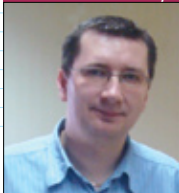


Из PowerDistribution – в Automation!

(Окончание. Начало в #6/2009)

Александр Смирнов (COLLA, Ltd.)

alex@colla.lv



В этом номере мы завершаем рассказ о взаимодействии программных продуктов датской компании PC|SCHEMATIC AS, а именно, PowerDistribution и PC|SCHEMATIC Automation, начатый в #6/2009.

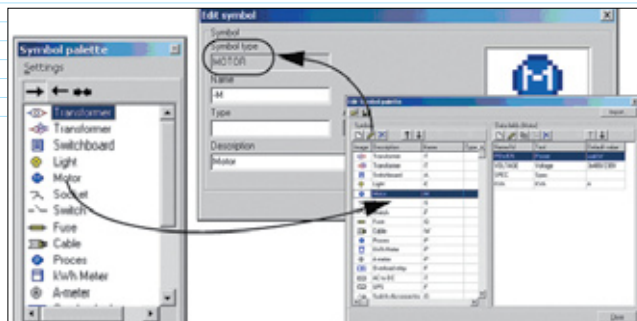


Рис. 7. Уточнить тип символа можно, запустив в PowerDistribution функцию редактирования палеты символов

Вставка кодов в шаблон проекта

При переносе данных из системы PCschematic PowerDistribution в шаблон проекта в формате системы Automation, отправной точкой послужит символ (элемент), который необходимо выбрать в рабочем окне, где представлена иерархия проектируемой сети, а затем нажать на клавишу “Создать чертеж”.

Давайте подробнее разберем, каким образом и какую информацию, связанную с элементом, можно перенести в шаблон. Существуют три различных способа переноса, которые мы рассмотрим ниже.

1 Перенос четырех основных значений полей данных символа

Если существует необходимость перенести данные основных полей данных символа (а именно – Name, Type, Article No и Description), то для каждого поля в окне данных символа в шаблоне Automation надо проделать следующую процедуру: ввести в эти поля специальные составные коды (см. таблицу).

Обозначение в зоне данных символа	Код
Имя (Name)	NAME
Тип (Type)	TYPE
Артикул № (Article No)	ARTICLE
Описание (Description)	FUNCTIONTEXT

Для примера возьмем данные из поля Name. Код состоит из четырех частей, и правила его образования следующие:

- 1-я часть – знак @, который указывает, что данные будут извлекаться из PowerDistribution;
- 2-я часть – описание типа символа из палитры (в нашем случае – MOTOR);
- 3-я часть – знак “точка”, который служит разделителем;
- 4-я часть – обозначение графы данных в PowerDistribution (в нашем случае – NAME).

В результате у нас получается @MOTOR.NAME (рис. 6).



Рис. 6. Так должен выглядеть составной код для импорта данных поля Name из PowerDistribution

При необходимости уточнить тип символа следует запустить функцию редактирования палеты символов; вся информация будет доступна в открывшемся окне (рис. 7).

2 Перенос значений дополнительных полей данных

Как уже говорилось ранее, описание элемента в проекте PowerDistribution может содержать гораздо больше информации о компоненте, чем в проекте Automation, поскольку эти данные могут быть нужны для осуществления требуемых расчетов. При необходимости переноса такой информации в шаблон, нам придется идти тем же путем, что был описан выше. Исключение составляет лишь шаг, где вместо обозначения поля необходимо ввести значение из поля в столбце Name/Id.

К примеру, если необходимо перенести значение дополнительного поля данных Type (тип символа) для элемента Switchboard (распределительный щит, тип символа – SWBOARD), обозначение графы данных (Name/Id) будет следующим: TYPE_ID (рис. 8).

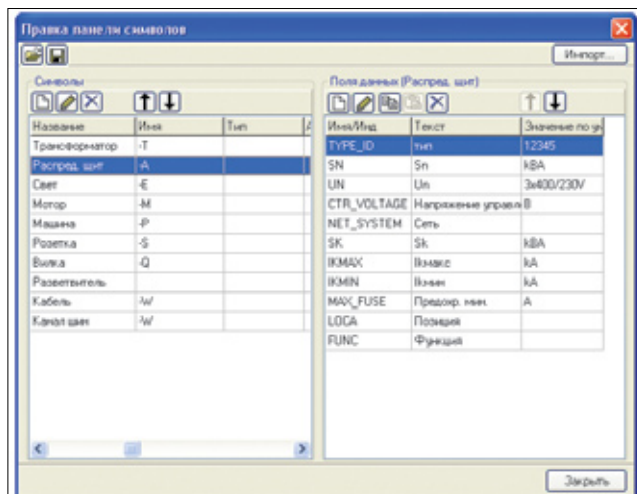


Рис. 8. Для переноса дополнительных полей данных служат столбцы Name/Id (Имя/Инд)

В данном случае полный составной код должен выглядеть так: @SWBOARD.TYPE_ID

3 Перенос данных о тех элементах, которые не представлены в главном окне

Иногда могут возникать ситуации, когда результирующий проект в *Automation* должен содержать элементы, которых в *PowerDistribution* нет (соответственно, нет и связанных с ними данных), поскольку их присутствие там не обязательно для решения поставленных перед проектировщиком задач.

Для такого вида переноса нам понадобится создать так называемый построчный (*inline*) файл, что можно сделать с помощью стандартной программы *Notepad* или любого другого текстового редактора.

Рассмотрим процесс создания такого файла по шагам:

- запускаем текстовый редактор (пусть это будет *Notepad*) и в первой строчке вводим *Name#Type*;
- далее вводим позиционное обозначение первого символа в шаблоне *Automation*, который выступает в качестве отправной точки в *PowerDistribution*, и сразу за ним добавляем #*INLINE* (к примеру, если используется символ *-P1*, то строчка должна выглядеть так: *P1#INLINE*);
- теперь аналогичным образом набираем строки с символами, отсутствующими в проекте-источнике;
- сохраняем файл.

При этом нужно учитывать, что имя этого файла должно строго совпадать с именем шаблона (за исключением расширения, которое всегда должно быть *.DAT*). К примеру, если название шаблона — *Eldemo.pro*, то построчный файл будет называться *Eldemo.dat*, и он должен находиться в той же папке.

Заполнение полей данных, указанных в “построчном” файле

В тот момент, когда мы захотим создать проект для элементов той ветки проекта сети, которая содержит неучтенные компоненты, не оказывающие физического влияния на распределение сигнала (тока, воздуха, воды) в сети (например, датчики), необходимо воспользоваться заранее подготовленным “построчным” файлом со списком этих символов. Файл активизируется при щелчке мышкой на первом из символов, после чего мы, во-первых, увидим в окне “Черчение с шаблонов” этот список, а, во-вторых, сможем внести изменения в обозначения (рис. 9).

Порядок поиска и обработки элементов, данные о которых переносятся в проект

Когда выбран символ в окне проекта, выбран шаблон проекта и нажата кнопка “Генерация чертежа”, программа начинает считывать информацию с нужных полей данных элемента в проекте *PowerDistribution* для осуществления дальнейшего переноса.

Далее система просматривает все поля данных символов или свободные текстовые поля в проекте *Automation* и определяет поля, содержащие составной код. Рассмотрим этот процесс на примере по шагам.

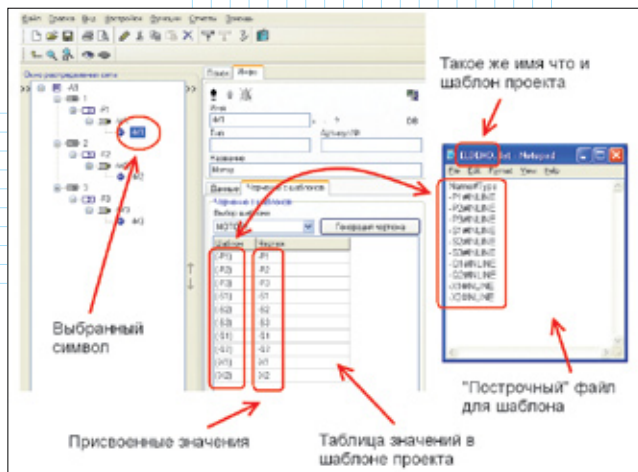


Рис. 9. В окне “Черчение с шаблонов” можно изменить обозначения элементов

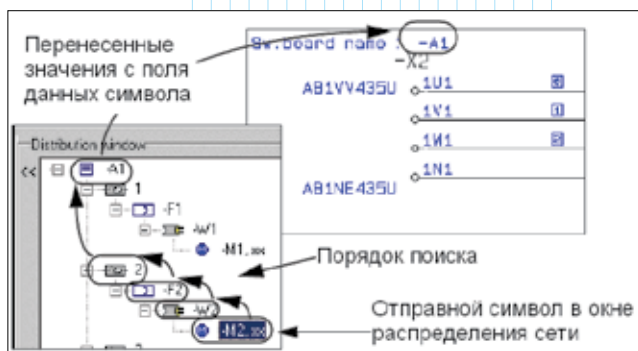


Рис. 10. Процесс поиска и переноса информации с полей данных символа

1 В первую очередь определяется тип символа, с которого необходимо “забрать” информацию (предположим, что тип символа — *SWBOARD*).

2 После этого программа производит поиск символа такого типа в рабочем окне *PowerDistribution*, где представлена иерархия сети.

3 Поиск осуществляется по дереву проекта, начиная с символа, принятого за отправную точку в шаблоне. В нашем случае это будет мотор *-M2*.

4 Если тип символа не совпадает с типом, указанным для *-M2*, то рассматривается следующий символ, стоящий иерархически выше. В нашем примере это будет кабель *-W2*, который тоже не совпадает по типу. Следующий символ — реле *-F2*, и с ним дело обстоит точно так же...

5 Поиск будет продолжаться до тех пор, пока программа не найдет элемент с нужным типом.

В нашем случае это будет распределительный щит *-A1* на самой вершине иерархического дерева сети. В конце концов, он будет найден, и информация из его полей данных будет перенесена в проект *Automation* (рис. 10).

6 Подобная процедура повторяется для всех полей, содержащих составные коды.

По окончании процедур поиска и переноса данных создается проект в формате системы *PC|SCHEMATIC Automation*.