

УДК 681.5

UDC 681.5

**ГРАФИЧЕСКИЙ МЕТОД
ОТОБРАЖЕНИЯ ВНУТРЕННЕЙ
СТРУКТУРЫ ПРОЕКТА В
ИНТЕГРИРОВАННОЙ
SCADA-СИСТЕМЕ TRACE MODE**

**GRAPHIC DISPLAY METHOD OF
INTERNAL PROJECT STRUCTURE AT
INTEGRATED SCADA-SYSTEM
TRACE MODE**

*А. Г. Поздняков, В. Х. Аль-Тибби,
В. Н. Анисимов*

*A. G. Pozdniakov, W. H. Al-Tibbi,
V. N. Anisimov*

Донской государственный технический университет, Ростов-на-Дону, Российская Федерация

Don State Technical University, Rostov-on-Don, Russian Federation

Pozdn2000@mail.ru
nb1979@mail.ru

Pozdn2000@mail.ru
nb1979@mail.ru

Статья посвящена изучению основ работы SCADA-системы TRACE MODE. Взаимодействие различных элементов структуры узла проекта рекомендуется рассматривать в форме их графического отображения. Предлагаемая методика апробирована на примере создания нового проекта и может быть интересна для специалистов, приступающих к работе со SCADA-системой TRACE MODE.

This article focuses on the fundamentals of operation of SCADA-system TRACE MODE. The interaction of different elements of a structure node in the project is recommended to consider in the form of their graphical display. The proposed method is shown on the example of a new project creation.

This article is intended for technical professionals who work with SCADA-system TRACE MODE.

Ключевые слова: SCADA, Trace mode, графическая структура.

Keywords: SCADA, Trace mode, graphic structure

Введение. При разработке вновь создаваемых проектов в интегрированной среде Trace Mode и при анализе уже существующих требуется определенный опыт для понимания внутренней структуры проекта и определении взаимосвязей его отдельных компонентов. При изучении инструментальной системы на начальном этапе и для улучшения понимания взаимодействия между компонентами проекта возникает необходимость в более наглядном представлении внутренней структуры проекта. Для этого предлагается графическое представление отдельных компонентов проекта и их внутренней взаимосвязи.

Методика отображения структуры проекта. Эффективность предлагаемого метода создания и графического отображения компонентов проиллюстрируем на примере проекта, в котором требуется вывести на экран фигуру Лиссажу в виде окружности [1,2].

Рассмотрим подробнее этапы разработки проекта.

1) Создадим новый проект со стандартным уровнем сложности;

2) Добавим в слое «Системы» узел RTM;

3) В слое «Шаблоны экранов» создадим новый шаблон Экран;

4) Методом drag&drop перетащим вновь созданный экран из шаблонов на созданный узел монитора реального времени (RTM). При этом в узле RTM появится канал «Экран» типа CALL, служащий для вызова только что созданного шаблона экрана.

5) Для реализации фигуры Лиссажу откроем шаблон экрана на редактирование и в таблице «Аргументы экрана» создадим два аргумента экрана X и Y. Тип аргументов — входной (In), а тип данных — вещественный (Real);

6) На экране разместим графический элемент тренд XY. В его свойствах в закладке «Кривые» создадим кривую 1 и привяжем X и Y к одноименным аргументам экрана;

7) Фигуру Лиссажу будем создавать, подавая на вход X тренда синусоиду, а на вход Y — эту же синусоиду, но с задержкой на 25 тактов работы программы. Для этого в слое «Источники/Приемники» добавим группу «Генераторы» и в ней создадим компонент «Синусоида»;

8) С помощью механизма drag&drop перетащим синусоиду в слой «Каналы» узла RTM. При этом будет создан канал типа Float с одноименным названием и одновременной привязкой к источнику сигнала. Таким образом, компонент синусоида через канал типа Float передается в аргумент канала «Вызов экрана». Отметим, что атрибуты для передачи данных есть только у канала типа «Вызов», а у самого шаблона экрана этих атрибутов нет, т.е. канал типа «Вызов» используется для передачи данных в шаблон экрана через свои атрибуты;

9) Для реализации задержки синусоидального сигнала в слое «Каналы» создадим компонент «Программа» [3]. При этом в правом окне навигатора появится канал «Программа» типа CALL и одновременно в слое «Шаблоны программ» создастся одноименный шаблон «Программа», который в проекте будет вызываться каналом «Программа». Связь канала «Программа» и «Шаблона программы», а также принцип их создания аналогичны принципу связи «Шаблона экрана» и канала «Вызов экрана»;

10) Во вкладке «Аргументы программы» создадим два аргумента: первый — ARG_000 (тип аргумента — In, тип данных — Real) привяжем к атрибуту «Реальное значение числового канала Синусоида», а второй аргумент — ARG_001 (тип аргумента — Out, тип данных — Real) привяжем к атрибуту «Входное значение числового канала типа Float» с именем «Задержка», который создадим вручную в слое «Каналы». Для реализации задержки напишем программу на языке FBD, которая будет состоять из девяти последовательно соединенных звеньев «Запаздывание» (находятся во вкладке «Выбор Палитры FBD блоков») через выход Q3 звеньев. Аргумент программы ARG_000 свяжем со входом INP первого звена запаздывания, а аргумент программы ARG_001 свяжем с выходом Q1 последнего звена. Для привязки задержанного и незадержанного синусоидальных сигналов к аргументам экрана откроем свойства экрана и во вкладке «Аргументы» привяжем аргументы X и Y к атрибутам «Реальное значение числовых каналов Задержка и Синусоида» соответственно.

Если представить содержание и взаимосвязи созданного проекта визуально, получим графическую структуру проекта, показанную на рисунке 1.

Числовой канал «Синусоида» является промежуточным звеном для передачи численных значений от источника «Синусоида» в каналы типа CALL «Экран» и «Программа», а числовой канал «Задержка» выполняет эту же функцию между каналами «Программа» и «Экран» типа CALL. Для обеспечения обмена данными между программой и каналами узла служат аргументы программы. Тип каждого аргумента определяет направление передачи данных. Так, если аргумент служит для передачи значения в программу, следует выбрать тип аргумента IN, а если для передачи из программы — тип аргумента OUT. В ряде случаев нужен аргумент, который будет передавать данные как в программу, так и из нее, тогда следует выбрать тип аргумента IN/OUT. Для обмена данными программы с тем или иным каналом необходимо произвести привязку каждого аргумента программы к тому или иному каналу (атрибуту канала).

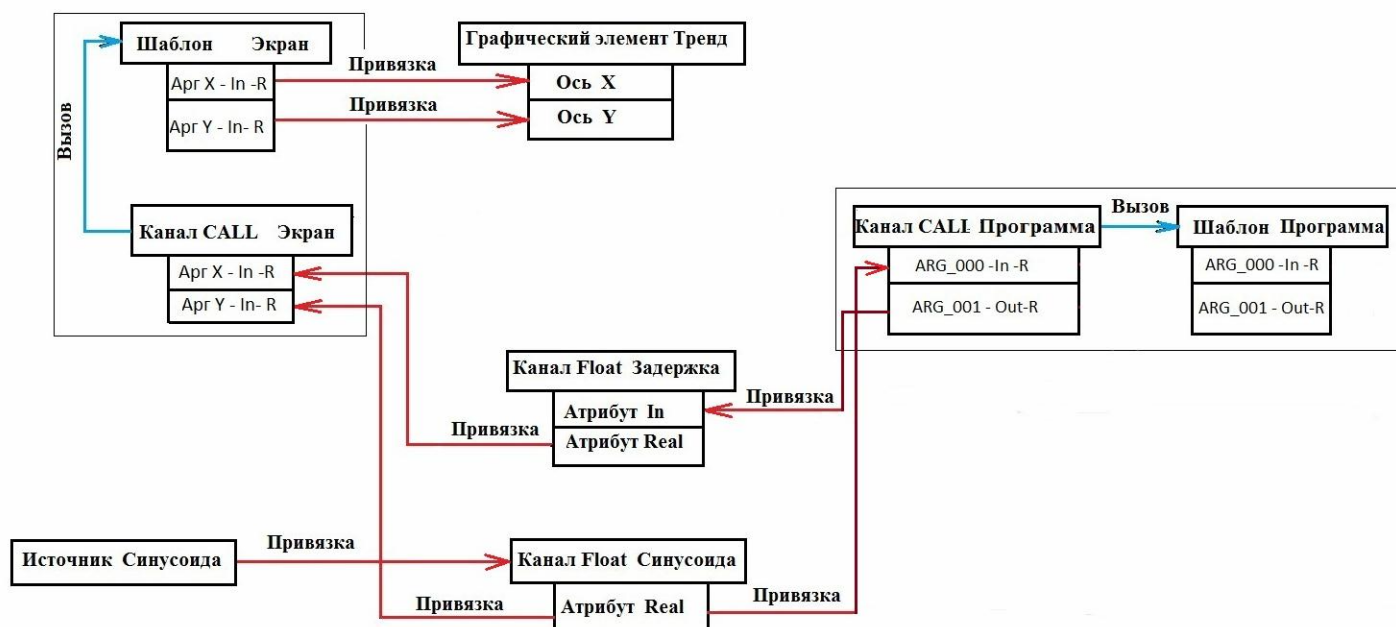


Рис. 1. Графическая схема структуры проекта в среде Trace Mode

Необходимо отметить, что аргументы типа «Out» привязываются к каналам через атрибуты типа «Входное значение», а аргументы типа «In» привязываются к каналам через атрибуты типа «Реальное значение».

Заключение. Создание графической структуры разрабатываемого проекта перед разработкой проекта в среде Trace Mode или параллельно с разработкой позволит улучшить понимание взаимодействия между компонентами проекта и избежать ошибок, связанных с привязками и внутренними связями. Кроме того, помощью графического отображения структуры проекта удобно производить анализ и выявление ошибок, допущенных при его разработке.

**Библиографический список.**

1. AdAstra Research Group. Быстрый старт Trace mode 6 / AdAstra Research Group. — AdAstra Research Group : eKnigi.org, 2007. — 151 с.
2. Пьявченко, Т. А. Автоматизированные информационно-управляющие системы / Т. А. Пьявченко, В. И. Финаев. — Таганрог : Изд-во ТРТУ, 2007. — 271 с.
3. Пьявченко, Т. А. Проектирование АСУТП в SCADA-системе / Т. А. Пьявченко. — Таганрог : Изд-во ТРТУ, 2007. — 84 с.