

Nestor Dias Pereira Neto

**Desenvolvimento de um co-processador de
vídeo em FPGA para integração com o Robot
Operating System - ROS**

Salvador

30 de abril 2022

Nestor Dias Pereira Neto

Desenvolvimento de um co-processador de vídeo em FPGA para integração com o Robot Operating System - ROS

Esta Dissertação de Mestrado foi apresentada ao Programa de Pós Graduação em Engenharia Elétrica da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica.

Universidade Federal da Bahia - UFBA

Escola Politécnica

Programa de Pós-Graduação em Engenharia Elétrica

Orientador: Wagner Oliveira

Coorientador: Paulo César

Salvador

30 de abril 2022

Nestor Dias Pereira Neto

Desenvolvimento de um co-processador de vídeo em FPGA para integração com o Robot Operating System - ROS/ Nestor Dias Pereira Neto. – Salvador, 30 de abril 2022-

35p. : il. (algumas color.) ; 30 cm.

Orientador: Wagner Oliveira

Dissertação (Mestrado) – Universidade Federal da Bahia - UFBA
Escola Politécnica

Programa de Pós-Graduação em Engenharia Elétrica, 30 de abril 2022.

1. Palavra-chave1. 2. Palavra-chave2. 2. Palavra-chave3. I. Orientador. II. Universidade xxx. III. Faculdade de xxx. IV. Título

Nestor Dias Pereira Neto

Desenvolvimento de um co-processador de vídeo em FPGA para integração com o Robot Operating System - ROS

Esta Dissertação de Mestrado foi apresentada ao Programa de Pós Graduação em Engenharia Elétrica da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica.

Trabalho aprovado. Salvador, 24 de novembro de 2012:

Wagner Oliveira
Orientador

Professor
Convidado 1

Professor
Convidado 2

Salvador
30 de abril 2022

Resumo

Segundo a ??, 3.1-3.2), o resumo deve ressaltar o objetivo, o método, os resultados e as conclusões do documento. A ordem e a extensão destes itens dependem do tipo de resumo (informativo ou indicativo) e do tratamento que cada item recebe no documento original. O resumo deve ser precedido da referência do documento, com exceção do resumo inserido no próprio documento. (...) As palavras-chave devem figurar logo abaixo do resumo, antecedidas da expressão Palavras-chave:, separadas entre si por ponto e finalizadas também por ponto.

Palavras-chave: latex. abntex. editoração de texto.

Abstract

oihbqptipbõq4tnpot4photnj4yojnj4ynojp

Keywords: latex. abntex. text editoration.

Sumário

1	INTRODUÇÃO	11
1.1	Justificativa	13
1.2	Objetivos	13
1.2.1	Objetivo Geral	13
1.2.2	Objetivos Específicos	13
1.3	Organização	14
I	REFERENCIAIS TEÓRICOS	15
2	CYCLONE V: SYSTEM ON A CHIP - SOC	17
2.1	Field Programmable Gate Array - FPGA	17
2.2	Hard processor ARM	17
2.2.1	Embedded Linux	17
2.3	Kit de desenvolvimento DE10-nano	17
3	ROBOT OPERATING SYSTEM - ROS	19
3.1	Sistema multiagentes	19
II	DESENVOLVIMENTO	21
4	ARQUITETURA DO SISTEMA	23
5	PACOTE ROS (CLIENTE)	25
6	SERVIDOR	27
III	RESULTADOS	29
7	RESULTADOS ALCANÇADOS	31
8	CONCLUSÃO	33
	REFERÊNCIAS	35

1 Introdução

Nos últimos anos novas técnicas para construção de robôs tem estado em muita evidência, em especial áreas como robótica móvel e robótica colaborativa tem chamado bastante atenção dos pesquisadores. Uma das principais características dessas áreas é exigência de um alto grau de percepção do ambiente que rodeia o robô, além de uma execução mais precisa em seus movimentos, isto devido ao fato de que as atividades desempenhadas por robôs estão exigindo um nível cada vez maior de interação com as atividades desempenhadas por seres humanos.

Outro ponto que sempre teve bastante importância no desenvolvimento de novos robôs é a autonomia, tanto do ponto de vista do consumo energético quando na tomada de decisões, superar o nível de autonomia atual continua sendo um dos principais objetivos na pesquisa e desenvolvimento de novos robôs.

As consequências desses trabalhos podem ser percebida em robôs com

2 - a busca por sistemas com um grau maior de autonomia A robótica tem se caracterizado pelo grande nível de percepção do ambiente e pelos sistemas complexos de controle do movimentos

O ROS como um framework que esta se tornando o padrão no desenvolvimento de robótica

1 - Fazer um link com a complexidade dos sistemas robóticos modernos e o ROS

Agrupar inúmeros "blocos" de softwares usados em robótica, fornecer drivers para hardwares específicos (sensores e atuadores), gerenciar troca de mensagens entre os nós que fazem parte do sistema, são as funções do ROS. Essas características fazem com que o ROS seja reconhecido com um pseudo sistema operacional (PYO et al., 2017). Dessa maneira o ROS se tornou muito ágil no desenvolvimento de novas aplicações para robótica. Usando nós já desenvolvidos e testados por outros desenvolvedores podemos criar novos sistemas completos apenas gerenciando esses nós na rede interna do ROS. Essa abordagem fez com que o número de pacotes para o ROS cresça em uma taxa muito rápida, desde o ano de seu lançamento, 2007, até 2012 o ROS aumentou de 1 para 3699 pacotes (YAMASHINA et al., 2005).

Com essa distribuição de tarefas através de vários nós podemos criar sistemas cada vez mais complexos, apenas inserindo novos nós na rede, essa rede é gerenciada pelo ROS Master, que é apenas mais um nó do sistema, mas com a função de ser um servidor de nome e serviços para o restante dos nós. Ele identifica os nós na rede, assim todos os nós podem se comunicar com os outros através de conexões peer-to-peer, Figura 1. Para

desenvolver novas aplicações para o crescente grupo de pacotes ROS, o desenvolvedor deve respeitar os protocolos de comunicação da rede, as bibliotecas do ROS facilitam este trabalho, por já fornecer funções prontas para o desenvolvimento de novos códigos compatíveis e que possam se registrar na rede. Detalhes dos protocolos e interno podem ser visto em (ROS, 2011a), (ROS, 2018) e (ROS, 2011b).

«««< HEAD Por se tratar de um hardware configurável o FPGA é ideal para processamento digitais de sinais. O potencial que os FPGAs possuem para melhorar o ===== **Desenvolvimento com fpga, SoC. dificuldade e maior tempo de desenvolvimento**

1 - explicar a ideia do FPGA e a dificuldade do desenvolvimento

O potencial que os FPGAs possuem para melhorar o »»»> 688eea1c4a8a8466b873451fbfb594eb047e8a desempenho de sistemas que utilizam processamento digitais de sinal é conhecido já algum tempo, as possibilidades de paralelismo, criação de estruturas de DSP dedicadas à aplicação, são recursos muito interessantes que a possibilidade do hardware configurado oferecem. Em contra partida as facilidade de desenvolvimento encontradas em aplicações que fazem uso de softwares não são encontradas nas mesmas proporções no mundo do hardware, sendo assim:

Juntar fpga com robótica através do ros dando foco na facilidade de desenvolvimento

2 - Fala sobre o FPGA escolhido

explicar a parte da comunicação entre o computador/ros e o SoC

melhorar a comunicação, comunicação eficiente, pacote pronto e de fácil integração com qualquer sistema ros,

O mais genérico possível para se enquadrar a qualquer projeto é que o desenvolvedor tenha interesse em incluir um FPGA ao sistema

Definir o problema (a pergunta)

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

Este problema é o que o trabalho vai tentar responder, podendo assim outros pesquisadores possam usar os benefícios do uso do hardware dedicado integrados ao benefícios que o ROS fornecem aos sistemas robóticos.

1.1 Justificativa

Nos últimos anos novas técnicas para construção de robôs tem sido bastante estudadas, em especial uma área que tem sido bastante explorada é a robótica móvel. A principal características que tem sido buscada é cada vez fornecer mais autonomia ao sistemas robóticos o que torna seus softwares cada vez mais complexos, o que aumenta a necessidade do uso de processadores muito mais poderosos, consequentemente aumentando muito o consumo de energia. Entretanto a busca por mais autonomia, diz respeito também às baterias, que são as fontes de energia da maioria dos robôs móveis, o que provoca uma verdadeira briga entre poder de processamento e baixo consumo.

Sendo assim, o FPGA pode ser uma ótima alternativa para solucionar os problemas de aumento do poder de processamento em conjunto com baixo consumo de energia. Meyer-Baese (2007) descreve algumas vantagens dos FPGAs modernos para uso em processamento digitais de sinais, como as cadeias de fast-carry usadas para implementar MACs de alta velocidade e o paralelismo tipicamente encontrado em dedign implementados em FPGA. Por essas características o FPGA necessita de frequências menores de trabalho para alcançar desempenho equivalente ou superior às soluções baseadas em processadores, tornando a dissipação de energia consideravelmente menor.

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver uma solução para estabelecer comunicação entre *Field Programmable Gate Array - FPGA*, configurado como um co-processador de vídeo.

1.2.2 Objetivos Específicos

- Estudar os assuntos relevantes ao projeto: Verilog HDL, embedded linux, Cyclone V, TCP/IP Stack, ROS;
- Conhecer com detalhes os protocolos da rede TCP/IP usada para comunicação interna dos nós e serviços ROS;
- Implementa distribuição embedded linux para processador ARM;
- Estabelecer comunicação entre o ROS e o Cyclone V, através da tecnologia Gigabit Ethernet;
- Testar aplicações de processamento de vídeo em hardware em conjunto com ROS;
- Avaliar a performance com a inclusão do FPGA ao sistema.

1.3 Organização

No primeiro capítulo...

Parte I

Referenciais teóricos

2 Cyclone V: System on a Chip - SoC

2.1 Field Programmable Gate Array - FPGA

2.2 Hard processor ARM

2.2.1 Embedded Linux

2.3 Kit de desenvolvimento DE10-nano

3 Robot Operating System - ROS

3.1 Sistema multiagentes

Parte II

Desenvolvimento

4 Arquitetura do sistema

Para conseguirmos estabelecer a comunicação entre o computador e o SoC precisamos efetuar programação de sockets e bibliotecas específicas para trabalho em redes, desenvolver um pacote ROS para disponibilizar os dados recebidos através da interface de rede para os outros pacotes ROS do sistema robótico, além de um programa rodando no HPS do SoC para estabelecer esta comunicação entre a interface de rede da placa De10-nano e a aplicação sendo executada no FPGA. Já esta aplicação que estará embarcada no FPGA contido no SoC deverá ser descrita por alguma linguagem de descrição de hardware, como por exemplo, verilog ou VHDL.

Todas essas etapas descritas anteriormente são necessárias para a construção completa do sistema proposto, o que torna o desenvolvimento da solução completa um desafio devido às diferentes ferramentas de software e hardware necessárias para sua conclusão. Tendo em vista este problema, a solução foi idealizada para conter o maior grau de modularidade possível, ou seja, cada uma dessas etapas será tratada com um projeto independente, apenas tendo cuidado para garantir a correta comunicação entre cada uma delas.

que por sua vez deverá conter a aplicação desenvolvida em alguma linguagem de descrição de hardware .

5 Pacote ROS (cliente)

6 Servidor

Parte III

Resultados

7 Resultados Alcançados

8 Conclusão

Referências