

Язык непрерывных функциональных схем CFC.

Часть 2

Федор Титов, инженер OBEH

Во второй части статьи продолжим рассказ о свойствах и настройках редактора CFC программного пакета CODESYS.

Свойства редактора CFC

Рабочая область редактора CFC имеет ряд полезных настроек. Можно настроить точки сетки (*Инструменты – Опции – CFC-редактор – Вид*) или оставить по умолчанию белый фон. Для задания фиксированного размера рабочей области должен присутствовать программный код редактора на рабочей области. Делаем <клик> на рабочей области и заходим в меню *CFC – Редактировать рабочий лист*. По умолчанию размер области подстраивается под расположение элементов программы. В правом нижнем углу рабочей области находятся кнопки управления визуальным представлением кода программы: масштабирование (<Ctrl> + <скроллинг мыши>), лупа, захват рабочей области редактора. Лупа существенно упрощает работу с крупными проектами: при выводе всей программы на рабочую область элементы становятся слабо различимыми, и лупа позволяет отчетливо видеть код, не прибегая к масштабированию рабочей области.

Общие свойства блоков

В первой части статьи [АиП, 2020, №2, стр. 33-35] был рассмотрен способ

добавления элемента программы на рабочую область редактора. Существует еще способ добавления не одного, а сразу нескольких однотипных элементов. Для этого, зажав клавишу <Ctrl>, нужно кликнуть мышью по выбранному элементу на панели инструментов, переместить курсор мыши (в виде креста) на рабочую область редактора и добавить необходимое количество элементов нажатием ЛКМ.

При работе с большими проектами часто возникает необходимость перемещать группы блоков по рабочей области редактора. Для этого нужно обвести группу блоков. Обычно с первого раза сделать это не удастся, так как какой-нибудь элемент не попадает в выделение, и операцию приходится повторять. Намного удобнее создать несколько групп блоков, которые будут перемещаться как одно целое. Для этого выделяется группа блоков, на одном из них нажимается ПКМ и выбирается пункт *Создать группу* в меню *Группа*. Группу можно расформировать, выполнив команду *Разгруппировать*.

Редактор CFC предоставляет пользователям возможность кастомизации

проектов. Для этого в меню *Инструменты* в разделе *Опции* переходим на вкладку *CFC-редактор, Вид* и устанавливаем галочку *Иконка элемента*. После этого у стандартных функций и ФБ появятся пиктограммы, и на вкладке *Изображение* в *Свойствах элемента* появится возможность добавить графическое изображение пользовательским функциям и ФБ (рис. 1).

Условный оператор на языке CFC и работа с логическими переменными

Реализация условий на языке CFC – часто задаваемый вопрос пользователей. Данная задача решается средствами редактора CFC в два клика: необходимо нажать ПКМ на блоке и выбрать пункт *EN/ENO*. Блоку добавится «условный» вход EN и выход ENO типа *BOOL*. Когда на вход EN приходит значение *TRUE*, то блок становится активен, значение на выходе ENO устанавливается *TRUE*. Если на вход EN пришло *FALSE*, то блок «замораживается», на выходе ENO – значение *FALSE*. «Заморозка» блока означает, что значения на выходе остаются постоянными и рав-

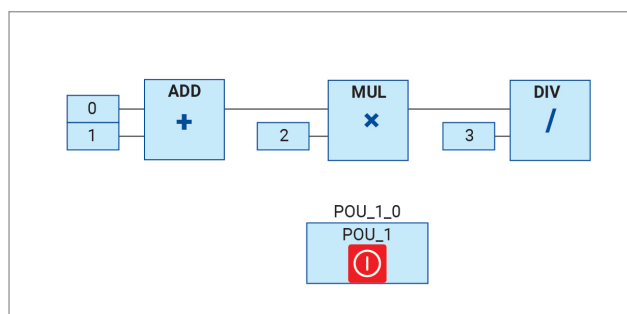


Рис. 1. Кастомизация программы

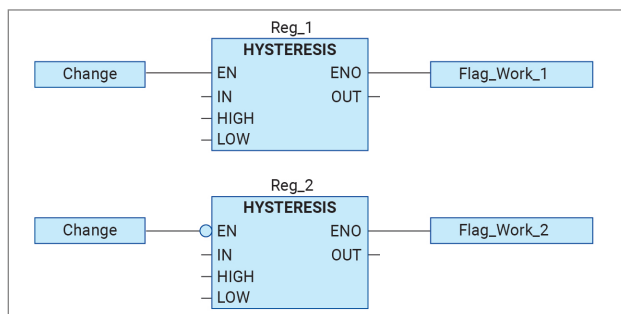


Рис. 2. Вход/выход EN/ENO

ными значениям в последнем цикле контроллера перед выключением блока. Об этом важном моменте нужно всегда помнить. Например, недостаточно отключить блок управления при работе с регуляторами (ПИД,

гистерезис), его выходы необходимо обнулить, в противном случае есть вероятность подачи сигнала на открытие задвижки или на нагрев ТЭНа, что может привести к поломке оборудования.

При работе с переменными типа BOOL возникают задачи инвертировать их значения. Для этого нажимаем ПКМ на входе или выходе блока, работающего с логической переменной, и выбираем пункт *Инверсия*. Примером инверсии «условного» входа EN/выхода ENO служит переключение режима работы двух блоков по сигналу от одной переменной (рис. 2).

Линии связи

Начиная с версии CODESYS V3.5 SP3, проблема с наложением линий связи друг на друга при их близком рас-

положении или на элементы схемы осталась в прошлом. Чтобы не тратить время на перетаскивание элементов схемы и перестроение линий связи, введена функция трассировки. Если автоматическая трассировка не справилась, и возникла ситуация с наложением или некорректным построением линий связи, на помощь приходит ручная трассировка. Для этого требуется нажать на кнопку трассировки и выбрать *Установить все соединения* (рис. 3).

Еще одна опция для работы с линиями связи редактора CFC – изменение цвета. В крупных проектах, содержащих большое число линий связи и различные типы данных, разбиение по цветам помогает избежать ошибок при построении схем и отладить уже готовую программу. Изменить цвета линий можно в меню *Инструменты*, в разделе *Опции*, на вкладке *CFC-редактор, Вид*.

Обратные связи

В первой части статьи была рассмотрена автоматическая нумерация блоков в соответствии с порядком данных, которая начала действовать с версии CODESYS V3.5 SP15. Рассмотрим частные случаи автоматической нумерации блоков, охваченных обратной связью (ОС). В первом примере (рис. 4) порядок элементов, охваченных ОС, совпадает с их расположением на рабочей области редактора и с порядком их выполнения: последовательно слева направо.

Во втором примере элементы, охваченные ОС, расположенные в шахматном порядке (рис. 5), нумеруются построчно: сначала элементы верхней строки, затем – нижней.

Аналогичная построчная нумерация элементов, охваченных ОС, показана на рис. 6. Данное свойство автоматической нумерации присуще только элементам, охваченным ОС. При ее отсутствии автоматическая нумерация происходит строго в соответствии с потоком данных.

В версии CODESYS V3.5 SP15 вместе с автоматической нумерацией элементов программы появилась возможность установить порядок выполнения элементов с обратной

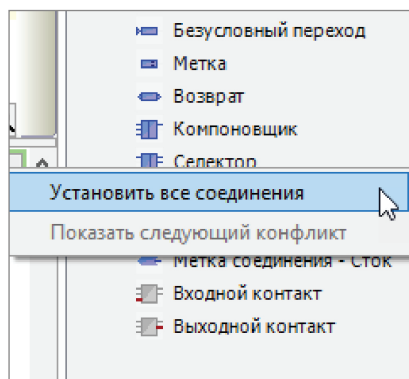


Рис. 3. Ручная трассировка

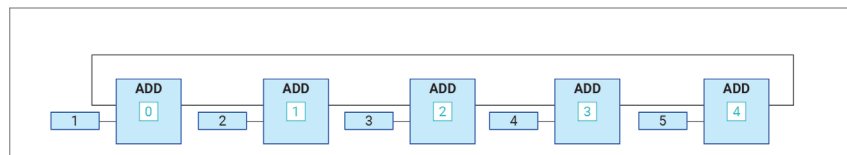


Рис. 4. Пример 1: обратная связь

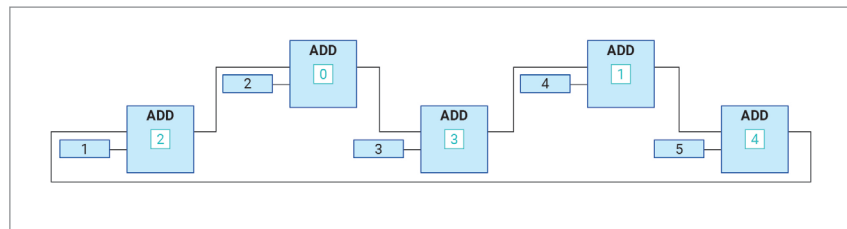


Рис. 5. Пример 2: обратная связь

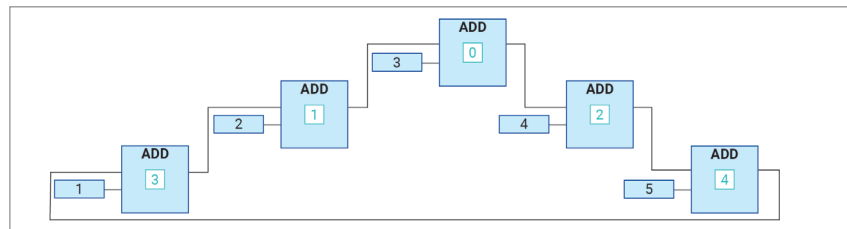


Рис. 6. Пример 3: обратная связь

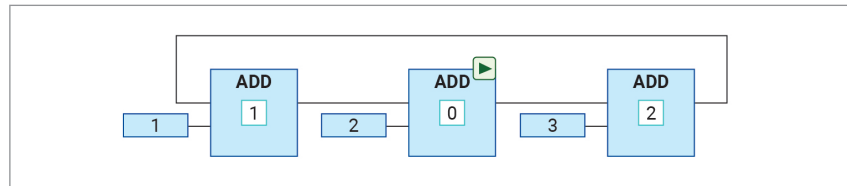


Рис. 7. Назначение порядка элементов в ОС

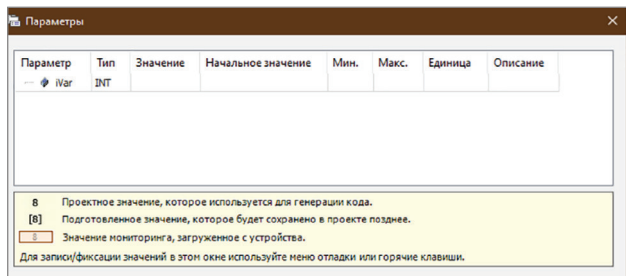


Рис. 8. Диалоговое окно Параметры в ФБ

Device.Application.PLC_PRG		
Выражение	Тип	Значение
usiVar1	USINT	100
usiVar2	USINT	200
uiVar	UINT	300

1	uiVar := usiVar1 + usiVar2; RETURN	
---	------------------------------------	--

Рис. 9. Сложение на ST

связью (рис. 7). По умолчанию порядок выполнения элементов с ОС осуществляется в соответствии с потоком данных – слева направо. Чтобы изменить порядок, следует нажать ПКМ на элементе, который должен являться начальным, в открывшемся меню выбрать *Порядок выполнения*, далее пункт *Задать начало обратной связи*, тогда над выбранным элементом появится значок Play.

Важно помнить, что при назначении какого-либо элемента начальным в группе элементов с ОС, остальные элементы не нумеруются согласно рассмотренному выше свойству автоматической нумерации. Чтобы избежать ошибок, связанных с порядком выполнения блоков, необходимо внимательно следить за расположением элементов на рабочей области редактора.

Функциональные блоки

Информация этой части статьи важна для многих пользователей. Речь пойдет о разделе VAR_INPUT CONSTANT в функциональных блоках. В раздел записываются пользовательские константы так же, как и в VAR CONSTANT, только с одним отличием: при запуске приложения значения переменных из раз-

дела VAR CONSTANT изменять нельзя, а из раздела VAR_INPUT CONSTANT – можно.

Важно отметить, что изменять значения констант из раздела VAR_INPUT CONSTANT можно только в программном комплексе CODESYS. Начальное значение переменных можно задать двумя способами: присвоить конкретное значение или привязать внешнюю переменную программы.

Значение внешней переменной записывается в переменную из раздела VAR_INPUT CONSTANT один раз при запуске приложения. На функциональных блоках в коде программы отображается кнопка *Параметры*, открывающая диалоговое окно, в котором можно задать значения переменных из раздела VAR_INPUT CONSTANT (рис. 8). Кнопка активна и при запуске приложения. В процессе работы программы на контроллере переменные VAR_INPUT CONSTANT изменять нельзя.

Конверсия данных на CFC

Рассмотрим разные подходы к выполнению арифметических и других операций на языках ST и CFC.

При операции сложения на языке ST складываются две переменные с типами USINT (usiVar1=100 и usiVar2=200),

результат присваивается переменной типа UDINT (UINT), тогда операция сложения выполнена верно (uiVar=300) (рис. 9).

Если ту же операцию реализовать на языке CFC, то результат выполненной операции будет неверным (44 вместо 300) (рис. 10). Ошибка является следствием того, что к выходному контакту блока суммы внутри самого блока привязана неявная переменная, тип которой определяется «наибольшим» из входных переменных.

Таким образом, результат суммы переменных сначала формируется внутри самого блока (значение неявной переменной), а уже после присваивается переменной, стоящей на выходе блока.

Следовательно, при сложении двух переменных типа USINT и ожидаемом результате типа UDINT (UINT) возможно переполнение внутренней неявной переменной, что в конечном итоге приводит к неверному результату. Это решается применением явной конверсии типов хотя бы для одной из входных переменных из типа USINT в UDINT (UINT), тогда тип внутренней неявной переменной установится UDINT (UINT) (рис. 11). ■

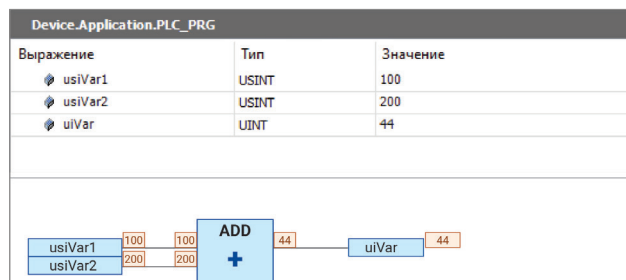


Рис. 10. Сложение на CFC типов данных без конверсии

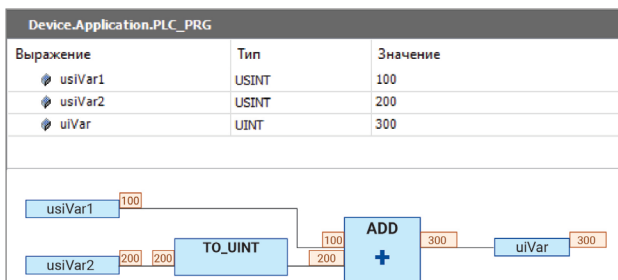


Рис. 11. Сложение на CFC типов данных с конверсией