

Trabalho

Título em Português: Controladores para sistemas embarcados de átomos frios

Título em Inglês: Controladores para sistemas embarcados de átomos frios

Autor: Anderson Eduardo de Carvalho Filho

Instituição: Universidade de São Paulo

Unidade: Instituto de Física de São Carlos

Orientador: Daniel Varela Magalhães

Área de Pesquisa / SubÁrea: Física Atômica e Molecular

Agência Financiadora: USP - Programa Unificado de Bolsas

CONTROLADORES PARA SISTEMAS EMBARCADOS DE ÁTOMOS FRIOS

Anderson Eduardo de Carvalho Filho

Prof. Dr. Daniel Varela Magalhães

Instituto de Física de São Carlos/Universidade de São Paulo

anderson.carvalho52@usp.br

Objetivos

O principal objetivo do trabalho é o desenvolvimento de uma arquitetura para a operação de um padrão de frequência embarcado, onde os subsistemas possuem processamento próprio e podem se comunicar com um sistema supervisor em mais alto nível de protocolo. Objetivos secundários dizem respeito ao entendimento de operação de um experimento de referência de frequência com átomos frios e as exigências para uma operação embarcada. Além disso, explorar computadores em módulo para operar como sistema supervisor personalizado. A definição de diferentes formas de comunicação entre os subsistemas e o módulo supervisão também estará em foco, tendo em vista as especificidades de operação, volume de dados e necessidades de ajustes a cada subsistema

Métodos e Procedimentos

O projeto teve início com a utilização da BeagleBone Black Rev C fornecida pela universidade, e a seleção cuidadosa de uma distribuição LinuxAM335x 12.2 2023-10-07 IoT. [1] A escolha foi baseada em critérios como estabilidade e compatibilidade com o hardware. Após a instalação e testes de conectividade e periféricos, foi realizada uma análise detalhada das interfaces de comunicação, como GPIO, UART, I2C e SPI. Essa exploração do

hardware permitiu um entendimento claro das capacidades e limitações da plataforma, estabelecendo uma base sólida para o desenvolvimento subsequente.

Na etapa de comunicação, o protocolo CAN foi selecionado por sua robustez e adequação a ambientes industriais. Para o desenvolvimento do software, optou-se pela linguagem Python, com o uso da biblioteca Adafruit-BBIO, o que facilitou a integração de periféricos e acelerou o processo de desenvolvimento. Essa combinação de escolhas técnicas visa garantir um sistema embarcado eficiente e confiável, alinhado às necessidades do projeto.

Resultados

Com base na escolha do protocolo CAN, delineou-se um plano futuro para o desenvolvimento do projeto. Isso inclui a implementação prática do sistema embarcado utilizando o protocolo selecionado, juntamente com o desenvolvimento do sistema supervisor personalizado. O plano também prevê etapas adicionais, como a integração dos subsistemas, testes de validação e ajustes finais do sistema.

Em resumo, a escolha do protocolo CAN foi fundamentada em uma análise abrangente e alinhada com os objetivos e requisitos do projeto. Sua seleção proporciona uma base sólida para o desenvolvimento bem-sucedido

do sistema embarcado e a realização dos objetivos.

Com a definição bem-sucedida do protocolo de comunicação e a justificativa clara da escolha, os passos seguintes do projeto foram implementados com êxito, resultando na criação prática do sistema embarcado na BeagleBone Black. Os subsistemas individuais foram desenvolvidos e integrados de forma eficiente, utilizando um módulo CAN para facilitar a comunicação entre eles. A programação e configuração dos dispositivos periféricos foram realizadas com sucesso, e os protocolos de comunicação específicos.

Além disso, se teve êxito em utilizar outros protocolos em conjunto com o CAN. O protocolo I2C nativo da beaglebone foi implementado como uma adição complementar. Combinar **I2C** e **CAN** em um sistema embarcado pode trazer uma série de vantagens, aproveitando os pontos fortes de cada protocolo para diferentes necessidades. O **CAN** é ideal para comunicação em rede com múltiplos dispositivos distribuídos, oferecendo alta taxa de transferência, robustez, e uma excelente capacidade de operar em ambientes ruidosos ou industriais. Ele tem um bom funcionamento para comunicação de longo alcance entre subsistemas que requerem alta confiabilidade e tolerância a falhas.

Por outro lado, o **I2C** é ótimo para comunicação local entre dispositivos periféricos de baixa velocidade, como sensores e displays, que estão fisicamente próximos. Ele é simples de implementar e consome poucos pinos, o que é vantajoso quando há limitações de hardware. Usar I2C em conjunto com o CAN permite que o sistema utilize o CAN para a comunicação em redes mais amplas e complexas, enquanto o I2C pode ser usado para gerenciar pequenos componentes de forma eficiente.

A combinação dos dois protocolos otimizou o desempenho do sistema: o **CAN** gerencia a comunicação crítica e de longa distância, enquanto o **I2C** lida com a comunicação rápida

e de baixa complexidade entre periféricos próximos. Dessa forma, é possível manter a flexibilidade e eficiência no controle de dispositivos, sem sobrecarregar um único protocolo com todas as responsabilidades.

Finalmente, testes foram conduzidos para validar o funcionamento do sistema como um todo. E utilizando um como receptor um arduino Nano, conseguiu-se realizar um bus em 0,7 Ms. A comunicação entre subsistemas, a integridade de dados, Os resultados desses testes confirmaram o alto desempenho e a confiabilidade do sistema, assegurando sua operação eficaz em um ambiente prático. Com a implementação e validação concluídas com sucesso, o sistema embarcado é válido para operar um sistema de comunicação, podendo ser ainda mais otimizado.

Conclusões

O desenvolvimento do projeto utilizando a BeagleBone Black Rev C foi altamente satisfatório, com a escolha de ferramentas como a linguagem Python e a biblioteca Adafruit-BBIO proporcionando um ambiente de desenvolvimento eficiente e flexível. A instalação do sistema operacional e a exploração detalhada do hardware permitiram uma integração robusta dos periféricos e interfaces de comunicação. A implementação do protocolo CAN em conjunto com o I2C nativo como barramento de comunicação cumpriu seu papel para garantir a troca eficiente de informações entre os diversos subsistemas.

Referências

[1] **BEAGLEBOARD.** *BeagleBoard.org Distros.* Disponível <https://www.beagleboard.org/distros>. Acesso em: 20 de out 2024.

Anderson Eduardo de Carvalho Filho

Prof. Dr. Daniel Varela Magalhães

São Carlos Institute of Physics/University of São Paulo

anderson.carvalho52@usp.br

Objectives

The main objective of this work is the development of an architecture for the operation of an embedded frequency standard, where the subsystems have their own processing and can communicate with a supervisory system at a higher protocol level. Secondary objectives include understanding the operation of a cold atom frequency reference experiment and the requirements for embedded operation. Additionally, it aims to explore module computers to operate as a customized supervisory system. The definition of different forms of communication between the subsystems and the supervision module will also be a focus, considering the specificities of operation, data volume, and the need for adjustments in each subsystem.

Materials and Methods

The main objective of this work is the development of an architecture for the operation of an embedded frequency standard, where the subsystems have their own processing and can communicate with a supervisory system at a higher protocol level. Secondary objectives include understanding the operation of a cold atom frequency reference experiment and the requirements for embedded operation. Additionally, it aims to explore module computers to operate as a customized supervisory system. The definition of different

forms of communication between the subsystems and the supervision module will also be a focus, considering the specificities of operation, data volume, and the need for adjustments in each subsystem.

Results

Based on the choice of the CAN protocol, a future plan for project development was outlined. This includes the practical implementation of the embedded system using the selected protocol, along with the development of a customized supervisory system. The plan also includes additional stages, such as subsystem integration, validation tests, and fine-tuning of the system. In summary, the choice of the CAN protocol was based on a comprehensive analysis and aligned with the project's objectives and requirements. Its selection provides a solid foundation for the successful development of the embedded system and the achievement of objectives.

With the successful definition of the communication protocol and a clear justification for the choice, the subsequent steps of the project were successfully implemented, resulting in the practical creation of the embedded system on the BeagleBone Black. The individual subsystems were developed and integrated efficiently, using a CAN module to

facilitate communication between them. The programming and configuration of peripheral devices were successfully carried out, along with the specific communication protocols.

Additionally, success was achieved in using other protocols in conjunction with CAN. The BeagleBone's native I2C protocol was implemented as a complementary addition. Combining I2C and CAN in an embedded system can bring several advantages, leveraging the strengths of each protocol for different needs. CAN is ideal for network communication with multiple distributed devices, offering high transfer rates, robustness, and excellent capability to operate in noisy or industrial environments. It works well for long-distance communication between subsystems that require high reliability and fault tolerance. On the other hand, I2C is great for local communication between low-speed peripheral devices, such as sensors and displays, that are physically close. It is simple to implement and consumes few pins, which is advantageous when hardware limitations exist. Using I2C in conjunction with CAN allows the system to use CAN for broader and more complex network communication, while I2C can be used to manage small components efficiently.

The combination of the two protocols optimized the system's performance: CAN manages critical and long-distance communication, while I2C handles fast and low-complexity communication between nearby peripherals. This way, it is possible to maintain flexibility and efficiency in device control without overloading a single protocol with all responsibilities.

Finally, tests were conducted to validate the system's operation as a whole. Using an Arduino Nano as a receiver, a bus of 0.7 Ms was achieved. Communication between subsystems and data integrity were confirmed.

The results of these tests confirmed the system's high performance and reliability, ensuring its effective operation in a practical environment. With the successful implementation and validation, the embedded system is validated to operate a communication system and can be further optimized.

Conclusions

The project development using the BeagleBone Black Rev C was highly satisfactory, with the choice of tools such as the Python language and the Adafruit-BBIO library providing an efficient and flexible development environment. The installation of the operating system and detailed exploration of the hardware allowed for a robust integration of peripherals and communication interfaces. The implementation of the CAN protocol in conjunction with the native I2C as a communication bus fulfilled its role in ensuring efficient information exchange between the various subsystems.

References

- [1] BEAGLEBOARD. BeagleBoard.org Distros. Available at <https://www.beagleboard.org/distros>. Accessed on: Oct 20, 2024.