

Universidade Federal da Bahia  
Programa de Pós-Graduação em Engenharia Elétrica

---



# **Comunicação entre Robot Operating System - ROS e SoC com FPGA integrado**

Autor: Nestor Dias Pereira Neto

Orientador: Prof. Dr. Wagner Luiz Alves de Oliveira

Coorientador: Prof. Dr. Paulo César Farias

Salvador, 6 de dezembro de 2022

# Agenda

1. Introdução
2. Parte I: Referenciais Teóricos
3. Parte II: Desenvolvimento
4. Parte III: Resultados

# Introdução

---

- Projetos em robótica têm exigido maior percepção do ambiente.
- Maior poder de processamento demanda maior consumo de energia.
- FPGAs possuem potencial para melhorar desempenho de sistemas computacionais.
- MACs de alta velocidade, processamento paralelo e baixas frequências de trabalho são comuns em FPGA.
- O uso do FPGA pode contribuir com ganho de poder de processamento associado ao baixo consumo.

- Apesar de oferecer grandes vantagens, o FPGA é pouco incentivado em projetos de robótica.
- Atualmente o framework ROS está se consolidando como o padrão na criação de novas plataformas robóticas.
- O ROS é considerado um sistema operacional para robôs.
- Desenvolver uma solução para estabelecer comunicação entre FPGA e o ROS.

## **Como estabelecer a comunicação entre o ROS e um sistema de processamento auxiliar embarcado em um FPGA?**

Este problema é o que este trabalho busca resolver, possibilitando assim, o uso de aceleração por hardware através do FPGA no desenvolvimento de novos projetos de robótica.

## Objetivo geral

- Desenvolver uma solução para estabelecer comunicação entre *Field-Programmable Gate Array - FPGA*, e o *Robot Operating System - ROS*.

## Objetivos específicos

- Estudar teoria dos assuntos relevantes ao projeto: Verilog HDL, Embedded Linux, Cyclone V, TCP/IP Stack, ROS;
- Estudar conceitos de programação de sockets em Iniguagem C++;
- Implementar distribuição Embedded Linux para processador ARM embarcado no SoC Cyclone V da Intel;
- Implementar distribuição Embedded Linux para processador ARM.
- Estabelecer comunicação entre o ROS e o Cyclone V, através da tecnologia Gigabit Ethernet;
- Avaliar o desempenho da rede entre o computador e o protótipo após a inclusão do FPGA ao sistema.

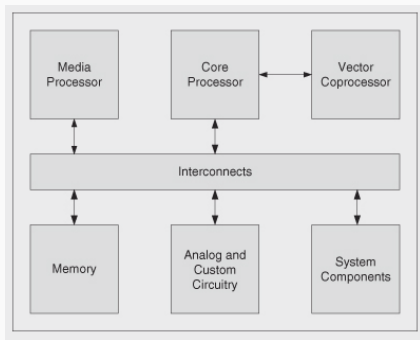


# Parte I: Referenciais Teóricos

---

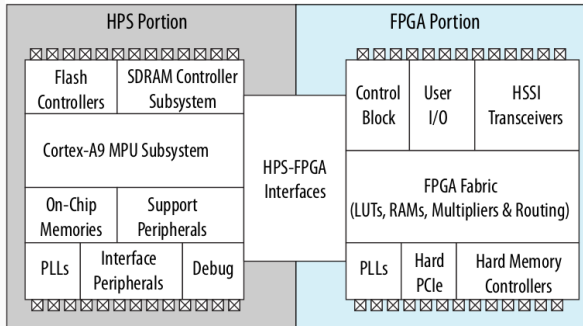
# Cyclone V SoC-FPGA

SoC é um acrônimo de *System-on-a-Chip* ou apenas *System on Chip*. Um SoC pode combinar diferentes elementos, em diferentes configurações, para formar um sistema completo.



# Cyclone V SoC-FPGA

A Intel fornece uma linha de produtos classificados como SoC-FPGA, os quais se caracterizam por possuir um rede de FPGA integrados a um processador ARM Cortex A9.



Esta estratégia de interconexão entre o HPS e o FPGA do Cyclone V em um único circuito integrado oferece:

- Largura de banda de pico de mais de 100 Gbps;
- Coerência de dados integrada;
- Significativa economia de energia do sistema, eliminando caminhos de E/S externos entre o processador e o FPGA.

## Processador ARM Cortex-A9

- É uma linha de processadores ARM de uso geral, otimizados para alcançar maior desempenho aliado a um baixo consumo.
- Possui estrutura interna configurável, o que oferece flexibilidade ideal para o desenvolvimento de um novo SoC.

Arquitetura	Armv7-A
Multicore	1-4 cores
Suporte ISA	Extensão DSP
	Ponto Flutuante (Opcional)
MMU	Armv7 MMU
Características	Arquitetura flexível com caches configuráveis
	Desempenho 50% maior do que o Cortex-A8

## Fild-Programmable Gate Array-FPGA

- Dispositivos semicondutores construídos através de uma matriz de blocos lógicos configurável, os CLBs.
- Interconectáveis a partir de conexões Programáveis.
- Hardware configuravel pelo usuário.
- Poderosas ferramentas de prototipagem.

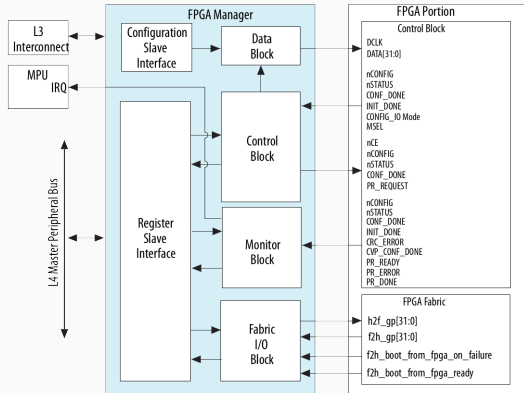
## Hardware Processor System

A família Cyclone V SoC-FPGA é composta por duas partições, uma malha de FPGA e um *Hard Processor System* (HPS). Os principais módulos do HPS são:

- *Microprocessador Unit* (MPU), subsistema com um ou dois ARM Cortex-A9.
- Controladores de memória Flash.
- Controladores de SDRAM.
- Sistema de interconexão.
- *On-chip memory*.
- *Phase-locked loops* (PLLs).

## Hardware Processor System: FPGA Manager

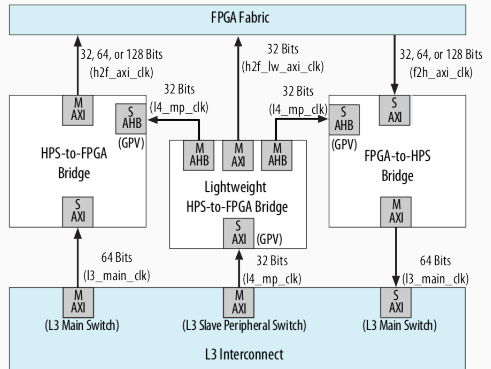
- Configuração do FPGA.
- 32 sinais de uso geral.
- Status do FPGA.
- Interrupções para o HPS a partir do FPGA.
- Resetar o FPGA.





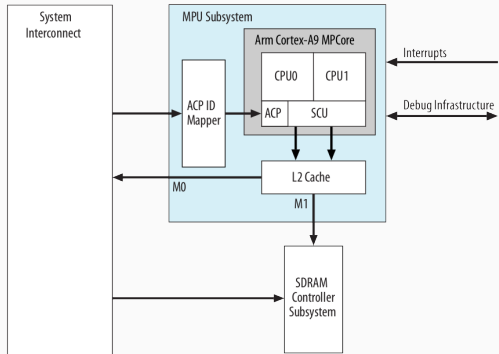
## Hardware Processor System: HPS-FPGA Bridge

- O HPS pode se comunicar com o FPGA.
- Podem alcançar larguras de banda superiores a 100Gbps
- O HPS possui três pontes.
- A comunicação funciona nos dois sentidos.



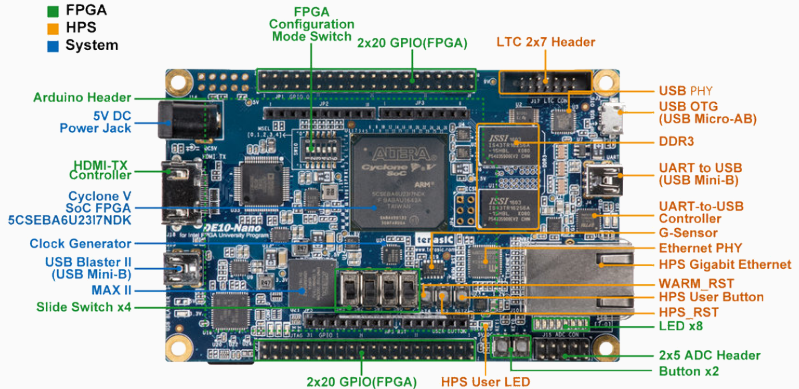
## Hardware Processor System: Cortex A9 MPU

O *Microprocessor Unit Subsystem* (MPU) presentes SoCs da família Cyclone V incluem um Arm Cortex-A9 MPCore, com processador de uso geral de 32 bits com um ou dois núcleos.



# Cyclone V SoC-FPGA

## Kit de desenvolvimento DE10-nano



## Um sistema operacional para robôs

“O Robot Operating System (ROS) é um conjunto de bibliotecas de software e ferramentas que te auxiliam na construção de aplicações em robótica. De *drivers* ao estado da arte de algoritmos, e com poderosas ferramentas de desenvolvimento, ROS tem o que você precisa para seu próximo projeto de robótica. E tudo é open source.”

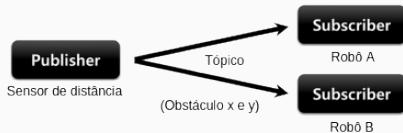
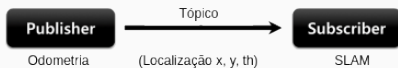
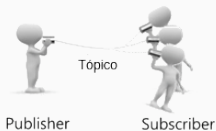
## Vantagens do ROS: Tópico

- **Recursos Nativos:** O ROS oferece, de forma nativa, muitos recursos prontos, testados e validados pela comunidade.
- **Ferramentas de desenvolvimento:** O ROS é disponibilizado com uma grande quantidade de ferramentas para debugging, visualização, incluindo ferramentas para simulação.
- **Suporte a sensores e atuadores:** Muitos dos sensores e atuadores usados na robótica já são suportados pelo ROS e o número de dispositivos compatíveis aumenta a cada ano.
- **Múltiplas linguagens:** O framework ROS pode ser programado em algumas linguagens de programação modernas.

## Sistema de arquivos ROS

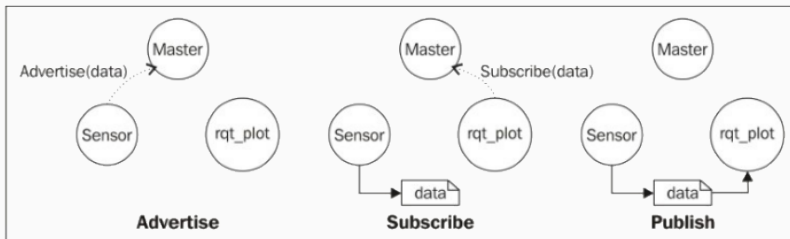
- **Pacotes:** Pacotes são a unidade de software principal do ROS.
- **Manifesto do pacote:** O arquivo `package.xml` é conhecido como manifesto do pacote, ele fornece os metadados a respeito do pacotes.
- **Arquivo `.msg`:** É o arquivo de descrição de um tipo de mensagem.
- **Arquivo `.srv`:** É o arquivo de descrição de um tipo de serviço.
- **Repositórios:** Pacotes ROS são compartilhados usando algum tipo de sistema de controle de versão, como o git.

## Componentes ROS: Tópicos ROS



# Robot Operating System-ROS

## Componentes ROS: ROS Master





## Componentes ROS: Parâmetros ROS

- São variáveis globais que podem ser usadas por nós.
- O principal objetivo dos parâmetros é fornecer ao sistema a capacidade de se adaptar a cenários distintos de maneira ágil.
- Os parâmetros são criados com valores padrões

## Parte II: Desenvolvimento

---

## Programação de rede:

- Efetuar programação de sockets e bibliotecas específicas para trabalho em redes.

## Aplicação ROS:

- Desenvolver um pacote ROS para disponibilizar os dados recebidos dos outros pacotes ROS do sistema para o SoC.

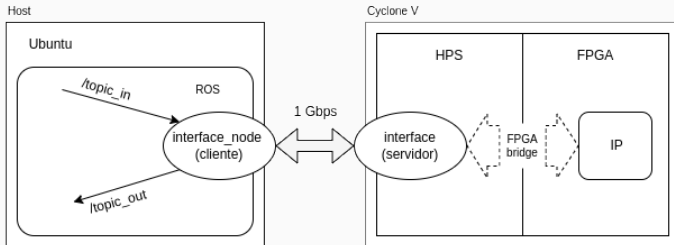
## Aplicação HPS-SoC:

- Estabelecer a comunicação entre a interface de rede da placa E10-nano através de um programa executado no HPS do SoC, mantendo uma conexão entre o HPS e o FPGA.

# Arquitetura do Sistema

## Modelo cliente-servidor

- A comunicação entre o *host* e a placa DE10-nano através de uma rede gigabit.
- Possui estrutura que permite dividir o esforço computacional.
- Contribui de forma significativa para a modularização do sistema.

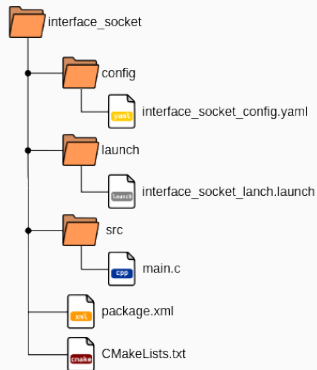


## Biblioteca de comunicação - libinterfacesocket

- Para manter o padrão do desenvolvimentos dos códigos tanto do cliente quanto do servidor.
- Classe foi desenvolvida como um módulo à parte e compilada como uma biblioteca estática.
- Alterações ou correções de erros, sem alterações nos códigos do servidor ou do cliente.
- Programação da biblioteca foi realizada com base em sockets.
- A programação de sockets em C++ possibilita um alto nível de otimização da comunicação.
- Código fonte disponível no Github.

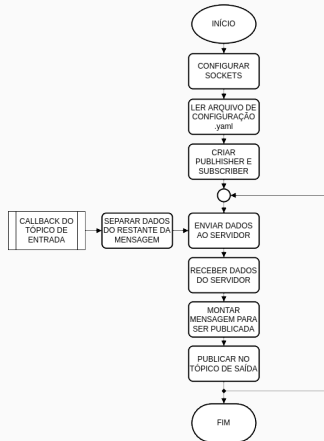
# Pacote ROS (cliente)

No desenvolvimento de aplicações com ROS deve ser respeitada a estrutura de diretórios de um pacote ROS, portanto o desenvolvimento do cliente deve levar em consideração essa estrutura.



# Pacote ROS (cliente)

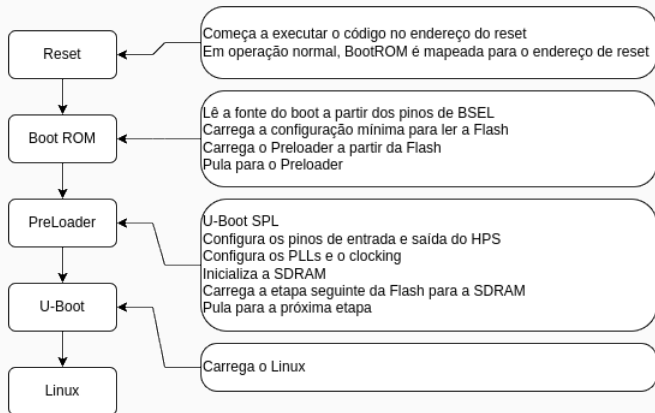
- Configurações iniciais e leitura dos parâmetro.
- Cria o publisher e subscriber.
- Callback ler um tópico enviado por outro nó ROS.
- Faz o tratamento dos dados recebidos e os envia a SoC.
- Recebe dados do SoC e monta as mensagem ROS para ser publicado em um tópico



Neste trabalho o programa servidor disponibiliza ao cliente a interface com o FPGA presente no SoC. Assim, ao receber o pedido do cliente, o servidor deve ser capaz de enviar os dados vindos do cliente à unidade de processamento embarcada no FPGA e em seguida devolver esses dados já processados ao cliente



## Processo de BOOT



## Distribuição Linux rsyocto

O rsyocto foi a distribuição linux escolhida, desenvolvida exclusivamente para SoCs da Intel com FPGA integrado. Algumas das suas características são:

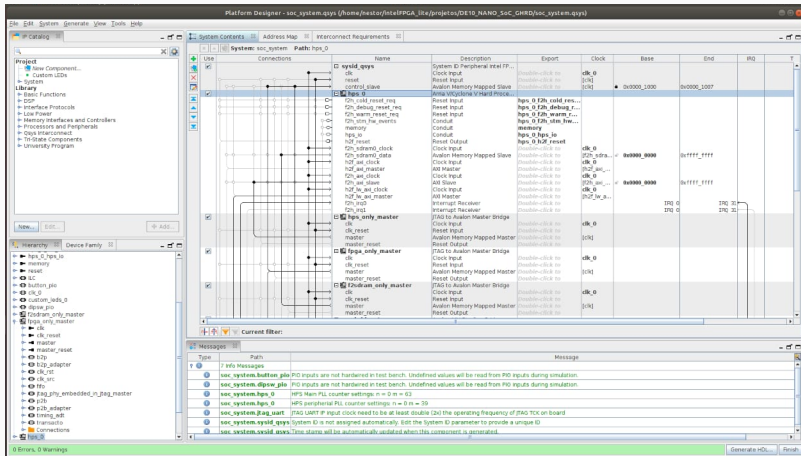
- Linux Kernel 5.11.
- Acesso total ao Dual-Core ARM (ARMv7-A) Cortex-A9.
- Configuração da malha FPGA durante o boot e com comandos Linux simples.
- Todas as interfaces entre o HPS e o FPGA estão habilitadas e prontas para uso.
- Ferramentas para interagir com a malha do FPGA a partir das ARM AXI HPS- toFPGA bridges.

## Servidor: Interface socket server

- Baixar, compilar e instalar a libinterfasesocket no linux embarcado.
- Mantém uma porta aberta esperando a conexão com o cliente.
- Dados recebidos do cliente são enviados para serem processados pelo FPGA e em seguida devolvidos ao cliente.
- O acesso do servidor ao FPGA é feito através de mapeamento de endereços do linux.



**Servidor: Interface socket server**



## Parte III: Resultados

---

Foram idealizados dois ensaios: o primeiro com o objetivo de testar o fluxo completo de troca de dados entre um nó ROS e o FPGA; o segundo ensaio teve o objetivo de testar um fluxo grande de dados trafegando entre o ROS e o SoC.

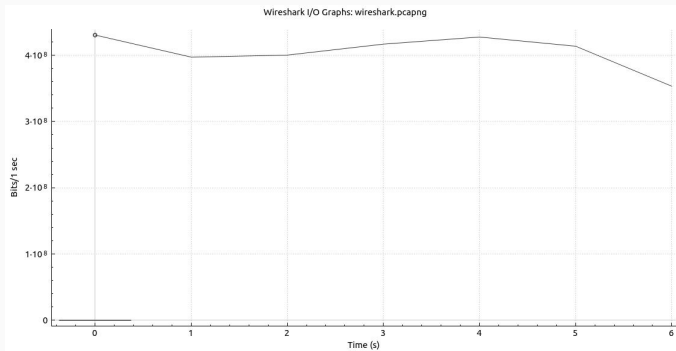
## Teste Comunicação:

- verificar a comunicação completa entre o nó ROS e um IP sendo executado no FPGA.
- foi projetado um circuito simples que calcula o dobro de um número.

## Teste Rede:

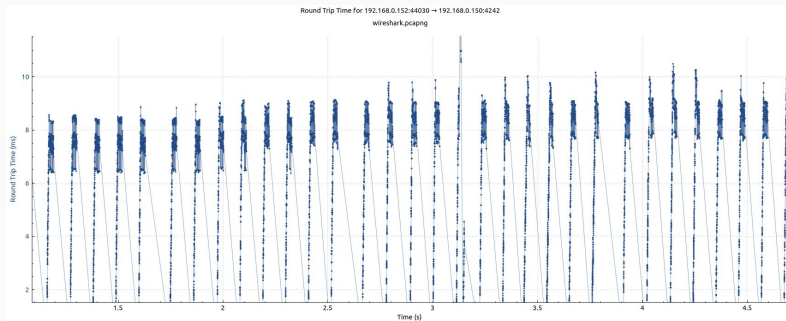
- testar a sobrecarga de dados na rede e avaliar o seu comportamento.
- *Stream* de vídeo com resolução de 1280x720 em 15 fps.

## Velocidade de transmissão



# Resultados Alcançados

## Atrasos gerados durante a comunicação





## Conclusão

Os resultados alcançados para os dois ensaios foram considerados satisfatórios, apesar de, na transferência das imagens, ter sido observado um pequeno delay, menor do que 10 ms.

## Trabalhos futuros

Estudos futuros podem dar ênfase a blocos específicos do trabalho, otimizando a comunicação e deixando o sistema ainda mais genérico.

Trabalho publicado no 28º IBERCHIP Workshop, promovido pela IEEE Circuits and Systems Society - CASS, Santiago no Chile, 2022.



ALTERA.

**SoC FPGA ARM Cortex-A9 MPCore Processor Advance Information Brief.**

Altera Corporation, 2 2012.



INTEL.

**Fpgas cyclone® v e fpgas soc, 2022.**



MAHTANI, A., SÁNCHEZ, L., FERNÁNDEZ, E., AND MARTINEZ, A.

**Effective Robotics Programming with ROS, 3 ed.**

Packt Publishing Ltd, Birmingham, 2016.



ROS.

**Pckages.**

Open Source Robotics Foundation, 2019.



TERASIC.

**DE10-Nano User Manual, 2.2 ed.**

Terasic Inc, 1 2020.

rev. B2/C Hardware.



XILINX, A.

**Field programmable gate array (fpga), 2022.**



YAMASHINA, K., OHKAWA, T., OOTSU, K., AND YOKOTA, T.

**Proposal of ros-compliant fpga component for low-power robotic systems: case study on image processing application.**

*2nd International Workshop on FPGAs for Software Programmers (FSP 2015)* (2015), 62–67.