

# Sistemas Embarcados

PROF. JOSENALDE OLIVEIRA

TADS UFRN

[josenalde@eaj.ufrn.br](mailto:josenalde@eaj.ufrn.br)

<https://github.com/josenalde/embeddedsystems>

# Software (sw) + Hardware (hw): projeto diferenciado (co-design)!



Fonte: Grupo de pesquisa em Sistemas Embarcados - UFSC  
[http://ppgeel.posgrad.ufsc.br/files/2017/12/foto\\_artistica\\_agritec.jpg](http://ppgeel.posgrad.ufsc.br/files/2017/12/foto_artistica_agritec.jpg)

# Objetivos

O(a) discente **ampliará a visão de possibilidades de desenvolvimento de software** para plataformas embarcadas com co-desenvolvimento hardware-software. Será apresentado a ferramentas, metodologias de modelagem, análise de requisitos e representação próprios destes sistemas, com aquisição de conhecimentos básicos herdados da automação de processos, incluindo o desenvolvimento de interfaces homem-máquina para monitoramento, controle e auxílio à tomada de decisão.

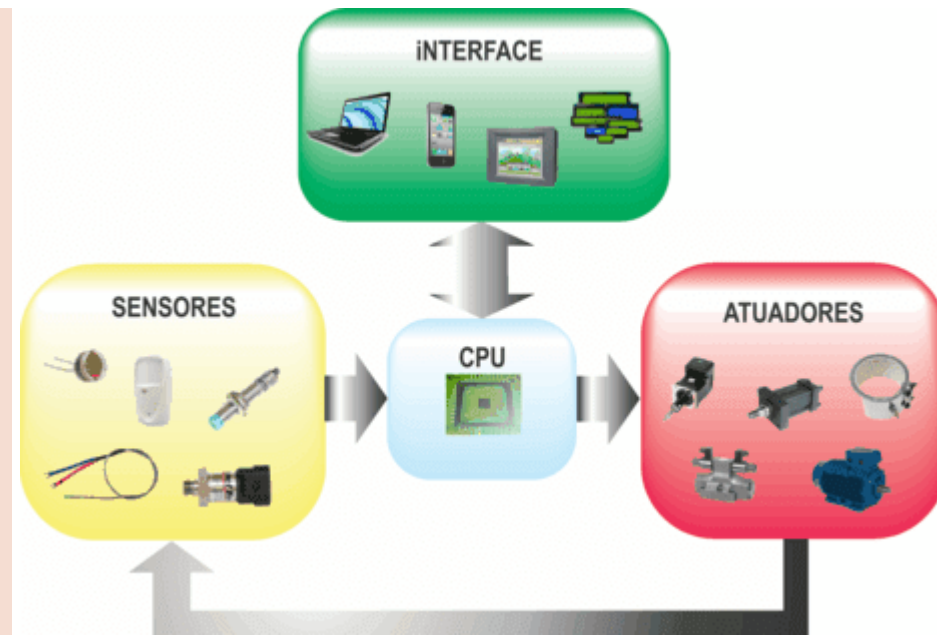
# Conteúdo

Introdução aos Sistemas Embarcados e Plataformas de Desenvolvimento; Metodologia de Desenvolvimento de Sistemas Embarcados; Implementação de Módulos de Hardware em FPGA; Introdução aos Controladores Digitais; Processadores Embarcados Reconfiguráveis em FPGA; Introdução ao Controle Automático de Processos; Amostragem de Sinais; Controladores PID (Proporcional-Integral-Derivativo); Introdução a Sistemas Operacionais de Tempo Real; Desenvolvimento de Projetos.



# Conceitualmente...

- Eletrônica e software **dentro de um produto**, ao contrário de um computador de uso geral
- Combinação de hardware e software, com a possibilidade de integração de partes mecânicas e outras, projetada para realizar uma função dedicada; Podem fazer parte de um sistema ou produto maior, como exemplo o sistema de freios ABS de um carro
- Requisitos e restrições variáveis (segurança, confiabilidade, tempo real, flexibilidade, suporte à radiação, vibrações, umidade, tempo de resposta, precisão, etc).



## CATEGORIAS

- 1) **TEMPO REAL:** aplicações de armazenamento, automotivas, industriais, redes etc. (Free RTOS etc.)
- 2) **PLATAFORMAS:** dispositivos com SO abertos, como Linux, Android, Chrome etc.
- 3) **SEGURANÇA:** smart cards, placas SIM, terminais de pagamento
- 4) **CRÍTICOS:** tempos de resposta altamente exigentes



# Plano de curso

**Unidade I:** plataformas de desenvolvimento e suas características (microcontroladores, microprocessadores, SoC, FPGA, DSP, SCB); demanda por energia, conectividade e tempo de resposta crítico em sistemas embarcados; modelagem grafCet, máquina de estados, Ladder e aplicação em controladores digitais de processos em FPGA; hardware reconfigurável para SoC e sistema operacional embarcado;

**Unidade II:** conceitos de automação de processos, malhas aberta e realimentação, tempo de amostragem, reconstrução de sinais e noções de técnicas de processamento de sinais, controladores liga-desliga e PID;

**Unidade III:** sistema operacional de tempo real e sistemas embarcados, divisão de tarefas por núcleos do dispositivo



FPGA – DE2



ESP32



Raspberry

# Competências e habilidades

Relacionar algoritmos sequenciais, paralelos e diagramas com implementação em linguagem de descrição de hardware; plotagem e visualização de dados de processos; aplicar conceitos de automação de processos e sistemas de controle no desenvolvimento de aplicações; desenvolver a integração de componentes de hardware e software em projetos de sistemas embarcados, selecionando a plataforma/tecnologia alvo mais adequada em função dos requisitos.

## Referências

Notas de aula .pdf disponibilizadas de autoria do docente (slides)

[egrafcet.utad.pt](http://egrafcet.utad.pt)

COSTA, Cesar da. **Projeto de Circuitos Digitais com FPGA**. Érica, 2013.

OLIVEIRA, Sergio. **Internet das Coisas com ESP8266, Arduino e Raspberry Pi**. São Paulo: Novatec, 2017.

MONK, Simon. **Programando o raspberry pi: primeiros passos com python**. Novatec, 2013.

MONK, Simon. **Programação com Arduino II: passos avançados com sketches**. Porto Alegre: Bookman, 2015.

[www.embarcados.com.br](http://www.embarcados.com.br)

MEMBREY, Peter. **Learn Raspberry pi with Linux**, 2013.

# Pré-requisito: sistemas digitais

noções de níveis lógicos (HIGH, LOW),

funções e portas lógicas (AND, OR, NOT, XOR, NAND, NOR),

clock (bordas de subida/descida),

representação binária e hexadecimal,

codificação, decodificação,

mux, demux,

circuitos combinacionais (incluindo aritméticos)

sequenciais (contadores, temporizadores),

IDE para desenvolvimento (Quartus – FPGA, Arduino IDE ou VS Code),

teste em placa física FPGA ou CPLD (No lab. Altera DE2 – Cyclone II)

## Software Selector

Select by Version   Select by Device   Select by Software

**Devices**

- Cyclone 10 LP
- Cyclone V
- Cyclone IV E
- Cyclone IV GX
- Cyclone IV
- Cyclone III LS
- Cyclone III
- Cyclone II**
- Cyclone I
- MAY Series

Quartus Edition	Version Listing
Subscription Edition	↓ 13.0, Service Pack 1
Web Edition	↓ 13.0, Service Pack 1



Jan2021: v20.4 (lite)

No lab. (2020.2): CycloneII 2C35: v13sp1 (web)

[Manual DE2 e pinagens](#)

# Ferramentas (sugeridas)



**e-GRFCET+**  
[egrfcet.utad.pt](http://egrfcet.utad.pt)



[WEG Clic02](#) – Ladder + Simulação

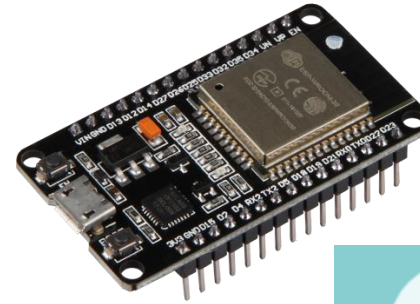





# Ferramentas (sugeridas)



 Visual Studio Code



 Visual Studio Code



Arduino (Microsoft) extensão



Arduino IDE 1.8.13

# Fato: variedade de soluções/dispositivos

Porém, a tecnologia/plataforma para desenvolver um projeto de sistema embarcado é ampla, a depender da aplicação, expertise do projetista, custo etc.

FPGA, uC (8051, Atmegaxxx, PIC, MSP (Texas) etc.), DSP (dsPIC etc.), SOC (system on chip – on demand computer), embedded computers com OS (Raspberry, beaglebone, Asus Tinker Board etc)

Ao dispositivo que rodará o software acrescenta-se o hardware da aplicação (entrada/saída)



*...ficar atento à próxima edição*

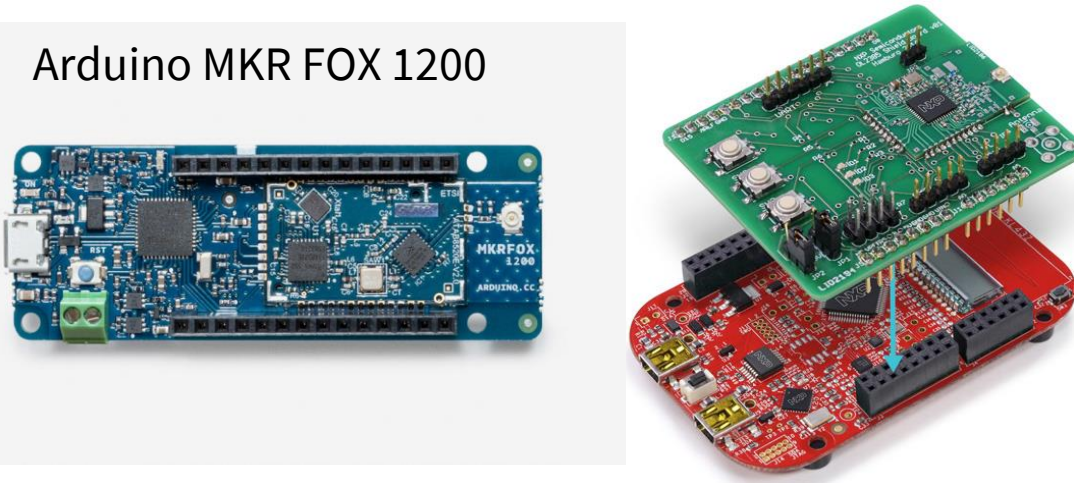
## Embedded Systems Competition (ESC)

- A cada ano plataforma de hardware diferente
- Consulta de pinagem, APIs (funções etc.)
- Em 2020 (open platform)

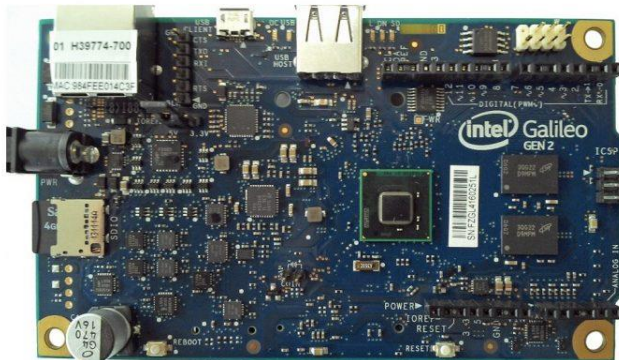
# Exemplos:

**2017:** OM2385/SF001 - OL2385 Wireless sub-GHz Transceiver SIGFOX Development Kit with KL43Z

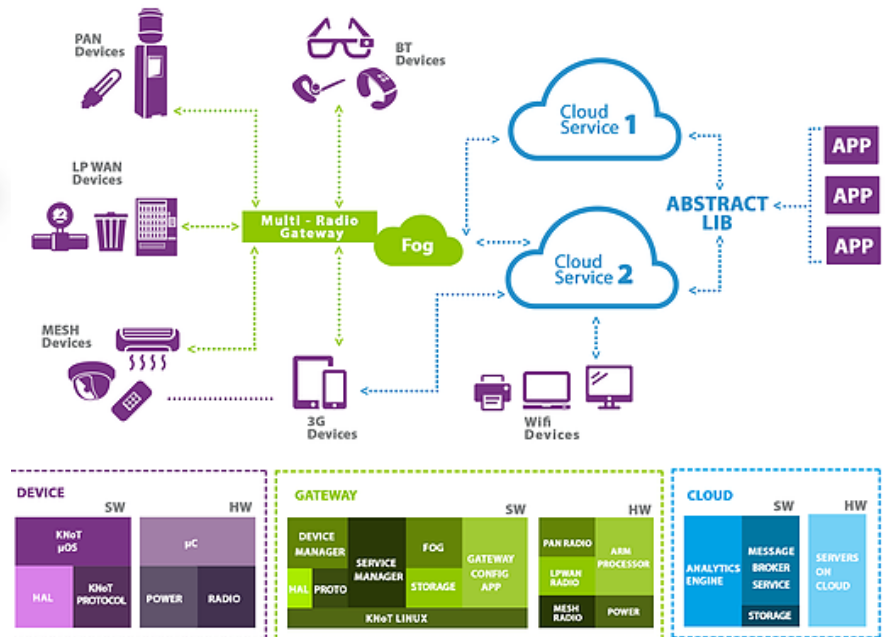
Arduino MKR FOX 1200



**2016:** Intel® Galileo Gen 2 (Elvis) e FPGA DE2-115, seletiva na UFRN



**2018:** Hackaton CSBC Natal (KNoT)



Dica: [TCC Almeida, João G. Q. \(2018\)](#)  
KNoT, IoT, automação, controle de processos

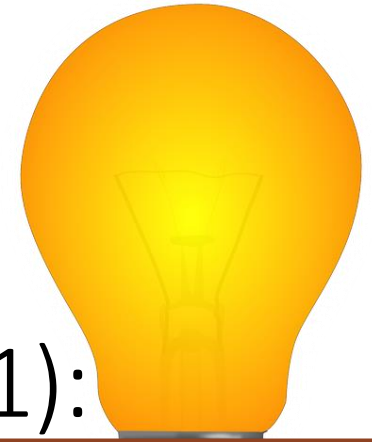
Raspberry  
Arduino  
Knot gateway  
Knot Thing

# Sites com projetos - inspiração:

[randomnerdtutorials.com](http://randomnerdtutorials.com)

<http://www.raspberrypiportugal.pt/os-melhores-projetos-raspberry-pi/>

<https://create.arduino.cc/projecthub>



## Projetos anteriores (18.1, 18.2, 19.1):

- Reconhecimento audível de seções em supermercados
- Estação Meteorológica com Smart Sensor para Cálculo de Evapotranspiração (anemômetro)
- *Indicador Sonoro e de Orientação em Viveiros de Plantas Medicinais para Deficientes Visuais*
- *Sistema de Contagem de Gado com Porteira Automática (Entrada/Saída) e Monitoramento de Bem-estar Animal em Ambiente Cercado*
- *Separação e Filtragem de Sinais de Áudio de Insetos Associados à Pragas Agrícolas para Construção de Armadilhas Inteligentes (afinador eletrônico)*
- Sistema Automático de Recirculação de Água em Hidroponia
- [SmartHive](#) (vídeo Tela Rural 15.10.2018)
- Implementação de controlador BELBIC em C++/Arduino Uno aplicado a sistema oscilatório
- [Esteira com CLP-FPGA e interface web de configuração para temporização](#)
- Auto sintonia de controlador PID
- Controle de rotação de ventoinha por proximidade bluetooth (ESP32)
- ...

