

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский ядерный университет «МИФИ»

Обнинский институт атомной энергетики –

филиал федерального государственного автономного образовательного учреждения высшего
образования «Национальный исследовательский ядерный университет «МИФИ»
(ИАТЭ НИЯУ МИФИ)

Отделение Интеллектуальные кибернетические системы
Направление подготовки Информатика и вычислительная техника

Научно-исследовательская работа
Анализ кодогенераторов для CANopen

Студент группы ИВТ-Б22 _____ Карасев Н. А.

Руководитель
инженер-программист _____ Жильцов Д. И.

Обнинск, 2025 г

РЕФЕРАТ

Работа 9 стр., 0 табл., 0 рис., 0 ист.

Ключевые слова: CAN, CANOPEN, RUST

Написать нормальный реферат в конце

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
ЗАКЛЮЧЕНИЕ	8

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В настоящем отчете о НИР применяют следующие термины с соответствующими определениями:

Позже

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

Позже

ВВЕДЕНИЕ

Для введения в столько конкретную тему стоит рассказать что вообще что такое CAN.

Допустим, вы являетесь инженером-электронщиком и разрабатываете различные электронные механизмы. У этих механизмов вполне могут быть разнесены некоторые элементы, например какой-то датчик находится в одном месте, а блок обработки сигналов - в другом. В таком случае самым тривиальным решением будет взять и соединить их проводами! Однако такой подход не всегда является оптимальным и рано или поздно вы столкнётесь с проблемой вездесущности этих самых проводов и кабелей. Огромные траты материалов на проводку - не самая большая проблема, намного хуже, на мой взгляд - обслуживать потом такую систему, разобраться среди десятков и сотен различных проводов крайне сложно. Немного подумав, вы решаете объединить какие-то провода в жгуты, а следующим логическим шагом является переход от соединений "точка-точка" к шинной архитектуре, где по общей линии передаются сигналы между различными устройствами.

Но теперь вы сталкиваетесь с другой проблемой - как научить устройство принимать только те сигналы которые назначались конкретно ему? Можно ввести какое-нибудь мультиплексирование по времени, но как быть с системами реального времени в которых дорога каждая секунда или крайне высока цена ошибки? Одним из способов заставить десятки электронных блоков в машине или промышленной установке обмениваться данными по одной общей шине так, чтобы это было надёжно, предсказуемо по времени и устойчиво к помехам - является введение шины CAN.

Для решения вышеописанных проблем CAN предлагает простое и в то же время мощное решение - задание каждому сообщению своего идентификатора. В такой системе:

Каждый узел обрабатывает только то сообщение, которое назначалось конкретно ему;

Арбитраж происходит без разрушения кадра - при передаче сигналов от нескольких узлов одновременно победит то сообщение, у которого идентификатор приоритетнее

CAN самостоятельно контролирует корректность данных на уровне канала

Однако CAN - всего лишь шина, он даёт транспорт для коротких сообщений, но в сами сообщения он не лезет - для этого нужен какой-то надстро-ечный протокол на более абстрактном уровне. Здесь и возникает CANopen.

CANopen задаёт общий прикладной каркас: определяет, как устройства описывают свои параметры, как ими управлять, как передавать “процессные” данные, как диагностировать аварии, и как сеть в целом живёт от включения питания до штатной работы. Этот протокол

Дописать введение и нормально подвести к конкретным решениям:

- *Zencan*
- *canopen*
- *canopen_rust*
- *Вероятнее всего рассмотреть решения на C*

ЗАКЛЮЧЕНИЕ