Эмулятор ProtoEMU

Редакция от 10.10.2026 · docs.kb-agava.ru

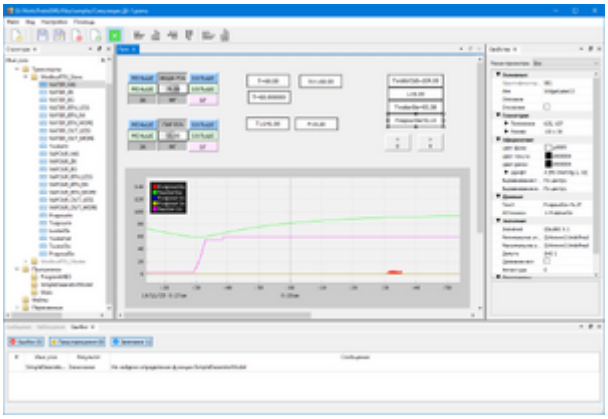
Содержание

1 Возможности эмулятора 2
  1.1 Основные 2
  1.2 Протоколы 2
  1.3 Визуализация 2
  1.4 Отладка 3
  1.5 Другие возможности 3
2 Лицензирование 3
3 Дистрибутивы 3
4 Версии 3
5 Документация 3
6 Принцип работы 4
  6.1 Структура пульта 5
7 Отладка с эмулятором 5
8 Создание отладочного стенда и эмуляция оборудования 5
9 Автоматизация и симуляция в проекте 6
  9.1 Использование программ для симуляции и других задач 6

Эмулятор ProtoEMU - специализированное приложение для эмуляции устройств, работающих по протоколам Modbus-RTU, Modbus-TCP.

Эмулятор позволяет заменить реальные устройства, используемые в системах автоматизации, диспетчеризации или других, их виртуальными программными аналогами. Это позволяет повысить удобство и скорость разработки информационных и управляющих систем, взаимодействующих с оборудованием, а так же исключить необходимость отладки систем на объекте, используя действующие установки.

1 Возможности эмулятора



ProtoEMU - симуляция деаэратора

1.1 Основные

- Эмуляция физических входных и выходных сигналов.
- Эмуляция внутренних переменных, доступных через карту регистров.
- Ручное управление состоянием устройства с помощью различных виджетов.
- Автоматическое управление состоянием устройства с помощью встроенных программ и подпрограмм.

1.2 Протоколы

ProtoEMU поддерживает следующие протоколы обмена (транспорты) с реальными устройствами:

- Modbus-RTU в режимах ведущий и ведомый.
- Modbus-TCP в режимах ведущий и ведомый.

1.3 Визуализация

1. Возможность создания нескольких экранов (форм) для визуализации состояния эмулируемого устройства.
2. Разнообразные виджеты для отображения значений и управления состоянием эмулируемого устройства:
    - Аналоговый индикатор.
    - Цифровой индикатор.
    - Кнопка.
    - График.

1.4 Отладка

1. Проверка корректности конфигурации виртуальных устройств.
2. Просмотр значений параметров в ходе эмуляции.
3. Ведение журналов событий, возникающих при работе эмулятора, в том числе журнала обмена с устройствами, включая весь трафик. Возможность фильтрации событий журнала "на лету".

1.5 Другие возможности

1. Поддержка автоматической идентификации параметров регистров на основе файлов идентификаторов.
2. Запись значений параметров в архив в течение сеанса эмуляции для последующего анализа и распечатки.

Полное описание эмулятора содержится в документе Руководство пользователя ProtoEMU.

2 Лицензирование

Доступно два способа лицензирования:

1. Аппаратная лицензия (USB ключ). Программу можно свободно использовать на любом компьютере при условии подключения ключа.
2. Программная лицензия. Данный способ лицензирования не требует наличия USB ключа, но требует доступа в сеть Internet. Привязка лицензии осуществляется к компьютеру.

3 Дистрибутивы

- Дистрибутивы с аппаратной лицензией: https://files.kb-agava.ru/s/m3trHHoaM4DwoGF
- Дистрибутивы с программной лицензией: https://github.com/taushkanov/ProtoEMU/releases

4 Версии

1. Перечень изменений ProtoEMU 1.0.

5 Документация

Полный перечень документации ProtoEMU размещен на странице-каталоге.

6 Принцип работы

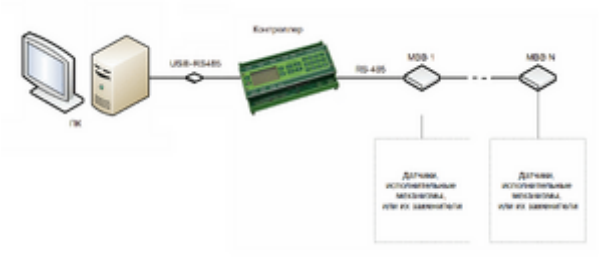


Рис. 1 Классическая схема стенда для отладки ПО

Классическая схема устройства стенда, на котором производится отладка программного обеспечения, приведена на рисунке 1. В ней участвуют контроллер, модули ввода-вывода и подключаемые к ним датчики, исполнительные механизмы или их заменители (тумблеры, лампочки-индикаторы и т. д).

Применение эмулятора ProtoEMU позволяет максимально упростить устройство отладочного стенда и даже сделать его портативным!

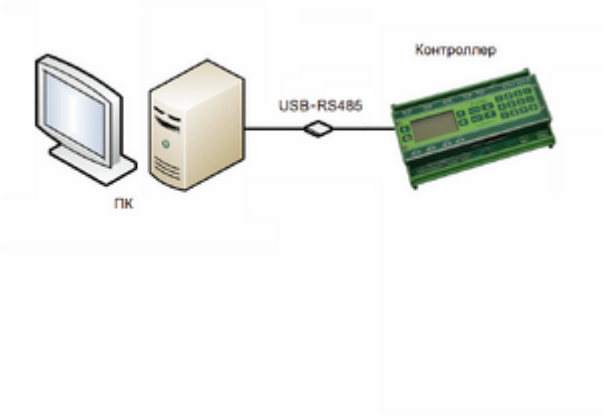


Рис. 2 Схема стенда с эмулятором для отладки ПО с подключением по RS-485

Из первоначальной схемы в нем остается только контроллер с отлаживаемым программным обеспечением, к которому через преобразователь USB-RS485 подключается персональный компьютер с эмулятором (см. рисунок 2).

При использовании эмулятора не требуется адаптировать отлаживаемое ПО, чтобы оно работало с эмулятором. Для отлаживаемого ПО замена реального оборудования на эмулятор прозрачна.

Дополнительно к имитации реального оборудования эмулятор позволяет имитировать поведение реального оборудования с помощью встраиваемых в эмулятор алгоритмов: в ответ на установку контроллером определенных выходных сигналов эмулятор позволяет запрограммировать изменение значений других сигналов. С помощью такой функции можно автоматизировать и существенно ускорить проверку и отладку программного обеспечения контроллера.

Для использования эмулятора необходимо создать проект (пульт), описывающий эмулируемую установку.

6.1 Структура пульта

Пульт содержит следующие типы узлов:

- Транспорт - выполняет сбор и первичную обработку внешних сигналов.
- Формы с виджетами - выполняют визуализацию собранных данных и взаимодействие с пользователем.
- Программы (необязательны) - выполняют дополнительную обработку собранных данных или реализуют алгоритмы, автоматизирующие взаимодействие с внешним оборудованием.
- Файлы (необязательны) - содержат идентификаторы, хранящие информацию о связи с внешними сигналами.
- Переменные (необязательны) - реализуют связь с переменными в программах.

7 Отладка с эмулятором

При использовании эмулятора процесс отладки выглядит следующим образом:

- контроллер производит обращение к модулям ввода-вывода по линии связи RS-485, которая подключена к эмулятору;
- эмулятор принимает запрос от контроллера и дает ответ, содержащий информацию о состоянии эмулируемых датчиков и исполнительных механизмов;
- при необходимости изменения состояний датчиков или исполнительных механизмов пользователь вводит эти изменения в эмулятор или эмулятор самостоятельно изменяет состояния с помощью заложенных алгоритмов.

8 Создание отладочного стенда и эмуляция оборудования

Для работы с эмулятором необходимо:

1. Создать в эмуляторе новый проект (пульт).
2. Добавить в проект транспорт нужного типа из перечня поддерживаемых.
3. Добавить в транспорт на основе описания эмулируемого устройства набор нужных "источников" - узлов проекта, содержащих передаваемые контроллером значения. Например, для транспортов типа Modbus, источниками являются регистры Modbus.
4. Добавить в проект форму, на которой разместить виджеты, отображающие значения источников и позволяющие управлять их значениями.

На этом этапе создания проекта уже можно в ручном режиме управлять состоянием "виртуального" устройства с помощью виджетов.

Далее можно реализовать в проекте функции автоматического управления состоянием виртуального устройства с помощью набора алгоритмов - программ и обработчиков событий в виджетах.

Виртуальное устройство с помощью программ способно самостоятельно менять свое состояние в зависимости от внешних воздействий.

9 Автоматизация и симуляция в проекте

9.1 Использование программ для симуляции и других задач

При необходимости автоматического изменения переменных, состояния входов и выходов по различным алгоритмам в проекте можно использовать программы.

Программа - отдельный узел, содержащий реализацию алгоритмов на диалекте языка C++ с помощью следующий конструкций:

- Функция.
- Класс с методами, конструкторами и пр.

Пример программы:

```
int SimulateBoiler()
{
    boiler.update(5);

    g_dTemperature = boiler.getTemperature();
    g_dFuelRate = boiler.getCurrentFuelRate();
    g_dHeatOutput = boiler.getCurrentHeatOutput();

    g_dTime += dt;
    Twater = g_dTemperature;

    return 0;
}

SimpleModel boiler(T_setpoint);
```

Подробно язык программирования ProtoEMU и его использование для автоматизации работы описан в отдельном руководстве.