

В прошлых номерах «itech - интеллектуальные технологии» (№ 2, 3) мы детально рассмотрели подходы в разработке ПО для контроллера ЭЛСИ-ТМ с использованием системы программирования OpenPCS 2004 Automation Suite for IEC 61131-3. Сегодня мы познакомим вас с новыми методами и средствами.



# РАБОТА С «ЖЕЛЕЗОМ» ШАГ ВТОРОЙ

**Н**екоторое время назад разработчики «ЭлеСи» задумались над улучшением методов разработки ПО для контроллера ЭЛСИ-ТМ. И в результате совместных усилий «ЭлеСи» и infoteam родился интегрированный набор инструментов.

16 марта 2006 года infoteam порадовал нас новой версией OpenPCS 2004 Automation Suite 5.2.1, а 7 апреля вышел релиз OpenPCS 2006 Automation Suite версия 5.2.2 с поддержкой ElsyTMMManager. Доступна русская версия.

На небольшом примере мы покажем, как создавать ПО с использованием более простой и понятной схемы.

Для работы нам понадобится следующее:

## Инструменты:

- конфигуратор ElsyTMMManager версии 04.01;
- шаблоны v 03.54;
- система программирования infoteam OpenPCS 2006 Automation Suite for IEC 61131-3 версии 5.2.2;
- поддержка ЭЛСИ-ТМ HWElsyTM\_v0505.cab.

## Материалы (необязательно):

- контроллер ЭЛСИ-ТМ.

Для начала установим OpenPCS 2006 Automation Suite for

Вячеслав **Агафонов**,  
компания ЭлеСи

IEC 61131-3 версии 5.2.2 или выше. Так же установим редактор конфигураций контроллера (ElsyTMMManager) v.04.01.

В OpenPCS через Extras-Tools-Driver Install установим HWElsyTM\_v0504.cab и через Extras-Tools-Licence лицензируем. Все это детально описано в предыдущих номерах itech – интеллектуальные технологии (№ 2,3).

Далее отойдем от уже привычной схемы и воспользуемся новой технологией.

## КОНФИГУРАЦИЯ

Начнем с запуска среды программирования OpenPCS.

Затем создадим новый проект с использованием одного из прототипов ЭЛСИ-ТМ (File-New-Project), например «Проект с участием одного контроллера ЭЛСИ-ТМ» (рис. 1).

Далее двойным щелчком по узлу config.esp вызываем редактор конфигураций.

Приступаем к формированию набора модулей контроллера:

- модуль 00\_main (добавлен по умолчанию);
- модуль 15\_orcs (добавлен по умолчанию);
- добавляем модуль 11\_empty (рис. 2).

С сигналами в конфигурации контроллера нужно произвести следующие действия:

- выходные сигналы с I1 по U4 модуля 11\_empty маршрутизируем в модуль OpenPCS (рис. 3);
- добавляем в модуль 15\_orcs шесть выходных сигналов раз-

ных типов: сигнал с именем I1\_empty и типом I1, с именем I2\_empty и типом I2, I4\_empty с типом I4, U1\_empty должен иметь тип U1, U2\_empty – тип U2 и, наконец, U4\_empty – тип U4 (рис. 4);

■ маршрутизируем созданные сигналы в модуль I1\_empty, начиная с номера 1 (рис. 5).

Информация в контроллере (сигналы контроллера) и программа пользователя, создаваемая в среде OpenPCS, связаны между собой через так называемые «прямоадресуемые переменные». Именно через такие переменные программа OpenPCS получает доступ к аналоговым входам контроллера, дискретным выходам и прочим информационным сущностям контроллера. При этом объявление переменной содержит адрес сигнала внутри контроллера и имя, по которому можно обращаться к сигналу из программы пользователя. В описываемой идеологии для всего ресурса прямоадресуемые переменные глобальны и объявляются в одном файле EtmDGLob.roe. При этом в каждой программе, использующей такие переменные, они должны быть объявлены как VAR\_EXTERNAL.

Таким образом, следующее производимое действие – команда «Обновить Direct Global» в программе ElsyTMManager. При этом объявления переменных, находящиеся в файле EtmDGLob.roe, приводятся в соответствие с конфигурацией контроллера. Эта же команда поможет обновить объявления VAR\_EXTERNAL в файлах программ, использующих прямоадресуемые переменные.

### СОЗДАНИЕ ПРИЛОЖЕНИЯ

Создаем исполняющую программу, например, на языке ST (рис. 6).

Эта программа формирует в сигналы разных типов и отправляет их в модуль empty, который все пришедшие сигналы, согласно конфигурации, передает обратно на oreprcs. Oreprcs принимает эти сигналы, переключивается с преобразованием типа в следующий по типу исходящий сигнал и снова отправляет в модуль empty. Последний принятый сигнал m11\_U4 инкрементируется и переключивается в стартовый сигнал startsint. Далее все повторяется по циклу. Создаем ресурс (рис. 7).

Проверяем тип задачи (рис. 8).

Компилируем и линкуем.

Еще раз убедимся, что IP-адрес в конфигурации контроллера корректный.

Дальнейший материал можно использовать только при наличии контроллера.

### СОЗДАНИЕ СОЕДИНЕНИЯ

Для того чтобы среда OpenPCS «знала», с каким контроллером предстоит работать, нужно создать «соединение». Это нужно сделать с помощью меню PLC – Connections. В создаваемом соединении важно указать номер порта 9988 и IP-адрес контроллера (он должен совпадать с указанным в конфигурации).

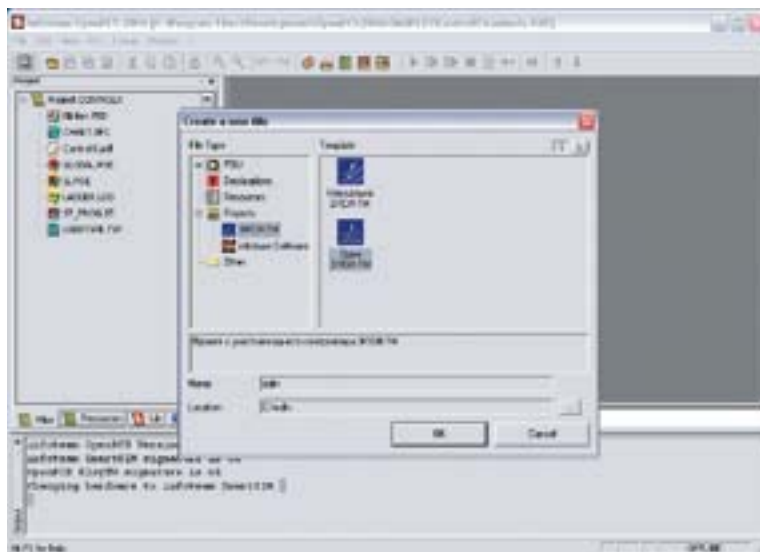


Рис. 1. Создание нового проекта

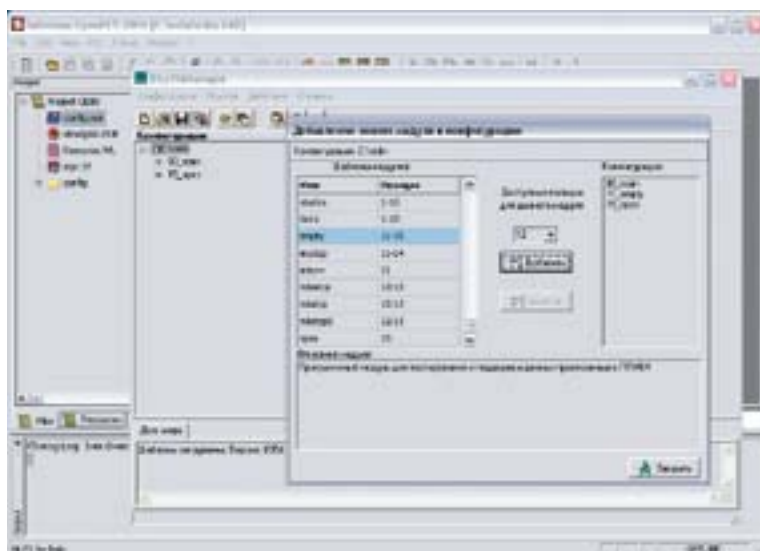


Рис. 2. Формирование набора модулей

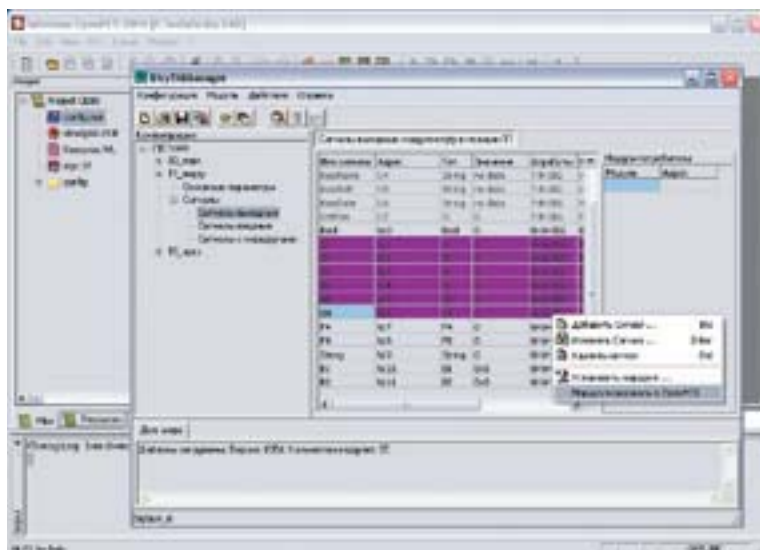


Рис. 3. Маршрутизация сигналов в OpenPCS

◀ работа с «железом» шаг второй

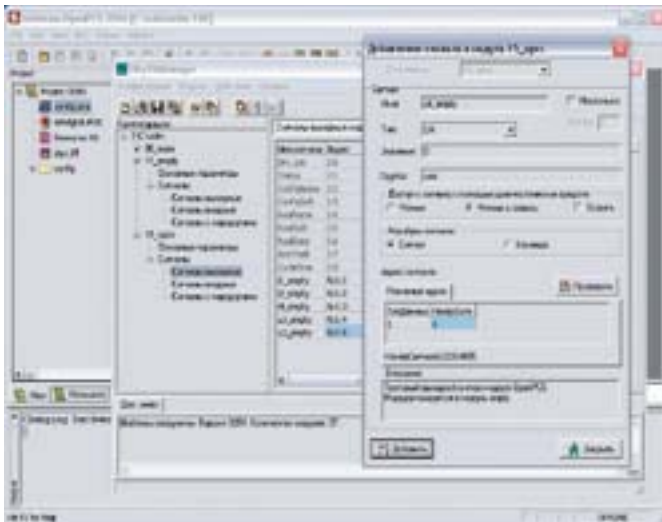


Рис. 4. Добавление выходных сигналов

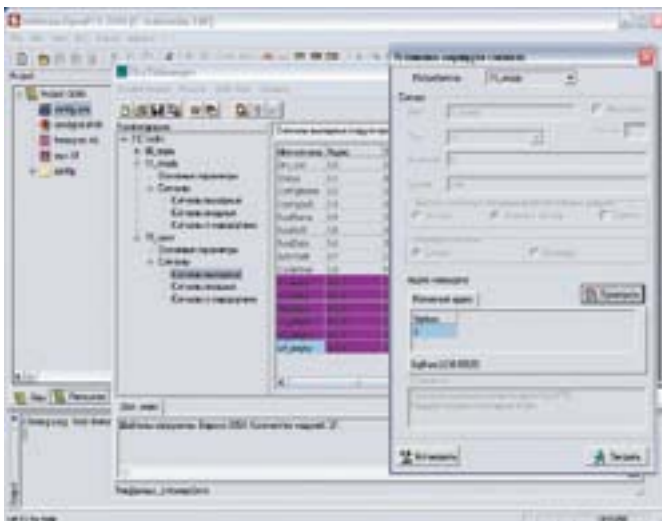


Рис. 5. Маршрутизация сигналов из OpenPCS

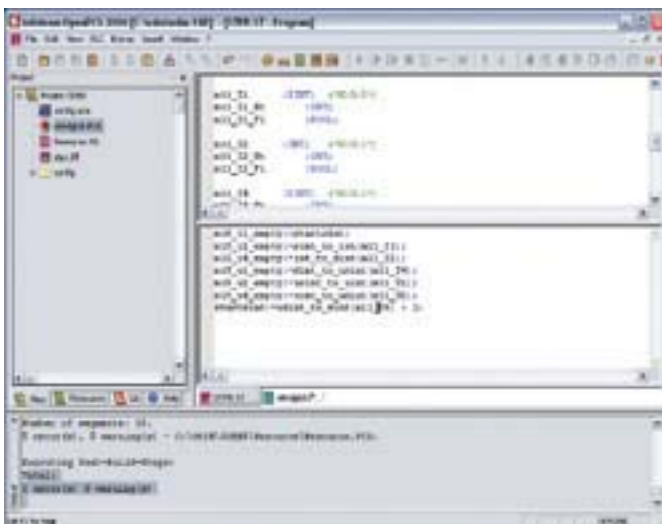


Рис. 6. Написание управляющей программы

Затем в закладке resources нам нужно выбрать в свойствах (properties) текущего ресурса Network connection имя нашего соединения (рис. 9).

**ЗАГРУЗКА ПРОЕКТА**

Загрузка проекта теперь осуществляется еще проще. Не нужно загружать отдельно конфигурацию ПЛК и исполняющую программу. Все выполняется одним действием при нажатии Online. После этого нужно просто стартовать приложение и проверить его работу по изменению переменных. ▶



Рис. 7. Выбор драйвера соединения

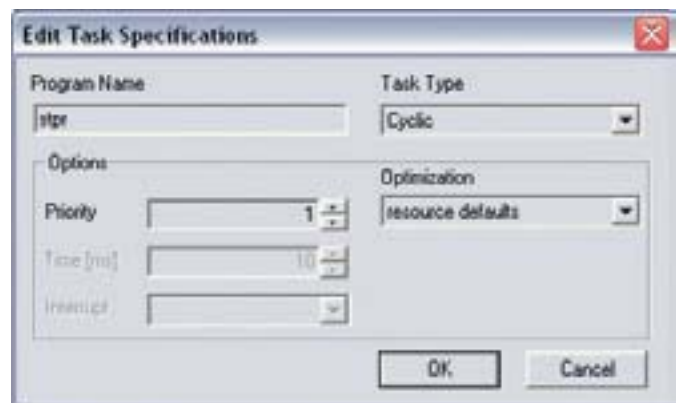


Рис. 8. Настройка исполнения задачи



Рис. 9. Настройка ресурса