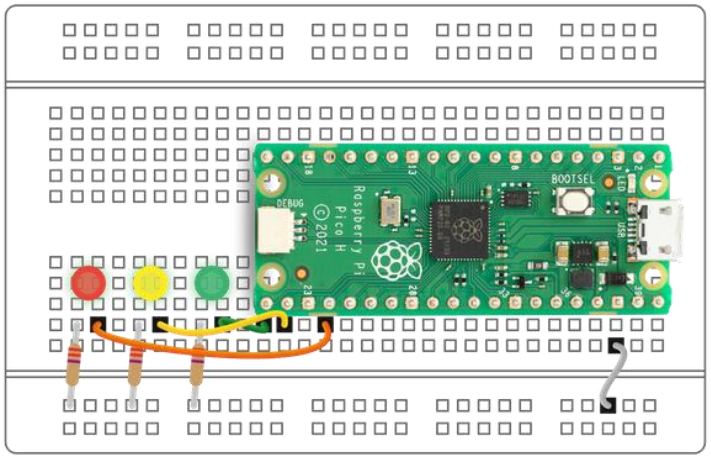


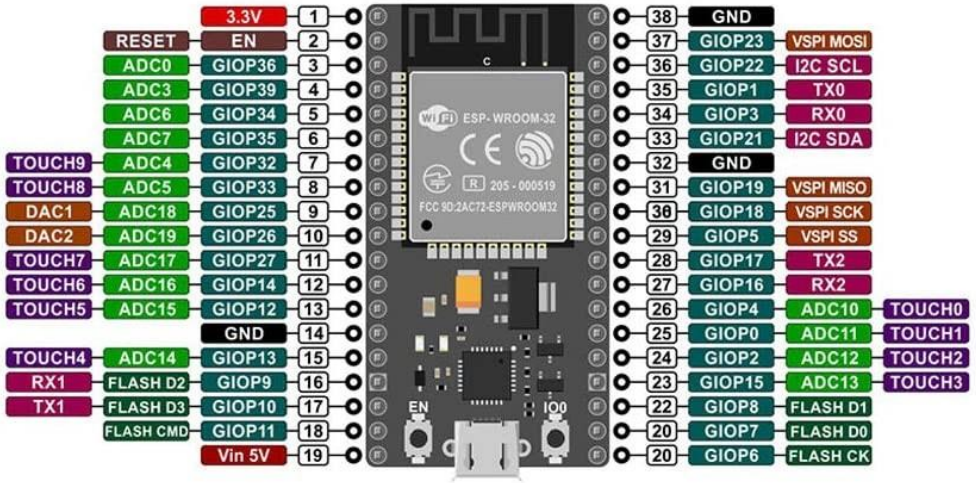




Qual plataforma usar para projetos da disciplina?

- **Microcontroladores de 32 bits – demanda por alta resolução em conversor A/D, Wi-Fi, Bluetooth, câmera (baixa capacidade) , tempo real – ESP32, Raspberry Pico (IDEs intuitivas como Arduino IDE, comunidade Raspberry)**

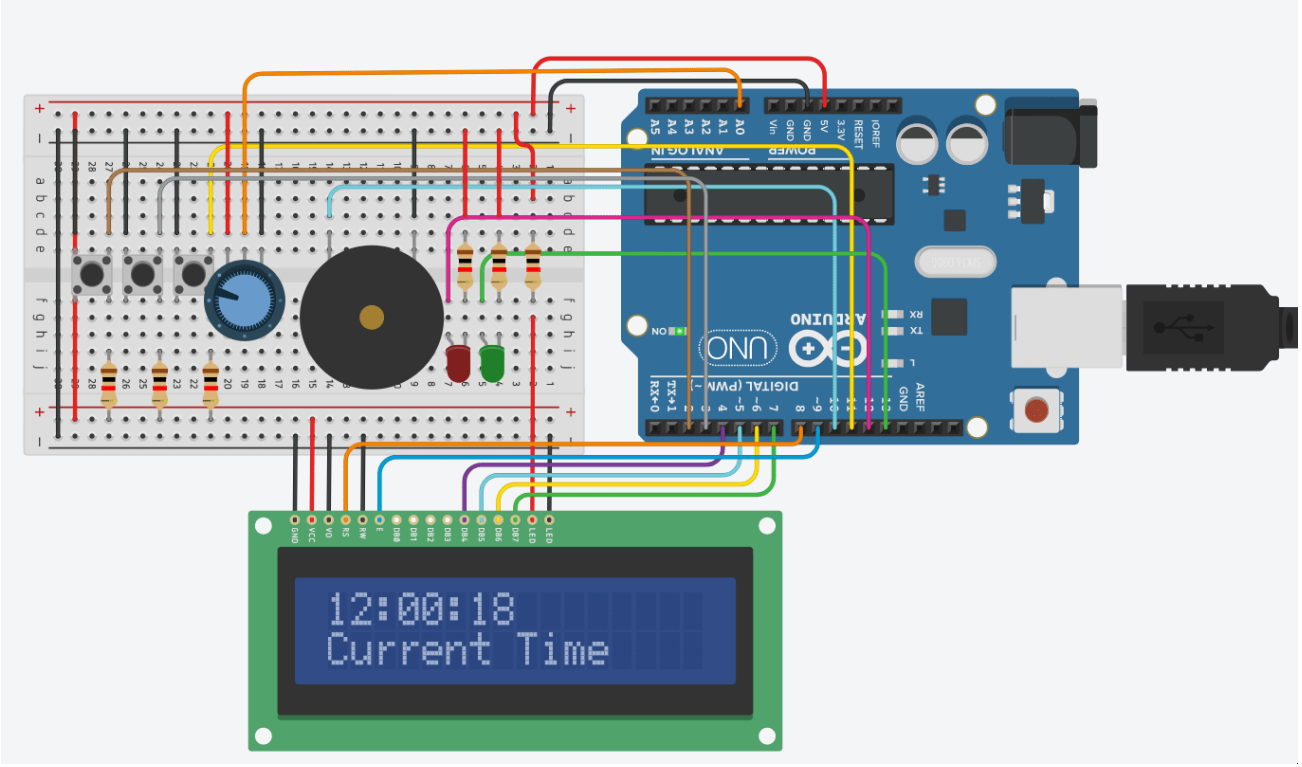




Qual plataforma usar para projetos da disciplina?

- **Ferramentas complementares: simuladores**

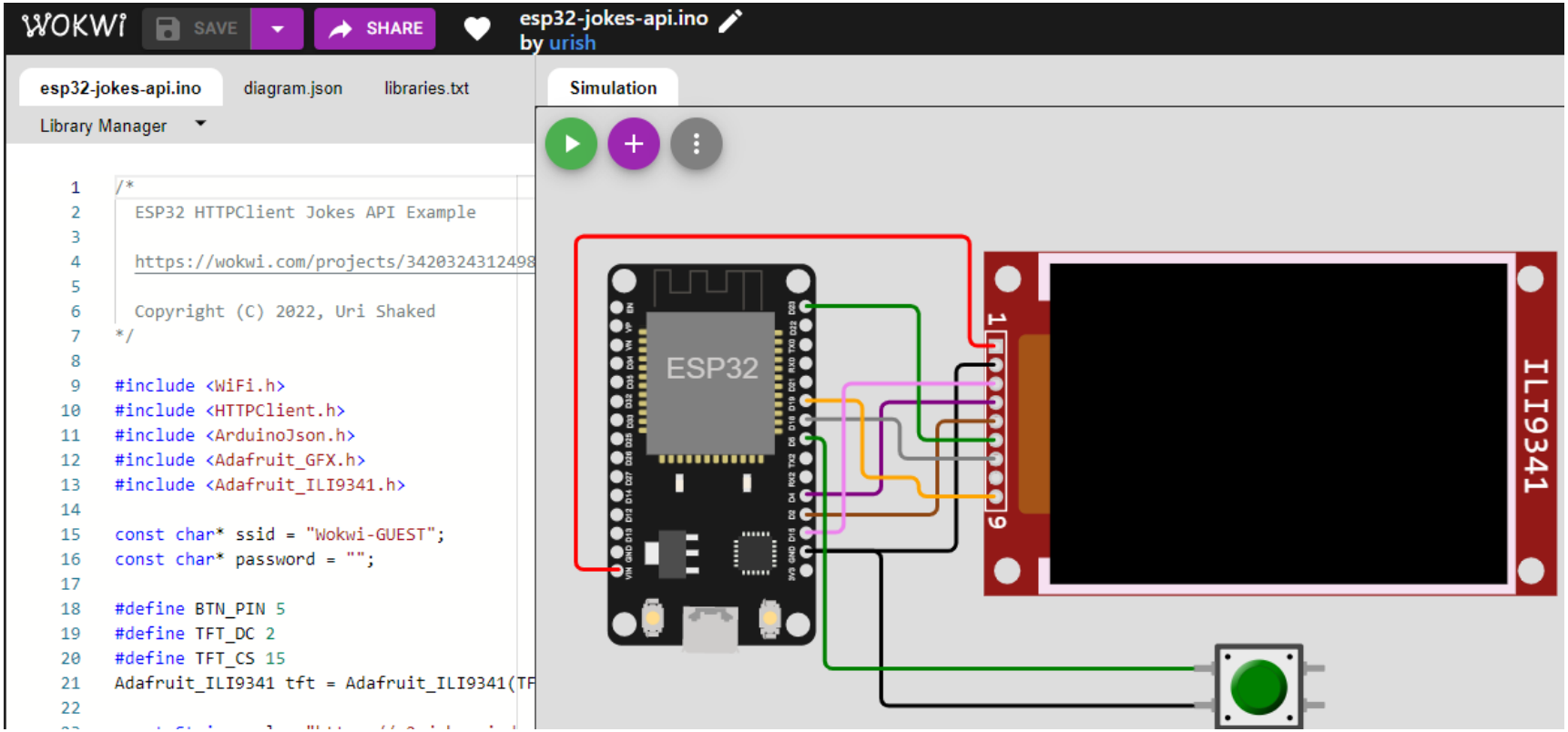
TinkerCAD – Arduino UNO



Qual plataforma usar para projetos da disciplina?

- **Ferramentas complementares: simuladores**

Wokwi – ESP32, Raspberry Pico, e Arduino (vários modelos)

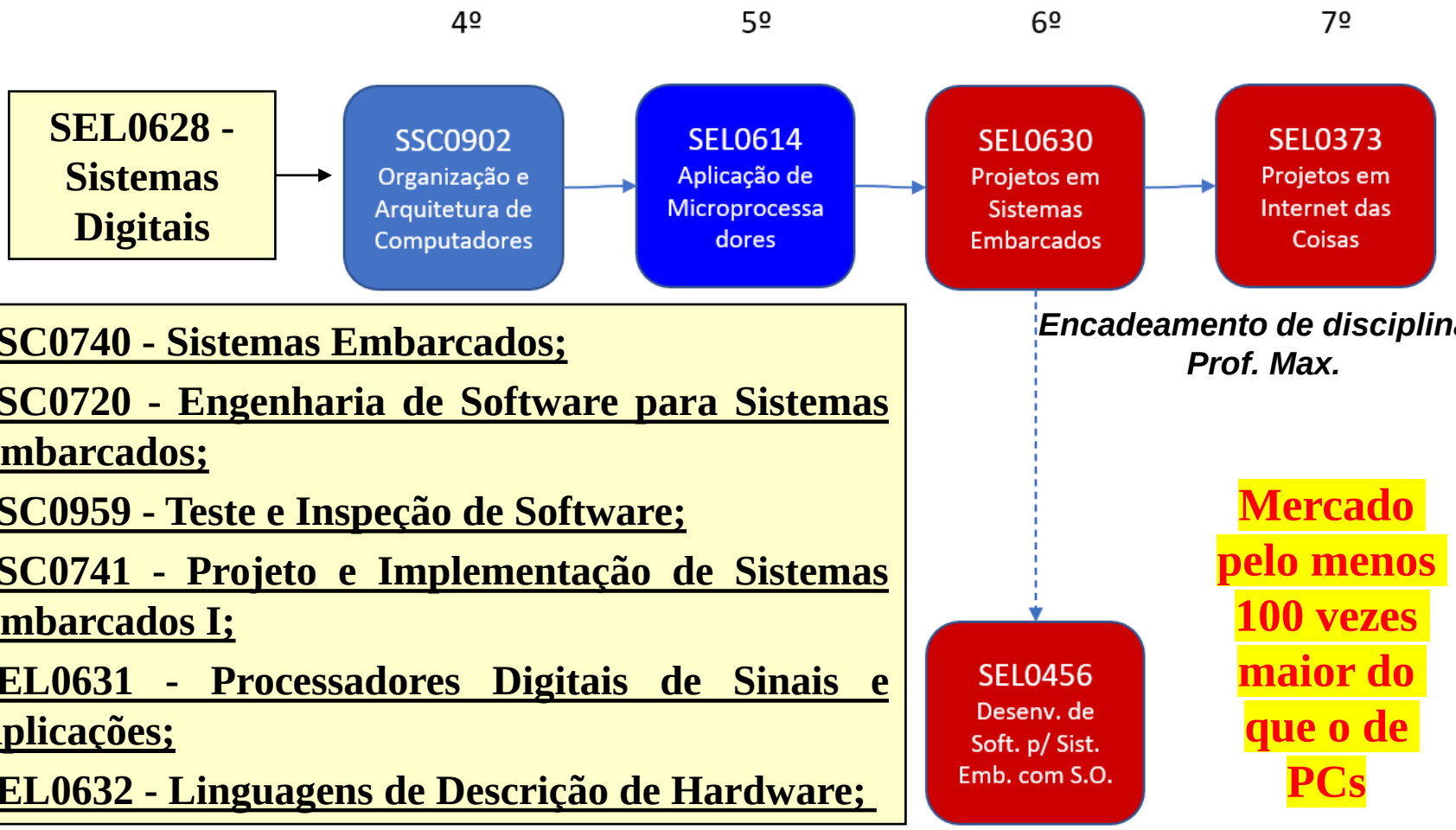


Considerações sobre as soluções

- Plataformas embarcadas como Arduino (microcontrolador), Raspberry Pi Pico (microcontrolador), Raspberry Pi (SBC), Beagle Bone (SBC):
  - ✓ **São ideais para prototipagem, testes, projetos acadêmicos, projetos integradores, teste preliminar para um produto.**
  - ✓ **Não são ideais como sistema embarcado de um produto comercial, onde será necessário suporte, reposição de componentes, garantia, tempo (2 anos, 10 anos, 20 anos de uso). Neste quesito, soluções como Raspberry Pi, por exemplo, não irão oferecer tal suporte, pois a cada ano versões diferentes são lançadas, com upgrades, e sem compatibilidade com versões anteriores em alguns aspectos.**
- O sistema embarcado de produtos deve considerar **o custo de fabricação, linha de produção, prototipagem, confecção da placa versus quantidade a ser produzida**, que irá determinar qual solução escolher (construir tudo do zero, usar um CoM, usar uma SBC, usar uma plataforma com microcontrolador). Ainda assim, **existem soluções de CoM, microcontrolador, e SBCs com maior suporte (NXP, Toradex,) para aplicações em produtos (vide NXP Freedom/Kinetis; Toradex CoMs, Colibri e Carrier boards).**
- Entretanto, para projetos no âmbito do curso de engenharia de computação, Arduino, Raspberry Pi (SBC), Raspberry Pico, e ESP32, simuladores como TinkerCAD e Wokwi são as melhores soluções nos primeiros anos do curso, pois existem as maiores comunidades envolvidas (principalmente Arduino e Raspberry Pi), com documentação, exemplos, projetos, recursos open source etc**., o que não é ofertado em plataformas profissionais da NXP ou Toradex.**

Sistemas embarcados no curso (ênfase)

Engenharia de Computação é umas das áreas mais aderentes à sistemas embarcados e IoT. Os sistemas embarcados dependem da engenharia de computação!



Como se atualizar

- **Páginas dos fabricantes (Ex.: NXP, Toradex, ST, Espressif, Intel, Microchip etc.)**
- **Páginas das comunidades (Ex.: Arduino, Raspberry Pi Foundation, Beagle Bone)**

- **Carreira de sistemas embarcados: https://careers.toradex.com**

- **Portal Embarcados (noticias, destaques, artigos, tutoriais, relatórios sobre o mercado):**

https://embarcados.com.br

Referências

- Arduino CC – Disponível em: https://www.arduino.cc
- Expressif – Disponível em: https://www.espressif.com/en
- Microchip – Disponível em: https://www.microchip.com
- Mikroe – Disponível em: https://www.mikroe.com
- Verle, M. -PIC Microcontrollers Programming in C – Mikroe. Disponível em: https://www.mikroe.com/ebooks/pic-microcontrollers-programming-in-c
- Verle, M. -PIC Architecture and Programming of 8051 MCUs – Mikroe. Disponível em: https://www.mikroe.com/ebooks/architecture-and-programming-of-8051-mcus
- Raspberry Pi Foundation. Disponível em: https://www.raspberrypi.org - https://www.raspberrypi.com
- William Stallings- Arquitetura e Organização de Computadores 10ed Pearson 2017.
- Wolf. M. "Computer as Components", 4th ed. M.K. 2016.
- Zelenovsky, R. & Mendonça, A. Microcontroladores: Programação e Projeto com a Família 8051 – MZ 2013

- Obrigado pela atenção!

pedro.oliveiracjr@usp.br

FIM

Coffee Break