

# Редактирование свойств блока и тестирование

## программы

# 5

### Что описывает эта глава?

После создания и программирования блоков данных и кодовых блоков проверьте и отредактируйте свойства блоков. Позднее это даст Вам возможность точно идентифицировать блок и делать заключения о его возникновении. Это может оказаться очень полезным при устранении ошибок.

В инкрементном редакторе имеется возможность тестирования отдельного блока, редактируемого в прикладной программе, находящейся в центральном процессоре. При этом Вы можете исполнять программу последовательно команда за командой. Эта возможность тестирования, называемая статусом программы, окажет Вам поддержку во время пуска системы в эксплуатацию при дополнительной проверке важных процессов, а также при устранении ошибок.

### Обзор главы

В разделе	Вы найдете	на стр.
5.1	Редактирование свойств блоков	5–2
5.2	Тестирование программы на AWL - обзор	5–4
5.3	Настройка отображения статуса программы	5–5
5.4	Установка условия запуска	5–6
5.5	Выбор среды тестирования и запуск/остановка статуса программы	5–7



**Имя и семейство блока** Задание имени и семейства для блоков дает возможность классификации разработанных Вами блоков. Так, например, Вы можете включить в семейство “Regler”(“Регулятор”) все блоки, программирующие работу регулятора.

**Версия блока** Из этого атрибута Вы можете получить информацию о том, в какой версии STEP 7 был разработан блок. Блоки версии 1 Вы должны конвертировать, чтобы включить их в программу версии 2. Это можно сделать в SIMATIC Manager с помощью команды меню **Datei ▶ Version 1– Projekt öffnen...** (Файл ▶ Открыть проект версии 1...).  
Блоки версии 1 нельзя использовать с мультитекстными блоками. Вы должны сгенерировать из них исходный файл, который затем компилируется в блоки версии 2. Дальнейшую информацию можно получить из Руководства пользователя /231/.

**Атрибуты блока** Под заголовком Bausteinattribute (Атрибуты блока) Вы можете найти следующие записи:

- Атрибут блока **”KNOW HOW Schutz”** указывает на то, что блок защищен, и имеет следующее действие:
  - операторная часть блока невидима;
  - в таблице описания переменных не отображаются временные и статические переменные;
  - при генерации из защищенного блока исходного файла последний содержит только входные, выходные и проходные параметры, но никаких команд;  
Указание: Если эти исходные файлы снова компилируются в блок, то операторная часть теряется, так как старый блок переписывается!
- Вы не можете редактировать свойства блока.
- Если установлен атрибут **”Standardbaustein”** (“Стандартный блок”), то речь идет о защищенном стандартном блоке фирмы Siemens. Это отображается с помощью соответствующей записи слева внизу.
- Атрибут **”READ\_ONLY”** (“Только для чтения”) говорит о том, что блок защищен от записи. Это имеет смысл для блоков данных с постоянными значениями, которые не могут быть изменены.
- Атрибут **”Unlinked”** (“Не связан”) может встретиться только у блоков данных. Он означает, что блок данных не загружается из загрузочной памяти в рабочую память CPU. Из этого блока нельзя сгенерировать текстовый файл. Доступ к блокам данных в загрузочной памяти происходит через SFC, которые копируют в рабочую память только содержимое DB. Этим достигается лучшее использование рабочей памяти, которая, таким образом, содержит только данные, релевантные для исполнения программы.

---

#### Указание

Для своего блока Вы можете ввести атрибуты защиты блока, защиты от записи и несвязанности только при программировании в форме исходного файла. Если Ваш блок создан с помощью инкрементного редактора, то конвертируйте его в исходный файл, в который Вы можете вносить атрибуты. После компиляции исходного файла в блоки последние оказываются соответствующим образом защищенными.

---

## 5.2. Тестирование программы на AWL – обзор

**Возможность тестирования** Вы можете протестировать программу на AWL, отобразив в выбранных Вами столбцах для каждой команды состояние статуса программы (VKE, Statusbit - бит состояния) или содержимое соответствующих регистров. Отображение статуса программы обновляется циклически.

**Предпосылки** Для возможности отображения статуса программы должны быть выполнены следующие предпосылки:

- Вы должны были сохранить блок без ошибок и затем загрузить его в CPU.
- CPU работает, обработка прикладной программы идет.
- Вы открыли блок в режиме online.

**Принципиальная последовательность действий** На рис. 5–2 показана принципиальная последовательность действий для наблюдения статуса программы:

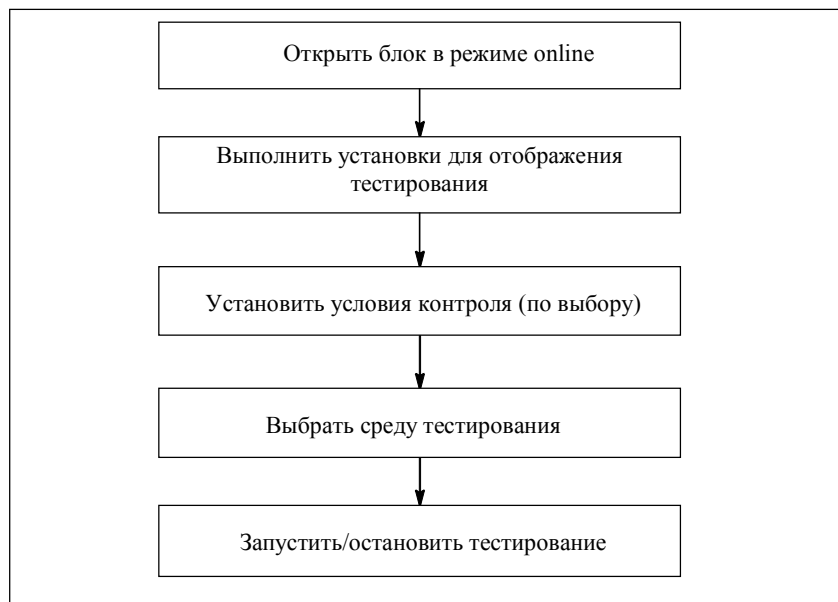


Рис. 5-2. Последовательность действий при тестировании кодового блока в AWL

### 5.3. Настройка отображения статуса программы

**Последовательность действий** Перед запуском статуса программы на AWL установите, какую информацию Вы хотели бы получить. Для этого откройте с помощью команды меню **Extras ► Einstellungen** (Дополнительные функции ► Настройка) диалоговый блок “AWL”.

**Возможные установки** Вы можете отобразить следующие поля статуса:

Таблица 5–1. Сокращения в установке статуса программы

При отметке ...	отображается:
Statusbit (бит состояния)	бит состояния, т.е. бит 2 слова состояния
VKE	результат логической операции, т.е. бит 1 слова состояния; указывает результат логической операции или арифметического сравнения.
Standardstatus (стандартный статус)	содержимое слова таймера, слова счетчика или Akku 1, если в команде имеют место соответствующие операции.
AR1, AR2	содержимое соответствующих адресных регистров при косвенной адресации через регистры (внутризонной или с указанием в регистре области памяти)
Akku2	содержимое Akku 2
DBR1, DBR2	содержимое регистра блока данных, первого или второго открытого блока данных.
Indirekt (косвенная)	косвенная ссылка на память; задание указателя (адреса), а не содержимого адреса; возможно только при косвенной адресации через память, но не при косвенной адресации через регистр.

С помощью кнопки ”Vorgabe” (“Предварительная установка”) Вы всегда получите бит состояния, VKE и стандартный статус.

## 5.4. Установка условия запуска

### Фон

С помощью условия запуска Вы можете установить среду вызова тестируемого блока. Тестирование выполняется только тогда, когда выполнено установленное условие запуска.

**Последовательность** Условия запуска можно установить через команду меню **Test ► действий Aufrufumgebung** (Тест ► Среда вызова).

Рис. 5-3. Установка условия запуска

### Значение условий запуска

Три возможных установки имеют следующее значение:

- **Условие запуска отсутствует:** Среда запуска тестируемого блока не играет роли. Однако, если Вы вызываете один и тот же блок в различных местах программы, то Вы не сможете отличить, для какого вызова отображается статус.
- **Путь вызова:** Здесь может быть указан путь, через который должен быть вызван тестируемый блок, чтобы запустить запись статуса. Вы можете ввести три последних уровня вызовов перед тестируемым блоком.
- **Открытые блоки данных:** Здесь устанавливается Среда вызова путем указания одного или двух блоков данных. Запись статуса происходит, если тестируемый блок был вызван с указанными смотря по обстоятельствам блоками данных.

## 5.5. Выбор среды тестирования и запуск/остановка статуса программы

### Выбор среды тестирования

В режиме online различают две ситуации, в которых Вы можете тестировать свою программу:

- В среде тестирования **”Prozeß”** (“Процесс”) Вы тестируете свою программу online при условиях управления процессом. При этом статус команд в программных циклах, выполняемых в цикле многократно, выясняется **только при первом проходе**. Этот режим характеризуется наименьшей загрузкой времени цикла.
- В среде тестирования **“Labor”** (“Лаборатория”) Вы тестируете свою программу также online в лабораторных условиях. Однако в программных циклах статус команд, которые в цикле выполняются многократно, выясняется **при каждом проходе**. В этом режиме возможна существенная загрузка времени цикла, так как удлинение времени цикла зависит от количества проходов и количества наблюдаемых команд.

Среду тестирования Вы выбираете командой меню **Test ▶ Testumgebung ▶ Labor/Prozeß** (Тест ▶ Среда тестирования ▶ Лаборатория/процесс).

### Запуск и остановка программы

Запуск и остановка статуса программы осуществляется через команду **статуса** меню **Test ▶ Beobachten**(Тест ▶ Наблюдение). Статус программы отображается только для области, видимой в данный момент в редакторе.

Adresse	Dekl.	Name	Typ	Anfangswert	Kommentar
0.0	in	dur_g p	S5TIME	S5T#0MS	
2.0	in	del r p	S5TIME	S5T#0MS	
4.0	in	starter	BOOL	FALSE	

U?	1	1	6	
U #starter	1	1	6	
U #t_next_red_car	0	0	6	
O #condition	0	0	6	
)	0	1	6	
UN #t_dur_r_car	0	0	6	
= #condition	0	0	6	

		VKE	STA	STANDARD
UN #condition	1	0	6	
= #g_car	1	1	6	

		VKE	STA	STANDARD
U #condition	0	0	6	
L S5T#3S	0	0	300	
SE #t_dur_y_car				

Рис. 5-4. Статус программы на AWL (пример)

## Проверка времени

Активизация режима тестирования увеличивает время цикла. При **цикла** превышении установленного времени CPU переходит в STOP, если Вы не запрограммировали OB 80.

Установленное и текущее время цикла Вы можете отобразить и проверить с помощью команды меню **Zielsystem ► Baugruppenzustand** (Контроллер ► Состояние модуля). Максимальное время цикла при необходимости Вы можете изменить для целей тестирования при параметризации аппаратуры в свойствах CPU.