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, 5 de dezembro de 2022

Agenda

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

Introdução

Contextualização

- 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.

Contextualização

- Apesar de oferecer grande vantagens o FPGA em projetos de robótica é pouco incentivado.
- 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.

Problema

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.

Objetivos

Objetivo geral

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

Objetivos

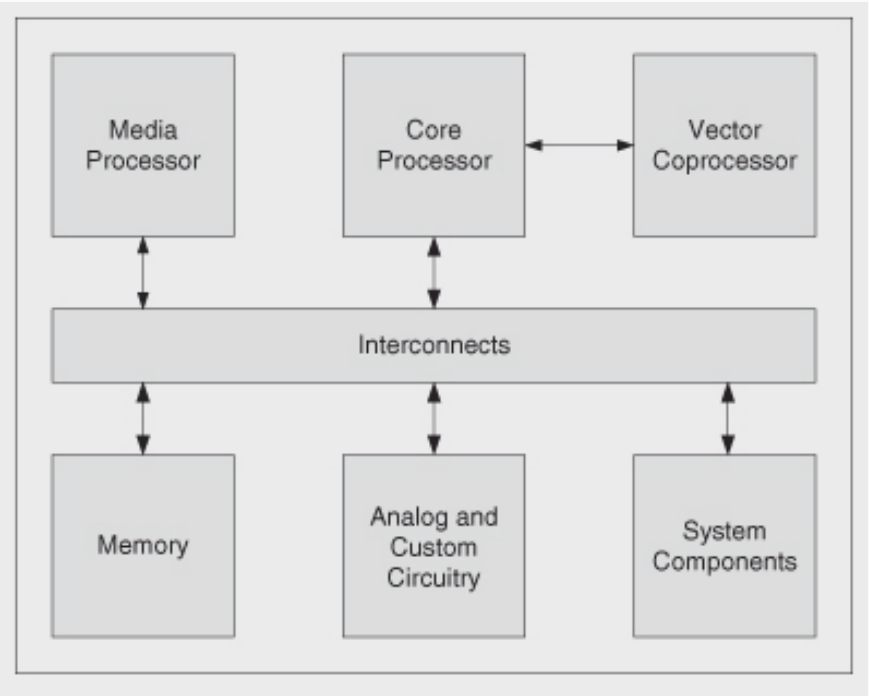
Objetivos específicos

- Estudar teoria dos assuntos relevantes ao projeto: Verilog HDL, Embedded Linx, Cyclone V, TCP/IP Stack, ROS;
- Estudar conceitos de programação de sockets em liguagem 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 Teórico

Cyclone V SoC-FPGA

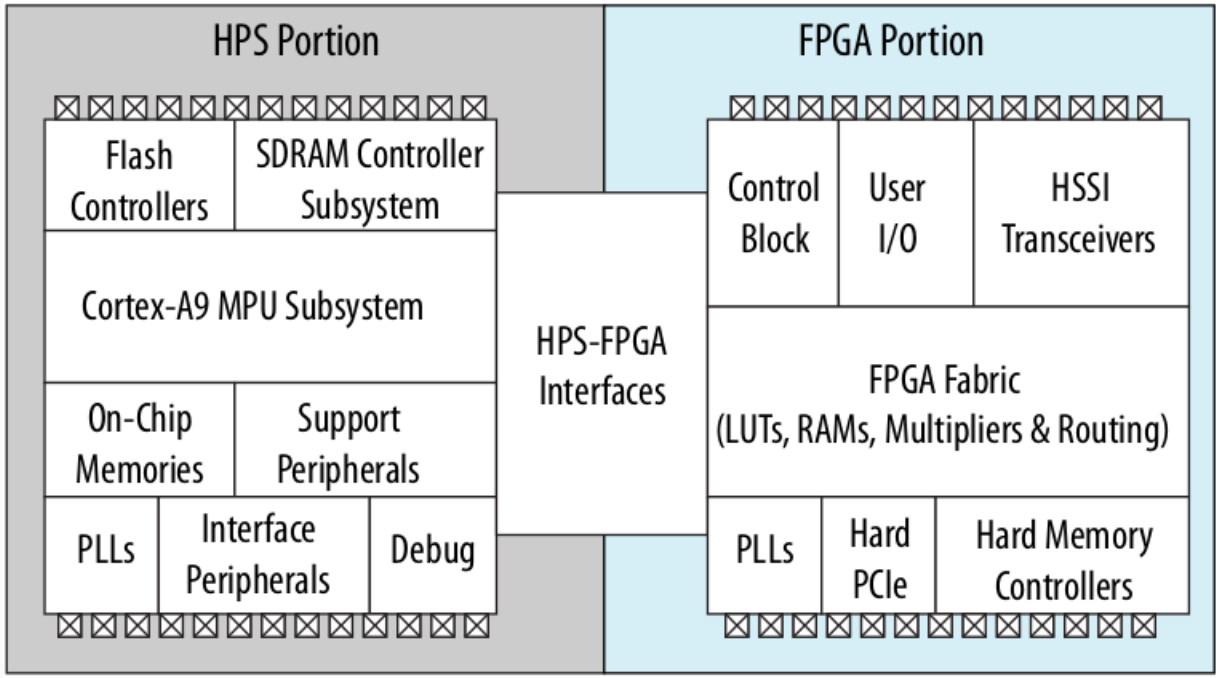
SoC é um acrônimo de *System-on-a-Chip* ou apenas *System on Chip*. Um SoC pode combinar diferentes elementos, em diferentes configurações, para formar um sistema completo.



Fonte:

Cyclone V SoC-FPGA

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



Fonte:

Cyclone V SoC-FPGA

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 alcançarem maior desempenho aliado a um baixo consumo.
- Possue estrutura interna configuravel, 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
Caracteristicas	Arquitetura flexisível com caches configuráveis
	Desempenho 50% maior do que o Cortex-A8

Fonte:

Cyclone V SoC-FPGA

Fild-Programmable Gate Array-FPGA

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

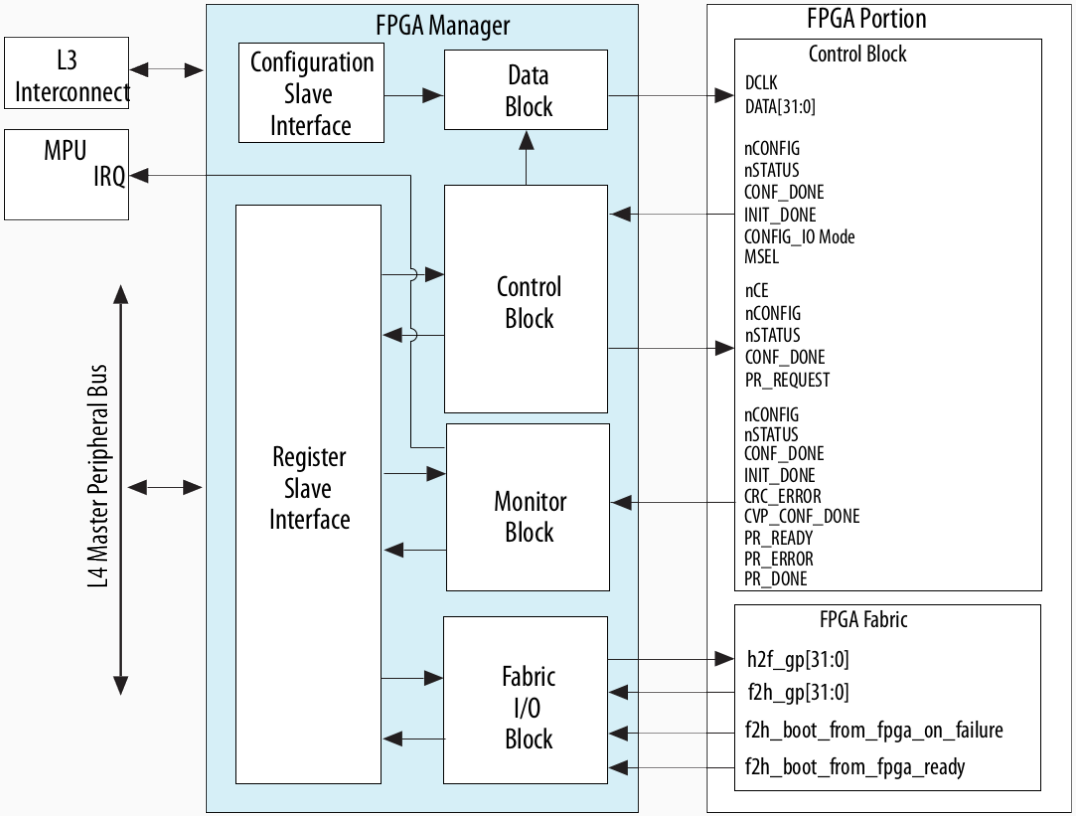
Hardware Processor System

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).

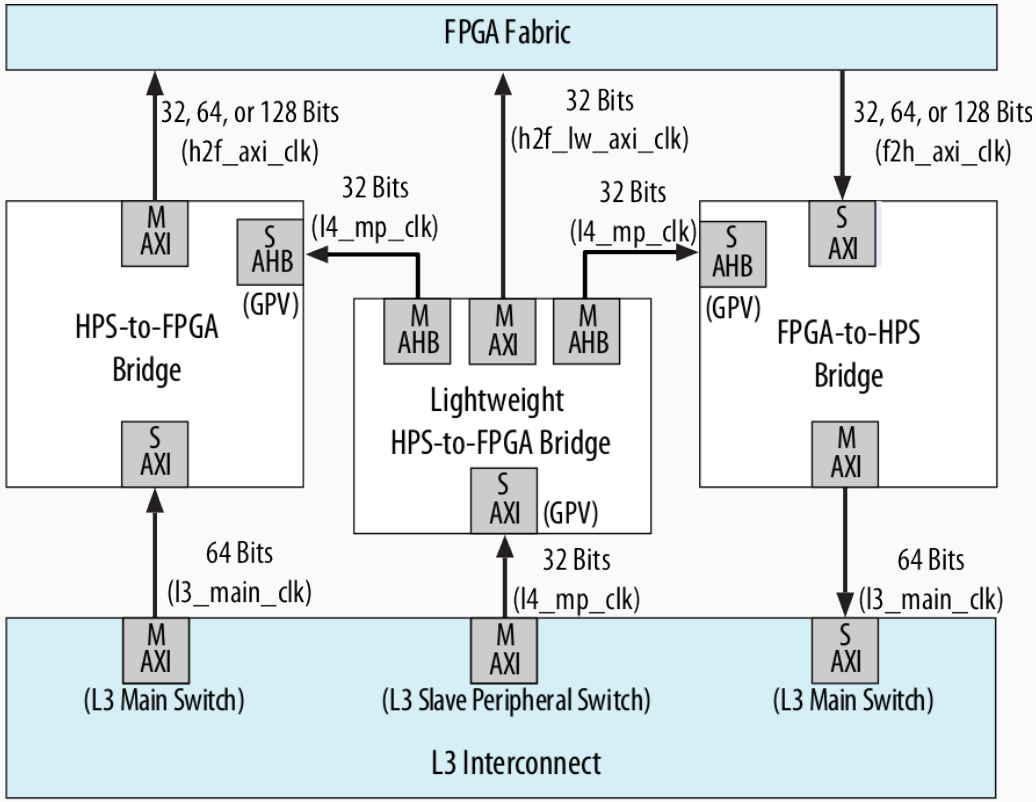
Hardware Processor System: FPGA Manager

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



Fonte:

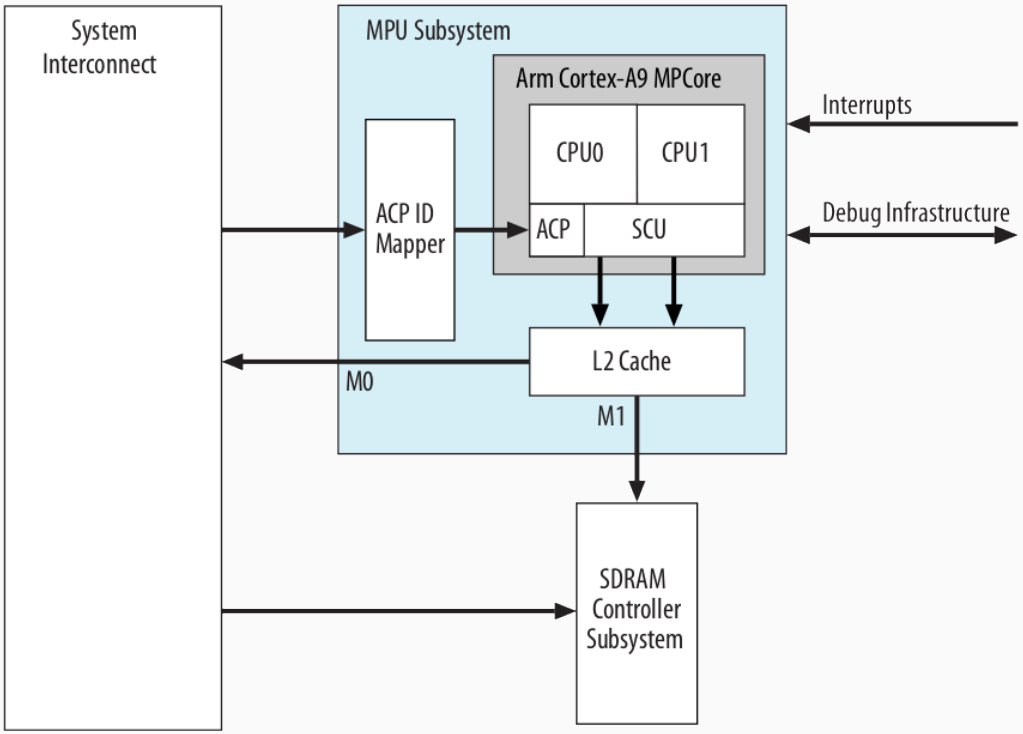
Hardware Processor System: HPS-FPGA Bridge



Fonte:

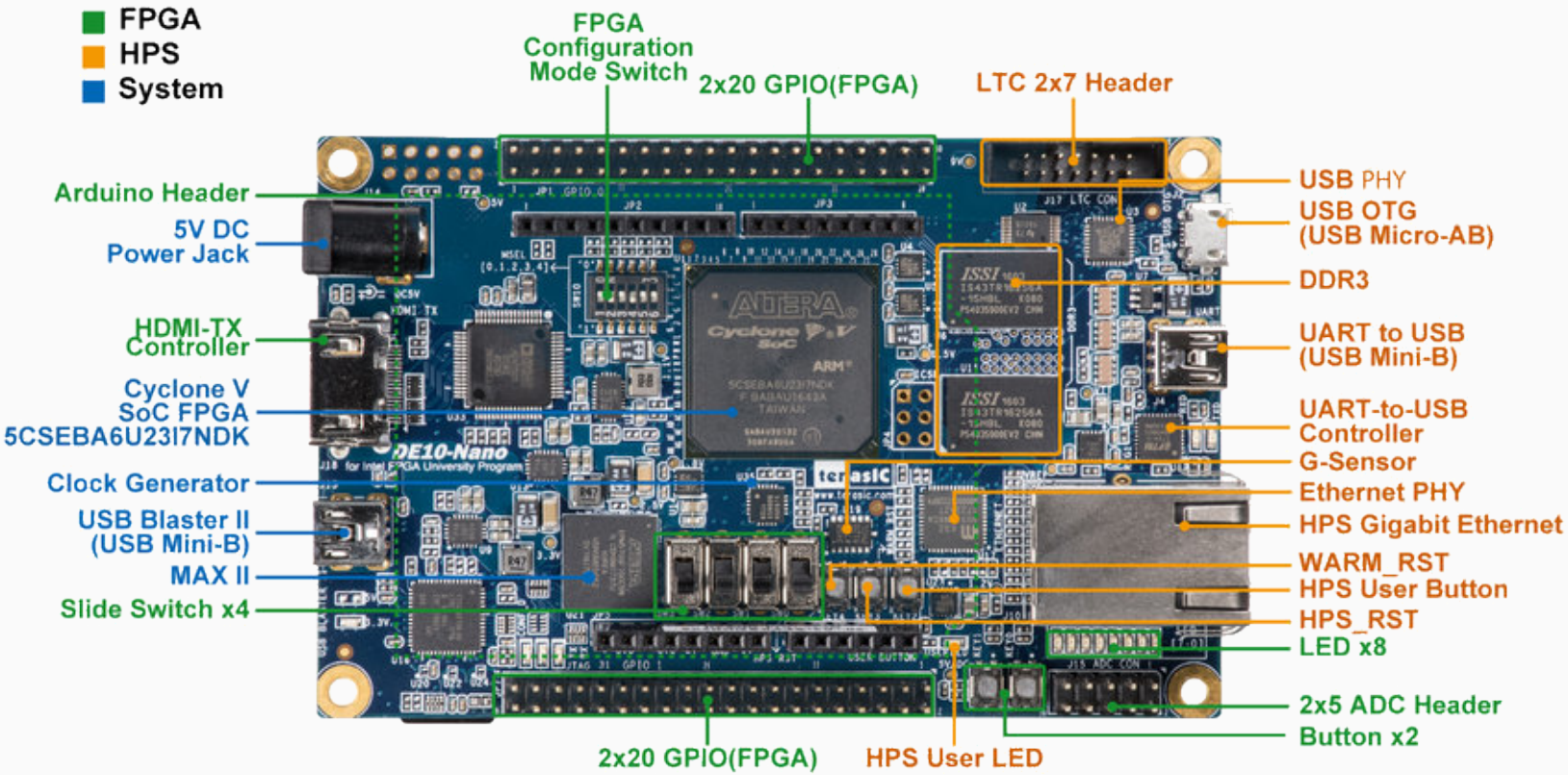
Hardware Processor System: Cortex A9 MPU

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.



Fonte:

Kit de desenvolvimento DE10-nano



Fonte:

Robot Operating System-ROS

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"

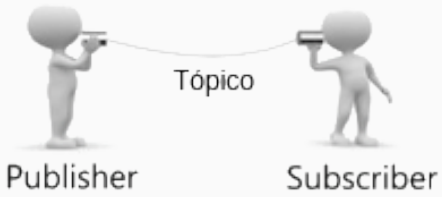
Vantagens do ROS: Tópico

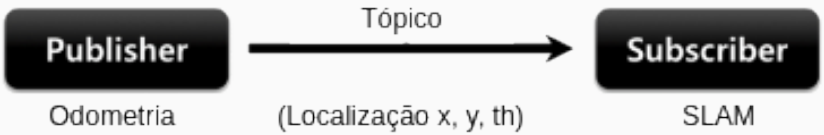
- **Recursos Nativos:** O ROS oferece, de forma nativa, muitos recursos prontos, testados e validados pela comunidade.
- **Ferramentas de desenvolvimento:** O ROS é disponibilizado com uma grande quantidade de ferramentas para debugging, visualização, incluindo ferramentas para simulação.
- **Suporte 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.

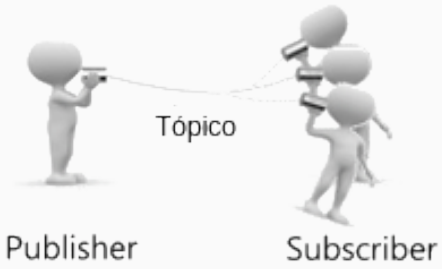
Sistema de arquivos ROS

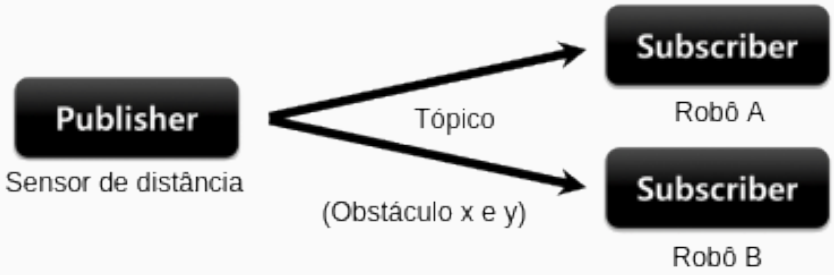
- **Pacotes:** Pacotes são a unidade de software principal do ROS.
- **Manifesto do pacote:** O arquivo package.xml é conhecido como manifesto do pacote, ele fornece os metadados a respeito do pacotes.
- **Arquivo .msg:** É o arquivo de descrição de um tipo de mensagem.
- **Arquivo .srv:** É o arquivo 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



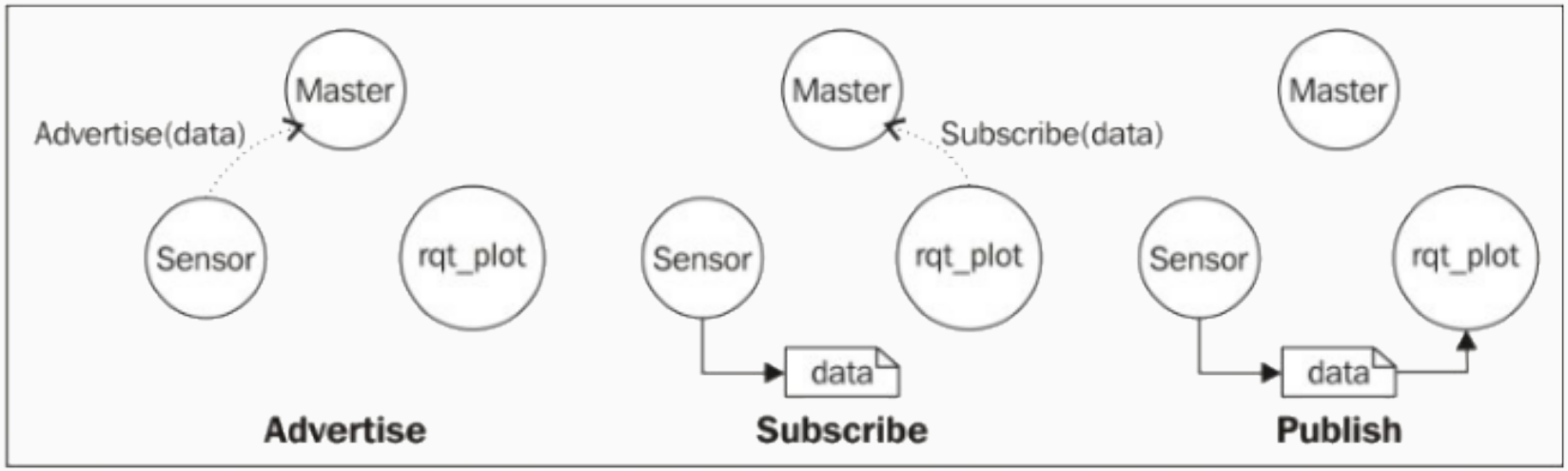






Fonte:

Componentes ROS: ROS Master



Fonte:

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

Arquitetura do Sistema

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.

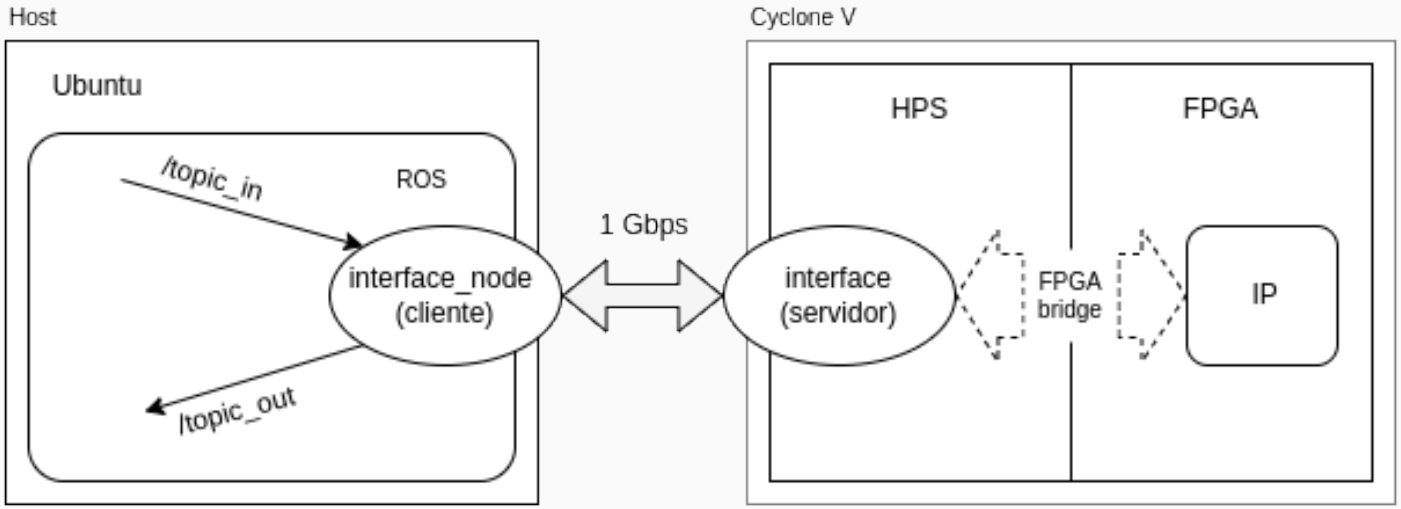
Aplicação HPS-SoC:

- 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.



Fonte:

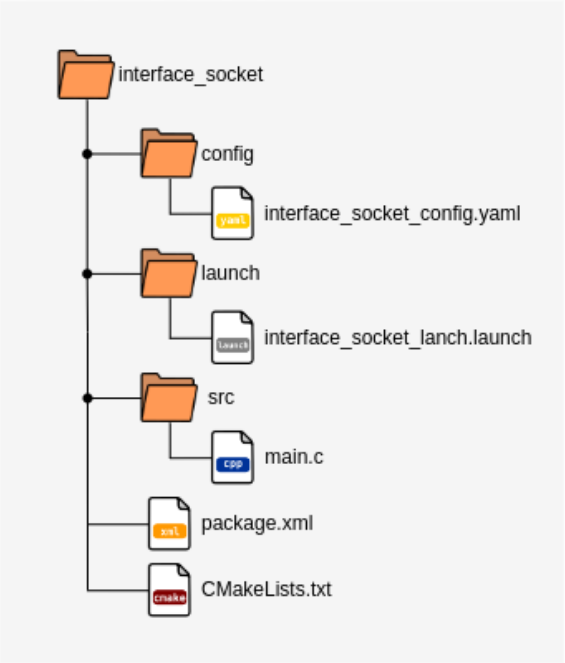
Arquitetura do Sistema

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)

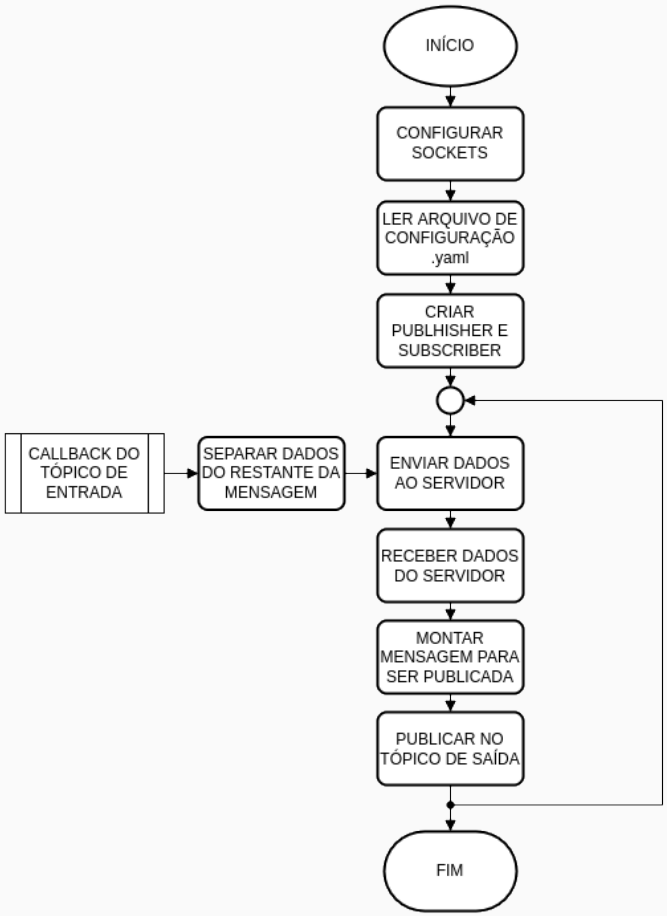
No desenvolvimento de aplicações com ROS deve ser respeitada a estrutura de diretórios de um pacote ROS, portanto o desenvolvimento do cliente deve lever em consideração essa estrutura.



Fonte:

Pacote ROS (cliente)

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



Fonte:

Parte III: Resultados

Cronograma

Metas físicas

1. Levantamento bibliográfico sobre os assuntos mais relevantes do projeto: ROS, Nios II, Verilog HDL, RTOS, TCP/IP Stack, Sockets.
2. Estudo detalhado do protocolo de comunicação entre os nós no ROS.
3. Desenvolvimento do sistema base do Nios II no Platform Designer.
4. Implementação do RTOS no sistema base.
5. Testes de comunicação TCP/IP entre o PC e o sistema embarcado no FPGA.
6. Desenvolvimento de uma aplicação de processamento de vídeo em FPGA.
7. Avaliação do desempenho do sistema proposto.
8. Elaboração da dissertação e publicação dos resultados.

Cronograma

Metas	Meses											
	1	2	3	4	5	6	7	8	9	10	11	12
Levantamento Bibliográfico	⊛											
Estudo protocolos ROS		⊛	⊛	⊛	⊛	⊛						
Desenv. do Nios II			⊛	⊛								
Implementação do RTOS				⊛	⊛	⊛						
Testes de Comunicação						⊛	⊛	⊛				
Desenv. do coprocessador								⊛	⊛	⊛	⊛	
Avaliação do desempenho										⊛	⊛	⊛
Elaboração da dissertação						⊛	⊛	⊛	⊛	⊛	⊛	⊛

Referencias i

Barry, R.

Effective Robotics Programming with ROS, 3 ed.

Packt Publishing, Birmingham, 2016.

Barry, R.

Mastering the FreeRTOS Real Time Kernel, 141204 ed.

Real Time Engineers Ltd, 2016.

Chu, P. P.

Embedded SoPC Design with Nios II Processor and Verilog Examples.

Wiley, Hoboken, 2012.

Referencias ii

Meyer-Baese, U.
Digital Signal Processing with Field Programmable Gate Arrays, 3 ed.
Springer, Nova York, 2007.

Nongnu.
Lwip - lightweight ip stack: Overview.
Savannah, 2018.

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) 1, 1 (2005), 62–67.