



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

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

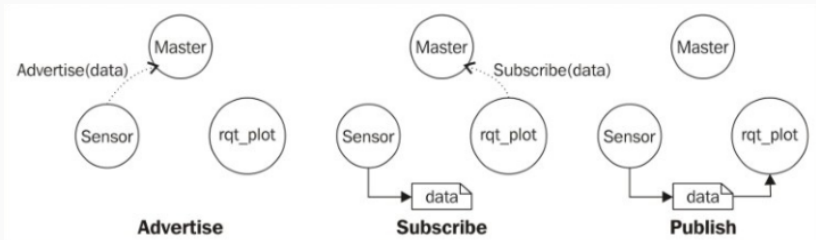
1. Introdução
2. Parte I: Referenciais Teórico
3. Parte II: Desenvolvimento
4. Parte III: Resultados

Introdução

- ROS um pseudo sistema operacional para robótica.
- Proporciona reutilização de um grande número de nós já desenvolvidos e testados por terceiros.
- Pode-se criar novos sistemas completos apenas gerenciando esses nós na rede interna do ROS.
- O número de pacotes para o ROS cresceu em uma taxa muito rápida, desde o ano de seu lançamento, 2007, até 2012 o ROS aumentou de 1 para 3699 pacotes.

- Com a distribuição de tarefas através de vários nós, pode-se criar sistemas cada vez mais complexos, apenas inserindo novos nós na rede ROS.
- ROS Master tem a função de ser um servidor de nome e serviços para o restante dos nós. Ele identifica os nós na rede.

Figure 1: ROS Master



Fonte:[1]

- Por se tratar de um hardware configurável o FPGA é ideal para processamento digitais de sinais.
- O FPGA está próximos a revolucionar o processamento digital de sinais, assim como os DSPs fizeram algumas décadas atrás.
- As facilidade de desenvolvimento encontradas em aplicações que fazem uso de softwares não são encontradas nas mesmas proporções no mundo do hardware.

- Como estabelecer a comunicação entre o ROS e um sistema de processamento de vídeo embarcado em um FPGA?
- O estabelecimento da comunicação entre o FPGA e o ROS e a execução do processamento de vídeo de forma paralela, embarcada em um FPGA, pode melhorar o desempenho do sistema?

- Nos últimos anos, novas técnicas para construção de robôs têm sido bastante estudadas e a robótica móvel tem recebido grande atenção.
- Busca por maior autonomia, sistemas mais complexos.
- FPGA uma opção para aumento de desempenho computacional, combinado com baixo consumo.
- Poucas pesquisas relacionadas ao tema.
- Reaproveitamento da pesquisa em outros projetos das mais diversas áreas.

Objetivo geral

- Desenvolver uma solução para estabelecer comunicação entre *Field-Programmable Gate Array* - *FPGA*, configurado como um coprocessador de vídeo e o *Robot Operating System* - *ROS*, avaliando o impacto desta aplicação ao sistema.

Objetivos específicos

- Estudar os assuntos relevantes ao projeto: Verilog HDL, RTOS, Nios II, TCO/IP Stack, ROS.
- Conhecer com detalhes os protocolos da rede TCP/IP usada para comunicação interna dos nós e serviços ROS.
- Desenvolver plataforma com Nios II como base para o andamento do projeto.
- Implementar um sistema operacional de tempo real - RTOS na plataforma base.
- Estabelecer comunicação entre o ROS e o sistema Nios II (embarcado no FPGA) através da tecnologia Gigabit Ethernet.
- Testar aplicações de processamento de vídeo em hardware em conjunto com ROS.
- Avaliar a desempenho com a inclusão do FPGA ao sistema.

Parte I: Referenciais Teórico

Para alcançar o objetivo desta pesquisa será necessário desenvolver um sistema, configurado em FPGA, que seja capaz de conectar-se a uma rede TCP/IP. O sistema contará com:

- Soft processor Nios II;
- Sistema operacional de tempo real - RTOS;
- Biblioteca TCP/IP stack.

Parte II: Desenvolvimento

Procedimentos Metodológicos

A pesquisa será realizada em duas fases:

Primeira fase:

- Levantamento de informações teóricas sobre as tecnologias relacionadas com o tema.

Segunda fase

- Serão desenvolvidos procedimentos, técnicas, algoritmos, circuitos e de todos os procedimentos práticos necessários para alcançar o objetivo da pesquisa.

Plano de trabalho

- Será apresentado com detalhes nas metas físicas na seção cronograma.

Materiais e infra-estrutura disponível

- Para desenvolvimento do trabalho será utilizado o kit de desenvolvimento DE2-115 da Terasic, que conta com um FPGA Intel EP4CE115 da família Cyclone IV. Inicialmente os teste com o ROS serão no ambiente de simulação Gazebo.

Matérias cursadas

Todos os créditos obrigatórios com disciplinas já foram concluídos.

- Processamento Digitais de Sinais (PPGESP IFBA).
- Processadores Digitais de Sinais - 8,5.
- Inteligência Artificial - 8,0.
- Robótica Móvel - 9,5.
- Processamento Estatístico de Sinais - 8,4.
- Componentes de Processadores Digitais de Sinais - 8,1.

Atividades desenvolvidas

Algumas atividades já foram concluídas.

- Revisão bibliográfica, estudo de trabalhos relacionado.
- Conhecimento das ferramentas utilizadas.
- Testes com sistema Nios II.
- Implementação do FreeRTOS no NiosII.

Parte III: Resultados

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 | | | | | | (*) | (*) | (*) | (*) | (*) | (*) | (*) |



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.



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.